

ISSN 2087-7897
ISSN (ONLINE) 2460-5344
Volume 10, Nomor 1, Januari 2020

Jurnal Ilmiah
Sisfotenika



STMIK Pontianak



IndoCEISS



Coris

JURNAL SISFOTENIKA
ISSN : 2087 – 7897
ISSN (ONLINE) : 2460 – 5344
Volume 10, Nomor 1, Januari 2020, hlm. 1 - 126

Pelindung dan Penyandang Dana
Ketua Yayasan Harapan Bersama Pontianak

Penanggung jawab
Ketua STMIK Pontianak

Editor-In-Chief
David, S.Kom.,M.Cs.,M.Kom., STMIK Pontianak

Associate Editor
Prof. Dra.Sri Hartati, M.Sc.,Ph.D, Universitas Gadjah Mada
Prof. Dr. Ir. Edi Abdurachman, MS., M.Sc., Universitas Bina Nusantara
Prof. Drs. Slamir M.Comp.Sc.,Ph.D, Universitas Jember
Prof. Ir. Zainal Arifin Hasibuan, MLS., Ph.D, Universitas Indonesia
Andrew Fiade, M.Kom, Universitas Islam Negeri Syarif Hidayatullah Jakarta

Section Editor
Susanti Margaretha Kuway, S.Kom., M.Kom, STMIK Pontianak
Gusti Syarifudin, ST., MMSI., M.Kom, STMIK Pontianak
Gat, S.Kom., M.Kom, STMIK Pontianak
Ir. Junaedi ,M.Sc, Universitas Tanjungpura
I Dewa Ayu Eka Yuliani, S.Kom.,M.Kom, STMIK Pontianak

Editor
Wahyu Sindu Prasetya, S.Kom.,M.Kom, STMIK Pontianak
Handy Kusuma, S.Kom., MMSI, STMIK Pontianak
Ponti Harianto Soetardi, S.Kom.,M.Cs, STMIK Pontianak

Jurnal Ilmiah SISFOTENIKA diterbitkan oleh LPPM dan IndoCEISS. Frekuensi Terbit
Tengah Tahunan (2 kali dalam setahun, yaitu Bulan Januari dan Juli)

Alamat Redaksi:
STMIK Pontianak
Jl. Merdeka 372, Pontianak, Kalimantan Barat
No. Telp 0561-735555, No. Fax 0561-737777
Website : www.sisfotenika.stmikpontianak.ac.id
E-mail : Sisfotenika@stmikpontianak.ac.id
Sisfotenika@gmail.com

JURNAL SISFOTENIKA
ISSN : 2087 – 7897
ISSN (ONLINE) : 2460 – 5344
Volume 10, Nomor 1, Januari 2020, hlm. 1 - 126

DAFTAR ISI

Implementasi Decision Tree Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu <i>Christin Nandari Dengen, Kusrini, Emha Taufiq Luthfi (Universitas Amikom Yogyakarta)</i>	1 - 12
Implementasi Hill Cipher Pada Kode Telepon dan Five Modulus Method dalam Mengamankan Pesan <i>Patmawati Hasan, Selviana Yunita, Dony Ariyus (STMIK Sepuluh Nopember Jayapura)</i>	13 - 23
Sistem Pendukung Keputusan Menentukan Bibit Padi Terbaik Menggunakan Metode Gap Kompetensi <i>Elvis Pawan, Agung Jasuma, Achmad Yusron Arif, Kusrini (STMIK Sepuluh Nopember Jayapura)</i>	24 - 36
Sistem Pakar Deteksi Dini Kesehatan Mental Menggunakan Metode Dempster-Shafer <i>Annisa Rahmadhani, Fauziah, Andri Aningsih (Universitas Nasional)</i>	37 - 49
Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04 <i>Kresno Wibowo, Iskandar Fitri, Deny Hidayatullah (Universitas Nasional)</i>	50 - 61
Perancangan Neur-O: Sistem Pakar Penyakit Saraf berbasis Ontologi <i>Christopel H. Simanjuntak, Dirko G. S. Ruindungan (Politeknik Negeri Manado)</i>	62 - 72
Rancangan dan Pengujian Perangkat Lunak Menggunakan Web Service pada Sistem Informasi Akademik <i>Ceria Asa Malinda, Sandy Kosasi (STMIK Pontianak)</i>	73 - 86
Penerapan Jaringan Syaraf Tiruan Backpropagation Dalam Sistem Pakar Diagnosa Virus TORCH <i>Erwin Gunawan Walker, David (STMIK Pontianak)</i>	87 - 102
Pemilihan Paket Wedding Menggunakan Metode Electre <i>Endah Sri Subrata, Susanti Margaretha Kuway, I Dewa Ayu Eka Yuliani (STMIK Pontianak)</i>	103 - 114
Perancangan Perangkat Lunak Enkripsi SMS Menggunakan Algoritma RC6 Dan Rijndael Pada Smartphone <i>Benny Djoede Kristianto, Gat, Gusti Syarifudin (STMIK Pontianak)</i>	115 - 126

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga pada akhirnya Jurnal Ilmiah SISFOTENIKA Volume 10, Nomor 1, Januari 2020 dapat diterbitkan. Jurnal Ilmiah SISFOTENIKA merupakan Jurnal Keilmuan bidang Sistem Informasi dan Teknologi Informasi yang memuat tulisan-tulisan ilmiah mengenai penelitian-penelitian murni dan terapan di bidang Sistem Informasi dan Teknologi Informasi serta ulasan-ulasan penerapan ilmu di bidang terkait lainnya.

Pada terbitan kali ini, terdapat sepuluh tulisan yang dimuat pada jurnal ini oleh penulis Universitas Amikom Yogyakarta, STIMIK Sepuluh Nopember Jayapura, Universitas Nasional, Politeknik Negeri Manado dan STMIK Pontianak. Untuk kedepannya dengan semakin eksisnya jurnal ini, diharapkan banyak pihak-pihak lain baik di Kalimantan Barat ataupun dari luar daerah yang menyumbangkan tulisannya untuk diterbitkan pada Jurnal Ilmiah SISFOTENIKA ini.

Jurnal ini memuat berbagai makalah, pada makalah pertama mengulas tentang Implementasi Decision Tree Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu. Makalah kedua memuat Implementasi Hill Cipher Pada Kode Telepon dan Five Modulus Method dalam Mengamankan Pesan. Makalah ketiga mengulas tentang Sistem Pendukung Keputusan Menentukan Bibit Padi Terbaik Menggunakan Metode Gap Kompetensi. Makalah keempat memuat tentang Sistem Pakar Deteksi Dini Kesehatan Mental Menggunakan Metode Dempster-Shafer. Makalah kelima mengulas mengenai Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04. Makalah keenam memuat topik Perancangan Neur-O: Sistem Pakar Penyakit Saraf berbasis Ontologi. Makalah ketujuh mengulas tentang Rancangan dan Pengujian Perangkat Lunak Menggunakan Web Service pada Sistem Informasi Akademik. Makalah kedelapan berisikan artikel mengenai Penerapan Jaringan Syaraf Tiruan Backpropagation Dalam Sistem Pakar Diagnosa Virus TORCH. Makalah kesembilan memuat topik mengenai Pemilihan Paket Wedding Menggunakan Metode Electre. Makalah terakhir memuat topik Perancangan Perangkat Lunak Enkripsi SMS Menggunakan Algoritma RC6 Dan Rijndael Pada Smartphone.

Untuk terbitan jurnal SISFOTENIKA edisi volume 10 nomor 1 telah dilakukan beberapa penyesuaian dengan merujuk kepada hasil kesepakatan CORIS (Cooperation Computer Research Inter-University), yang beranggotakan STMIK Pontianak, STMIK Tasikmalaya, STMIK STIKOM Bali, Universitas Potensi Utama Medan, STMIK Dipanegara Makassar, Universitas Amikom Yogyakarta, Universitas Klabat Manado, Universitas Dian Nuswantoro, STMIK Raharja, STMIK Adhi Guna Palu, STMIK Sepuluh Nopember Jayapura dan Universitas Bina Nusantara yang bekerja sama dengan IndoCEISS (Indonesian Computer, Electronics and Instrumentation Support Society). Sejumlah penyesuaian yang dilakukan meliputi teknik penulisan dan penyuntingan jurnal, format halaman dan tata letak informasi untuk sebuah jurnal yang baik, penggunaan bahasa jurnal, dan manajemen jurnal.

Tak lupa dewan redaksi mengucapkan terima kasih kepada semua pihak yang telah mendukung terbitnya jurnal SISFOTENIKA Volume 10, Nomor 1 ini. Dewan redaksi berupaya menerbitkan jurnal SISFOTENIKA ini sesempurna mungkin, akan tetapi tidak tertutup kemungkinan masih banyak kekurangannya dengan demikian Dewan redaksi membutuhkan kritik dan saran dari pembaca jurnal ilmiah SISFOTENIKA untuk mendukung proses pengembangannya dan perbaikan jurnal menuju jurnal yang berkualitas.

Pontianak, Januari 2020

Dewan Redaksi Jurnal SISFOTENIKA

Implementasi Decision Tree Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu

Implementation of Decision Tree for Prediction of Student Graduation On Time

Christin Nandari Dengen^{*1}, Kusri², Emha Taufiq Luthfi³

^{1,2,3}Universitas Amikom Jl. Ring Road Utara, Condong Catur Yogyakarta, Tlp (0274) 884 201

^{1,2,3}Magister Teknik Informatika, Universitas Amikom, Yogyakarta

e-mail: ^{*1}christin.dengen@students.amikom.ac.id, ²kusri@amikom.ac.id,

³emhataufiqluthfi@amikom.ac.id

Abstrak

Fakultas Ilmu Komputer dan Teknologi Informasi (FKTI) merupakan salah satu fakultas baru dalam lingkup Universitas Mulawarman. Mahasiswa yang diterima setiap tahun semakin meningkat, namun tidak semua mahasiswa dapat lulus tepat waktu. Dalam penelitian ini, dengan memanfaatkan data mining dapat mendukung dan membantu mahasiswa untuk memprediksi kelulusan. Adapun tujuan dari penelitian ini adalah untuk membantu mahasiswa untuk memprediksi kelulusan agar dapat meminimalisir mahasiswa yang tidak lulus tepat waktu. Dalam prediksi kelulusan mahasiswa tepat waktu menggunakan data kelulusan mahasiswa tahun 2013 dengan menggunakan atribut jenis kelamin, ipk, predikat kelulusan dan toefl. Penelitian ini, mengimplementasikan decision Tree untuk prediksi kelulusan mahasiswa yang didukung dengan simulasi menggunakan aplikasi WeKa untuk menghitung dan mendapatkan nilai akurasi. Dari hasil simulasi WeKa, maka didapatkan tingkat akurasi penggunaan decision tree terhadap prediksi kelulusan mahasiswa sebesar 60%. Dengan adanya penelitian ini, diharapkan dapat membantu pihak Program Studi Teknik Informatika Universitas Mulawarman agar membimbing mahasiswa sehingga dapat lulus tepat waktu.

Kata kunci— *Prediksi Kelulusan, Data Mining, Decision Tree*

Abstract

The Faculty of Computer Science and Information Technology (FKTI) is one of the new faculties within the scope of Mulawarman University. Students are accepted every year, but not all students can graduate on time. In this research, utilizing data mining can support and help students to predict graduation. The purpose of this study is to help students predict graduation in order to minimize students who do not graduate on time. In predicting student graduation on time using 2013 student graduation data using the attributes of gender, GPA, graduation predicate and TOEFL. This study implemented a decision tree for student graduation predictions supported by simulations using the WeKa application to calculate and get accuracy values. From the results of WeKa simulation, the accuracy of decision tree use on student graduation predictions is 60%. With this research, it is expected to help the Mulawarman University Informatics Engineering Study Program to guide students so they can graduate on time.

Keywords— *Graduation Prediction, Data Mining, Decision Tree.*

1. PENDAHULUAN

Mahasiswa salah satu parameter penting dalam sebuah evaluasi program studi[1]. Kehadiran mahasiswa, prestasi yang dicapai mahasiswa dan profil kelulusan seharusnya mendapatkan perhatian yang serius. Mahasiswa yang diterima setiap tahun semakin banyak , namun tidak sedikit yang dapat menyelesaikan studi dengan tepat waktu[1].

Dalam mencapai sebuah kelulusan, ada tahapan atau proses yang harus dilalui oleh setiap mahasiswa seperti menyelesaikan sejumlah mata kuliah, kuliah kerja nyata, praktik kerja lapangan maupun seminar tugas akhir. Tahapan atau proses tersebut dilakukan dalam waktu yang telah ditentukan oleh pihak universitas [1].

Data tentang mahasiswa yang lulus terus bertambah di tiap tahunnya dan menumpuk seperti data yang terabaikan karena jarang digunakan[2]. Dalam system pendidikan mahasiswa merupakan aset penting dalam sebuah institusi pendidikan, untuk itu perlu diperhatikan tingkat kelulusan mahasiswa tepat waktu. Prediksi kelulusan mahasiswa dapat digunakan lebih lanjut untuk membantu universitas maupun fakultas dalam mengevaluasi dan memperbaiki system pembelajaran sehingga universitas dapat menghasilkan lulusan yang berkualitas[2]

Dari latar belakang tersebut, adapun tujuan dari penelitian ini yaitu menghasilkan penelitian yang membentuk *Decision Support System (DSS)*. Hasil dari analisa diterapkan sebagai dasar perancangan pada system pengambil keputusan untuk mengidentifikasi dan mengklasifikasi kelulusan mahasiswa tepat waktu dan tidak tepat waktu. Manfaat dari penelitian ini yaitu diharapkan dengan adanya penelitian ini dapat menjadi bahan pembelajaran bagi pihak fakultas dalam membimbing mahasiswa agar dapat lulus tepat waktu.

Data terkait mahasiswa yang lulus dapat memberikan informasi yang berguna bagi universitas maupun program studi jika dimanfaatkan secara maksimal [1]. Salah satu cara memanfaatkan data tentang mahasiswa yang lulus yaitu dengan mengolah menjadi *data mining*[2]. Data mining merupakan sebuah proses untuk mendapatkan informasi yang berguna dari dari gudang basis data besar[3]. Proses *data mining* dapat ditemukan pola atau aturan yang dapat digunakan untuk menghasilkan suatu informasi seperti prediksi kelulusan mahasiswa[2].

Pemilihan metode sangat berpengaruh dalam melakukan tahap penelitian. *CRISP-DM* menyediakan standar proses baku untuk *data mining* yang dapat diterapkan ke dalam strategi pemecahan masalah umum pada bisnis atau pada unit penelitian[4]. Prediksi kelulusan telah banyak diterapkan dalam beberapa penelitian, salah satunya pada jurnal “*Matching Preprocessing Methods for Improving the Prediction of Student's Graduation*”. Penelitian [5] membahas prediksi kelulusan siswa untuk membantu dalam meningkatkan kualitas system pendidikan dengan membandingkan dua metode preprocessing yaitu algoritma SMOTE dan relief. Dalam penelitian ini menggunakan 544 data set yang diperoleh dari system informasi pendaftaran pada Universitas Teknologi Bangkok. Data set di kelola dengan empat pembelajaran berbasis aturan (DT, OneR, PART dan DTNB), dengan hasil percobaan bahwa DTNB memberikan peningkatan presisi, daya ingat ukuran-f dan rata-rata g dibandingkan dengan metode lain.

Decision tree merupakan salah satu algoritma yang dapat digunakan dalam pembentukan pohon keputusan[6]. Dalam penelitian ini menggunakan *decision tree*, karena sesuai kebutuhan dimana hasil akhir berupa keputusan terhadap prediksi kelulusan mahasiswa. *Decision tree* juga memiliki tingkat akurasi yang lebih tinggi saat diaplikasikan untuk jumlah data yang besar dibandingkan algoritma pohon keputusan yang lain[6] dan dapat menghasilkan pohon keputusan yang mudah diinterpretasikan[7].

Dalam penelitian ini akan membahas aspek yang berpengaruh dalam prediksi kelulusan mahasiswa tepat waktu dengan mengimplementasikan *decision tree* dan dilakukan pengujian untuk mengukur tingkat akurasi menggunakan *confusion matrix* pada aplikasi WeKa.

2. METODE PENELITIAN

2.1 Prosedur Pengumpulan Data

Dataset yang digunakan diperoleh berdasarkan data primer yang didapatkan langsung dari narasumber[8]. Narasumber dalam penelitian ini adalah Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Mulawarman dan bidang kemahasiswaan dan alumni untuk mendapatkan data kelulusan dari *database*.

Tabel 1. Contoh Dataset Kelulusan Mahasiswa

Nama	Jenis Kelamin	IPK	Predikat	TOEFL	Keterangan
Dwi Kinasih Widiyati	P	>3.5	Cum Laude	>400	TW
Fajar Maitari	P	>3.5	Sangat Memuaskan	>400	TTW
Ketaren, Muhammad Hafizh	L	>3.5	Sangat Memuaskan	<400	TTW
Hadriana Agus Shifa	L	>3.5	Sangat Memuaskan	>400	TTW
Abraham Ody Watulingas	L	>3.5	Sangat Memuaskan	>400	TW
Azriana Sari	P	>3.5	Sangat Memuaskan	>400	TTW
Randi Saputra	P	<3.5	Sangat Memuaskan	>400	TTW
Danny Agniawan	L	<3.5	Sangat Memuaskan	<400	TTW
Lara Silvia	L	>3.5	Sangat Memuaskan	<400	TTW

2.2 Desain Penelitian

Penulis mengaplikasikan model standarisasi data mining yaitu *CRISP-DM* (*Cross Industry Standar Process for Data Mining*), dengan beberapa tahapan sebagai berikut[4]:

1) Pemahaman Bisnis

Penulis mengedepankan penentuan tujuan proyek dan kebutuhan lengkap dalam memahami proses bisnis atau tujuan bisnis mengenai penelitian tersebut. Setelah itu, penelitian tersebut mampu mengolah sebuah data yang didapatkan dari program studi teknik informatika universitas mulawarman menjadi sebuah formula atau rules.

2) Pemahaman Data

Penelitian ini menggunakan sumber data primer, yaitu data-data yang didapatkan dari *database* FKTI Universitas Mulawarman dan data sekunder, yaitu data-data yang didapatkan secara tidak langsung berupa buku, jurnal dan lain sebagainya.

3) Pengolahan Data

Pengolahan data berlangsung dari data mentah menjadi data bentuk hasil normalisasi. Berikut penjelasan dalam tahapan pengolahan data.

- Data cleaning and data selection*, merupakan tahapan awal dalam memproses *data mining*. Kegiatan ini bertujuan untuk membuang atribut yang tidak dibutuhkan agar lebih efisien.
- Data integration and data transformation*, merupakan tahap selanjutnya dalam proses *data mining* yaitu memindahkan *database* dan memproses transformasi data. Pada data kelulusan mahasiswa, data dari file yang berbeda seperti data kelulusan tahun 2013 disusun menjadi satu kesatuan dalam file.
- Data relection*, yaitu penggunaan data sesuai dengan atribut dan jumlah *record* yang dibutuhkan. Jadi, terdapat beberapa bagian *field* yang perlu dihilangkan karena tidak lengkapnya data atau kurang relevan dalam penelitian.

4) Pemodelan

Tahap awal dilakukan pemrosesan data set seperti, konversi data dan *replacing data*, kemudian dilakukan pemodelan menggunakan algoritma C4.5.

5) Evaluasi

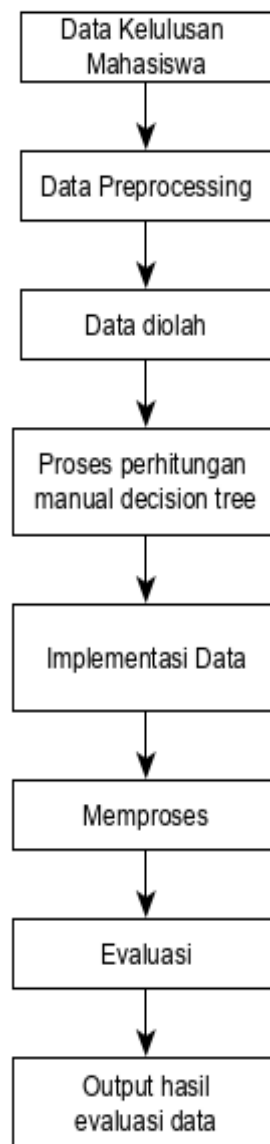
Dilakukan validasi, yaitu pengukuran akurasi dari hasil sebuah pemodelan algoritma yang digunakan dengan memanfaatkan *tools WeKa*.

6) Penyebaran

Penulis menghasilkan penelitian yang membentuk *Decision Support System (DSS)*. Hasil dari analisa diterapkan sebagai dasar perancangan pada system pengambil keputusan untuk mengidentifikasi dan mengklasifikasi kelulusan mahasiswa tepat waktu dan tidak tepat waktu.

2.3 Metode yang diusulkan

Desain metode pada penelitian ini menggunakan *decision tree* untuk melakukan perhitungan klasifikasi dan prediksi.



Gambar 1. Metode yang diusulkan

Berdasarkan gambar 1, maka langkah-langkah metode dapat dijelaskan sebagai berikut:

- Data yang akan diimplementasikan sebagai data penelitian adalah data kelulusan pada tahun 2013.
- Data diolah dengan cara pre-processing, yaitu dengan seleksi data, transformasi data dan reduksi data. Pengolahan tersebut dilakukan supaya data terhindar dari *missing value* (data yang kurang lengkap).

- c. Tahap selanjutnya setelah dilakukan data pre-processing, maka akan menghasilkan data *training* yang siap diolah di tahap berikutnya.
- d. Pada tahap ini dilakukan perhitungan manual dengan mulai menghitung *entropy* dan *gain* berdasarkan data pre-processing.
- e. Hasil dari perhitungan tersebut diselaraskan dengan hasil yang dilakukan dalam *tools WeKa*.
- f. Dalam *tools* tersebut dilakukan proses mengolah data sesuai dengan algoritma yang digunakan yaitu algoritma C4.5.
- g. Dilakukan simulasi untuk melihat presentase akurasi data yang di dapat dari algoritma tersebut.
- h. Ouput akan memperlihatkan hasil dari pengolahan data tersebut dalam aplikasi *WeKa*.

3. HASIL DAN PEMBAHASAN

Untuk melakukan pengujian model tersebut diuji pada data kelulusan mahasiswa tahun 2013. Data tersebut akan dibuat pohon keputusan untuk menentukan aspek yang berpengaruh dalam prediksi kelulusan mahasiswa.

3.1 Perhitungan Manual

Berikut perhitungan manual prediksi kelulusan mahasiswa tepat waktu di Program Studi Teknik Informatika Universitas Mulawarman.

1. Mengambil sampel data acak

Sampel data acak digunakan sebagai data awal untuk melakukan perhitungan manual decision tree dengan menggunakan data kelulusan tahun 2013 pada Program Studi Teknik Informatika Universitas Mulawarman.

Tabel 2. Data Training Tahap 1

Nama	JK	IPK	Masa Studi	Predikat	TOEFL	Ket
Dwi Kinasih Widiyati	P	> 3,5	4 tahun	Cum Laude	> 400	TW
Fajar Maitari	P	> 3,5	> 4 tahun	Sangat Memuaskan	> 400	TTW
Ketaren, Muhammad Hafizh	L	> 3,5	> 4 tahun	Sangat Memuaskan	< 400	TTW
Hadriana Agus Shifa	L	> 3,5	> 4 tahun	Sangat Memuaskan	> 400	TTW
Abraham Ody Watulingas	L	> 3,5	4 tahun	Sangat Memuaskan	> 400	TW
Azriana Sari	P	> 3,5	> 4 tahun	Sangat Memuaskan	> 400	TTW
Randi Saputra	P	< 3,5	> 4 tahun	Sangat Memuaskan	> 400	TTW
Danny Agniawan	L	< 3,5	> 4 tahun	Sangat Memuaskan	< 400	TTW
Lara Silvia	L	> 3,5	> 4 tahun	Sangat Memuaskan	< 400	TTW
Umayatul Choerohnur	P	> 3,5	4 Tahun	Sangat Memuaskan	> 400	TW

2. Menentukan entropy tahap 1

Setelah menentukan data awal, langkah selanjutnya yaitu melakukan perhitungan dengan mencari nilai entropy dari masing-masing data, menggunakan rumus entropy [9] :

$$\text{Entropy (S)} = \sum_{i=1}^k -p_i * \log_2 p_i \dots\dots\dots(1)$$

Berdasarkan rumus (1) maka dapat dilakukan perhitungan seperti :

$$\begin{aligned} & \left(- \left(\frac{26}{50} \right) * \log_2 \left(\frac{26}{50} \right) + \left(- \left(\frac{24}{50} \right) * \log_2 \left(\frac{24}{50} \right) \right) \right) = 0.998845536 \\ \text{Jenis Kelamin (P)} & = 0.721928095 \\ \text{Jenis Kelamin (L)} & = 0.918295834 \\ \text{IPK (>3.5)} & = 0.877962001 \\ \text{IPK (<3.5)} & = 0 \end{aligned}$$

Predikat Cumlaude = 0
 Predikat Sangat M = 0.979868757
 Predikat Memuaskan = 0

3. Menentukan gain tahap 1

Setelah melakukan perhitungan entropy, langkah selanjutnya yaitu melakukan perhitungan nilai gain dari masing-masing atribut.

$$\text{Gain (A)} = \sum_{i=1}^k \frac{|S_i|}{|S|} \times \text{Entropi}(S_i) \dots\dots\dots(2)$$

Berdasarkan rumus gain, maka diperoleh perhitungan sebagai berikut :

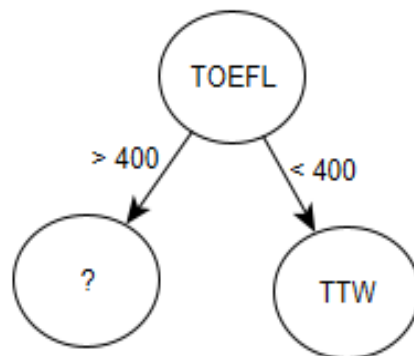
Gain (Jenis Kelamin)
 0.99884553 – ((20/50*0.721928095)+(30/50*0.918295834)) = 0.159097
 Gain IPK
 0.99884553 – ((37/50*0.877962001)+(13/50*0)) = 0.349154
 Gain Predikat
 0.99884553 – ((11/50*0)+(36/50*0.979868757)+(3/50*0)) = 0.29334
 Gain Toefl
 0.99884553 – ((34/50*0.787126586)+(16/50*0)) = 0.463599

Setelah ditentukan data awal, langkah selanjutnya yaitu melakukan perhitungan dengan mencari nilai entropy dan gain dalam sebuah table agar lebih rinci dan jelas.

Tabel 3. Node Tree Tahap 1

Atribut	Nilai atribut	Total	TW	TTW	Entropy	Gain
		50	26	24	0.998845536	
Jenis Kelamin						0.159096798
	P	20	16	4	0.721928095	
	L	30	10	20	0.918295834	
IPK						0.349153655
	> 3,5	37	26	11	0.877962001	
	< 3,5	13	0	13	0	
Predikat						0.293340031
	Cumlaude	11	11	0	0	
	Sangat M	36	15	21	0.979868757	
	Memuaskan	3	0	3	0	
TOEFL						0.463599457
	> 400	34	26	8	0.787126586	
	< 400	16	0	8	0	

Berdasarkan tabel 3, dapat dilihat bahwa nilai entropy dan gain sudah sesuai aturan antar 0 sampai 1. Setelah dilakukan perhitungan nilai entropy dan gain, maka didapatkan hasil gain tertinggi sebagai akar yaitu nilai gain dari TOEFL sebesar 0,463599457. Hasil pohon keputusan sebagai berikut:



Gambar 2. Pohon Keputusan Node 1

4. Mengambil Data Training Tahap 2

Setelah dilakukan tahap 1 maka, langkah selanjutnya yaitu melakukan perhitungan ulang dengan menggunakan data baru dengan atribut TOEFL dengan nilai >400 karena memiliki nilai gain terbesar pada perhitungan sebelumnya.

Tabel 4. Data Training Tahap 2

Nama	JK	IPK	Predikat	TOEFL	Ket
Dwi Kinasih Widiyati	P	> 3,5	Cum Laude	> 400	TW
Fajar Maitari	P	> 3,5	Sangat Memuaskan	> 400	TTW
Hadriana Agus Shifa	L	> 3,5	Sangat Memuaskan	> 400	TTW
Abraham Ody Watulingas	L	> 3,5	Sangat Memuaskan	> 400	TW
Azriana Sari	P	> 3,5	Sangat Memuaskan	> 400	TTW
Randi Saputra	P	< 3,5	Sangat Memuaskan	> 400	TTW
Umayatul Choerohnur	P	> 3,5	Sangat Memuaskan	> 400	TW

5. Menentukan Entropy Tahap 2

Perhitungan entropy tahap 2 sebagai berikut :

$$\left(-\left(\frac{25}{34}\right) * \log_2\left(\frac{25}{34}\right) + \left(-\left(\frac{9}{34}\right) * \log_2\left(\frac{9}{34}\right)\right)\right) = 0.833764907$$

Jenis Kelamin (P)	=	0.543564443
Jenis Kelamin (L)	=	0.981940787
IPK (>3.5)	=	0.757878463
IPK (<3.5)	=	0
Predikat Cumlaude	=	0
Predikat Sangat Memuaskan	=	0.954434003

6. Menentukan Gain Tahap 2

Setelah melakukan perhitungan entropy, dilanjutkan dengan menghitung nilai gain dari masing-masing atribut, sebagai berikut :

$$\text{Gain Jenis Kelamin} = 0.833764907 - ((16/34 * 0.543564443) + (19/34 * 0.981940787)) = 0.029238$$

$$\text{Gain IPK} = 0.833764907 - ((32/34 * 0.757878463) + (2/34 * 0)) = 0.120468$$

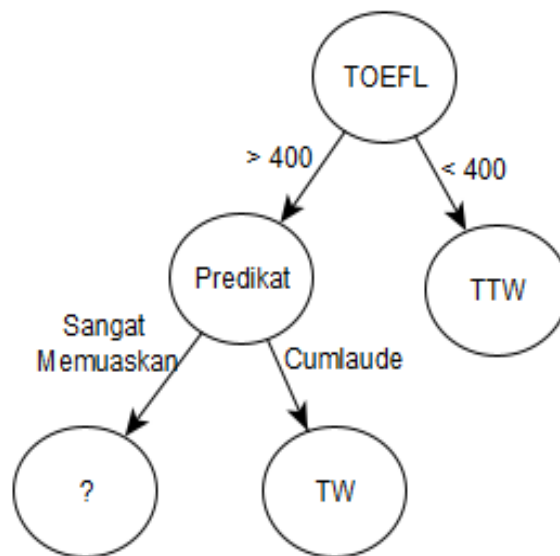
$$\text{Gain Predikat} = 0.833764907 - ((10/34 * 0) + (24/34 * 0.954434003)) = 0.160047$$

Agar lebih memudahkan untuk membaca perhitungan entropy dan gain, disajikan seperti pada tabel 5.

Tabel 5. Nilai Entropy dan Gain Tahap 2

Atribut	Nilai Atribut	Total	TW	TTW	Entropy	Gain
TOEFL	> 400	34	25	9	0.833764907	
Jenis Kelamin						0.029238259
	P	16	14	2	0.543564443	
	L	19	11	8	0.981940787	
IPK						0.120467531
	> 3,5	32	25	7	0.757878463	
	< 3,5	2	0	2	0	
Predikat						0.160046787
	Cumlaude	10	10	0	0	
	Sangat Memuaskan	24	15	9	0.954434003	

Berdasarkan tabel 5, dapat diketahui bahwa nilai gain tertinggi ada pada atribut predikat dengan nilai 0,160046787. Maka yang akan digunakan sebagai acuan untuk perhitungan selanjutnya adalah predikat dengan nilai sangat memuaskan. Atribut predikat juga digunakan sebagai lanjutan dari pohon keputusan pada gambar 3 sebagai berikut.



Gambar 3. Pohon Keputusan Node 2

7. Menentukan Data Training Tahap 3

Setelah didapatkan pohon keputusan sampai pada tahap 2, maka kembali dilakukan perhitungan ulang untuk menentukan data baru berdasarkan hasil sebelumnya, seperti pada:

Tabel 6. Data Training Tahap 3

Nama	JK	IPK	Predikat	TOEFL	Ket
Fajar Maitari	P	> 3,5	Sangat Memuaskan	> 400	TTW
Hadriana Agus Shifa	L	> 3,5	Sangat Memuaskan	> 400	TTW
Abraham Ody Watulingas	L	> 3,5	Sangat Memuaskan	> 400	TW
Azriana Sari	P	> 3,5	Sangat Memuaskan	> 400	TTW
Umayatul Choerohnur	P	> 3,5	Sangat Memuaskan	> 400	TW

8. Menentukan Nilai Entropy Tahap 3
Perhitungan entropy dilakukan berdasarkan data training tahap 3.

$$\left(-\left(\frac{15}{24}\right) * \log_2\left(\frac{15}{24}\right) + \left(-\left(\frac{9}{24}\right) * \log_2\left(\frac{9}{24}\right)\right)\right) = 0.954434003$$

- Jenis Kelamin (P) = 0.918295834
- Jenis Kelamin (L) = 0.979868757
- IPK (>3.5) = 0
- IPK (<3.5) = 0

9. Menentukan Nilai Gain Tahap 3
Setelah melakukan perhitungan entropy, dilanjutkan perhitungan gain tahap 3 seperti dibawah ini :

Gain Jenis Kelamin
 $0.954434003 - ((12/24 * 0.918295834) + (12/24 * 0.979868757)) = 0.005351708$

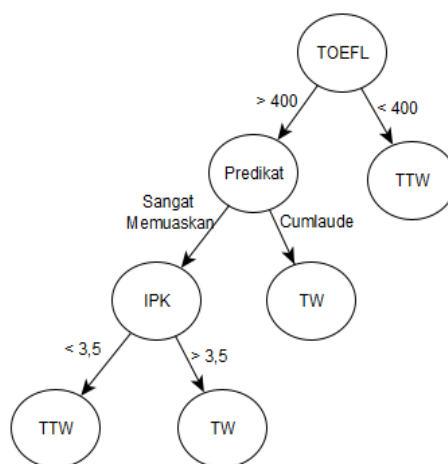
Gain IPK
 $0.954434003 - ((22/24 * 0) + (2/24 * 0)) = 0.954434003$

Agar lebih memudahkan untuk membaca perhitungan entropy dan gain, disajikan seperti pada tabel 7.

Tabel 7. Nilai Entropy dan Gain Tahap 3

Atribut	Nilai atribut	Total	TW	TTW	Entropy	Gain
Predikat	sangat memuaskan	24	15	9	0.954434003	
Jenis Kelamin						0.005351708
	P	12	8	4	0.918295834	
	L	12	7	5	0.979868757	
IPK						0.954434003
	>3,5	22	22	0	0	
	< 3,5	2	0	2	0	

Berdasarkan tabel 7, dapat diketahui bahwa nilai gain tertinggi yaitu IPK dengan nilai 0,954434003. Maka atribut IPK dapat dijadikan node selanjutnya, dimana jika IPK < 3,5 akan lulus TTW dan jika IPK > 3,5 akan lulus TW. Berikut hasil pohon keputusan yang dibentuk pada gambar 4.



Gambar 4. Pohon Keputusan

Gambar 4 merupakan hasil akhir dari pohon keputusan setelah dilakukan perhitungan algoritma C4.5. Dari gambar 4 dapat dilihat bahwa TOEFL < 400 maka lulus TTW, namun jika TOEFL > 400 maka akan masuk dalam node predikat dimana jika predikatnya *cumlaude* maka akan lulus TW, jika predikat sangat memuaskan akan masuk dalam node IPK. Jika IPK mahasiswa > 3,5 maka lulus TW, namun jika < 3,5 maka lulus TTW.

3.2 Pengujian Tingkat Akurasi Algoritma C4.5

Percobaan yang dilakukan pada implementasi algoritma C4.5 dalam penelitian ini menggunakan aplikasi WeKa. Data yang diolah menggunakan data tahun kelulusan 2013. Atribut yang digunakan yaitu jenis kelamin, ipk, predikat kelulusan, dan toefl. Dari hasil pengujian algoritma C4.5 pada aplikasi WeKa, didapatkan tingkat keberhasilan 100% dan error 0%, seperti pada gambar 5.

Correctly Classified Instances	5	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.1448		
Root mean squared error	0.187		
Relative absolute error	29.0893	%	
Root relative squared error	37.5459	%	
Total Number of Instances	5		

Gambar 5. Tingkat Keberhasilan

Berkaitan dengan data training yaitu data kelulusan tahun 2013 yang berisi atribut jenis kelamin, ipk, predikat dan toefl, kemudian dilakukan pemodelan menggunakan *decision tree*. Selanjutnya dilakukan pengujian tingkat akurasi menggunakan *confusion matrix*. *Confusion Matrix* menggunakan tabel matriks untuk melakukan pengujian dan dataset hanya terdiri dari dua kelas yaitu positif dan negative[10]. Berdasarkan pemodelan tersebut diperoleh hasil dalam bentuk *confusion matrix* pada gambar 6.

```

=== Confusion Matrix ===
      a b  <-- classified as
      3 0 | a = TW
      0 2 | b = TTW
    
```

Gambar 6. *Confusion Matrix*

Didapatkan hasil dari *confusion matrix* yaitu untuk jumlah *true positif* sebanyak 3, jumlah *true negative* sebanyak 0, jumlah *false positif* 0 dan jumlah *false negative* sebanyak 2. Dari hasil tersebut dapat dianalisis bahwa nilai akurasi prediksi kelulusan mahasiswa tepat waktu menggunakan perantara aplikasi WeKA sebesar 60%. Adapun perhitungan manual menentukan akurasinya adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% \\
 &= \frac{(3+0)}{(3+0+0+2)} * 100\% \\
 &= \frac{3}{5} * 100\% \\
 &= 0.6 * 100\% \\
 &= 60\%
 \end{aligned}$$

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan bahwa:

1. *Decision tree* dapat diterapkan untuk prediksi kelulusan mahasiswa tepat waktu di Program Studi Teknik Informatika Universitas Mulawarman dengan tingkat akurasi sebesar 60%.
2. Penelitian ini dapat membantu pihak program studi untuk memprediksi kelulusan mahasiswa tepat waktu.

5. SARAN

Berdasarkan hasil penelitian ini, maka peneliti menyarankan beberapa hal untuk penelitian selanjutnya yaitu menggunakan lebih banyak data kelulusan agar tingkat akurasi dapat lebih tinggi dan dapat menambahkan atribut terkait kelulusan mahasiswa tepat waktu.

DAFTAR PUSTAKA

- [1] M. Ridwan, "Sistem Rekomendasi Proses Kelulusan Mahasiswa Berbasis Algoritma Klasifikasi C4.5," *J. Ilm. Inform.*, vol. 2, no. 1, pp. 105–111, 2017.
- [2] R. Putri and I. Waspada, "Penerapan Algoritma C4 . 5 pada Aplikasi Prediksi Kelulusan Mahasiswa Prodi Informatika," *Khazanah Inform.*, vol. 4, no. 1, pp. 1–7, 2018.
- [3] P. M. Hasugian, "Penguujian Algoritma Apriori Dengan Aplikasi Weka Dalam," *J. Mantik Penusa*, vol. 1, no. 2, pp. 98–103, 2017.
- [4] A. P. Fadillah, "Penerapan Metode CRISP-DM untuk Prediksi Kelulusan Studi Mahasiswa Menempuh Mata Kuliah (Studi Kasus Universitas XYZ)," *J. Tek. Inform. dan Sist. Inf.*, vol. 1, no. 3, pp. 260–270, 2018.
- [5] P. D. Prachuabsupakij Wanthanee, "Matching Preprocessing Methods for Improving the Prediction of Student ' s Graduation Wanthanee Prachuabsupakij *, Pafan Doungpaison," pp. 33–37, 2016.
- [6] Kusrini and E. T. Luthfi, *Algoritma Data Mining*. ANDI OFFSET, 2009.
- [7] D. H. Kamagi and S. Hansun, "Implementasi Data Mining dengan Algoritma C4.5 untuk Memprediksi Tingkat Kelulusan Mahasiswa," *J. Ultim.*, vol. 6, no. 1, pp. 15–20, 2014.
- [8] N. Z. Rahma and A. Setyono, "Penerapan Algoritma C4.5 Dalam Memprediksi Kesiapan Siswa SMP IT PAPB Semarang Menghadapi Ujian Nasional," *Sisfotenika*, vol. 1, 2018.
- [9] R. A. Raharjo, "Kajian Komparasi Penereapan Algoritma C4.5 Neural Network dan SVM Dengan Teknik PSO Untuk Pemilihan Karyawan Teladan PT.XYZ," *J. String*, vol. 1, no. 3, pp. 345–356, 2017.
- [10] R. Geometry and G. Analysis, "Penerapan Algoritma C4.5 Pada Program Klasifikasi Mahasiswa Dropout," pp. 139–147.

Implementasi Hill Cipher Pada Kode Telepon dan Five Modulus Method dalam Mengamankan Pesan

Implementation of Hill Cipher in Telephone Code and Five Modulus Method for Securing Messages

Patmawati Hasan^{*1}, Selviana Yunita², Dony Ariyus³

¹ STIMIK Sepuluh Nopember Jayapura, ^{2,3} Universitas Amikom Yogyakarta

¹Jl. Ardipura II No. 22 B Polimak – Jayapura, Tlp (0967) 533400, ²Jl. Ring Road Utara, Yogyakarta, Tlp (0274) 884201

e-mail: ¹patmawatihasan@gmail.com, ²selviana.yunita.ax@gmail.com, ³dony.a@amikom.ac.id

Abstrak

Perkembangan teknologi yang canggih di era ini semakin pesat, pesan-pesan yang disembunyikan atau dirahasiakan dari orang-orang yang ingin mengakses pesan rahasia tersebut perlu dijaga keamanan dengan cara mengubah pesan tersebut menjadi sandi rahasia dan menyisipkan menggunakan teknik kriptografi dan steganografi. Sistem keamanan yang dibangun menggunakan kombinasi algoritma Hill cipher dan Five Modulus Method dengan tujuan membuat keamanan yang kuat dan berlapis tanpa mengurangi atau merusak pesan teks pada gambar. Sistem yang dibangun menggunakan bahasa pemrograman Matlab. Algoritma yang digunakan memiliki kelebihan dalam mengenkripsi data berdasarkan penelitian-penelitian terdahulu yang berhasil menggunakan teknik kriptografi dan steganografi. Hill Cipher menggunakan matriks berukuran ($m \times m$) sebagai kunci dalam mengenkripsi dan dekripsi data pada kode telepon yang telah dimodifikasi dengan tujuan meningkatkan kemananan. Sedangkan untuk steganografi menggunakan Five Modulus Method dengan memeriksa seluruh piksel dalam blok ($k \times k$) dan mengubah setiap piksel menjadi sebuah angka yang habis dibagi 5. Hasil dari penelitian ini berupa sistem keamanan yang menggunakan Kombinasi Hill Cipher dan Five Modulus Method dalam Mengamankan Pesan Teks. Berdasarkan hasil tingkat akurasi yang dilakukan pengujian sebanyak lima (5) kali terhadap plaintext yang berbeda didapatkan tingkat akurasi sebesar 100% yaitu tidak ditemukan perbedaan. Selanjutnya dilakukan pengembangan agar tidak hanya dapat enkripsi text saja, tetapi dapat enkripsi file atau media lainnya.

Kata kunci— Hill cipher, Five Modulus Method, Kode Telepon

Abstract

The development of sophisticated technology in this era is increasingly rapid, messages that are hidden or kept secret from people who want to access these secret messages need to be kept safe by changing the message into a secret password and inserting it using cryptographic and steganographic techniques. The security system is built using a combination of the Hill cipher algorithm and the Five Modulus Method with the aim of creating a strong and layered security without reducing or damaging text messages in the image. The system was built using the Matlab programming language. The algorithm used has the advantage of encrypting data based on previous studies that have successfully used cryptographic and steganographic

techniques. Hill Cipher uses a matrix of size $(m \times m)$ as a key in encrypting and decrypting data in a modified telephone code with the aim of increasing security. Whereas for steganography using the Five Modulus Method by examining all pixels in a block $(k \times k)$ and converting each pixel into a number that has been divided 5. The results of this research are a security system that uses a combination of Hill Cipher and Five Modulus Method in Securing Text Messages. Based on the results of the accuracy level which was tested five times (5) times for different plaintext obtained an accuracy level of 100% that is not found differences. Further development is carried out so that it can not only encrypt text, but can encrypt files or other media.

Keywords— *Cryptography, Steganography, Hill cipher, Five Modulus Method*

1. PENDAHULUAN

Dalam kecanggihan teknologi saat ini, proses berkirim pesan menjadi jauh lebih cepat dan mudah. Namun, perkembangan ini juga berdampak pada masalah keamanan, dimana informasi yang dikirimkan rentan mengalami pencurian, pembajakan, serta diketahui oleh pihak yang tidak bertanggung jawab. Menghadapi masalah tersebut, diperlukan suatu metode agar informasi yang dikirimkan hanya dapat diterima oleh pihak yang memiliki kepentingan. Salah satu metode pengamanan pesan adalah melalui kriptografi dan steganografi.

Kriptografi adalah suatu cabang ilmu yang mempelajari bagaimana informasi yang dikirimkan tetap aman dan terjaga kerahasiaannya. Dua konsep utama kriptografi adalah enkripsi dan dekripsi [1]. Sedangkan steganografi adalah sebuah tindakan menyembunyikan pesan rahasia didalam suatu media, dengan sebagian besar sistem memanfaatkan kelemahan perseptual manusia. Steganografi seringkali disalah artikan dengan kriptografi dikarenakan keduanya sama-sama digunakan untuk melindungi informasi rahasia. Namun, jika keduanya digunakan secara bersamaan akan menggandakan perlindungan terhadap informasi [2]. Perbedaan utama dari kriptografi dan steganografi adalah kriptografi berfokus dalam menjaga isi pesan tetap rahasia, sedangkan steganografi berfokus dalam menjaga keberadaan pesan rahasia [3]. Berdasarkan pendapat diatas, maka penting untuk menggunakan kombinasi kriptografi dan steganografi demi memperoleh perlindungan terhadap informasi yang lebih optimal.

Dalam penelitiannya, Qazi menawarkan modifikasi algoritma *hill cipher* dengan melakukan penambahan transposisi, substitusi, serta pemindahan fungsi kiri dan kanan. Transposisi dan Substitusi plaintext dilakukan pada masing-masing matrik $n \times n$. Setelah itu dilakukan pemindahan fungsi kiri dan kanan pada proses enkripsi. Penelitian dilakukan dengan menggunakan Matlab. Hasil penelitian menunjukkan jika algoritma yang ditawarkan memiliki hasil output yang lebih baik dalam hal ukuran file ketika dibandingkan dengan algoritma tradisional *hill cipher* [4].

Penelitian yang dilakukan oleh Prasad melakukan modifikasi pada matrik hill cipher dan affine hill cipher dengan menggunakan modulasi bilangan prima. Persamaan linear yang homogen dan bilangan prima digunakan menghasilkan matrik self-invertible yang dapat mengurangi kompleksitas komputasi pada saat dekripsi. Penelitian menunjukkan hasil bahwa algoritma ini cocok untuk gambar hitam putih dan berwarna [5].

Penelitian menggunakan Five Modulus Method dalam menyembunyikan pesan rahasia pada sebuah gambar. Keuntungan utama dari dari algoitma ini adalah menjaga ukuran gambar tetap konstan ketika pesan rahasia meningkatkan ukuran gambar. Pengujian dilakukan dengan cara menangkap sinyal *noise* puncak dalam suatu gambar. Diketahui berdasarkan nilai PSNR, gambar yang memiliki pesan rahasia memiliki nilai PSNR lebih tinggi. Algoritma ini sangat efisien untuk menyembunyikan data didalam gambar [6].

Alkadi melakukan penelitian untuk keamanan hybrid steganografi pada private cloud platform, dengan memanfaatkan algoritma Five Modulus Method. Pada algoritma yang ditawarkan, dilakukan proses *embedded* steganografi dan *cloud computing*, sehingga dapat menghasilkan standar keamanan yang lebih baik. Dilakukan proses *inverted* pada piksel dengan menggunakan *Five Modulus Method*. Menggunakan Software as a Service (SaaS) Document Management, gambar disimpan dan dibagikan kepada penerima yang akan mengurangi proses seperti *upload* dan *download*. Dengan adanya SaaS dapat lebih menghemat biaya, keamanan jaminan, serta lebih mudah terukur dalam proses berbagi pada *cloud computing* [7].

Ariyus melakukan penelitian dengan mengembangkan dari penelitian terdahulu berupa kombinasi dari 4 teknik substitusi, salah satunya dengan menggunakan algoritma *hill cipher*. Cipher text kemudian disembunyikan dalam suatu gambar dengan algoritma *Least Significant Bit (LSB)*. Penelitian ini memberikan penambahan sebelum proses substitusi standar sehingga meningkatkan tingkat keamanan meskipun enkripsi dilakukan dengan teknik yang sederhana. Hasil penelitian ini menunjukkan jika penelitian ini hanya dapat diterapkan pada media sosial seperti Google+, Telegram, dan Google Drive karena terjadinya perubahan terhadap ukuran gambar. Metode yang diusulkan tidak dapat digunakan pada media sosial seperti Instagram, Facebook, WhatsApp, dan Pinterest karena media sosial tersebut otomatis merubah ukuran gambar menjadi lebih kecil dari ukuran sebenarnya [8].

Penelitian lain mengenai modifikasi *Hill Cipher* dilakukan oleh Qasem. *Hill Cipher* sendiri merupakan kriptografi klasik yang berdasarkan pada aljabar linear dengan transformasi linear sederhana menggunakan matrik. Proses enkripsi dan dekripsi pada *hill cipher* menggunakan perkalian matrik yang berpotensi menghabiskan waktu, dan membuat hal ini menjadi salah satu masalah yang banyak dipelajari. Dalam penelitian ini, Qasem mengimplementasikan *Message Passing Interface (MPI)* dan metode *MapReduce* untuk mendemonstrasikan efektivitasnya dalam mempercepat algoritma *hill cipher* dengan algoritma paralel pada sistem *multicore*. Hasil simulasi menemukan jika angka efisiensi MPI dan *MapReduce* adalah 93.71% dan 53.43%, mengindikasikan hasil yang lebih baik dibandingkan dengan metode konvensional [9].

Joshi dalam penelitiannya mengusulkan suatu pendekatan baru dalam dunia steganografi dengan memanfaatkan kombinasi antara nilai pada karakter ASCII dengan nilai RGB pada suatu pixel, sehingga suatu karakter tunggal dapat tersimpan dalam sebuah pixel. Tujuan utama dari penelitian ini adalah menyediakan kapasitas maksimum pixel yang dapat dimiliki oleh suatu gambar. Pendekatan yang diusulkan adalah dengan cara mengambil karakter pertama dari suatu pesan dan membagi nilai karakter ASCII karakter tersebut menjadi tiga segmen. Kemudian tiga angka dari nilai karakter yang kemudian ditempatkan pada 3 pixel yang ada, yaitu RGB (Red, Green, and Blue). Sehingga akhirnya kombinasi dari ketiga angka ini dengan nilai RGB ini tidak akan memberikan perubahan kepada gambar yang awal [10].

Selviana juga dalam penelitian melakukan modifikasi algoritma *hill cipher* dan *twofish*, pada proses enkripsi dan dekripsi file. Implementasi dari modifikasi algoritma *hill cipher* dengan kode wilayah yang kemudian di enkripsi dengan menggunakan metode *Twofish* sehingga kriptanalis tidak bebas mengakses keamanan data. Enkripsi berupa penyandian data dan informasi menjadi sesuatu yang tidak terbaca oleh pihak yang tidak berhak dan juga bermanfaat dalam memberikan tambahan informasi terkait proses enkripsi dan dekripsi yang terjadi pada algoritma *twofish*. Hasil dari enkripsi dengan algoritma ini adalah file yang dapat diakses melalui aplikasi notepad yang berisi simbol acak. Namun pada penelitian modifikasi algoritma *hill cipher* dan *twofish* ini hanya menggunakan Teknik kriptografi untuk menyembunyikan file [11].

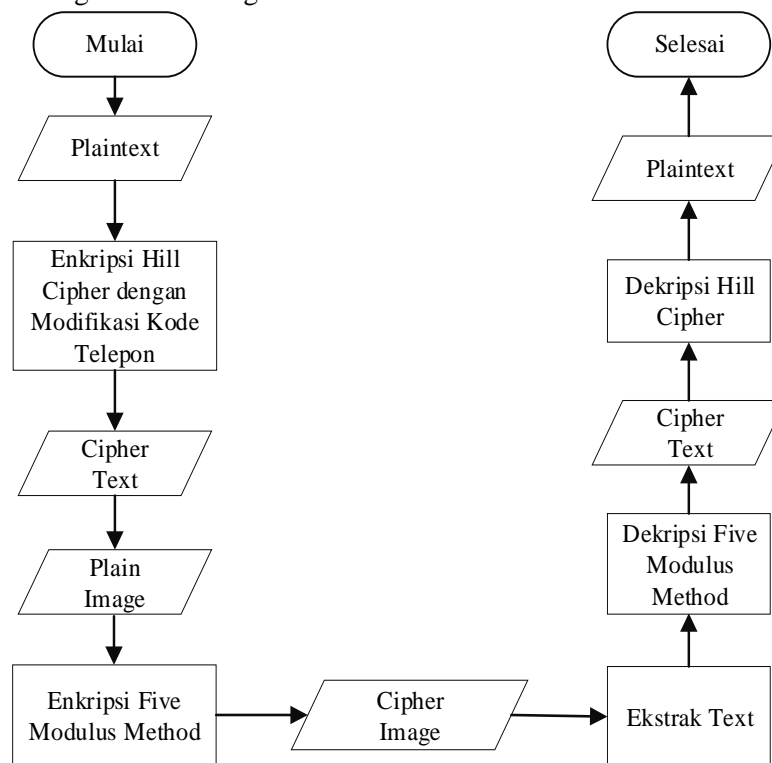
Berdasarkan penelitian yang telah dilakukan sebelumnya, penelitian ini menawarkan modifikasi algoritma dengan menggunakan algoritma *hill cipher* dan *Five Modulus Method* sebagai kombinasi algoritma dalam pengamanan pesan yang menggunakan teknik kriptografi dan steganografi. Tahap awal adalah melakukan enkripsi informasi dengan menggunakan

algoritma *hill cipher* yang dikombinasikan dengan kode telepon. Setelah itu, *cipher text* akan disisipkan pada sebuah gambar dengan menggunakan algoritma *Five Modulus Method*. Pengujian dilakukan dengan melakukan percobaan enkripsi pesan teks pada sistem sebanyak lima kali dan membandingkan dengan perhitungan manual.

2. METODE PENELITIAN

2.1. Alur Penelitian

Alur penelitian untuk menyembunyikan pesan rahasia menggunakan kombinasi algoritma *Hill cipher* dan *Five Modulus Method* dalam meningkatkan keamanan berlapis pada pesan teks dapat dilihat gambar 1 sebagai berikut:



Gambar 1 Alur Penelitian

Pada gambar 1 Alur penelitian dibuat gambaran alur data secara detail sesuai dengan penggunaan algoritma sebagai berikut:

1. Hal pertama yang dilakukan adalah memasukkan *plaintext* yang akan di enkripsi.
2. *Plaintext* yang di enkripsi menggunakan algoritma *Hill cipher* dengan kunci kode telepon yang telah dimodifikasi.
3. Kemudian didapatkan *cipher text* yang nantinya akan disisipkan kedalam gambar.
4. Menyiapkan *plain image* yang akan disisipkan *cipher text*.
5. Setelah *plain image* dimasukkan kedalam system kemudian di enkripsi menggunakan algoritma *Five Modulus Method*.
6. Kemudian didapatkan *cipher image* yang didalamnya sudah di ekstrak Text
7. Mendekripsikan *cipher image* untuk mendapatkan *chipper text*.
8. Langkah selanjutnya mendekripsikan *chipper text* menggunakan algoritma *Hill cipher*.
9. Akhir dari alur data tersebut adalah sebuah *plain text*.

2.2. Algoritma Hill Cipher

Pada tahun 1929 algoritma *Hill Cipher* diciptakan oleh Lester S. Hill. Teknik kriptografi ini bermaksud untuk menciptakan *cipher* (kode) yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. *Hill Cipher* merupakan salah satu algoritma kriptografi kunci simetris yang memiliki beberapa kelebihan dalam enkripsi data. Algoritma *Hill Cipher* menggunakan matriks berukuran $m \times m$ sebagai kunci untuk melakukan enkripsi dan dekripsi [12]. Hill cipher dapat digolongkan sebagai kriptografi *polyalphabetic* yang dapat dikategorikan sebagai blok cipher, karena teks yang akan diproses dibagi menjadi blok-blok dengan ukuran tertentu. Menggunakan matriks berukuran $m \times n$ sebagai kunci untuk melakukan enkripsi dan dekripsi. Dasar teori matriks yang digunakan dalam Hill Cipher antara lain adalah perkalian antar matriks dan melakukan invers pada matriks [13]. Matriks K yang menjadi kunci ini harus merupakan matriks yang invertible, yaitu memiliki inverse K^{-1} . Kunci harus memiliki invers karena matriks K^{-1} tersebut adalah kunci yang digunakan untuk melakukan dekripsi.

Dalam melakukan enkripsi pada algoritma Hill Cipher dilakukan blok per blok plainteks. Ukuran blok tersebut sama dengan ukuran matriks kunci. Sebelum membagi teks menjadi deretan blok-blok, plainteks mula-mula dikonversi menjadi angka, misalnya A=0, B=1, sampai Z=25. Secara matematis, proses enkripsi pada Hill Cipher adalah [14]:

$$C = K \cdot P$$

C = Ciphertext

K = Kunci

P = Plaintext

2.3. Algoritma Five Modulus Method

Ide utama dari Five modulus method adalah berdasarkan pada satu konsep yaitu karakteristik umum pada sebagian besar gambar adalah pixel-pixel yang berdekatan saling berhubungan. Karena itu, untuk bi-level images, pixel yang saling berdekatan memiliki kecenderungan menjadi sama dengan piksel yang asli. Five Modhulus Method membagi gambar menjadi setiap blok $k \times k$ pixel. Diketahui setiap pixel pada bi-level gery images memiliki nilai antara 0 sampai 255. Meskipun jika dapat diubah setiap nilai yang ada pada rentang tesaerbut menjadi nilai yang dapat dibagi 5, hal tersebut tidak akan mempengaruhi sistem penglihatan manusia. Konsep dasar dari FMM adalah dengan memeriksa seluruh pixel diblok $k \times k$ dan mengubah setiap pixel menjadi angka yang dapat dibagi menjadi 5 berdasarkan algoritma dibawah ini [6]:

```
If Pixel mod 5 = 4
Pixel=Pixel+1
Else if Pixel mod 5 = 3
Pixel=Pixel+2
Else if Pixel mod 5 = 2
Pixel=Pixel-2
Else if Pixel mod 5 = 1
Pixel=Pixel-1
```

2.4. Kode Telepon

Pada kunci dalam algortima *Hill Cipher* yang digunakan untuk menenkripsi *plaintext* yaitu kode telepon yang telah dimodifikasi dengan tujuan meningkatkan keamanan *plaintext* agar tidak diketahui orang-orang yang tidak bertanggung jawab. Berikut kunci kode telepon yang telah dimodifikasi pada table 1:

Tabel 1. Kunci Kode Telepon

Kode Telepon	Modifikasi Kunci	Keterangan
03-25	$\begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}$	Sangkapura (Bawean)
04-31	$\begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix}$	Tomohon
04-35	$\begin{bmatrix} 3 & 1 \\ 5 & 4 \end{bmatrix}$	Limboto
04-23	$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$	Makale
04-19	$\begin{bmatrix} 1 & 1 \\ 9 & 4 \end{bmatrix}$	Jeneponto
04-17	$\begin{bmatrix} 1 & 1 \\ 7 & 4 \end{bmatrix}$	Malino

3. HASIL DAN PEMBAHASAN

3.1 Algoritma Hill Cipher

a. Enkripsi Menggunakan Algoritma Hill Cipher

Pesan rahasia yang dijadikan *plaintext* (P A S C A S A R J A N A) dengan modifikasi kunci perblok dengan matrix (2 x 2) sebagai berikut :

$$K1 = PA = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}$$

$$K2 = SC = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix}$$

$$K3 = AS = \begin{bmatrix} 3 & 1 \\ 5 & 4 \end{bmatrix}$$

$$K4 = AR = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

$$K5 = JA = \begin{bmatrix} 1 & 1 \\ 9 & 4 \end{bmatrix}$$

$$K6 = NA = \begin{bmatrix} 1 & 1 \\ 7 & 4 \end{bmatrix}$$

Dalam proses penyandian harus menetapkan kunci matriks (2 x 2) terlebih dahulu untuk melakukan proses penyandian. Pesan yang disandikan maksimal 66 karakter tiap karakter harus berada diantara A-Z yang berjumlah 25 huruf dalam proses penyandian ini huruf besar dan kecil tidak dibedakan. Tahap-tahap enkripsi *plaintext* (PASCASARJANA) diantaranya:

Hal pertama yang dilakukan adalah mengkonversi deretan huruf *plaintext* menjadi deretan angka seperti Tabel 2 berikut:

Tabel 2. Konversi Deretan Huruf *Plaintext* menjadi Deretan Angka

Huruf	A	B	C	D	E	F	G	H	I	J	K	L	M
Angka	0	1	2	3	4	5	6	7	8	9	10	11	12
Huruf	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Angka	13	14	15	16	17	18	19	20	21	22	23	24	25

Maka *plaintext* (PASCASARJANA) berbentuk seperti Tabel 3 berikut:

Tabel 3. Perubahan *plaintext*

<i>Plaintext</i>	P	A	S	C	A	S	A	R	J	A	N	A
Angka	15	0	18	2	0	18	0	17	9	0	13	0

Kemudian membagi deretan angka menjadi sebuah blok matrix (1 x 2) seperti berikut:

$$\text{Blok 1} = \begin{bmatrix} P \\ A \end{bmatrix} = \begin{bmatrix} 15 \\ 0 \end{bmatrix}$$

$$\text{Blok 2} = \begin{bmatrix} S \\ C \end{bmatrix} = \begin{bmatrix} 18 \\ 2 \end{bmatrix}$$

$$\text{Blok 3} = \begin{bmatrix} A \\ S \end{bmatrix} = \begin{bmatrix} 0 \\ 18 \end{bmatrix}$$

$$\text{Blok 4} = \begin{bmatrix} A \\ R \end{bmatrix} = \begin{bmatrix} 0 \\ 17 \end{bmatrix}$$

$$\text{Blok 5} = \begin{bmatrix} J \\ A \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \end{bmatrix}$$

$$\text{Blok 6} = \begin{bmatrix} N \\ A \end{bmatrix} = \begin{bmatrix} 13 \\ 0 \end{bmatrix}$$

Selanjutnya mengenkripsikan modifikasi kunci perblok dengan matrix (2 x 2) yang di jumlahkan dengan *plaintext* blok matrix (1 x 2) menggunakan algoritma *Hill Cipher* untuh mendapatkan *chipertext* seperti berikut:

$$\text{Blok 1} = \begin{bmatrix} P \\ A \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix} \times \begin{bmatrix} 15 \\ 0 \end{bmatrix} = \begin{bmatrix} 30 \\ 75 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 4 \\ 23 \end{bmatrix} = \begin{bmatrix} E \\ X \end{bmatrix}$$

$$\text{Blok 2} = \begin{bmatrix} S \\ C \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \times \begin{bmatrix} 18 \\ 2 \end{bmatrix} = \begin{bmatrix} 56 \\ 26 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 4 \\ 0 \end{bmatrix} = \begin{bmatrix} E \\ A \end{bmatrix}$$

$$\text{Blok 3} = \begin{bmatrix} A \\ S \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 5 & 4 \end{bmatrix} \times \begin{bmatrix} 0 \\ 18 \end{bmatrix} = \begin{bmatrix} 18 \\ 72 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 18 \\ 20 \end{bmatrix} = \begin{bmatrix} S \\ U \end{bmatrix}$$

$$\text{Blok 4} = \begin{bmatrix} A \\ R \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 0 \\ 17 \end{bmatrix} = \begin{bmatrix} 17 \\ 68 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 17 \\ 16 \end{bmatrix} = \begin{bmatrix} R \\ Q \end{bmatrix}$$

$$\text{Blok 5} = \begin{bmatrix} J \\ A \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 9 & 4 \end{bmatrix} \times \begin{bmatrix} 9 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ 81 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 9 \\ 3 \end{bmatrix} = \begin{bmatrix} J \\ D \end{bmatrix}$$

$$\text{Blok 6} = \begin{bmatrix} N \\ A \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 7 & 4 \end{bmatrix} \times \begin{bmatrix} 13 \\ 0 \end{bmatrix} = \begin{bmatrix} 13 \\ 91 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 13 \\ 13 \end{bmatrix} = \begin{bmatrix} N \\ N \end{bmatrix}$$

Maka hasil *chipertextnya* adalah (E X E A S U R Q J D N N).

b. Dekripsi Menggunakan Algoritma Hill Cipher

Pesan teks yang disandikan sebelumnya akan disisipkan kedalam gambar menggunakan Teknik steganografi. Kemudian chipertext (E X E A S U R Q J D N N) tersebut dikembalikan seperti semula atau didekripsikan menggunakan algoritma hill cipher sebagai berikut:

Diketahui:

Blok 1 = $\begin{bmatrix} E \\ X \end{bmatrix}$ dengan $K1 = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}$, maka

1. $\text{Det } K = (2 \times 3) - (1 \times 5) = 1$

2. Nilai invers Modulo $1^{-1} \text{ mod } 26$

$$K = 1 = \frac{26(1)+1}{1} = 27$$

3. Invers Kunci

$$K^{-1} = \begin{bmatrix} 3 & -1 \\ -5 & 2 \end{bmatrix}$$

4. Mariks kunci *hill cipher*

$$27 \times \begin{bmatrix} 3 & -1 \\ -5 & 2 \end{bmatrix} = \begin{bmatrix} 81 & -27 \\ -135 & 54 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 3 & 25 \\ 21 & 2 \end{bmatrix}$$

5. Dekripsi *hill cipher*

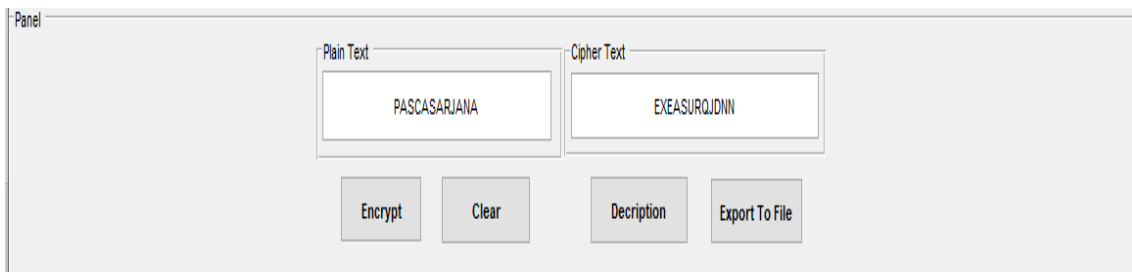
$$\begin{bmatrix} 3 & 25 \\ 21 & 2 \end{bmatrix} \times \begin{bmatrix} 4 \\ 23 \end{bmatrix} = \begin{bmatrix} 587 \\ 130 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 15 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 15 \\ 0 \end{bmatrix} = \begin{bmatrix} P \\ A \end{bmatrix}$$

Selanjutnya dengan proses yang sama dari blok 1 untuk mendapatkan hasil blok 2 sampai dengan blok 6 menggunakan algoritma *hill cipher*.

c. Implementasi Enkripsi Menggunakan Algoritma Hill Cipher

Dengan menggunakan Bahasa Pemrograman MATLAB hasil algoritma *Hill Cipher* dapat dilihat pada Gambar 1 :

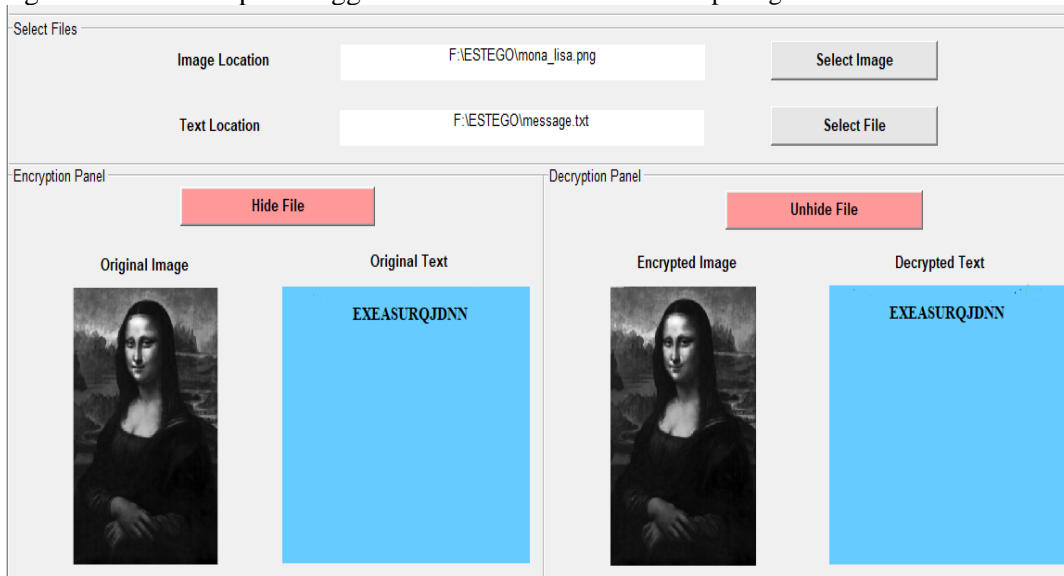


Gambar 1 Hasil Enkripsi Hill Cipher

Pada gambar 1 terdapat buttom encrypt, clear, description dan Export to File, fungsi enport to file adalah dengan menyimpan chipertext ke dalam sebuah notepad yang akan disisipkan kedalam gambar menggunakan algoritma Five Modhulus Method.

3.2 Algoritma Five Modhulus Method

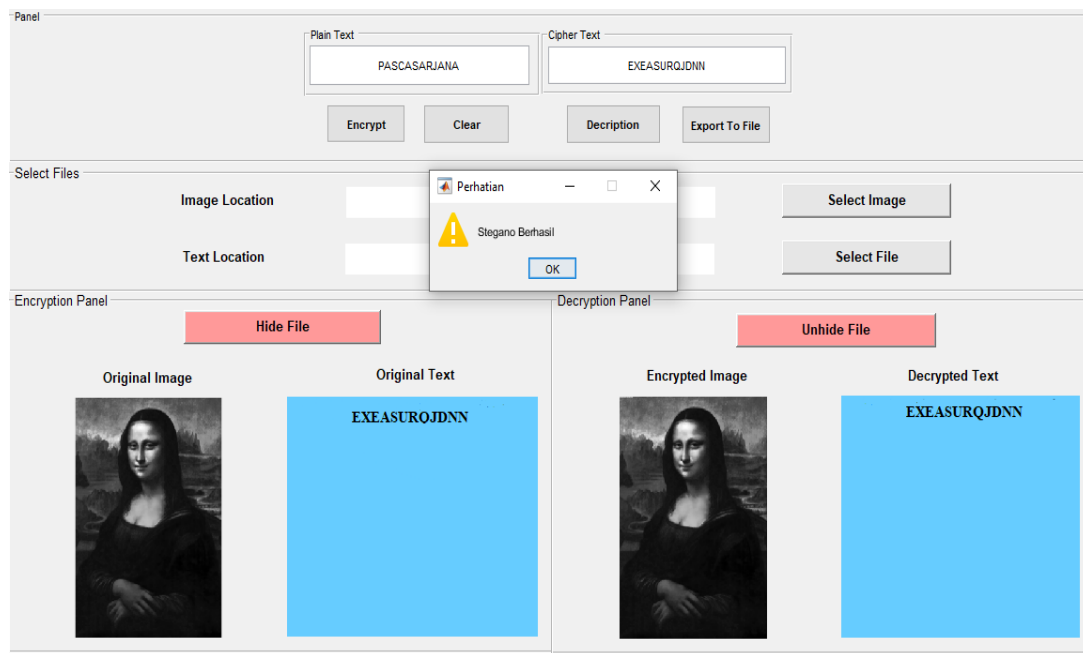
Algoritma Five Modhulus Method membagi gambar menjadi setiap blok $k \times k$ pixel. Diketahui setiap pixel pada bi-level gery images memiliki memiliki nilai antara 0 sampai 255. Meskipun jika dapat diubah setiap nilai yang ada pada rentang tersebut menjadi nilai yang dapat dibagi 5. Berikut enkripsi menggunakan Five Modhulus Method pada gambar 2.



Gambar 2. Hasil enkripsi dan deskripsi Algoritma Five Modhulus Method

Pada gambar 2 *ciphertext* yang disimpan kedalam notepad telah disisipkan kedalam gambar monalisa.png dengan dimensions 250 x 360 pixel. Besaran gambar sebelum disisipkan *chipertext* adalah 38,9 Kb namun setelah disisipkan *chipertext* berubah menjadi 55,8 Kb namun tetap memiliki dimensions yang sama.

Selanjutnya tampilan keseluruhan dari kombinasi *Hill Cipher* dan *Five Modulus Method* dapat dilihat pada Gambar 3.



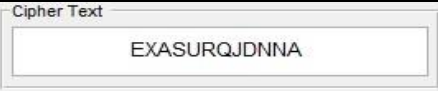
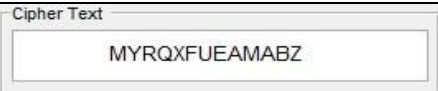
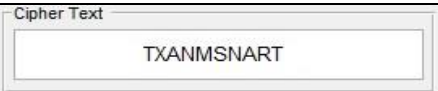
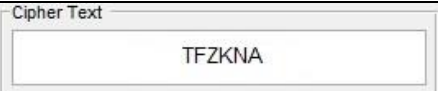
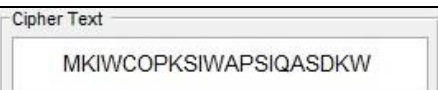
Gambar 3 Kombinasi Hill Cipher dan Five Modulus Method

Pada Gambar 3 menampilkan keseluruhan fungsi dari masing-masing algoritma yang dibuat dalam satu form menu.

3.3 Pengujian

Pengujian akurasi dengan mencocokkan hasil enkripsi *hill cipher* pada sistem dengan data perhitungan manual. Pengujian dilakukan sebanyak lima (5) kali yaitu dengan memasukkan *plaintext* yang berbeda pada tiap-tiap pengujian dapat dilihat pada tabel 4.

Tabel 4. Pengujian Tingkat Akurasi

No	Plaintext	Hasil enkripsi Manual	Hasil enkripsi Sistem	Hasil Uji (%)
1	PASCASARJANA1	EXEASURQJDNNA		100
2	MODIFIKASI 12.	MYRQXFUEAMABZ		100
3	AAAAAAAAAA	TXANMSNART		100
4	ATURAN	TFZKNA		100
5	AMIKOM, KAMPUS WUNGU.	MKIWCOPKSIWAPSIQASDKW		100

Rata-rata hasil uji akurasi

100%

Berdasarkan pengujian pada Tabel 4 dengan kunci kode telepon tersebut tidak ditemukan perbedaan hasil enkripsi maupun dekripsi sehingga didapatkan tingkat akurasi untuk algoritma ini sebesar 100 %.

Kemudian peneliti melakukan pengujian gambar pada Five Modulus Method dengan mencoba 4 gambar dengan ukuran dan tipe gambar yang berbeda didapatkan perubahan kapasitas gambar yang tidak terlalu besar pada tabel 5.

Tabel 5. Hasil Pengujian Gambar

No	Nama File	Ukuran File Asli (Byte)	Hasil Enkripsi Ukuran Gambar (Byte)	Selisih Ukuran Gambar (Byte)
1	Mona_lisa.png	39.881	57.211	17.330
2	Earth_(16530938850).jpg	1.051.749	1.241.899	190.150
3	giphy.gif	208.614	320.453	111.839
4	smiley.bmp	354.870	474.436	119.566

Hasil pengujian gambar pada tabel 5 menampilkan perbedaan ukuran ketika disisipkan teks yang sudah di enkripsi. Selisih pada setiap gambar ditampilkan untuk mengetahui ukuran file yang telah disisipkan teks.

4. KESIMPULAN

Pada penelitian ini, beberapa kesimpulan yang diperoleh adalah:

- Pada awalnya plaintext di enkripsi dengan menggunakan algoritma hill cipher dengan kunci menggunakan kode telepon. Hasil ciphertext kemudian akan disisipkan pada gambar dengan menggunakan algoritma Five Modulus Method.
- Berdasarkan hasil tingkat akurasi yang dilakukan pengujian sebanyak lima (5) kali terhadap *plaintext* yang berbeda didapatkan tingkat akurasi sebesar 100% yaitu tidak ditemukan perbedaan hasil enkripsi ataupun dekripsi.
- Ukuran gambar pada Five Modulus METHOD yang telah disisipkan pesan rahasia tidak banyak mengalami perubahan dengan melakukan empat (4) kali pengujian untuk ukuran file gambar yang berbeda-beda. Namun selalu mendapatkan penambahan ukuran file bila disisipkan pesan yang sudah di enkripsi melalui hill cipher.

5. SARAN

Saran untuk penelitian selanjutnya adalah, dapat dilakukan pengembangan terhadap metode yang di usulkan, sehingga tidak hanya dapat enkripsi text saja, tetapi dapat enkripsi file, serta steganografi yang dilakukan tidak terbatas pada file image saja, namun dapat dilakukan dalam file yang berbentuk video dan media lainnya.

DAFTAR PUSTAKA

- [1] Dony Ariyus, 2008, Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi, Andi Offset.
- [2] Swain G. and Lanka S. K., 2012, A Quick review of Network Security and Steganography, *International Journal of Electronics and Computer Science Engineering*,

- vol. 1, no. 2, pp.426-435.
- [3] Wang H and Wang S., 2004, Cyber warfare: Steganography vs. Steganalysis, Communications of the ACM, vol. 47, no. 10..
 - [4] Qazi, F., Khan, F.H., Agha, D.S., Khan, S.A., Rehman, S.U., 2019, Modification in Hill Cipher For Crptographic Application, *3C Tecnologia. Glosas de Innovacion aplicadas a la pyme*, ISSN: 2254-4143.
 - [5] Prasad, MG. V., Sundarayya, P., 2016, Generalized Self-Invertiblekey Generation Algorithm by using Reflection Matrix in Hill Cipher and Affine Hill Cipher, *International journal of pharmacy & Technology*. ISSN; 0975-766X.
 - [6] Jassim, F. A., 2013, A Novel Steganography Algorithm for Hiding Text in Image using Five Modulus Method, *International Journal of Computer Applications*, Volume 72 No. 17.
 - [7] Alkadi, I., Robert, S., 2017, Application and Implementation of Secure Hybrid Steganography Algorithm in Private Cloud Platform. *Journal of Computer Science Application and Information Technology*. DOI: <http://dx.doi.org/10.15226/2474-9257/2/2/00105>.
 - [8] Ariyus, D., Ardiansyah., 2019, Optimization Substitution Cipher and Hidden Plaintext in Image Data Using LSB Method, *Journal of Physics: Conference Series*, Volume 1201.
 - [9] Qasem, M.H., Qataweh, M., 2018, Parallel Hill Cipher Encryption Algoritm, *Inetrnational Journal of Computer Application*, Volume 179, No. 19.
 - [10] Joshi, K., 2018, A New Approach of Text Steganography Using ASCII Values, *International Journal of Engineering Research & Technology (IJERT)*, Vol. 7 Issue 5.
 - [11] S. Yunita, P. Hasan, and D. Ariyus, 2019, Modifikasi Algoritma Hill Cipher dan Twofish Menggunakan Kode Wilayah Telepon, *Sisfotenika*, vol. 9, no. 2, pp. 213–224.
 - [12] Jane Irma Sari, Sulindawaty, Hengki Tamando Sihotang, 2017, Implementasi Penyembunyian Pesan Pada Citra Digital dengan Menggabungkan Algoritma Hill Cipher dan Metode Least Significant BIT (LSB), *Jurnal Mantik Penusa*, Volume 1 No 2 Desember 2017
 - [13] Forouzan, Behrouz, 2006, Cryptography and Network Security, McGraw-Hill.
 - [14] Kaharuddin, E. Pawan, and D. Ariyus, 2019, Kombinasi Arnold Cat Map dan Modifikasi Hill Cipher Menggunakan Kode Bunyi Beep BIOS PHOENIX, *Sisfotenika*, vol. 9, no. 2, pp. 159–168.

Sistem Pendukung Keputusan Menentukan Bibit Padi Terbaik Menggunakan Metode Gap Kompetensi

Decision Support Systems Determine the Best Rice Seeds Using the Competence Gap Method

Elvis Pawan^{*1}, Agung Jasuma², Achmad Yusron Arif³, Kusri⁴

¹Program Studi Teknik Informatika, STIMIK Sepuluh Nopember Jayapura Tlp (0967) 533799

^{2,3,4}Magister Teknik Informatika, Universitas Amikom Yogyakarta Tlp (0274) 884201

e-mail: *elvispawan09@gmail.com, agung.jasuma@students.amikom.ac.id,

achmadyusronarif@gmail.com, kusrini@amikom.ac.id

Abstrak

Pemerintah sedang mempersiapkan Indonesia mencapai swasembada pangan, akan tetapi perlu kita ketahui bahwa persolaan utama yang akan menghambat cita-cita tersebut tercapai adalah jika produksi hasil pertanian padi menurun terutama di daerah terpencil. Ini diakibatkan karena kurangnya pengetahuan petani dalam menentukan bibit padi yang cocok untuk ditanam pada lahan pertanian. Penelitian ini diharapkan dapat membantu petani secara khusus yang belum mengerti menentukan bibit padi yang baik. Selain itu hasil penelitian ini dapat membantu pemerintah mewujudkan swasembada pangan di Indonesia. Dalam merancang SPK ini menggunakan metode Gap Kompetensi dengan memasukkan beberapa kriteria seperti, produksi padi perhektar, ketahanan terhadap penyakit, harga bibit, batas tanam, dan kebutuhan tanah, kriteria tersebut akan diolah sesuai aturan metode Gap Kompetensi untuk menghasilkan sebuah rekomendasi bibit terbaik. Penelitian ini menghasilkan SPK, kriteria utama digunakan ketahanan terhadap penyakit dan harga bibit yang diberikan nilai prioritas sebesar 70%, kriteria lainnya sebesar 30%. Berdasarkan kriteria tersebut diperoleh tiga varietas padi yang direkomendasikan oleh sistem yaitu pertama jenis Kalimasada, kedua Inpasari 42 GSR dan ketiga jenis Mapan P-05. Bagi peneliti selanjutnya diharapkan dapat membandingkan beberapa metode serta menggabungkan kedalam sistem pakar untuk tingkat akurasi yang lebih baik.

Kata kunci— SPK, Gap Kompetensi, Bibit, Padi

Abstract

Government is preparing Indonesia to achieve food self-sufficiency, but we need know the main process that will hamper these ideals achieved if the production of paddy agriculture decreases, especially remote areas. This is due to lack of knowledge of farmers in determining which rice seeds are suitable for planting on agricultural land. This research is expected to help farmers specifically who do not understand the determination of good rice seeds. In addition, the results of this study can help the government realize food self-sufficiency in Indonesia. In designing this SPK using the Competence Gap method by including several criteria such as rice production per hectare, disease resistance, seed prices, planting limits, and land requirements, these criteria will be processed according to the rules of the Competence Gap method to produce the best seed recommendation. This study resulted in SPK, the main criteria used for disease resistance and the price of seeds given a priority value of 70%, another criterion of 30%. Based these criteria, three rice varieties recommended by DSS, Kalimasada, both

Inpasar 42 GSR and the three types of Mapan P-05. The next researcher expected able to compare several methods and combine them into expert system for better accuracy.

Keywords— *DSS, Competency Gap, Seeds, Rice*

1. PENDAHULUAN

Salah satu kebijakan pemerintah pada pembangunan bidang pertanian khususnya tanaman pangan bertujuan untuk melestarikan swasembada pangan [1]. Beras yang bersumber dari padi merupakan salah satu makanan pokok bagi masyarakat Indonesia [2]. Sebagian besar mata pencaharian penduduk di Indonesia adalah bertani, tanaman pertanian yang paling banyak diolah adalah padi [2]. Sebagai Negara agraris negara Indonesia, merupakan Negara yang kaya akan sumber daya alam. Hingga kini mayoritas penduduk Indonesia telah memanfaatkan sumber daya alam untuk menunjang kebutuhan hidupnya dan salah satunya ialah menggantungkan hidupnya pada sektor pertanian [3]. Namun sering kali masyarakat atau petani kesulitan dalam menentukan bibit yang cocok untuk ditanam [3]. Beberapa alasan inilah yang mendorong dilakukannya penelitian ini.

Beberapa penelitian menentukan bibit tanaman yang baik telah dilakukan akan tetapi menggunakan metode yang berbeda pada penelitian ini, beberapa metode yang digunakan diantaranya metode *Analytical Hierarchy Process (AHP)*, *Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)*, *Simple Additive Weighting (SAW)* dan beberapa metode lainnya. Adapun penelitian mencari bibit tanaman padi terbaik dengan menggunakan metode AHP telah dilakukan dan memiliki kesimpulan bahwa metode AHP baik untuk digunakan dalam menentukan bibit tanaman padi terbaik akan tetapi pada penelitian tersebut memiliki beberapa kelemahan seperti kesimpulan SPK tidak memiliki wisdom dan sistem yang dibangun tidak sampai dengan implementasi melainkan hanya sebuah rancangan [3]. Sedangkan penelitian ini sampai pada tahap implementasi kedalam aplikasi SPK.

Penelitian tentang menentukan bibit padi unggul menggunakan metode *Simple Additive Weighting (SAW)* memiliki kesimpulan bahwa dengan adanya sistem pendukung keputusan menggunakan metode SAW maka petani memiliki panduan dalam menentukan bibit unggul akan tetapi pada penelitian tersebut memiliki kelemahan yakni wisdom serta rekomendasi bibit unggul yang dihasilkan dari SPK ini tidak ditampilkan [4]. Sedangkan penelitian ini memberikan hasil rekomendasi dan wisdom yang akan digunakan oleh petani sebagai acuan dalam menentukan bibit padi yang akan ditanam, ini merupakan hasil akhir dari sistem pendukung keputusan yang dirancang.

Penelitian tentang penentuan varietas padi terbaik menggunakan metode AHP dan TOPSIS telah dilakukan oleh peneliti sebelumnya dan memiliki kesimpulan akurasi kecocokan antara keluaran sistem dan pendapat ahli sebesar 83.33 % akan tetapi kelemahan pada penelitian tersebut hanya melakukan perankingan [5]. Perbedaan pada penelitian yang ini yaitu tidak hanya dilakukan perankingan pada sistem namun terdapat hasil rekomendasi berupa laporan yang dapat dicetak dari sistem, yang dapat digunakan petani sebagai rekomendasi untuk menentukan atau membeli bibit padi yang di inginkan.

Selain itu metode *Gap Kompetensi* telah beberapa kali diteliti oleh pakar terdahulu akan tetapi pada kasus yang berbeda misalnya penelitian penggunaan metode *profile matching* untuk menunjang keputusan kenaikan jabatan pada sebuah instansi pemerintah pada penelitian ini memberikan rekomendasi pegawai yang akan dipilih sebagai kandidat yang paling cocok untuk mendapatkan kenaikan jabatan, akan tetapi kelemahan penelitian tersebut tidak diimplementasikan kedalam bentuk aplikasi SPK sehingga perhitungan dan penentuan dilakukan secara manual [6]. Perbedaan penelitian ini dengan penelitian sebelumnya adalah hasil perhitungan manual akan di implementasikan kedalam aplikasi SPK sehingga benar-benar dapat dimanfaatkan oleh masyarakat atau petani.

Penelitian yang menggunakan bantuan metode *profile matching* pemilihan kandidat pejabat struktural pada sebuah instansi pemerintahan di kota Tarakan memiliki kesimpulan bahwa dengan memanfaatkan metode *profile matching* maka penilaian kandidat dapat dilakukan oleh sistem sehingga dapat membantu mengatasi masalah yang muncul yang sebelumnya dilakukan secara manual sehingga sistem ini dapat membantu meminimalisir terjadinya keputusan secara subjektif [7]. Perbedaan pada penelitian yang dilakukan adalah dalam hal implementasi pada penelitian terdahulu mengambil objek pada pemilihan kandidat pejabat struktural pada instansi pemerintahan sedangkan penelitian ini objeknya dilakukan pada penentuan bibit tanaman padi terbaik, hal yang dipertimbangkan sebagai acuan dari penelitian terdahulu menyatakan bahwa metode *profile matching* dapat membantu untuk memutuskan jalan keluar sebuah kasus dan terhindar dari hal-hal yang bersifat subjektif.

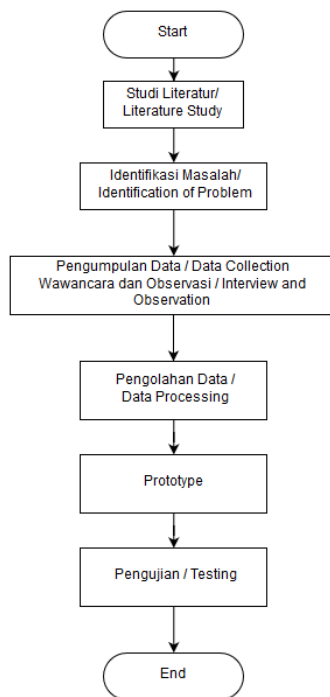
Penelitian lain dengan menggunakan metode *profile matching* pada pemindahan posisi karyawan dalam sebuah perusahaan, kesimpulan pada penelitian ini bukan berupa rekomendasi atau wisdom melainkan sebuah harapan pada masa yang akan datang, selain itu pengujian pada SPK yang dibangun tidak dijelaskan secara eksplisit [8]. Perbedaan pada penelitian ini bahwa kesimpulan berupa rekomendasi sebagai hasil akhir dari sistem SPK yang dirancang

Penelitian pemilihan marketing terbaik pada sebuah perusahaan perbankan menggunakan metode *profile matching* pada penelitian ini menyimpulkan bahwa metode *profile matching* dapat memberikan bantuan untuk menentukan seorang *marketing officer* yang terbaik akan tetapi kelemahan pada penelitian tersebut adalah pengujian tidak dilakukan [9]. Perbedaan pada penelitian ini dilakukan dua pengujian yaitu pengujian terhadap sistem dan pengujian terhadap *output* untuk benar-benar mengetahui akurasi dari hasil penelitian.

2. METODE PENELITIAN

2.1 Alur Penelitian

Dalam melakukan penelitian terdiri dari beberapa tahapan penting yang digambarkan dan dijelaskan seperti terlihat pada Gambar 1 Alur Penelitian. Hal ini mempunyai tujuan agar para pembaca dapat benar-benar memahami langkah-langkah dalam membuat sebuah sistem pendukung keputusan menggunakan metode Gap Kompetensi.



Gambar 1. Alur Penelitian

Berdasarkan langkah-langkah pada Gambar 1 Alur Penelitian, dijelaskan bahwa ada enam tahapan penting dalam melakukan penelitian ini, langkah pertama adalah studi literatur pada tahap ini peneliti mempelajari hal-hal yang berkaitan dengan topik pada penelitian, adapun sumber-sumber literatur berupa makalah, jurnal, hasil penelitian tesis, prosiding, arsip dan artikel yang memiliki ciri atau kesamaan topik yang dibahas. Langkah kedua dengan melakukan identifikasi masalah bertujuan untuk mengetahui permasalahan yang sedang terjadi dan memfokuskan penelitian pada kasus yang akan dipecahkan atau diselesaikan. Langkah ketiga pengumpulan data yang berkaitan dengan topik yang dibahas pada penelitian ini, data dikumpulkan dari berbagai sumber baik dari para ahli ataupun dari literatur-literatur. Langkah ke empat melakukan pengolahan data yang telah dikumpulkan serta melakukan perhitungan manual dengan mengikuti aturan metode Gap Kompetensi, hal tersebut bertujuan agar memudahkan pemindahan kedalam *prototype* atau aplikasi SPK. Langkah ke lima pembuatan *prototype* atau aplikasi untuk memperoleh representasi dari hasil penelitian dan dapat digunakan sebagai pembandingan perhitungan manual. Langkah ke enam melakukan pengujian atau *testing* terhadap akurasi hasil penelitian.

2.2 Gap Kompetensi

Gap merupakan sebuah perbedaan antara *profile* yang dimiliki oleh masing-masing alternatif dengan profile standar. Hal tersebut dapat dilihat pada persamaan 1 [10] [11].

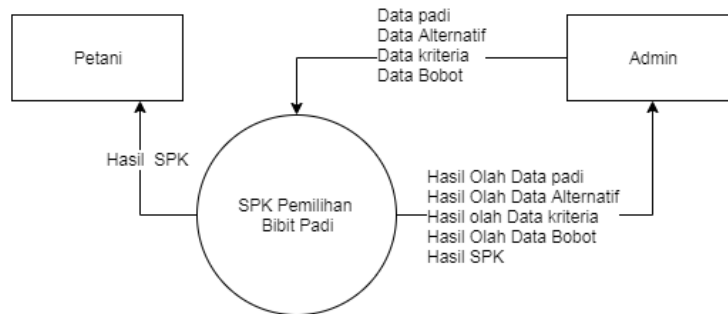
$$Gap = Profile Atribut - Profile Target \dots\dots\dots (1)$$

Hasil akhir dari sebuah gap kompetensi adalah rangking *varietas* padi yang akan ditanam, Penentuan rangkin berpatokan pada perhitungan kriteria yang diinputkan user pada sistem.

2.3 Perancangan Sistem

a. Data Flow Diagram (DFD)

Data flow diagram ialah data yang dimanfaatkan untuk menggambarkan langkah-langkah atau proses yang terjadi pada sistem yang sedang dirancang untuk dikembangkan. Dengan memakai model DFD aliran atau arus data dapat terdeteksi dengan mudah. Pada penggunaan model DFD dimulai dari diagram konteks selanjutnya diagram berjenjang, diagram level 1 sampai pada diagram terperinci, hal tersebut dapat disesuaikan dengan kompleksitas SPK yang dikembangkan [12]. Adapun diagram konteks pada SPK yang dirancang dapat dilihat pada Gambar 2.

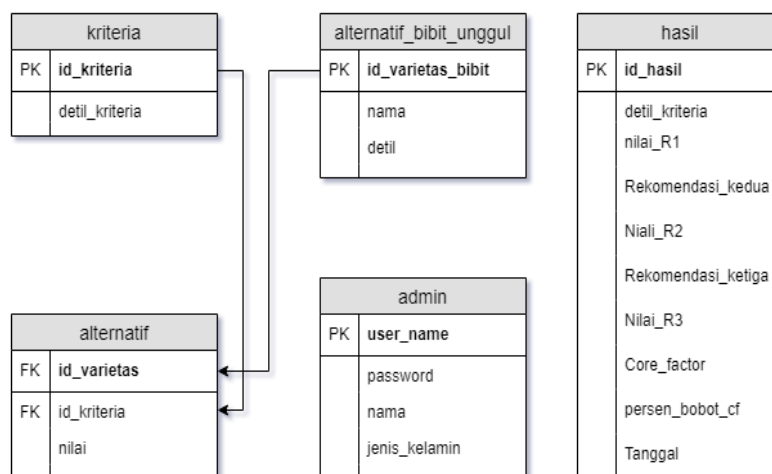


Gambar 2. Diagram Konteks

Gambar 2. Diagram Menjelaskan aliran atau arus data didalam sistem SPK yang dirancang sebagai contoh Admin dapat menginputkan pada sistem bebera bentuk data misalnya data alternative, data kriteria, data bobot, dan data padi sehingga sistem akan mengirimkan kembali hasil pengolahan data yang telah di inputkan oleh admin, baik itu berupa data yang di tampilkan di sistem maupun data yang dapat dicetak.

b. Perancangan Basis Data

Setelah pembuatan DFD selanjutnya mendesain rancangan basis data yang digunakan pada aplikasi SPK, adapun rancangan basis data Sistem pendukung keputusan Penentuan bibit padi terbaik menggunakan metode *Gap* Kompetensi dapat dilihat pada gambar 3. Rancangan Basis data SPK.



Gambar 3. Rancangan Basis Data SPK

Pada Gambar 3 menjelaskan terdapat lima tabel yang digunakan pada SPK ini diantaranya adalah tabel kriteria, alternatif, varietas_bibit_unggul, tabel admin dan tabel hasil. Terdapat tiga tabel yang berelasi yaitu tabel kriteria berelasi ke tabel alternative dengan Id_kriteria pada tabel kriteria sebagai *Primary Key* (PK) dan id_kriteria pada tabel alternatif sebagai *foreign key* (FK). Tabel varietas_bibit_unggul berelasi ke tabel alternatif dengan id_varietas_bibit sebagai *primary key* (PK) pada tabel varietas_bibit_unggul dan id_varietas pada tabel alternatif sebagai *foreign key* (FK). Sedangkan tabel admin dan tabel hasil berdiri sendiri dan tidak berelasi ke tabel lainnya.

3. HASIL DAN PEMBAHASAN

Hal yang paling penting dalam keberhasilan panen adalah menggunakan bibit yang baik dalam bercocok tanaman, itu dapat berdampak positif terhadap hasil panen karena dengan menggunakan bibit yang baik dapat mengurangi potensi tanaman terserang hama, panen dapat meningkat sampai dengan rata-rata 11 ton dalam satu hektar lahan, batas tanam juga dapat dilakukan beberapa kali, serta baik ditanam pada kondisi rata-rata pH tanah. Hal tersebut tentu sangat diharapkan oleh setiap petani akan tetapi permasalahan yang terjadi dilapangan ialah hanya sebaian kecil petani yang dapat menentukan jenis padi yang cocok ditanam.

Sistem pendukung keputusan dirancang agar dapat membantu seorang *user* untuk mengambil sebuah tindakan yang bermanfaat, sistem pendukung keputusan yang dirancang menggunakan metode *Gap* Kompetensi dengan memasukkan beberapa kriteria yang sering dijumpai dilapangan. Dengan demikian dengan adanya sistem pendukung keputusan ini diharapkan penentuan bibit yang baik tidak lagi menjadi hal yang sulit bagi petani. Dalam melakukan analisis pada penelitian ini menggunakan data serta fakta yang dijumpai dilapangan.

3.1 Daftar Varietas Bibit

Data varietas bibit padi yang telah diperoleh dan digunakan pada penelitian ini dapat dilihat pada tabel 1 varietas bibit.

Tabel 1. Varietas Bibit

Nama Bibit	Deskripsi
Inpasari 42 GSR	Varietas inbrida keluaran tahun 2016, rasanya enak dan pulen luar biasa, kerontokannya pun mudah sekali. Potensi hasilnya mencapai 7,11 ton
Sertani 13	Sertani 13 merupakan jenis padi dari MSP. Popularitasnya begitu meluas dari Sabang sampai Merauke. Sertani 13 adalah jenis terbaik yaitu 13A, 13B, dan 13C.
Mapan P-05	Padi hibrida merupakan unggulan berkualitas,Keunggulan utamanya adalah bobot tiap gabah yang tinggi dengan hasil > 8 ton / ha.
Padjadjaran	Padi padjadjaran ialah varietas keluaran 2018. Padi ini menjadi padi dengan hasil tertinggi untuk varietas yang dikembangkan pemerintah selama ini. potensi produktivitas 11 ton / ha, dengan rata-rata hasil 7,8 ton / ha.
Kabir 07	Kabir 07 adalah galur lokal Aceh.Hasil rata-rata diatas 8 ton / hektar.
Inpari 43 GSR	Varietas ini mempunyai gelar GSR (Green Super Rice) ini mampu memproduksi sampai 10 ton/ha.
M400	Mempunyai hasil hingga 400 bulir / malai menjadikan jenis padi ini bisa menghasilkan rata-rata diatas 8 ton / hektar dan potensi 11 ton. Selain usianya pendek hanya 90 harian m400 lebih mudah dalam hal perawatan.
IPB 3S	Padi ini merupakan hasil penelitian ahli Pertaian Bogor ini mempunyai potensi hasil di kisaran 10 ton lebih dengan rata-rata hasil 7 ton/ha.

Sistem Pendukung Keputusan Menentukan Bibit Padi Terbaik ...

PIM 1	Tinggi tanaman padi ini hampir 2 meter, banyak orang yang menyukainya karena hasilnya yang banyak. Dalam satu malai dapat mencapai 600 bulir.
Kalimasada	Potensi hasilnya pun sangat lumayan hingga 12 ton dengan rata-rata hasil mencapai 8,5 ton / hektar. Padi satu ini sangat mudah perawatannya dan lebih tahan penyakit.

Pada tabel 1 varietas padi yang digunakan untuk merancang sistem ini sebanyak sepuluh jenis yaitu Inpasari 42 GSR, Sertani 13, Mapan P-05, Padjadjaran, Kabir 07, Inpari 43 GSR, M400, IPB 3S, PIM 1, Kalimasada. Jenis bibit padi tersebut berasal dari berbagai daerah di Indonesia.

3.2 Alternative

Pada penelitian ini memiliki nilai alternative pada setiap varietas atau jenis padi adapun nilai alternative dapat dilihat pada tabel 2 tabel *alternative*.

Tabel 2. Tabel *Alternative*

Alternative	Produksi Padi Perhektar	Ketahanan terhadap penyakit	Harga Bibit	Batas Tanam Bibit	Kebutuhan Tanah
Inpasari 42 GSR	>8 ton/ hektar &<=10 ton/ hektar	Kurang (Tahan <2 jenis penyakit)	<30rb/Kg	Kualitas tetap jika ditanam lebih dari 1x	PH 6
Sertani 13	>10 ton / perhektar	Kurang (Tahan <2 jenis penyakit)	>=60rb/Kg	Kualitas kurang jika ditanam lebih dari 1x	PH 6
Mapan P-05	>10 ton / perhektar	Baik (tahan >2 jenis penyakit)	<30rb/kg	Kualitas sangat buruk jika ditanam lebih dari 1x	PH 6

Pada tabel 3.2 merupakan nilai *alternative* yang dimiliki oleh jenis varietas padi sebagai contoh Inpasari 42 GSR memiliki alternative yaitu lebih besar dari 8 ton/ha, kurang tahan terhadap penyakit, harga bibit dibawah 30 ribu/ kg, kualitas buruk jika ditanam lebih dari satu kali, dan tumbuh di lahan yang memiliki pH 6.

3.3 Bobot Kriteria

Dalam merancang sistem pendukung keputusan menggunakan metode Gap Kompetensi menggunakan beberapa kriteria yang dapat dijadikan sebagai pedoman untuk menentukan bibit terbaik dan sesuai dengan lahan ataupun kebutuhan tanam. Adapun kriteria yang digunakan dapat dilihat pada tabel 3 Bobot.

Tabel 3. Bobot Kriteria

Kriteria	Nilai =1	Nilai = 2	Nilai = 3
Produktivitas Padi	<=8 ton / hektar	>8 ton/ hektar dan <=10 ton/ hektar	>10 ton /hektar
Ketahanan Penyakit	Kurang (Tahan <2 jenis penyakit)	Sedang (tahan 2 jenis penyakit)	Baik (tahan >2 jenis penyakit)
Harga Bibit	>=60 rb/kg	<60rb/kg dan >=30 rb/kg	<30rb/ kg
Batas Tanam	Kualitas sangat buruk jika ditanam lebih dari 1 kali	Kualitas kurang jika ditanam lebih dari 1 kali	Kualitas tetap jika ditanam lebih dari 1 kali
Kebutuhan tanah	PH 5	PH 6	PH 7

Dengan memperhatikan tabel 3 terdapat lima kriteria yang digunakan untuk menentukan bibit padi terbaik kriteria pertama adalah produktivitas padi, kedua ketahanan terhadap penyakit, ketiga harga bibit keempat batas tanam bibit dan kelima adalah kebutuhan bibit terhadap tanah. Untuk memperoleh nilai target maka pada setiap kriteria diberikan nilai sebagai contoh jika Produktivitas padi mencapai rata-rata lebih dari 10 ton perhektar maka nilai

target yang diberikan adalah 3, sedangkan jika 8 sampai 10 ton perhektar diberikan nilai target adalah 2 dan jika lebih kecil dari 8 ton perhektar maka nilai target adalah 1.

3.4 Mencari Selisih Alternative dengan Target Ideal

Untuk mencari selisih nilai (*Gap*) menggunakan persamaan $Gap = Profile\ Atribut - Profile\ Target$, sebagai contoh jika user memilih kriteria ketahanan terhadap penyakit dan harga bibit sebagai prioritas dalam menentukan varietas bibit, dengan nilai prioritas sebanyak 70 % dan kriteria lainnya mendapat nilai sebanyak 30% hal tersebut di ilustrasikan pada tabel 4 Target Ideal

Tabel 4. Target Ideal

Kriteria	Kebutuhan	Nilai
Produksi Padi	>8 ton/ hektar dan <=10 ton/ hektar	2
Ketahanan terhadap Penyakit	Kurang (Tahan <2 jenis penyakit)	1
Harga Bibit	<60rb per KG && >=30 rb per KG	2
Batas Tanam Bibit	Kualitas tetap jika ditanam lebih dari 1x	3
Kebutuhan Tanah	PH 7	3

Pada tabel 4 merupakan ilustrasi antara kriteria yang tersedia, kebutuhan user serta nilai pada setiap kriteria. Kriteria prioritas adalah ketahanan terhadap penyakit dengan kebutuhan *user* varietas bibit tahan terhadap penyakit kurang dari dua, sehingga dengan kriteria tersebut diberikan nilai = 2, kriteria prioritas lainnya adalah harga bibit dengan harga minimal 30 rb/kg dan maksimal harga kurang dari 60 rb/kg sehingga kriteria tersebut diberikan nilai =2, dari kedua kriteria tersebut diberikan nilai prioritas sebesar 70% sedangkan kriteria lainnya diberikan nilai prioritas 30%.

3.5 Selisih Nilai (*Gap*)

Selisih yang diperoleh menggunakan rumus $Gap = Profile\ Atribut - Profile\ Target$ dapat dilihat pada tabel 5 Selisih Nilai (*Gap*).

Tabel 5. Selisih Nilai (*Gap*)

Alternative	Produksi Padi / Hektar	Ketahanan terhadap Penyakit	Harga Bibit	Batas Tanam Bibit	Kebutuhan Tanah
Inpasari 42 GSR	0	0	1	0	-1
Sertani 13	2	0	-1	-1	-1
Mapan P-05	1	2	1	-2	-1
Padjajaran	1	2	1	0	0
Kabir 07	-1	1	1	-2	-1
Inpari 43 GSR	1	1	-1	-1	0
M400	0	2	-1	0	-1
IPB 3S	1	2	-1	-1	0
PIM 1	1	2	0	-2	0
Kalimasada	0	0	0	-1	0

Setelah mengetahui selisih nilai, selisih tersebut dikonversi menjadi nilai baru dengan ketentuan jika 0 maka nilai baru = 3, jika 1 maka nilai baru = 2,5, jika -1 maka nilai baru = 2, jika 2 maka nilai baru = 1,5 dan jika selisih nilai -2 maka nilai baru = 1.

3.6 Nilai Baru

Hasil konversi dari tabel selisih nilai (*Gap*) pada tabel 3.4 diperoleh nilai baru seperti pada tabel 6 dibawah ini.

Tabel 6. Nilai Baru

Alternative	Produksi Padi / Hektar	Ketahanan terhadap Penyakit	Harga Bibit	Batas Tanam Bibit	Kebutuhan Tanah
Inpasari 42 GSR	3	3	2.5	3	3
Sertani 13	2	3	2	2	2
Mapan P-05	1.5	3	2.5	1	2
Padjajaran	2.5	1.5	2.5	3	3
Kabir 07	2	2.5	2.5	1	2
Inpari 43 GSR	2.5	2.5	2	2	3
M400	3	1.5	2	3	2
IPB 3S	2.5	1.5	2	2	3
PIM 1	2.5	1.5	3	1	3
Kalimasada	3	3	3	2	3

Pada tabel 6 merupakan nilai baru yang bermanfaat untuk menghitung nilai *core factor* dan *secondary factor* sebagai contoh varietas padi Inpasari 42 GSR memiliki 5 kriteria dengan nilai baru produksi padi/ hektar = 3, ketahanan terhadap penyakit = 3, harga bibit = 2,5, batas tanam bibit=3 dan kebutuhan tanah sebesar 3.

3.7 Perhitungan dan Pengelompokan Core dan Secondary Factor

Hasil nilai baru yang terdapat pada tabel 6 selanjutnya dilakukan perhitungan mencari nilai *core factor* dan *Secondary Factor*, untuk memperoleh hasil rangking atau rekomendasi bibit yang terpilih sebagai prioritas dalam memilih bibit. Hal tersebut dapat dilihat pada tabel 7 Hasil Rangking

Tabel 7. Hasil Rangking

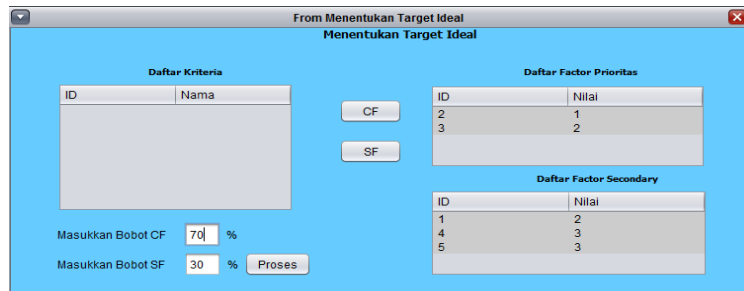
Alternative	Core Factor	Secondary Factor	Nilai Core Factor	Nilai Sekundari Factor	Hasil	Rangking
Inpasari 42 GSR	2.75	3	1.925	0.9	2.825	2
Sertani 13	2.5	2	1.75	0.6	2.35	
Mapan P-05	2.75	1.5	1.925	0.45	2.375	3
Padjajaran	2	2.83	1.4	0.85	2.25	
Kabir 07	2.5	1.66	1.75	0.5	2.25	
Inpari 43 GSR	2.25	2.5	1.575	0.75	2.325	
M400	1.75	2.66	1.225	0.8	2.025	
IPB 3S	1.75	2.5	1.225	0.75	1.975	
PIM 1	2.25	2.16	1.575	0.65	2.225	
Kalimasada	3	2.66	2.1	0.8	2.9	1

Pada table 7 merupakan rekomendasi atau wisdom yang dihasilkan dari sistem pendukung keputusan yang telah dirancang, dengan memilih kriteria ketahanan terhadap penyakit dan harga bibit, sebagai kriteria prioritas dan diberikan nilai prioritas sebesar 70%, sedangkan kriteria lainnya diberikan nilai prioritas sebesar 30%.

Dengan mengikuti aturan Gap kompetensi maka sistem pendukung keputusan memberikan 3 jenis varietas padi yang bisa ditanam, rekomendasi pertama adalah jenis bibit Kalimasada, rekomendasi kedua jenis Inpasari 42 GSR, dan rekomendasi ketiga adalah Mapan P-05.

3.8 User Interface

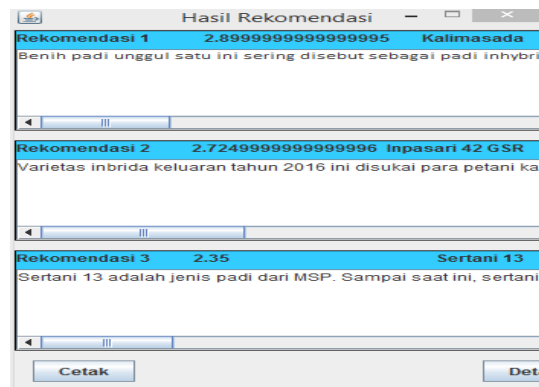
Untuk memanfaatkan sistem yang telah dibuat diawali dengan menentukan nilai target yang dapat dibuat atau diinput melalui modul menentukan target ideal seperti pada gambar 4.



Gambar 4. Modul Menentukan Target Ideal

Pada gambar 4 user dapat memilih kriteria yang sesuai kebutuhan untuk menjadi CF (*core factor*) dan SF (*Secondary Factor*), selain itu user akan diminta mengisikan bobot CF berupa besaran presentasi nilai yang akan diambil untuk *core factor* atau kriteria utama, sedangkan bobot SF akan otomatis terisi dengan nilai 100 dikurang bobot CF.

Data yang telah di inputkan akan di proses mengikuti aturan gap kompetensi dan selanjutnya diperoleh rekomendasi dari SPK, seperti pada gambar 5.



Gambar 5. Hasil Rekomendasi (*Wisdom*)

Pada gambar 5 menampilkan 3 varietas bibit padi terbaik beserta detail informasi pada masing-masing varietas bibit padi tersebut. Tiga varietas padi tersebut yang akan dijadikan rekomendasi varietas bibit yang sebaiknya di tanam.

3.9 Pengujian

Pengujian sistem bertujuan untuk mengetahui sistem pendukung keputusan dapat bekerja sesuai dengan tujuan perancangan.

a. Black Box Testing

Black box testing merupakan suatu metode pengujian perangkat lunak, melalui pengamatan terhadap hasil eksekusi melalui data uji serta memperhatikan fungsionalitas aplikasi perangkat lunak. Pada tabel 8 Hasil uji coba menggunakan metode *black box testing* pada sistem pendukung keputusan yang telah dirancang.

Tabel 8. *Black Box Testing*

No	Skenario Pengujian	Hasil Yang Diharapkan	Status
1	Pada Form Login memasukkan user name dan password sesuai dengan data yang terdapat pada database	Login Sukses dan menuju pada form menu utama	Valid/ sukses

2	Form login di isi user name dan password yang tidak terdapat pada database	Terdapat pemberitahuan username atau password salah dan tetap berada pada laman login	Valid/Sukses
3	Pada Sistem dapat dilakukan input dan menampilkan varietas padi	Sistem dapat menyimpan dan menampilkan data varietas padi	Valid/Sukses
4	Memasukkan atau memilih kriteria dan memberikan nilai prioritas sesuai dengan keinginan user	Sistem dapat menampilkan rekomendasi jenis atau varietas padi yang akan ditanam	Valid/Sukses
5	Menekan tombol proses sebelum nilai prioritas di isikan	Menampilkan pemberitahuan untuk mengisi nilai prioritas terlebih dahulu	Valid/Sukses
6	Menampilkan 3 rekomendasi varietas padi dengan nilai tertinggi	Menampilkan 3 data varietas padi yang memiliki nilai akhir tertinggi	Valid/Sukses

Pada tabel 8 memperlihatkan pengujian yang telah dilakukan terhadap prototype sistem pendukung keputusan yang telah dibuat. Pada masing-masing modul dilakukan pengujian berdasarkan kebutuhan fungsional, meskipun semua modul telah diuji akan tetapi hanya ditampilkan beberapa tindakan yang dilakukan pada modul yang dijadikan sebagai contoh, dan semua mendapatkan hasil yang valid atau sukses.

b. Pengujian Output

Setelah pembuatan *prototype* sistem pendukung keputusan selanjutnya dilakukan pengujian terhadap *output* software. Pengujian dilakukan dengan cara memasukkan data kriteria pada secara manual pada *microsoft excel* selanjutnya data yang sama di input pada sistem pendukung keputusan, apabila perhitungan manual dan perhitungan melalui sistem memiliki kesamaan maka pengujian berhasil, sebaliknya apabila berbeda maka pengujian gagal. Untuk memperjelas hasil pengujian dapat dilihat pada tabel 9 dibawah ini.

Tabel 9 Pengujian *Output*

Pengujian	Nama Kriteria	Perhitungan Manual	Perhitungan Sistem	Keterangan
Kasus 1	Core Factor : 70 %	1. Kalimasada	1.Kalimasada	Berhasil
	1.Harga Bibit	2. Inpasari 42 GSR	2. Inpasari 42 GSR	
	2. Ketahanan Terhadap Penyakit	3. Mapan P-05	3. Mapan P-05	
Kasus 2	Secondary Factor			Berhasil
	1. Produksi Padi	1. Kalimasada	1. Kalimasada	
	2. Batas Tanam	2. Inpasari 42 GSR	2. Inpasari 42 GSR	
Kasus 3	3. Kebutuhan Tanah	3. Inpari 43	3. Inpari 43 GSR	Berhasil
	Core Factor : 65 %	1.Inpasari 42	1. Inpasari 42 GSR	
	1. Harga Bibit	2. Kalimasada	2. Kalimasada	
Kasus 3	2. Produksi Padi	3.M400	3. Inpari 43 GSR	Berhasil
	Secondary Factor			
	1. Batas Tanam			
Kasus 3	2. Ketahanan Terhadap Penyakit			Berhasil
	3. Kebutuhan Tanah			

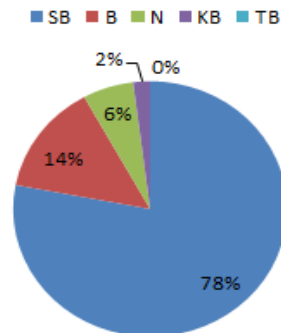
Pada tabel 9 merupakan hasil pengujian output antara sistem dan perhitungan manual sebagai contoh pada kasus pertama harga bibit dan ketahanan terhadap penyakit menjadi *core factor* dan produksi padi, batas tanam, serta kebutuhan tanah menjadi *secondary factor*, antara

perhitungan manual dan sistem memiliki *output* yang sama sehingga pengujian dinyatakan berhasil.

c. *Pengujian User Acceptance Test*

Pengujian dengan dimaksudkan dapat memberikan gambaran apakah sistem pendukung keputusan ini dapat dimanfaatkan sesuai dengan kebutuhan, dan dapat menyelesaikan permasalahan dengan baik. Pengujian user acceptance test dapat dilihat pada gambar 6.

Grafik Pegujian User Acceptance Test



Gambar 6. Pengujian User Acceptance Tes

Pada gambar 6 merupakan pengujian yang dilakukan terhadap lima orang responden yang bertugas sebagai *administrator*, jawaban yang diperoleh sebanyak 78% menjawab sangat benar (SB) dan 14 % menjawab benar (B), 6% netral (N), 2 % kurang benar (KB) dan 0% menjawab tidak benar (TB).

4. KESIMPULAN

Dari penelitian ini peneliti dapat menarik beberapa kesimpulan diantaranya adalah sebagai berikut :

- Sistem pendukung keputusan menggunakan metode gap kompetensi dapat memeberikan rekomendasi jenis atau varietas yang cocok ditanam berdasarkan kriteria yang di inputkan pada sistem.
- Berdasarkan kriteria batas tanam dan harga bibit dengan nilai prioritas sebanyak 70% dan kriteria lain mendapat nilai prioritas sebanyak 30% maka SPK memberikan tiga jenis padi yang cocok untuk ditanam, diantaranya Jenis kalimasada, Inpasari 42 GSR dan Mapan P-05.

5. SARAN

Pada penelitian ini penulis menyarankan beberapa hal kepada peneliti selanjutnya diantaranya sebagai berikut:

- Untuk mengetahui akurasi dari metode gap kompetensi sebaiknya dilakukan penelitian dengan membandingkan beberapa metode, dan menggabungkan SPK dengan sistem pakar.
- Dalam penelitian selanjutnya sabaiknya aplikasi dibuat dengan berbasis website atau android agar dapat di akses oleh banyak pihak yang membutuhkan.

DAFTAR PUSTAKA

- [1] N. F. Mayalibit, Suwanto, E. Rusdiyana, and A. Wijianto, "Rice Farmers Attitude To Superior Certified Seeds," *J. Sustain. Agric.*, vol. 2, no. 2, pp. 116–125, 2017.
- [2] A. Praba *et al.*, "Klasifikasi Prioritas Distrik Terhadap Ketahanan Pangan Menggunakan Metode Jaringan Syaraf Tiruan," *Telematika*, vol. 11, no. 2, pp. 1–11, 2018.
- [3] D. Kurniawan, "Design Of Selection Decision Support System Of Superior Rice Breed Using Ahp Method," *Konf. Mhs. Sist. Inf.*, no. 09, 2005.
- [4] Y. A. Prasetyo, "Selected Decision Support System Of Superior Rice Breed Using Simple Additive Weight (Saw) Method," *Simki-Techsain*, vol. 02, no. 06, pp. 1–10, 2018.
- [5] M. Rendra, H. Roisdiansyah, A. W. Widodo, and N. Hidayat, "Decision Support System for the Selection of Planting Rice Superior Varieties Using AHP and TOPSIS Methods," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 10, pp. 1058–1065, 2017.
- [6] F. Friyadie, "Use Of The Profile Matching Method For The Decision Support System Of Increasing Position In Government Institutions," *Paradigma*, vol. XVIII, no. March, pp. 75–80, 2016.
- [7] R. Astriratma, R. Wardoyo, and A. Musdholifah, "SPK Selection Recommendations for Structural Officials Using the Profile Matching Method (Case Study: Tarakan City Government)," *IJCCS*, vol. 11, no. 1, pp. 77–88, 2017.
- [8] T. Susilowati, E. Y. Anggraeni, W. Andewi, Y. Handayani, and A. Maseleno, "Using Profile Matching Method to Employee Position Movement," *Int. J. Pure Appl. Math.*, vol. 118, no. 7, pp. 415–423, 2018.
- [9] R. M. Wibowo, A. E. Permanasari, and I. Hidayah, "Decision Support Systems With Profile Matching Method In Selection Of Achievement Marketing Officer At Bri Katamso Yogyakarta," *Int. Conf. Sci. Technol. Humanit.*, pp. 115–124, 2015.
- [10] K. Nisa and E. Sutinah, "Profile Matching for Server and Network Maintenance Vendor Selection Decision Support Systems," *J. Inform.*, vol. 5, no. 2, pp. 262–269, 2018.
- [11] Kusriani, *Concept and Application of Decision Support Systems*. Yogyakarta: Andi Offset, 2007.
- [12] David, "Application of Forward Chaining in Expert Systems for Diagnosing Corn Pests and Plant Diseases," *Semin. Nas. Inform.*, 2014.

Sistem Pakar Deteksi Dini Kesehatan Mental Menggunakan Metode Dempster-Shafer

Mental Health Initial Detection Expert System Using Dempster Shafer Method

Annisa Rahmadhani*¹, Fauziah², Andri Aningsih³

^{1,2,3}Universitas Nasional; Jl. Sawo Manila, Telp: (021) 7806700

^{1,2,3}Jurusan Sistem Informasi, FTKI UNAS, Jakarta

e-mail: *rahmadhaniannisa44@gmail.com, fauziah@civitas.unas.ac.id,
andrianingsih@civitas.unas.ac.id

Abstrak

Kesehatan mental merupakan aspek penting yang juga harus diperhatikan sebagaimana kesehatan fisik. Kurangnya kesadaran publik akan kesehatan mental serta keterbatasan waktu dan biaya menjadi beberapa kendala seseorang untuk enggan untuk memeriksakan dirinya pada psikolog/pakar. Penelitian ini bertujuan untuk meningkatkan kesadaran masyarakat akan pentingnya kesehatan mental dan mendeteksi dini apakah seseorang memiliki kecenderungan untuk mengalami masalah pada kesehatan mentalnya dengan sistem pakar menggunakan metode dempster-shafer yang merupakan metode berdasarkan teori fungsi keyakinan dengan menggunakan variabel belief (keyakinan) dan variabel plausibility. Hasil dari penelitian ini adalah sebuah sistem pakar berbasis web yang dapat mendeteksi apakah seseorang memiliki kecenderungan masalah pada kesehatan mentalnya berdasarkan kondisi yang dialami oleh pengguna dan memberikan informasi terkait dengan kesehatan mental. Pengujian akurasi sistem perbandingan antara hasil sistem dan hasil pakar dengan menggunakan 100 data uji mendapatkan persentase 94% dan pengujian usability sistem berdasarkan 31 koresponden mendapatkan nilai 3.95 dari total nilai 5. Diharapkan pada penelitian selanjutnya dapat dikembangkan dengan mengintegrasikan sistem kepada psikolog di lokasi sekitar pengguna untuk konsultasi sehingga memudahkan pengguna dalam penanganan lanjut serta memberikan solusi yang lebih rinci terkait masalah pada kesehatan mental. Selain itu, diharapkan sistem dapat dikembangkan dari platform dari web menjadi berbasis android/ios agar memudahkan pengguna untuk mengakses sistem.

Kata kunci— Sistem Pakar, Kesehatan Mental, Dempster Shafer, Deteksi Dini, Berbasis Web

Abstract

Mental health is an important aspect that needs attention as much as physical health. Lack of public awareness of mental health and the deficiency time and money become some of the thought for someone to be reluctant to check himself with a psychologist / expert. This study aims to do an initial detection by detecting whether a person has a tendency to experience problems in his mental health with an expert system using the dempster-shafer method which is a method based on the theory of belief functions using the belief variable (belief) and plausibility variable.

The results of this study are a web-based expert system that can detect whether a person has a tendency for mental health problems based on the conditions experienced by the user and provide information related to mental health. Testing the accuracy of the system comparison between system result and expert result used 100 data and get 94% percentage and system usability testing based on 31 correspondents and get score 3.95 out of 5. We hope that further research can develop the system by integrating the system with psychologists in around the user's locations to consult so it is easier for users to provide more detailed solutions related to problems in mental health. In addition, it is expected that the system can be developed from the web platform to be based on Android / iOS that it will be easier for users to access the system.

Keywords— *Expert System, Mental Health, Dempster-Shafer, Initial Detection, Web Based*

1. PENDAHULUAN

Kesehatan mental merupakan aspek penting yang perlu diperhatikan sebagaimana kesehatan fisik. Seringkali kesehatan mental diabaikan oleh kebanyakan masyarakat karena berbeda dengan kesehatan fisik yang kasat mata dan dapat diidentifikasi dengan mudah[1]. Berdasarkan World Health Organization (WHO), kesehatan mental adalah keadaan dimana individu menyadari kemampuannya sendiri, dapat mengatasi tekanan hidup yang normal, dapat bekerja secara produktif dan subur, dan mampu memberikan kontribusi ke lingkungan sekitarnya. Berdasarkan hasil riskesdas (riset kesehatan dasar kementerian kesehatan) tahun 2018, terdapat kenaikan sebanyak 3,8% penduduk berumur > 15 tahun yang mengalami gangguan mental emosional di Indonesia. Kenaikan ini dapat dikatakan sebagai tanda bahwa selayaknya masyarakat harus meningkatkan kesadaran terhadap kesehatan mental. Karena kurangnya kesadaran akan kesehatan mental dalam kehidupan masyarakat Indonesia dan anggapan bahwa seseorang yang pergi ke psikolog merupakan seseorang yang kurang waras, seringkali individu yang sebenarnya mengalami masalah kesehatan mental enggan untuk mendapatkan pertolongan dari ahli atau menyangkal bahwa dirinya tidak memiliki masalah kesehatan mental. Keterbatasan waktu dan biaya juga menjadi salah satu kendala seseorang enggan untuk mendapatkan pertolongan dari ahli seperti psikolog. Oleh karena itu, sistem pakar berbasis web dibangun untuk digunakan sebagai deteksi dini dan memungkinkan pengguna untuk mengetahui apakah pengguna tersebut memiliki kecenderungan masalah pada kesehatan mentalnya atau tidak sebelum dibutuhkan konsultasi resmi dari ahli.

Pada penelitian sebelumnya membahas tentang sistem pakar untuk mendeteksi gangguan mental anak menggunakan metode Dempster-Shafer berbasis web yang menggunakan 40 data uji perbandingan antara hasil diagnosa sistem dan hasil diagnosa pakar dan menghasilkan tingkat akurasi sebesar 95% [2]. Penelitian lain penerapan metode Dempster-Shafer menggunakan 20 data uji perbandingan antara hasil diagnosa sistem dan hasil diagnosa pakar dan menghasilkan tingkat akurasi sebesar 85% [3]. Penelitian selanjutnya menggunakan 10 data uji perbandingan antara hasil diagnosa sistem dan hasil diagnosa pakar dan menghasilkan tingkat akurasi sebesar 80% [4]. Penelitian lain menggunakan 104 data uji antara hasil diagnosa sistem dan hasil diagnosa pakar dan menghasilkan tingkat sebanyak 94,23% [5]. Penelitian selanjutnya menggunakan 60 data uji perbandingan antara hasil diagnosa sistem dan hasil diagnosa pakar dan menghasilkan presentase keberhasilan sebanyak 88,33% [6]. Penelitian selanjutnya merupakan penelitian perbandingan antara metode Dempster-Shafer dan metode Certainty Factor yang menghasilkan perbandingan nilai akurasi 90% pada metode Dempster-Shafer yang didapatkan dari hasil diagnosa sistem pakar sedangkan nilai akurasi 85% pada metode Certainty Factor yang didapatkan dari hasil diagnosa sistem pakar [7]. Penelitian terakhir yang merupakan penelitian perbandingan antara metode Dempster-Shafer dan metode Certainty Factor yang menghasilkan nilai akurasi 85% pada metode

dempster shafer yang didapatkan dari hasil diagnosa sistem pakar sedangkan nilai akurasi 80% pada metode certainty factor yang didapatkan dari hasil diagnosa sistem pakar [8].

Penelitian ini menggunakan metode dempster shafer karena berdasarkan tinjauan pustaka yang dilakukan dengan membandingkan metode dempster-shafer dengan metode yang lain, metode dempster-shafer dinilai lebih akurat dalam mendiagnosa penyakit berdasarkan gejala-gejalanya berdasarkan nilai belief dari pakar. Dalam penelitian ini, aplikasi sistem pakar deteksi dini kesehatan mental dibuat dengan fitur login user, konsultasi, hasil konsultasi, informasi terkait kesehatan mental, dan histori terkait konsultasi yang telah dilakukan.

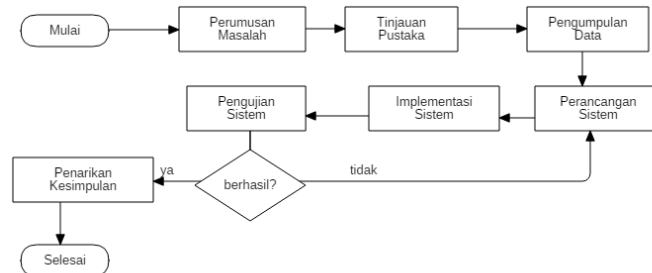
Agar masalah yang dibahas tidak menyimpang dari judul, sehingga penulis membatasi masalah yang akan dibahas seperti metode Dempster-Shafer diterapkan untuk merekomendasikan apakah seseorang terindikasi memiliki masalah pada kesehatan mental atau tidak, instrumen kesehatan mental yang dipilih user hanya kondisi yang dirasakan user selama 30 hari terakhir, dan perancangan program aplikasi sistem pakar ini menggunakan database management system MySQL dan Bahasa pemrograman PHP.

Tujuan dari penelitian ini adalah untuk merancang aplikasi sistem pakar berbasis web untuk merekomendasi apakah seseorang memiliki kecenderungan mempunyai masalah pada kesehatan mentalnya, informasi terkait dengan kesehatan mental serta meningkatkan kesadaran akan kemungkinan masalah pada kesehatan mental yang sebenarnya dialami pada diri individu namun tidak pernah disadari serta sistem ini dapat menghemat waktu dan biaya dan melakukan pengujian akurasi perbandingan antara hasil sistem dan hasil pakar serta pengujian usability pada sistem. Hasil dari penelitian ini diharapkan sistem ini dapat membantu masyarakat untuk merekomendasi kemungkinan apakah seseorang memiliki kecenderungan masalah pada kesehatan mentalnya, informasi terkait bagaimana menjaga kesehatan mental, dan pengujian akurasi perbandingan antara hasil sistem dan hasil pakar serta pengujian usability pada sistem.

2. METODE PENELITIAN

2.1 Alur Penelitian

Tahapan awal dalam penelitian ini adalah merumuskan masalah deteksi dini kesehatan mental yang ingin kita selesaikan dengan sistem pakar. Selanjutnya adalah mengumpulkan sumber bacaan dari berbagai jurnal, buku, dan sumber lain yang berkaitan dengan penelitian ini.. Setelah tinjauan pustaka, dilakukan pengumpulan data untuk menunjang fakta-fakta yang akan digunakan sebagai informasi dalam sistem ini dengan melakukan observasi dan wawancara pada pakar. Lalu tahap perancangan sistem yaitu meliputi perancangan desain arsitektur sistem, perancangan, basis data, antar muka, dan pengujian sistem . Setelah dirancang, sistem akan dibuat berdasarkan perancangan yang sudah dibuat. Selanjutnya, pengujian dilakukan untuk mengetahui kelayakan sistem. Ketika dalam pengujian, jika hasil tingkat keakuratannya tidak sesuai dengan harapan maka akan dilakukan analisa kembali pada perancangan sistem. Jika pengujian berhasil, kesimpulan akan diambil. Berikut gambar alur penelitian yang meliputi beberapa langkah:



Gambar 1. Alur Penelitian

2. 2 Dempster Shafer

Teori *dempster-shafer* adalah suatu teori matematika untuk pembuktian berdasarkan belief functions (fungsi kepercayaan) dan plausible reasoning (pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa [2]. Secara umum teori Dempster-Shafer ditulis dalam suatu interval:

[Belief, Plausibility]

Belief (Bel) merupakan ukuran kepercayaan evidence untuk mendukung suatu himpunan. Nilai *Belief* diberikan oleh pakar berdasarkan ilmu pengetahuan pakar terhadap jenis evidence. Nilai Bel ini berada dalam kisaran [0...1]. Jika nilai Bel = 0 artinya tidak ada evidence dan Bel = 1 artinya kepastian. Fungsi belief dapat diformulasikan dan ditunjukkan pada persamaan (1) [2].

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \quad (1)$$

Plausibility (Plau) juga bernilai 0 sampai 1, jika kita yakin terhadap X, maka dapat dikatakan bahwa $Bel(X)=1$, dan $Pls(X)=0$. *Plausibility* mengurangi tingkat kepercayaan dari evidence. *Plausibility* (Pls) dinotasikan pada persamaan (2).

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \subseteq X} m(Y) \quad (2)$$

dimana:

$Bel(X) = \text{Belief}(X)$.

$Pls(X) = \text{Plausibility}(X)$.

$m(Y) = \text{mass function}$.

Pada teori *Dempster-Shafer* adanya *frame of discernment* yang dinotasikan dengan θ dan *mass function* yang dinotasikan dengan m . *frame of discernment* adalah semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment*. Sedangkan *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (instrumen kesehatan mental), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m). Pada sistem ini, terdapat sejumlah *evidence* (instrumen kesehatan mental) yang akan digunakan pada faktor ketidakpastian dalam menentukan hasil deteksi dini. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination* pada persamaan (3) [2].

$$m_z(Z) = \frac{\sum_{X \cap Y = Z} m_x(X) \cdot m_y(Y)}{1 - K} \quad (3)$$

Dimana :

$m_z(Z)$ = *mass function* dari evidence z.

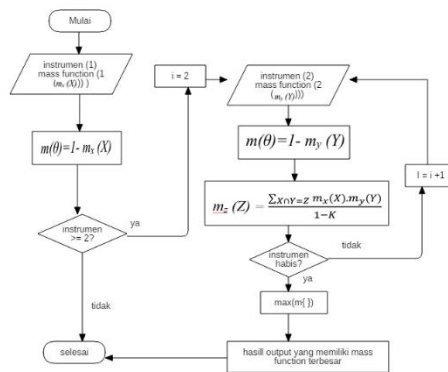
$m_x(X)$ = *mass function* dari evidence x.

$m_y(Y)$ = *mass function* dari evidence y.

$\sum_{X \cap Y = Z} m_x(X) \cdot m_y(Y)$ = jumlah dari irisan pada perkalian m_x dan m_y .

K = perkalian dari *mass function* yang mengalami konflik *evidence* bila irisannya kosong.

Berikut penjelasan alur metode *dempster-shafer* yang dapat dilihat pada gambar 2.



Gambar 2 Alur Metode Dempster-Shafer

Keterangan

i = instrumen (*evidence*).

m = nilai *mass function* / kepercayaan.

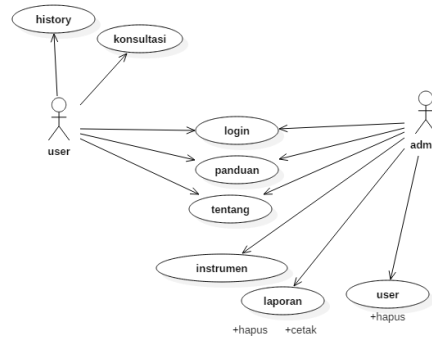
X dan Y = *output*.

I = iterasi

2.3 Perancangan Sistem

2.3.1 Usecase Diagram

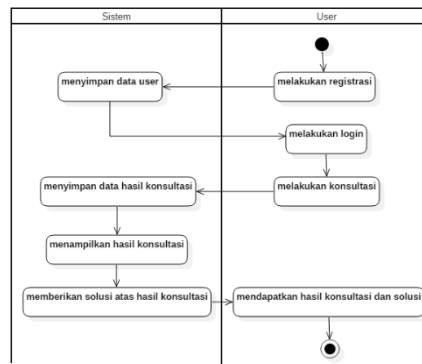
Pada gambar 3, usecase diagram menggambarkan bahwa admin mendapatkan hak akses penuh pada sistem seperti dapat mengakses instrumen, menghapus dan mencetak laporan, serta menghapus user. Sedangkan pada user hanya dapat hak akses untuk mengakses konsultasi dan histori konsultasi.



Gambar 3 Usecase Diagram

2.3.2 Activity Diagram

Pada gambar 4 adalah activity diagram dari sistem pakar deteksi dini kesehatan mental yang merupakan gambaran diagram alur aktivitas antar user dan sistem dari sistem pakar yang digunakan agar user dapat mengetahui dengan mudah bagaimana alur dari sistem tersebut.



Gambar 4 Activity Diagram

3. HASIL DAN PEMBAHASAN

3.1 Basis Pengetahuan

Basis pengetahuan merupakan komponen yang berisi pengetahuan yang berasal dari pakar, berisi sekumpulan fakta dan aturan[6]. Berikut merupakan pembentukan aturan output, instrumen kesehatan mental, dan nilai belief yang didapat dari hasil wawancara bersama psikolog berdasarkan persentase kemungkinan seseorang mengalami masalah kesehatan mental dalam setiap instrumen.

Tabel 1 Output Indikasi Kesehatan mental

Kode Output	Nama Output
O1	Terindikasi Memiliki Masalah Pada Kesehatan Mental
O2	Tidak Terindikasi Memiliki Masalah Pada Kesehatan Mental

Tabel 2 Instrumen dan Nilai Belief Kesehatan Mental

Kode	Instrumen	Belief
I1	Sering Menderita Sakit Kepala	0.8
I2	Tidak Memiliki Nafsu Makan	0.8
I3	Sulit Tidur	0.7
I4	Mudah Takut	0.5
I5	Merasa Tegang, Cemas, Atau Kuatir	0.6
I6	Tangan Mudah Gemetar	0.4
I7	Pencernaan Terganggu/Buruk	0.7
I8	Sulit Berpikir Jernih	0.5
I9	Merasa Tidak Bahagia	0.6
I10	Sering Menangis	0.5
I11	Merasa Sulit Untuk Menikmati Kegiatan Sehari-Hari	0.4
I12	Sulit Mengambil Keputusan	0.4
I13	Pekerjaan Sehari-Hari Terganggu	0.6
I14	Tidak Mampu Melakukan Hal-Hal Yang Bermanfaat Dalam Hidup	0.6
I15	Kehilangan Minat Pada Berbagai Hal	0.6
I16	Merasa Tidak Berharga	0.6
I17	Mempunyai Keinginan Untuk Mengakhiri Hidup	0.7
I18	Merasa Lelah Sepanjang Waktu	0.7
I19	Mengalami Rasa Tidak Enak Diperut	0.8
I20	Mudah Lelah	0.8
I21	Lebih Sering Menggunakan Alkohol/Zat Terlarang Dari Biasanya	0.7
I22	Merasa Seseorang Bermaksud Mencedakai Diri Anda	0.8
I23	Merasa Ada Sesuatu Yang Mengganggu Pikiran Anda	0.8
I24	Mendengar Suara-Suara Yang Tidak Didengar Orang Lain	0.8
I25	Mengalami Mimpi Bencana Atau Seakan Bencana Itu Muncul Kembali	0.8

I26	Menghindari Berbagai Kegiatan, Tempat, Orang, Atau Pikiran Yang Mengingatkan Akan Bencana Tersebut	0.7
I27	Kurang Tertarik Terhadap Teman-Teman Atau Kegiatan Sehari-Hari	0.8
I28	Merasa Sangat Sedih Apabila Berada Dalam Situasi Yang Mengingatkan Akan Bencana Tersebut?	0.9
I29	Sulit Menghayati Dan Mengeluarkan Perasaan	0.7

Dibawah ini terdapat tabel 3 yang berisikan *rules* terkait instrumen dan output.

Tabel 3 Rules terkait instrumen dan output

Kode	Nama Output	Rule
O1	Terindikasi Memiliki Masalah Pada Kesehatan Mental	I4 and I5 and I6 and I8 and I9 and I10 and I11 and I12 and I13 and I15 and I16 and I17 and I21 and I22 and I23 and I24 and I25 and I26 and I27 and I28 and 29
O2	Tidak Terindikasi Memiliki Masalah Pada Kesehatan Mental	I1 and I2 and I3 and I7 and I18 and 19 and 120

3.2 Analisis Metode Dempster-Shafer

Dilakukan pengujian konsultasi, user memilih 3 instrumen yang dirasakan selama 30 hari

1. Sering menderita terakhir yaitu:sakit kepala (I1)
2. Tidak memiliki nafsu makan (I2)
3. Mempunyai keinginan untuk mengakhiri hidup (I17)

Instrumen Pertama:

Instrumen 1 merupakan instrumen untuk output Tidak Terindikasi(O2) dengan nilai belief. Instrumen 1 akan menjadi mass function m1 dengan menggunakan persamaan (2), maka

$$m_1(O2) = 0.8$$

$$m_1(\theta) = 0.2$$

Instrumen Kedua:

Instrumen 2 merupakan instrumen untuk output Tidak Terindikasi(O2) dengan nilai belief. Instrumen 2 akan menjadi mass function m1 dengan menggunakan persamaan (2), maka

$$m_2(O2) = 0.8$$

$$m_2(\theta) = 0.2$$

Hasil mass function m_1 dan m_2 akan dihitung dengan menggunakan rumus kombinasi pada persamaan (3). Untuk mempermudah proses perhitungan dapat digunakan tabel seperti pada tabel 4.

Tabel 4 Perhitungan Iterasi Pertama Dempster-Shafer

	$m_2(O2) 0.8$	$m_2(\theta) 0.2$
$m_1(O2) 0.8$	(O2) 0.64	(O2) 0.16
$m_1(\theta) 0.2$	(O2) 0.16	(θ) 0.04

$$m_3(O2) = \frac{\sum_{X \cap Y = Z} m_x(X).m_y(Y)}{1-K} = \frac{(O2)}{1-K} = \frac{0.64+0.16+0.16}{1-0} = 0.96$$

$$m_3(\theta) = 1 - 0.96 = 0.04$$

Instrumen Ketiga:

Instrumen 3 merupakan instrumen untuk output Terindikasi(O1) dengan nilai *belief*. Instrumen 3 akan menjadi *mass function* m_4 dengan menggunakan persamaan (2), maka

$$m_4(O1) = 0.7$$

$$m_4(\theta) = 0.3$$

Hasil *mass function* m_3 dan m_4 akan dihitung dengan menggunakan rumus kombinasi pada persamaan (3). Untuk mempermudah proses perhitungan dapat digunakan tabel seperti pada tabel 5.

Tabel 5 Perhitungan Iterasi kedua Dempster-Shafer

	$m_4(O1) 0.7$	$m_4(\theta) 0.3$
$m_3(O2) 0.96$	(K) 0.672	(O2) 0.288
$m_3(\theta) 0.04$	(O1) 0.028	(\theta) 0.012

$$m_5(O1) = \frac{\sum_{X \cap Y = Z} m_x(X).m_y(Y)}{1-K} = \frac{(O1) 0.028}{1-0.672} = 0.085$$

$$m_5(O2) = \frac{\sum_{X \cap Y = Z} m_x(X).m_y(Y)}{1-K} = \frac{(O2) 0.288}{1-0.672} = 0.878$$

$$m_5(\theta) = 1 - (0.085+0.878) = 0.037$$

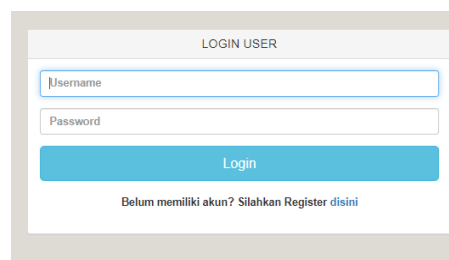
Nilai keyakinan yang terbesar terdapat pada (O2) yaitu tidak terindikasi memiliki masalah pada kesehatan mental dengan nilai 0.878 sehingga memiliki persentase 87.8% yang didapat dari $0.878 \times 100\%$.

3.3 Tampilan Sistem

Desain tampilan sistem pakar deteksi dini kesehatan mental menggunakan metode *dempster shafer* berbasis web dapat dilihat dari penjelasan berikut

3.3.1 Halaman Login

Halaman login merupakan halaman awal user sebelum melakukan konsultasi



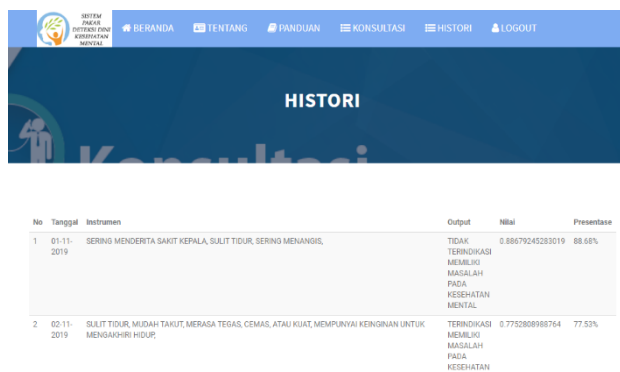
Gambar 5 Halaman Login

3.3.2 Halaman Beranda

Halaman beranda memuat menu yang dapat diakses.

3.3.5 Halaman Histori

Halaman histori menampilkan data rekap hasil konsultasi yang dilakukan oleh user sebelumnya



Gambar 9 Halaman Histori

3.4 Pengujian Sistem

Berdasarkan hasil pengujian akurasi perbandingan antara hasil sistem dan hasil konsultasi pakar melalui kuesioner yang diberikan kepada psikolog sebanyak 100 data uji dan diambil sampel sebanyak 40 data uji serta pada kolom diagnosa sistem terdapat persentase output konsultasi sehingga didapatkan hasil sebagai berikut:

Tabel 6 Hasil Simulasi Pengujian Akurasi

No	Indikator	Diagnosa Pakar	Diagnosa Sistem	Hasil
1	I1, I2, I17	Tidak Terindikasi	Tidak Terindikasi (87,8 %)	Sesuai
2	I1, I2, I14, I5	Tidak Terindikasi	Tidak Terindikasi (79.34%)	Sesuai
3	I3, I17, I20, I26, I27	Terindikasi	Terindikasi (94.24%)	Sesuai
4	I1, I3, I10	Tidak Terindikasi	Tidak Terindikasi (88.68%)	Sesuai
5	I28, I29	Terindikasi	Terindikasi (97%)	Sesuai
6	I1, I8, I9, I24	Terindikasi	Terindikasi (82.76%)	Sesuai
7	I3, I5, I17	Tidak Terindikasi	Terindikasi (68.75)	Tidak Sesuai
8	I3, I7, I10	Tidak Terindikasi	Tidak Terindikasi (83.49%)	Sesuai
9	I11, I14, I26	Terindikasi	Terindikasi (92.8%)	Sesuai
10	I22, I23	Terindikasi	Terindikasi (96%)	Sesuai
11	I19, I27, I28	Terindikasi	Terindikasi (90.74%)	Sesuai
12	I3, I4, I5, I8, I9, I10, I11, I14, I15	Terindikasi	Terindikasi (99.36%)	Sesuai
13	I5, I6, I9, I24	Terindikasi	Terindikasi (98.08%)	Sesuai
14	I16, I20, I27	Terindikasi	Terindikasi (69.7%)	Sesuai
15	I2, I7, I10	Tidak Terindikasi	Tidak Terindikasi (88.68%)	Sesuai
16	I9, I11, I14, I15, I24	Terindikasi	Terindikasi (99.23%)	Sesuai
17	I9, I18, I20	Tidak Terindikasi	Tidak Terindikasi (86.24%)	Sesuai
18	I4, I8, I9, I11, I12, I18	Tidak Terindikasi	Terindikasi (88.93%)	Tidak Sesuai
19	I14, I18	Tidak Terindikasi	Tidak Terindikasi (48.28%)	Sesuai
20	I12, I13, I23	Terindikasi	Terindikasi (95.2%)	Sesuai
21	I5, I6, I7, I8, I9, I10, I11, I12	Terindikasi	Terindikasi (97.18 %)	Sesuai
22	I17, I18, I19, I20, I21, I22, I23, I24, I25	Terindikasi	Terindikasi (98.81%)	Sesuai
23	I19, I26, I27	Terindikasi	Terindikasi (75.81%)	Sesuai
24	I26, I27, I28	Terindikasi	Terindikasi (99.4%)	Sesuai
25	I16, I19, I22	Terindikasi	Terindikasi (69.7%)	Sesuai
26	I17, I18, I19	Tidak Terindikasi	Tidak Terindikasi (98.8%)	Sesuai
27	I20, I21	Terindikasi	Tidak Terindikasi (54.55%)	Tidak Sesuai

28	I3, I5	Tidak Terindikasi	Tidak Terindikasi (48.28%)	Sesuai
29	I6, I7, I29	Terindikasi	Terindikasi (57.5%)	Sesuai
30	I22, I23, I26	Terindikasi	Terindikasi (98.8%)	Sesuai
31	I3, I12	Tidak Terindikasi	Tidak Terindikasi (58.33%)	Sesuai
32	I9, I10, I11, I12, I17, I22	Terindikasi	Terindikasi (99.57%)	Sesuai
33	I25, I26, I28	Terindikasi	Terindikasi (99.4%)	Sesuai
34	I3, I10, I19, I20	Tidak Terindikasi	Tidak Terindikasi (97.63%)	Sesuai
35	I3, I4	Terindikasi	Terindikasi (53.85%)	Sesuai
36	I21, I23, I29	Terindikasi	Terindikasi (998.2%)	Sesuai
37	I7, I8, I19	Tidak Terindikasi	Tidak Terindikasi (88.68%)	Sesuai
38	I1, I2, I3, I4, I5, I6, I7, I8, I9	Terindikasi	Tidak Terindikasi (86.92%)	Tidak Sesuai
39	I1, I7, I9	Tidak Terindikasi	Tidak Terindikasi (86.24%)	Sesuai
40	I7, I14, I17, I18, I27, I28, I29	Terindikasi	Terindikasi (99.21%)	Sesuai

Dari 20 data sampel pengujian akurasi perbandingan antara hasil sistem dan hasil pakar, maka didapatkan:

$$\frac{\text{Banyak data uji sampel yang sesuai}}{\text{Banyak data uji sampel}} \times 100\% \quad (4)$$

$$\frac{36}{40} \times 100\% = 90\%$$

Sedangkan berdasarkan hasil 100 data pengujian akurasi perbandingan antara hasil sistem dan hasil pakar pada tabel 6 didapatkan:

$$\frac{\text{Banyak data uji yang sesuai}}{\text{Banyak data uji}} \times 100\% \quad (5)$$

$$\frac{94}{100} \times 100\% = 94\%$$

Dari perhitungan persentase tingkat keakurasian sistem yang didapatkan dari 100 data uji perbandingan hasil sistem dan hasil pakar mengalami kenaikan persentase dibandingkan dari tingkat keakurasian sistem yang didapatkan dari 40 data uji sampel dan didapatkan sebanyak 94 data yang sesuai dari 100 data uji dan 6 data yang tidak sesuai dikarenakan instrumen yang dipilih masih sedikit dan kurang spesifik, sehingga dapat dikatakan bahwa sistem pakar deteksi dini kesehatan mental ini memiliki tingkat keberhasilan sebesar 94% yang berarti sistem sudah dapat berjalan dengan baik.

3.5 Pengujian Usability

Pada pengujian *usability* sistem menggunakan 9 pertanyaan dan dilakukan penyebaran kuesioner yang diberikan pada 31 koresponden, selanjutnya dilakukan rekap nilai pada setiap pertanyaan terhadap hasil kuesioner yang telah disebar. Berdasarkan persentase hasil *usability testing* dalam setiap pertanyaan memiliki skala 1-5, pada skala 1 (sangat tidak mudah), 2 (tidak mudah), 3 (cukup mudah), 4 (mudah), 5 (sangat mudah). Selanjutnya untuk mendapatkan nilai total dari seluruh aspek digunakan persamaan (6).

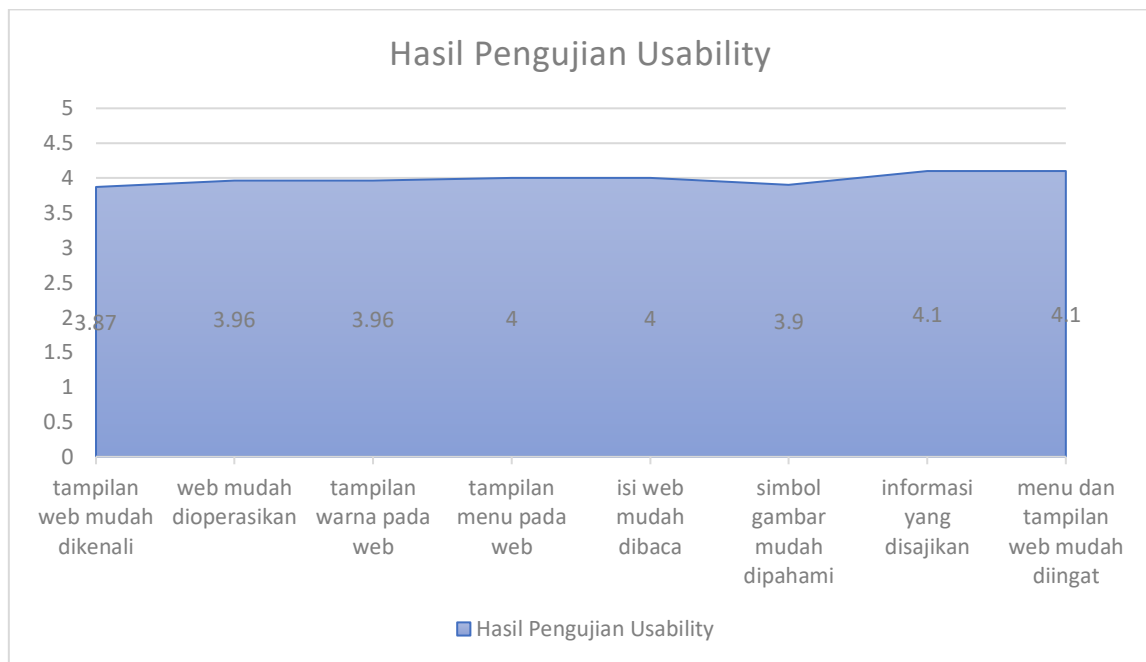
$$\frac{\text{Jumlah Nilai dari Setiap Pertanyaan}}{\text{Jumlah Pertanyaan}} \quad (6)$$

$$\frac{3.87+3.96+3.96+4+3.93+4+3.90+4.1+4.1}{9} = 3.95$$

maka diperoleh rekap nilai *usability* yang terlihat pada tabel 7.

Tabel 7 Hasil Pengujian Usability

NO	PERTANYAAN	NILAI
1	Apakah tampilan web mudah dikenali?	3.87
2	Apakah web mudah dioperasikan?	3.96
3	Apakah tampilan warna pada web enak dilihat & tidak membosankan?	3.96
4	Apakah tampilan menu dalam web mudah dikenali?	4
5	Apakah fitur halaman web mudah dicari?	3.93
6	Apakah isi web yang ada mudah dibaca?	4
7	Apakah simbol-simbol gambar mudah dipahami?	3.90
8	Apakah informasi yang disajikan sudah sesuai dengan kebutuhan dan mudah dipahami?	4.1
9	Apakah menu dan tampilan halaman web mudah diingat?	4.1
TOTAL		3.95



Hasil rekap nilai pengujian usability sistem menunjukkan keseluruhan pertanyaan memiliki nilai *usability* 3.95 dari nilai total range 5, sehingga dapat dikatakan bahwa sistem informasi yang disajikan kepada pengguna sudah baik dan mudah digunakan oleh semua kalangan.

4. KESIMPULAN

Dari hasil penelitian tentang sistem pakar deteksi dini kesehatan mental berbasis web ini, dapat diambil kesimpulan yaitu sistem ini memiliki keunggulan dibandingkan penelitian sebelumnya dimana sistem pada sisi pengguna memiliki data rekap histori hasil konsultasi yang dapat digunakan sebagai perkembangan hasil konsultasi terkait kesehatan mental pengguna. Selain itu, berdasarkan pengujian akurasi sistem perbandingan antara hasil sistem dan hasil pakar, sistem memiliki tingkat keberhasilan sebesar 94% dan pengujian usability yang didapatkan dari kuesioner memiliki nilai usability sebanyak 3.95 dari total nilai 5 yang artinya sistem informasi yang disajikan kepada pengguna cukup baik serta sistem pakar ini dinilai layak untuk direkomendasikan sebagai deteksi dini kesehatan mental dengan menggunakan metode Dempster-Shafer. Namun, sistem pakar ini tidak dapat dijadikan sebagai final decision. Konsultasi resmi dengan Pakar/Psikolog tetap menjadi faktor utama dalam penyelesaian masalah dalam kesehatan mental.

5. SARAN

Diharapkan pada penelitian selanjutnya dapat mengembangkan sistem dengan mengintegrasikan sistem kepada psikolog yang ada di lokasi sekitar pengguna melakukan konsultasi sehingga lebih memudahkan pengguna dalam melakukan penanganan lanjut serta memberikan solusi yang lebih rinci terkait penanganan masalah pada kesehatan mental. Selain itu, diharapkan sistem dapat dikembangkan dari sisi platform dari web menjadi berbasis android/ios agar semakin memudahkan pengguna untuk mengakses sistem.

DAFTAR PUSTAKA

- [1] Dumilah, A., Misnaniarti., Marisa, R, 2018. Analisis Situasi Kesehatan Mental Pada Masyarakat Di Indonesia Dan Strategi Penanggulangannya. *Jurnal Ilmu Kesehatan Masyarakat*, No.1, Vol.9, 1-10.
- [2] Dina, H., Fitri, B, 2018. Sistem Pakar untuk Mendiagnosis Gangguan Anak Menggunakan Metode Dempster Shafer. *J-COSINE*, No.2, Vol.2, 71-79.
- [3] Doddy, T.Y., Abdul, F., Sunardi, 2019. Implementasi Sistem Pakar Diagnosa Gangguan Kepribadian Menggunakan Metode Dempster Shafer. *Jurnal Sistem Informasi Bisnis*, No.1, Vol.9, 25-31.
- [4] Khairunnisa, N, Huzaeni, Akmalul, F, 2018, Sistem Pakar Gangguan kepribadian (Personality Disorders) Menggunakan Metode Dempster Shafer Berbasis Web, Proceeding Seminar Nasional Politeknik negeri Lhokseumawe, Medan, september 2018.
- [5] Reza, S., Cucu, S., Syamsul, B, 2018. Implementasi Metode Dempster Shafer pada Sistem Pakar Diagnosa Infeksi Penyakit Tropik Berbasis Web. *Jurnal Coding*, No.3, Vol.6, 97-106.
- [6] Aprilia, I.F., Tedy, R., Syamsul, B, 2018. Aplikasi Sistem Pakar Diagnosa Penyakit pada Anak Dengan Inference Forward Menerapkan Metode Dempster Shafer Berbasis Web. *Jurnal Coding*, No.3, Vol.6, 25-35.
- [7] Doddy, T.Y., Abdul, F., Sunardi, 2019. Comparative Analysis of Dempster Shafer Method and Certainty Factor Method On Personality Disorders Expert System. *Scientific Journal of Informatics*, No.1, Vol.6, 12-22.
- [8] Erwin, P, 2018. Comparative Analysis of Dempster Shafer Method and Certainty Factor Method For Diagnose Stroke Diseases. *International Journal of Artificial Intelligence Research*, No.1, Vol.2, 37-41.
- [9] David, 2016. Sistem Pakar Diagnosa Penyakit Gangguan Jiwa dengan Dempster Shafer. Konferensi Nasional Sistem Informasi (KNSI) 2016. pp.197-203.

Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04.

Implementation Load Balancing Web Server Using Apache on Ubuntu 16.04.

Kresno Wibowo*¹, Iskandar Fitri², Deny Hidayatullah³

^{1,2,3}Universitas Nasional; Jl. Sawo Manila, RT.14/RW.3, Ps. Minggu, Kec. Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12520, (021) 7806700

e-mail: *kresnowibowo4598@gmail.com, tektel2001@yahoo.com, faradeari@gmail.com

Abstrak

Saat banyak pengguna yang mengakses suatu situs web pada saat yang bersamaan. Maka saat itu bisa membuat web server menampung beban yang berlebih karena tidak bisa menampung lagi banyaknya request yang diterima. Dan itu biasa terjadi saat web server tunggal atau single. Namun dengan menggunakan metode load balancing Round Robin dan juga Haproxy sebagai load balancer dapat membantu mengatur pembagian beban pada setiap server. Dengan pengujian yang dilakukan pada 250 user, 500 user, 750 user, sampai 1000 user dan menggunakan web statis. Hingga dapat melihat penggunaan CPU yang digunakan dengan beban user yang diuji pada saat setelah melakukan. Hasil dari pengujian dengan menggunakan haproxy berjalan dengan baik daripada single server dan dua server. Dengan menggunakan apache benchmark dalam menghitung Throughput dan Time per request antara single server, dua server serta menjalankan semua server menggunakan Haproxy. Untuk pengujian presentasi pada CPU yang dipakai pada 250 user yaitu 31,9% yang terendah namun terdapat peningkatan pada pengujian 1000 user yaitu mencapai 54,5%. Hasil pengujian pada Dua server dalam pengujian 250 user, 500 user, 750 user, dan 1000 user menghasilkan hasil 62.5%-68.7%. Dan dengan pengujian pada single server pemakaian RAM dalam pengujian 250 user, 500 user, 750 user, serta 1000 user yaitu mencapai sekitar 91%-96%. Dengan hasil pengujian yang lebih baik menggunakan load balancing haproxy daripada menggunakan single server maupun dua server. Karena penggunaan single server dan dua server hampir mencapai overload. Yang berarti waktu yang dibutuhkan untuk menangani beban tersebut ialah load balancing menggunakan haproxy lebih baik daripada single server untuk lebih cepat menangani request, time per second dan penggunaan CPU.

Kata kunci — Apache, Haproxy, Load Balancing, Virtual Box

Abstract

When many users access a website at the same time. So when it can make the web server to accommodate excessive loads because it can no longer accommodate the number of requests received. And that usually happens when a single web server or single. However, using the Round Robin load balancing method and also Haproxy as a load balancer can help regulate load sharing on each server. With testing done on 250 users, 500 users, 750 users, up to 1000 users and using a static web. So you can see the CPU usage that is used by the user load that is

tested at the time after performing. The results of testing using haproxy run better than a single server and two servers. By using Apache benchmark in calculating Throughput and Time per request between a single server, two servers and running all servers using Haproxy. For testing the presentation on the CPU that is used on 250 users is 31.9% the lowest but there is an increase in testing 1000 users, reaching 54.5%. The test results on two servers in testing 250 users, 500 users, 750 users, and 1000 users produced 62.5% -68.7% results. And by testing on a single server RAM usage in testing 250 users, 500 users, 750 users, and 1000 users which reached about 91% -96%. With better test results using haproxy load balancing than using a single server or two servers. Because the use of a single server and two servers almost reached overload. Which means the time needed to handle the load is load balancing using haproxy better than a single server to more quickly handle requests, time per second and CPU usage.

Keywords — Apache, Haproxy, Load Balancing, Virtual Box

1. PENDAHULUAN

Pada zaman sekarang yang sudah semakin maju era teknologi khususnya pada kemajuan bidang jaringan. Perkembangan tersebut terbilang cukup pesat setiap waktu. Meningkatnya kebutuhan akan informasi menuntut akses yang cepat untuk mendapatkan informasi – informasi terkini, salah satunya yang paling dominan mempengaruhi kecepatan akses suatu alamat website tertentu adalah server penyedia layanan. Server bisa dibilang sebagai pelayan karna tugasnya menerima request dari klien, permintaan datang dari klien yang sudah terhubung pada server yang ingin diakses, server merupakan pusat pengelolaan, semakin banyak klien semakin berat kerja yang dilakukan oleh server, sehingga dibutuhkan spesifikasi komputer khususnya server harus baik sehingga mampu melayani permintaan klien yang cukup banyak. Penelitian sebelumnya yang menggunakan dua server untuk melakukan load balancing menggunakan nbl manager dilakukan oleh penulis pertama, dkk. Kemudian penelitian yang selanjutnya juga menggunakan metode nth yang dilakukan oleh penulis pertama, dkk [1].

Permasalahan pada umumnya tidak jarang pengelola server kurang mengelola dengan baik web server. Oleh sebab itu sering terjadi server down dan overload akibat banyaknya user yang mengakses secara bersamaan. Karena menggunakan satu server untuk menampung banyaknya permintaan. Karena menggunakan single server tidak ada server lain yang bisa membantu saat terjadi overload [2].

Load Balancing adalah suatu teknik untuk membagi beban pada koneksi secara seimbang, agar koneksi dapat berjalan lancar dan optimal, memaksimalkan throughput, memperkecil untuk terjadinya overload pada salah satu jalur koneksi Load balancing juga membagi beban kerja yang dilakukan secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal. dalam hal ini masalah pada pemerataan beban server menjadi salah satu solusi untuk dapat menjawab permasalahan yang ada diatas[3].

Penelitian ini bertujuan mengimplementasikan load balancing pada web server untuk menjamin kestabilan server dengan cara manajemen yang baik. Manfaatnya Dapat memperkecil untuk terjadinya overload pada webserver. Dan dengan menggunakan load balancing dengan Algoritma Round Robin yang dapat membagi kinerja web server dengan merata. Karena dapat menjaga kestabilan server karena menggunakan lebih dari satu server.

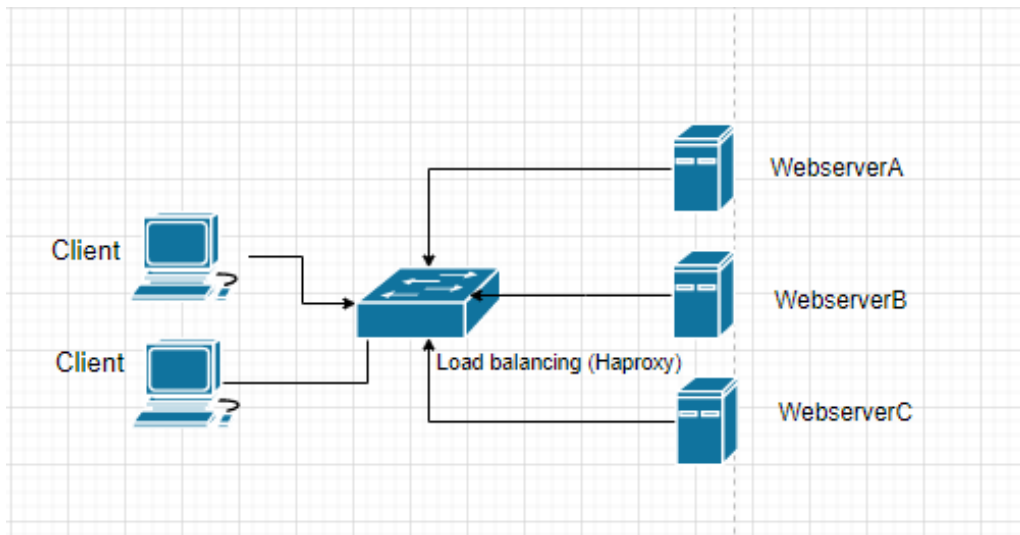
2. METODE PENELITIAN

Perancangan yang akan dibuat ialah membangun sebuah *web server* yang akan nantinya dijadikan *load balancer* yaitu *haproxy*. Metode yang diterapkan dalam penelitian ini menggunakan *Load balancing dengan Algoritma Round Robin*. *Haproxy* akan mengambil *source* dengan beban yang setara antara satu *server* dengan *server* lainnya.

Implementasi Load Balancing Web Server ...

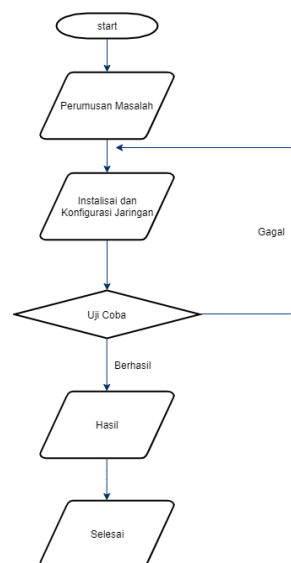
Apache adalah sebuah perangkat lunak *web server* yang menghubungkan antara *web* dengan *user*. *Apache* memudahkan pemilik *website* untuk membuat konten. *Apache* juga merupakan *software* lintas *platform* dan *server* nya dapat berjalan dengan baik.

Round Robin ialah sebuah *Algoritma* yang dipakai untuk mengambil *sourch* dengan beban yang setara antara *server* satu dengan *server* yang lainnya. Pembagian beban tersebut bertujuan agar saat banyak beban yang masuk secara bersamaan, kinerja server tidak down dan dapat berjalan dengan baik menerima beban.



Gambar 1. Topologi yang digunakan

Pada Gambar 1 menunjukkan jalan yang akan dilakukan pada sistem *load blancing* menggunakan 3 *server* dengan *OS ubuntu 16.04* yang masing-masing sudah diberi *web statis*. Lalu sebagai *load balancing* nya menggunakan *Haproxy* dengan *OS ubuntu 16.04* yang nantinya tersambung dengan *client*.



Gambar 2. Tahapan penilitan

Pada gambar 2 merupakan tahapan penelitian yang dilakukan mulai dari perumusan masalah lalu lanjut terhadap instalasi yang dilakukan seperti install *Ubuntu 16.04* lalu *Apache* sebagai *web server* dan *Haproxy* sebagai *Load Balancer*. Setelah itu melakukan konfigurasi terhadap tiga server yang diuji dengan *haproxy*. Lalu lakukan pengujian namun jika pengujian gagal kita kembali melakukan konfigurasi. Sampai pada jika uji coba yang dilakukan berhasil masuk kepada tahap hasil yang diuji lalu selesai.

2. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Pada proses pembuatan aplikasi absensi dalam penelitian ini dibutuhkan beberapa perangkat pendukung seperti perangkat lunak (*software*) dan perangkat keras (*hardware*), yang diantaranya:

1. Kebutuhan Hardware : Laptop Intel Core i3-6006U, 2.0GHz dengan RAM 8GB dan HDD 1TB
2. Kebutuhan Software : Virtualbox, Google Chrome, Draw.io, Ms. Excel.

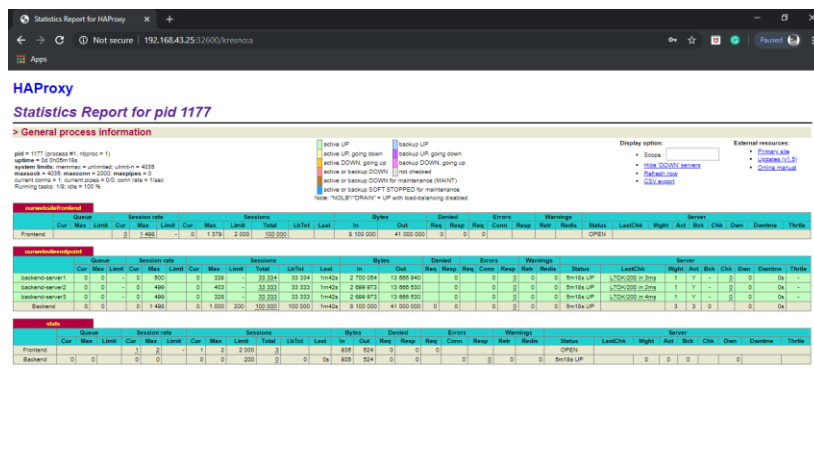
B. Tampilan Aplikasi

Pada proses ini tampilan web server dengan *ip address* yang digunakan untuk melakukan *Load Balancing*. *WebserverA* dengan *Ip Address* 192.168.43.10 sebagai server yaitu *Apache*. *WebserverB* dengan *Ip Address* 192.168.43.11 juga sebagai server yaitu *Apache*. *WebserverD* dengan *Ip Address* 192.168.43.13 juga sebagai server yaitu *Apache*. Lalu ada sebagai load balancer yaitu *webserverL* dengan *Ip Adress* 192.168.43.25 dengan *load balancer Haproxy* yang digunakan.

Tabel 1. IP Adress

Nama	Ip Address	Isi
webserverA	192.168.43.10	Apache
webserverB	192.168.43.11	Apache
webserverD	192.168.43.13	Apache
webserverL	192.168.43.25	Haproxy

Haproxy Stats ialah hasil yang akan ditampilkan saat sudah melakukan *load balancing web server*. Untuk mengetahui hasil tersebut maka terlebih dahulu melakukan *testing* terhadap *load balancer*.



Gambar 2. Statistik tampilan Haproxy

Gambar 2 Berikut adalah hasil yang ditampilkan dari hasil static yang nanti tiga web server tersebut akan dilakukan *load balancing* dengan memasukkan data yang akan diuji dengan diberi beban yang ingin diuji.

	Queue			Session rate			Sessions			Bytes		Denied		Errors		Warnings		Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bok	Chk	Dwn	Dwntme	Thrtle
backend-server1	0	0	-	0	500	0	338	-	33 334	33 334	1m42s	2 700 054	13 886 840	0	0	0	0	0	0	0	0	5m18s UP	L7OK/200 in 3ms	1	Y	-	0	0	0s	-
backend-server2	0	0	-	0	499	0	403	-	33 333	33 333	1m42s	2 869 973	13 886 830	0	0	0	0	0	0	0	0	5m18s UP	L7OK/200 in 2ms	1	Y	-	0	0	0s	-
backend-server3	0	0	-	0	499	0	328	-	33 333	33 333	1m42s	2 869 973	13 886 830	0	0	0	0	0	0	0	0	5m18s UP	L7OK/200 in 4ms	1	Y	-	0	0	0s	-
Backend	0	0	-	0	1 498	0	1 000	200	100 000	100 000	1m42s	8 100 000	41 000 000	0	0	0	0	0	0	0	0	5m18s UP		3	3	0	0	0	0s	-

Gambar 3. Hasil yang ditampilkan

Pada Gambar 3. Seperti yang dilihat pada gambar 3 ialah tampilan lebih jelas yang akan keluar setelah melakukan *load balancing* dengan menggunakan load balancer haproxy yaitu dapat menampilkan tiga server hasil dari pengujian.

C. Pengujian

Dalam proses pengujian yang akan dilakukan pada penelitian ini ialah memasukkan dari 250, 500, 750 sampai 1000 permintaan akses *user*. Pengujian tersebut meliputi hasil pengujian *request persecond*, *transfer rate*, dan penggunaan memory.

```

frontend ourwebsitefrontend
    bind *:80
    mode http
    default_backend ourwebsiteendpoint

backend ourwebsiteendpoint
    balance roundrobin
    option forwardfor
    http-request set-header X-Forwarded-Port [dst_port]
    http-request set-header X-Forwarded-Proto https if { ssl_fc }
    option httpchk HEAD / HTTP/1.1\r\nHost:localhost
    server backend-server1 192.168.43.11:8080 check
    server backend-server2 192.168.43.10:8080 check
    server backend-server3 192.168.43.25:8080 check

listen stats
    bind :32600
    stats enable
    stats uri /
    stats hide-version
    stats auth kresno:a
    
```

Gambar 4. Konfigurasi Haproxy

Pada gambar 4. Terlihat pada gambar 4 yaitu saat melakukan konfigurasi untuk menghubungkan antara *web server 1*, *web server 2*, *web server 3* dengan *load balancer*. Untuk melakukan *load balancing*, agar masuk ke konfigurasi pada Gambar 4 *nano /etc/haproxy/haproxy.cfg*, harus memasukkan *ip address web server* yang akan digunakan seperti Gambar 4 untuk *web server 1* -192.168.43.10, *web server 2* -192.168.43.11 *web server 3* -192.168.43.13 dan untuk haproxy sendiri memiliki *ip address* yaitu web serverL - 192.168.43.25.

```

root@ubuntu:/home/kresno# ab -n 10000 -c 1000 http://192.168.43.25/
    
```

Gambar 5. Apache benchmark

Pada gambar 5, terlihat pada gambar 5 ialah Untuk pengujian menggunakan *Apache Benchmark* memasukkan script seperti Gambar 5. Parameter *c* adalah untuk menguji jumlah *request* yang ingin diuji, sedangkan parameter *n* adalah jumlah koneksi yang nantinya dibuat untuk ke server tujuan, lalu masukkan *ip address* tujuan yang ingin diuji yaitu *load balancer* nya.

```

Server Software:      Apache/2.4.18
Server Hostname:     192.168.43.25
Server Port:         80

Document Path:       /
Document Length:     140 bytes

Concurrency Level:   1000
Time taken for tests: 14.219 seconds
Complete requests:   10000
Failed requests:     0
Total transferred:   4100000 bytes
HTML transferred:    1400000 bytes
Requests per second: 703.27 [#/sec] (mean)
Time per request:    1421.937 [ms] (mean)
Time per request:    1.422 [ms] (mean, across all concurrent requests)
Transfer rate:       281.58 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0    10  39.8      1   1000
Processing: 47  898  869.8    714 14097
Waiting:   40  896  869.9    713 14097
Total:    143  908  875.3    716 14164
    
```

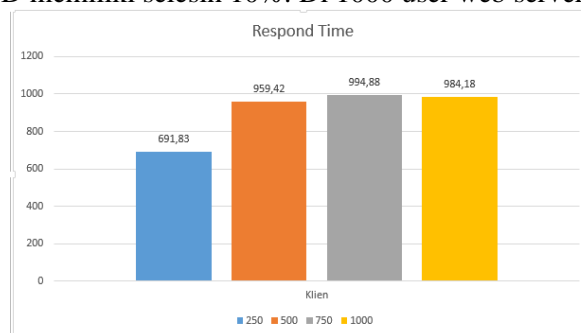
Gambar 6. Hasil Apache benchmark

Pada gambar 6, terlihat dari hasil pengujian dengan 1000 user ialah untuk hasil dari time per request menghasilkan angka 1.033, dan untuk transfer rate yaitu dengan hasil 387.49 (kbytes/sec), begitu juga dengan hasil dari *time takes for tests* menghasilkan 10.333 second, dan juga hasil dari *request per second* yaitu mencapai 967.77 (/sec).

Tabel 2. Pengujian kinerja RAM

Jumlah User	Nama Server	% of CPU	Memory
250	webserveraA	31,9%	227
	webserveraB	33,6%	225
	webserveraD	34,3%	226
500	webserveraA	42,6%	220
	webserveraB	42,3%	256
	webserveraD	47,2%	252
750	webserveraA	39,1%	275
	webserveraB	43,5%	274
	webserveraD	49,3%	273
1000	webserveraA	38,7%	256
	webserveraB	45,8%	274
	webserveraD	54,4%	277

Pada Tabel 2 adalah hasil penggunaan RAM dengan menggunakan htop setelah melakukan pengujian. Dari hasil pengujian % of RAM yang telah diringkas menjadi tabel agar mudah membacanya. Pembagian pada beban 3 *web server ubuntu* yang tersedia menggunakan *Haproxy* sebagai *load balancer* secara fungsional. Pada pengujian 250 user untuk pembagian beban yang hampir setara yaitu dengan hasil diangka 31-34%. Di 500 user juga hasil pembagian hampir setara yaitu dihasil 42-47%. Di 750 *user* mulai terjadi perbedaan karena web serverA menghasilkan 39,1% sementara web serverB 43,5% dan web serverD menghasilkan 49,3%. Dari web serverA dan D memiliki selesih 10%. Di 1000 user web serverA.



Gambar 6. Respond Time

Pada gambar 6, terlihat gambar 6 menunjukkan hasil dari saat melakukan *Apache Benchmark* untuk *request per second* yang pada pengujian 250 *user* yang berada pada angka 691.83 *per second*. untuk pengujian pada 500 *user* yaitu menghasilkan pada angka 959.43 *per second*. Pada pengujian untuk 750 *user* yaitu mencapai angka 994.88 *per second*. dan untuk pengujian terakhir yaitu pada 1000 *user* menghasilkan angka mencapai 984.18 *per second*.

Tabel 3. Pengujian single server

Nama	User	%ofCPU	Memori	Request per second
Webserver	250	91,9%	225	1061.31
Webserver	500	93,4%	229	1004.27
Webserver	750	95,5%	231	1060.485
Webserver	1000	96,0%	235	1313.786

Pada Tabel 3 menunjukkan kinerja *server* yang sangat besar jika menggunakan *single server*. Karena tidak ada yang bisa membantu saat banyak *user* mengakses dengan jumlah besar. Oleh karena itu akhirnya sudah hampir memenuhi kapasitas dan terjadi *overload*. Terlihat penggunaan yang digunakan pada pengujian 250 *user* yaitu mencapai angka 91,9%, dan pengujian 500 *user* juga mencapai hasil 93,4%, sementara untuk pengujian 750 *user* mencapai angka 95,5%, dan untuk pengujian 1000 *user* yaitu mencapai angka tertinggi yaitu 96,0%. Ini adalah hasil jika penggunaan *single server* saat melakukan menerima beban yang cukup besar. Angka yang dihasilkan cukup tinggi dikarenakan *server* hanya bekerja sendiri tanpa ada pembagian beban ke *server* lainnya.

Tabel 4. Pengujian dua server

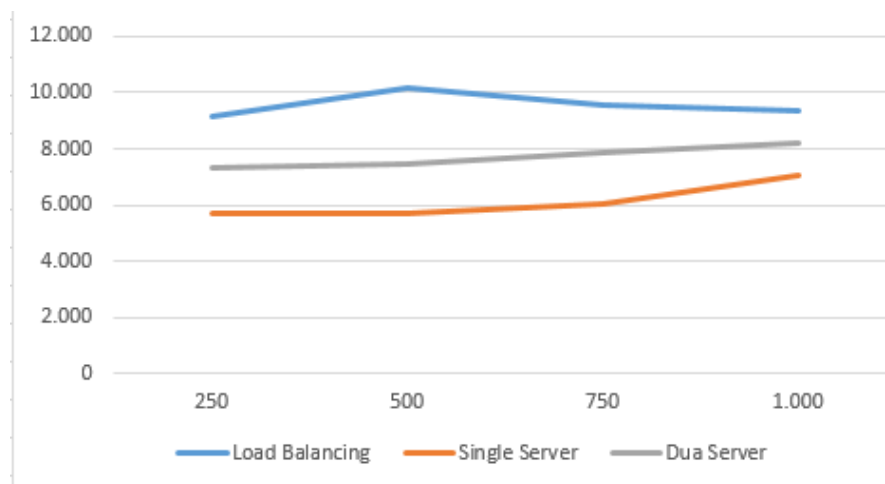
User	Nama	%ofCPU	Memori	Request per Second
250	webserverA	65,3%	223	1134.44
	webserverB	67,8%	225	
500	webserverA	63,4%	231	1087.50
	webserverB	66,2%	236	
750	webserverA	67,8%	232	1091.93
	webserverB	68,4%	229	
1000	webserverA	68,7%	237	1023.30
	webserverB	62,5%	235	

Pada tabel 4. Terlihat pengujian menggunakan *dua server* atau *satu server* yang dimatikan terlihat bahwa penggunaan RAM cukup meningkat jika tidak menggunakan semua *server*. Yaitu dengan hasil pengujian dari 250 *user* untuk *webserverA* mencapai 65,3% sementara untuk *webserverB* mencapai 67,8%. Untuk hasil pengujian dari 500 *user* untuk *webserverA* sendiri mencapai 63,4% sementara untuk *webserverB* mencapai 66,2%. Dan untuk hasil dari pengujian dari 750 *user* juga untuk *webserverA* mencapai 67,8% sedangkan untuk *webserverB* menghasilkan 68,4%. Dan untuk pengujian terakhir yaitu 1000 *user* untuk *webserverA* mencapai 68,7% dan hasil lebih rendah yaitu *webserverB* dengan hasil 62,5%. Meskipun menggunakan *dua server* tetap penggunaannya masih diatas 50% atau setengah lebih dari kapasitas seharusnya.

Tabel 5. Pengujian hasil load balancing

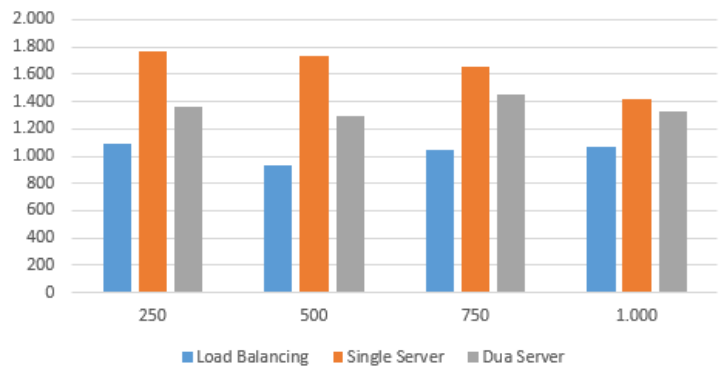
User	Time taken for test [second]	Request per second [#/sec]	Time per request [ms]	Transfer rate [kbytes/sec]
250	9.029	1083.40	0.923	433.78
500	9.060	1103.80	0.906	441.95
750	9.244	1081.76	0.924	433.13
1000	9.284	1077.16	0.928	431.28

Pada tabel 5 Menampilkan hasil dari *time taken for test*, *request per second*, *time per request*, dan juga *transfer rate*. Karena menggunakan *Algoritma Round Robin* maka bisa dilihat pembagian yang merata pada pengujian *load balancing*. Hasil pengujian pada tabel 5 dari *time taken for test* untuk pengujian 250 user ialah 9.029, 500 user ialah 9.060, 750 user ialah 9.244, dan untuk 1000 user mencapai 9.284. Dan hasil *request per second* dari 250 user ialah 1083.40/sec, 500 user mencapai 1103.80/sec, dan 750 user dengan hasil 1081.76/sec, dan untuk 1000 user mencapai 1077.16. untuk *time per request* hasil dari 250 user ialah 0.923, 500 user mencapai 0.906, 750 user mencapai angka 0.924, lalu untuk 1000 user mencapai angka 0.928. dan untuk hasil dari *transfer rate* dari hasil 250 user mencapai 433.78, untuk 500 user mencapai 441.95, dan 750 user mencapai 433.13, dan yang terakhir untuk 1000 user mencapai 431.28. Hasil tersebut cukup stabil saat melakukan load balancing karena pembagian yang cukup merata.



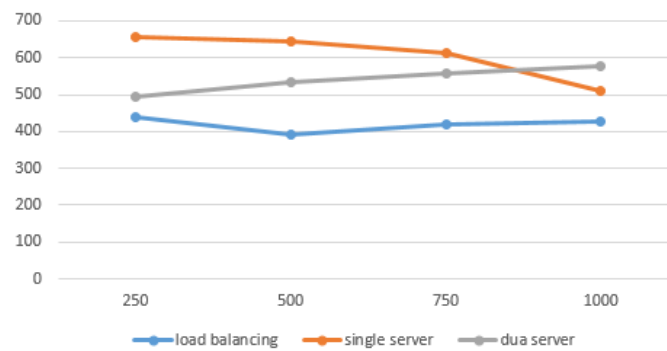
Gambar 7. Perbandingan Single Server dengan Load Balancing

Berdasarkan pada pengujian gambar 7, data yang telah dihasilkan dari waktu *respon* ialah diketahui dari 500 *client*, *single server* dapat melayani dengan baik setiap data yang masuk. Namun ketika diatas 500 *client* maka kinerjanya mulai berkurang akibat banyaknya data yang masuk. Begitupun dengan dua server tidak berbeda jauh dengan single server, karena pembagian bebannya hanya ke satu server tersisa walaupun tetap bisa membagi beban namun dapat berjalan dengan baik. Namun tidak dengan menggunakan *load balancing* karena ada pembagian beban yang dilakukan saat melayani jumlah user yang meningkat.



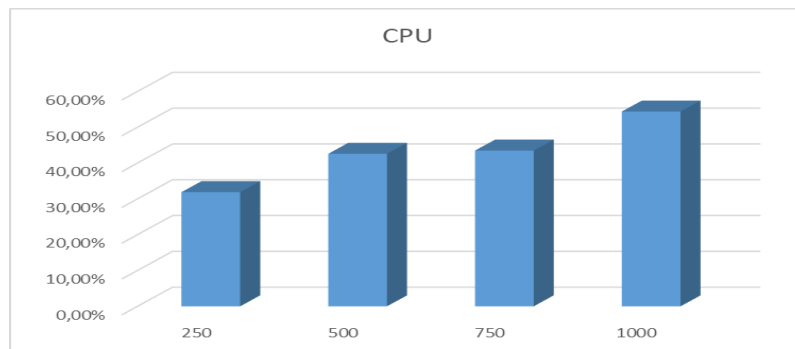
Gambar 8. Perbandingan *request*

Pada gambar 8 hasil dari pengujian *request*. Terlihat bahwa menggunakan *load balancing* sangat berjalan dengan baik dibandingkan dengan single server maupun dua server. Dengan menggunakan single server terlihat setiap request yang masuk maka sudah akan terjadinya overload dari server tersebut karena server hanya bekerja sendiri disaat masuknya request. Seperti halnya single server, dua server juga tidak berbeda jauh meskipun ada satu server lagi yang membantu disaat masuknya request. Dan menggunakan load balancing tiga server baru mulai dapat membagi beban dengan stabil dan berjalan cukup baik.



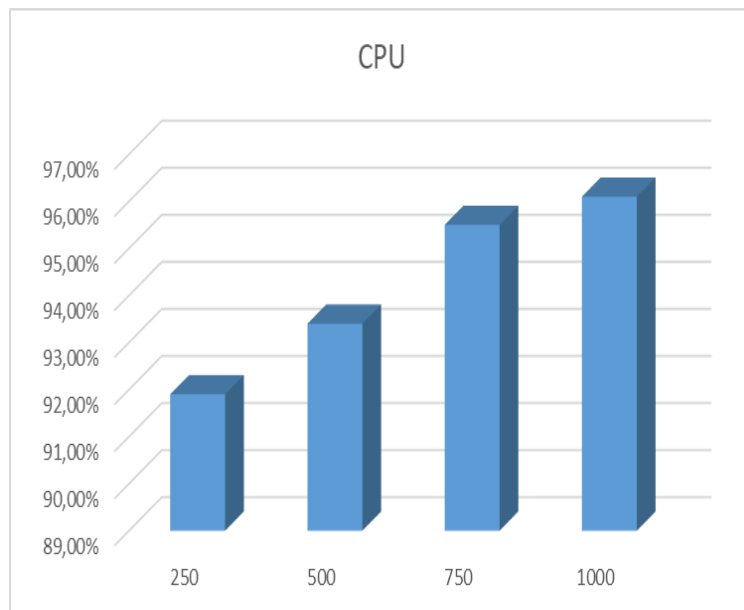
Gambar 9. Perbandingan Transfer Rate

Pada pengujian gambar 9. *Transfer Rate* yang menggunakan *load balancing* lebih stabil dan lebih kecil dibandingkan dengan *single server* yang cukup tinggi dan perbedaannya cukup terlihat. Sedangkan *dua server* berada diantara *single server* dan *load balancing*. Pada *load balancing* berada diangka 393.83-438.87[kbytes/sec]. dan untuk *dua server* 496.73-578.65[kbytes/sec]. sedangkan untuk pengujian *single server* yaitu 512.03-656.49[kbytes/sec].



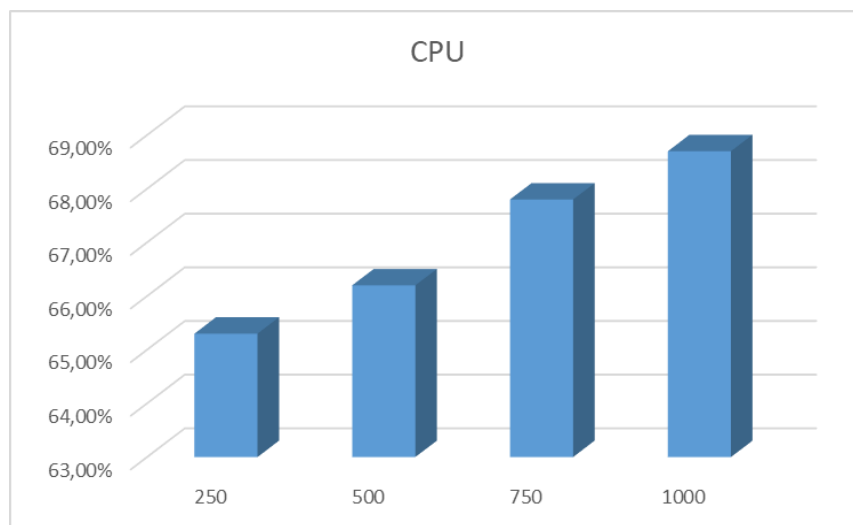
Gambar 10. Penggunaan Memori load balancing

Pada pengujian gambar 10. Penggunaan Memori pada pengujian load balancing terlihat lebih stabil dan tidak terlalu berbeda jauh. Oleh karena itu penggunaannya berjalan dengan baik dengan pengujian mulai dari 250 user yaitu 32,9%, 500 user mencapai 42.6%, pada 750 user berada pada angka 43.5%, dan untuk 1000 user 54.4%. setiap penggunaan yang lebih besar, maka terdapat peningkatan pemakaian memori pada setiap prosesnya.



Gambar 11. Penggunaan memori single server

Pada pengujian gambar 11. Pengujian pada *single server* terlihat pada pengujian tersebut bahwa penggunaan *single server* tidak berjalan dengan baik karena memiliki kapasitas yang tidak cukup besar. Seperti yang sudah diuji dengan 250 *user* yaitu berada pada angka 91,9%, sedangkan 500 *user* 93,4%, lalu 750 *user* mencapai 95,5%, dan untuk 1000 *user* mencapai 96,1%. terlihat pada pengujian menggunakan *single server* kapasitas yang digunakan hampir *overload*. Karena saat melakukan proses tidak ada yang dapat membantu saat banyaknya data yang masuk. Karena *single server* bekerja sendiri. Ini membuktikan bahwa *single server* tidak cukup untuk menampung data yang cukup banyak.



Gambar 12. Penggunaan memori dua server

Pada pengujian gambar 12. Terlihat pengujian pada *dua server* dapat membagi beban yang ditanggung walaupun masih di angka yang cukup tinggi yaitu masih diatas 50% yaitu yang berarti melebihi dari setengah kapasitas yang dapat digunakan. Seperti pengujian yang sudah dilakukan dengan pengujian mulai dari 250 user mendapatkan hasil 65,3%, 500 user mencapai 66,2%, dan 750 mencapai 67,8%, dan 1000 user mencapai 68,7%. Terlihat pada pengujian dua server pembagian beban cukup berjalan dengan baik walaupun hanya menggunakan *dua server*. Karena saat data masuk *server* tidak bekerja sendirian seperti halnya saat menggunakan *single server*.

4. KESIMPULAN

Dari hasil uji coba dalam pembahasan di atas, maka dapat disimpulkan:

- Secara fungsi *load balancing* menggunakan *Haproxy* berjalan dengan baik. Dengan menggunakan *tiga web server* dengan *Apache*.
- Untuk pengujian presentasi pada CPU yang dipakai pada 250 user yaitu 31,9% yang terendah namun terdapat peningkatan pada pengujian 1000 user yaitu mencapai 54,5%. Namun pada titik tertinggi itu *load balancing* menggunakan *Haproxy* berjalan dengan baik pada pembagian beban kepada 3 server tersebut. Dan dapat memperlihatkan bila semakin besar beban user yang mengakses, maka semakin besar juga kinerja *web* untuk bekerja.
- Untuk hasil pengujian menggunakan *single server* cukup terlihat beban yang besar terjadi. Terlihat pemakaian RAM dalam pengujian 250, 500, 750, 1000 user yaitu sekitar 91%-96% ini bisa terjadi *overload* karena penggunaan yang hampir 100% ketika dijalankan. Dan penggunaan *dua server* berada di sekitar 62.5%-68.7% pada hasil dari pengujian.

5. SARAN

Dari hasil penelitian yang telah dilakukan dengan menggunakan tiga server dengan satu load balancer cukup berjalan dengan baik. Yang data yang di uji muai dari 250, 500, 750 sampai 1000 user. Dengan hasil pada pengujian CPU yang dipakai pada pengujian 250 user yaitu 31,9% yang terendah namun mendapatkan peningkatan pada pengujian 1000 user 54,5%. Namun pada penggunaan *single server* terlihat kapasitas yang hampir *overload* karena server hanya bekerja

sendiri. Pada pengujian dua server cukup baik pembagian beban walaupun penggunaan juga masih diatas 50%. Namun saat dicoba melebihi 1000 mulai tidak dapat menampung banyaknya data yang diterima. Oleh karena itu disarankan jika ingin menggunakan kapasitas lebih besar disarankan untuk menambah lagi beberapa server agar dapat menampung data yang lebih besar lagi.

DAFTAR PUSTAKA

- [1]. Muhammdad Sholikin, Muhammad Akbar, Zaid Amin “Analisis Performasi Server Cluster Pada Load Balancing Web Server Menggunakan NBL MANAGER”.
- [2]. Denny Rachmawan, Dadan Irwan, Harum Argyawati, 2016, “Penerapan Teknik Load Balancing pada Web Server lokar dengan metode Nth menggunakan Mikrotik”
- [3]. Alam Rahmatulloh, Firmansyah MSN 2017, “Implementasi load balancing web server menggunakan haproxy dan sinkronisasi file pada sistem informasi akademik Universitas Siliwangi”.
- [4]. Syaqlia Azizah, Asmunin, 2017, “Implementasi Load Balancing web server menggunakan Haproxy”, Jurnal Management Informatika, Vol 8 ,No.01, 2017.
- [5]. Efendi Yusuf, Tengku A Riza, Tody Ariefianto, 2013, “Impelemtasi Teknologi Load Balancer denga web server nginx untuk mengatasi beban server”, STMIK AMIKOM Yogyakarta, Januari 2013.
- [6]. Fajar Zuhroni, Adian Fatchur Rochim, Eko didik Wudianto, “Analisa Performasi Layanan Kluster Server Menggunakan Penyeimbang Beban dan Virtualbox”. Jurnal Teknologi dan Sistem Komputer, Vol 3, No.4, Oktober 2015
- [7]. Abe Wisnu Syaputra, Setiawan Asseegaff 2017, “Analisis dan Implementasi Load Balancing dengan Metode NTH pada Jaringan Dinas Pendidikan Provinsi Jambi,” 2017 JurnalManajemenSistemInformasi Vol.2, No.4, Desember 2017
- [8]. Sampurna Dadi Riskiono, 2018, “Implementasi Metode Load Balancing dalam mendukung Sistem Kluster Server”.
- [9]. Syamsul Alam Haris, Hero Suhartono, Herlawati, 2018, “Menjaga Kestabilan Jaringan Load Balancing Nth dengan teknik Failover Pada PT. Jakarta Samudera Sentosa Jakarta”.
- [10]. Warman I, Andrian A. 2017. Analisis Kinerja Load Balancing Dua line Koneksi Dengan Metode Nth. Jurnal TEKNOIF 5(1): 56-62
- [11]. Putri Utami, Lindawati Lindawati, Suzan Zefi, 2017, “Optimalisasi load balancing dua ISP untuk manajemen bandwidth berbasis Mikrotik”.

Perancangan Neur-O: Sistem Pakar Penyakit Saraf berbasis Ontologi

Design of Neur-O: an Expert System neurological disease based on ontology

Christopel H. Simanjuntak^{*1}, Dirko G. S. Ruindungan²

¹Prodi D4 Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Manado, Manado

²Jurusan Teknik Elektro, Fakultas Teknik Elektro UNSRAT, Manado

e-mail: ^{*1} christopel.simanjuntak@polimdo.ac.id, ²dirko@unsrat.ac.id

Abstrak

Penyakit saraf sangat berbahaya jika menyerang seseorang sehingga perlu adanya cara untuk menyelesaikan masalah ini. Banyak cara dalam menanggulangi penyakit ini, salah satunya ialah membangun sistem pakar untuk mengedukasi masyarakat tentang bahaya penyakit ini. Sistem ini dirancang dengan metodologi *waterfall* dimana memiliki 4 fase utama sampai bisa menghasilkan produk. Fase *waterfall* ialah perencanaan, analisis, desain dan implementasi dan akhirnya sistem telah dibangun. Hasil dari penelitian ini adalah sistem pakar berbasis ontologi yang diberi nama Neur-O. Pada Neur-O ada beberapa kategori yaitu Nama Penyakit, Sinonim, Definisi, Epidemiologi, Gejala baik umum maupun dugaan, Penyebab umum dan pendukung (lain), Agen pembawa, Pemeriksaan utama dan lain, Terapi obat dan fisik serta Pencegahan. Evaluasi menggunakan *consistency checking* dan evaluasi pakar. Pada *consistency checking*, ontologi dievaluasi dengan menggunakan *inference engine* Pellet. Pada proses inferensi, ontologi yang dipakai tidak terdapat error dan memerlukan 5093 milidetik untuk inferensi dan Evaluasi pakar rata-rata nilai ada 4 dimana sistem sudah baik. Berdasarkan hasil yang didapat, diperlukan pengembangan lagi kedepan dimana menambahkan amnesia berupa keluhan pasien dan riwayat penyakit dan pada outputnya dikembangkan ada diagnose banding serta tingkat stadium penyakit.

Kata kunci— Sistem Pakar, Ontologi, Neur-O, Penyakit Saraf

Abstract

Nerve disease is very dangerous if it attacks someone so there needs to be a way to resolve this problem. Many ways to deal with this disease, one of which is to build a expert system to educate the public about the dangers of this disease. This system was designed with the waterfall methodology which has 4 main phases to produce a product. The waterfall phase is the planning, analysis, design and implementation and finally the system has been built. The results of this study are ontology-based expert system named Neur-O. In Neur-O there are several categories namely Disease name, Synonym, Definition, Epidemiology, Common and suspected symptoms, Common and supporting causes (other), Carrier agents, Primary and other examinations, Drug and physical therapy and Prevention. Evaluation using consistency checking and expert evaluation. In consistency checking, ontology is evaluated using the Pellet inference engine. In the inference process, the ontology used has no errors and requires 5093 milliseconds for expert inference and evaluation, on average there are 4 values where the system is good. Based on the results obtained, it is necessary to develop further in the future

where adding amnesia includes patient complaints and disease reporting and at the output developed there is a differential diagnosis and stage of disease

Keywords— *Expert System, Ontology, Neur-O, Neurological Diseases*

1. PENDAHULUAN

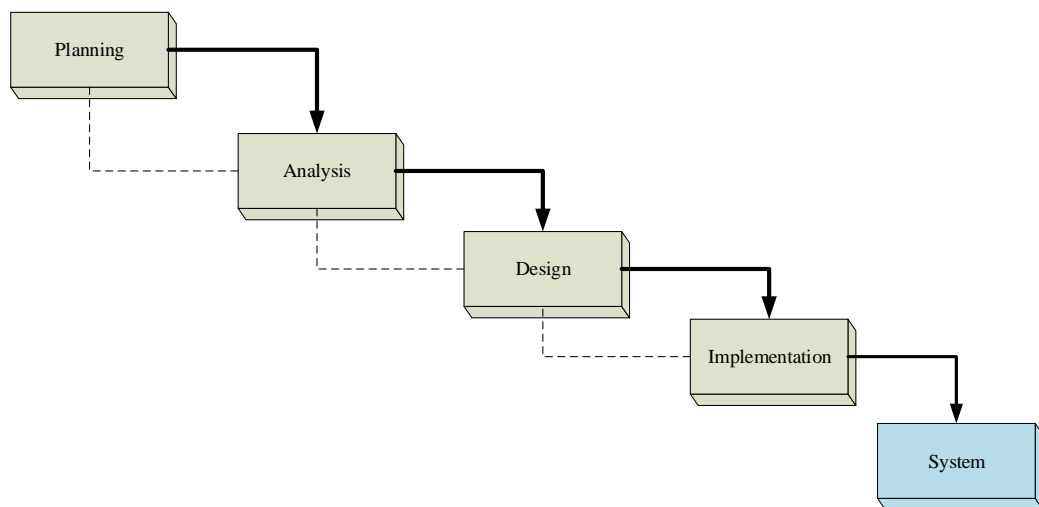
Penyakit saraf adalah penyakit yang bahaya untuk manusia. Menurut WHO, tahun 2005 hanya untuk penderita cerebrovascular disease mencapai 50785000 jiwa. Tahun 2015 penderita mencapai 53815000 jiwa dan diprediksikan mencapai 60864000 jiwa di seluruh dunia [1]. Penyakit saraf memberikan kontribusi sebesar 6.3% dari seluruh penyakit di dunia [1]. Stroke dan Penyakit jantung iskemik merupakan penyakit paling mematikan yang menyebabkan kematian tahun 2016 [2]. Dengan data yang ada menunjukkan bahwa penyakit saraf perlu menjadi perhatian untuk semua orang.

Dari beberapa solusi pemecahan masalah untuk penyakit saraf, sistem pakar untuk edukasi ke masyarakat adalah salah satunya. Sistem pakar adalah sistem yang digunakan untuk pemecahan suatu masalah dengan menggunakan pengetahuan dari pakar atau ahli dibidang tertentu. Sistem pakar telah digunakan diberbagai bidang misalnya saja pada transportasi [3][4], tanaman [5], hukum[6], maupun kesehatan [7]. Kendala sistem pakar yang dibuat saat ini adalah struktur data tiap sistem pakar berbeda-beda sehingga menyulitkan pengembangan sistem tersebut. Teknologi semantik adalah jawaban untuk masalah ini [8], [9]. Dengan teknologi ini, Ontologi yang dikembangkan dapat digunakan secara terus-menerus tanpa perlu mengubah struktur data yang telah dibuat. Penelitian ini memaparkan perancangan protipe sistem pakar yang diberi nama Neur-O dimana memanfaatkan ontologi [10] untuk keperluan edukasi. Sistem dikembangkan menggunakan metodologi waterfall [11][12] dan diberi nama Neur-O.

Ada beberapa penelitian yang telah dikembangkan yang menggunakan ontologi dalam sistem yang dikembangkan yaitu Kadek,dkk [17] yang merancang ontologi untuk teknologi penelusuran pustaka di SIRREF JTETI UGM. Pada penelitiannya Kadek membuat ontologi dengan metode METHONTOLOGY untuk membentuk ontologi. Pada penelitiannya, sistem yang dibangun memudahkan menelusuran user saat mencari pustaka pada perpustakaan. Virginie,dkk [18] meneliti penggunaan SWRL yang akan digunakan pada desain power plant. Peneliti mengguakan JESS engine untuk mengeseksekusi SWRL yang telah dibentuk. Pada penelitiannya, peneliti menemukan beberapa limitasi dari SWRL yaitu tidak bisa digunakan pada rule yang kompleks yang ada saat implementasi power plant. Cech,dkk melakukan penelitian [19] dengan memanfaatkan ontologi untuk membangun sistem pakar pada situasi emergency jika terjadi kecelakaan biologis. Sistem memberi arahan reaksi berdasarkan beberapa fase yang akan terjadi pada kecelakaan biologis misalnya identifikasi kecelakaan, karakteristik masalah pada keadaan darurat, membentuk potensial skenario, memilih probabilitas skenario yang akan terjadi, menentukan reaksi-reaksi paling layak saat skenarion tersebut dan memilih subset reaksi yang sesuai yang akan direalisasikan. Dirko ,dkk [20] dilakukan penelitian dalam merancang ontologi untuk nutrisi masa kehamilan. Dalam penelitiannya, ontologi dievaluasi menggunakan Schema Metric. Pada perancangannya dilakukan dengan menjabarkan atribut, relasi serta taksonomi yang akan dibangun. Hasil evaluasi menunjukkan nilai AR 2.42. Nilai IR yang didapat 3.14 dan RR 0.41.

2. METODOLOGI PENELITIAN

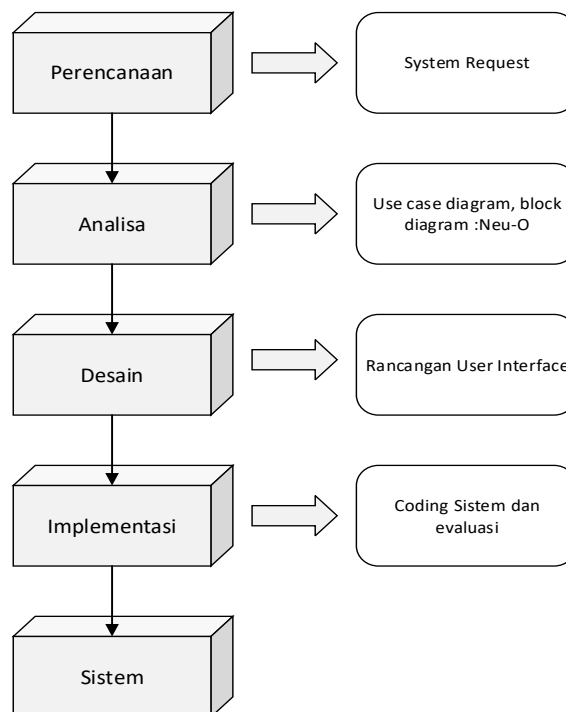
Dalam melakukan penelitian ini, metode mengacu pada metodologi pengembangan sistem *waterfall*[11] untuk membangun sistem pakar ini. Waterfall adalah salah satu metode pengembangan sistem [11]. Pada alur Waterfall ada 4 proses penting sehingga menghasilkan sistem atau protipe yang diinginkan.



Gambar 2. Fase Pengembangan Waterfall [11].

Dapat dilihat pada gambar 2 bahwa tiap fase pada metode dieksekusi satu persatu sampai akhirnya menjadi suatu sistem atau protipe yang diinginkan. Kalaupun ada langkah mundur ke fase sebelumnya maka akan sangat sulit. Kelebihan dari *waterfall* adalah waktu yang diperlukan untuk mengidentifikasi sistem sebelum diimplementasikan [11][12]. Pada *waterfall* ada beberapa fase yaitu *Planning*, *Analysis*, *Design*, *Implementation* sampai terbentuknya *System*.

Metodologi *waterfall* pada penelitian ini terdiri dari tahap perencanaan, analisa, desain, implementasi lalu menghasilkan sistem diinginkan.



Gambar 3. Alur dari Waterfall [11]

Pada gambar 3 menggambarkan bagaimana alur *waterfall* yang dipakai. Pada tahap perencanaan digambarkan dalam *system request*. Pada *system request* digunakan untuk menganalisa proses bisnis yang ada, mengidentifikasi kesempatan yang bisa dipakai untuk menyelesaikan masalah yang ada serta memaparkan ide yang digunakan pada sistem. Dalam *system request* ada beberapa hal yang perlu dipertimbangkan seperti: *project sponsor*, *business needs*, *business requirement*, *business value* dan *special issue*. *Project sponsor* ialah siapa yang membantu atau mensponsori penelitian ini. *Project sponsor* bisa dari pemerintah, swasta maupun dari pribadi. *Project sponsor* bisa memberikan bantuan finansial baik hanya bersifat membantu (tidak 100% finansial *project* disponsori) maupun *full funded*. *Business needs* ialah tujuan sistem tersebut dibangun. *Business requirement* berhubungan dengan keperluan atau keinginan dari pengguna yang digunakan untuk membuat fungsi pada sistem. *Business value* adalah nilai yang didapat dari pengguna jika sistem tersebut selesai dan sudah bisa berjalan dan *special issue* ialah catatan yang diperlukan agar sistem bisa berjalan dengan baik.

Setelah proses perencanaan, masuk ke tahap analisa dimana pada bagian ini digunakan *use-case diagram* dan *block diagram* untuk membantu dalam menganalisis kebutuhan sistem pakar yang dibangun. Pada proses ini dapat dilihat interaksi user dengan sistem yang berjalan kedepan. Dengan adanya proses ini, sistem yang dibangun dapat teridentifikasi setiap fungsinya berdasarkan relasi pada *use case* yang dibangun.

Pada proses desain, sistem didesain sesuai dengan hasil perencanaan. Desain aplikasi tersebut harus *user friendly* sehingga pengguna tidak bingung dalam menjalankan aplikasi yang telah dibangun. Pada desain aplikasi, dirancang tampilan user interface yang *simple* dan mengakomodir semua fungsi yang diharapkan.

Pada tahap implementasi, sistem dibangun menggunakan code program dan program IDE yang membantu proses implementasinya. Setelah proses ini selesai maka dilakukan evaluasi baik pada ontologi yang dibangun maupun aplikasi yang dibangun. Evaluasi pada ontologi digunakan pertanyaan kompetensi dan aplikasi yang dibangun dilakukan evaluasi dengan melakukan pengecekan untuk fungsi yang dibuat. Pada evaluasi juga diadakan validasi pakar supaya menjamin sistem tersebut sesuai dengan pengetahuan yang ada.

Ontologi

Pada dunia kecerdasan buatan, Ontologi menggambarkan relasi antar suatu objek serta hubungan yang terbangun antar objek yang ada [8]. World Wide Web Consortium (W3C) telah memberikan standard bahasa untuk ontologi yaitu OWL [9]. OWL mengakomodir atribut yang banyak diperlukan dalam membangun suatu ontologi. Beberapa diantaranya ialah Individuals, Properties dan Classes [9]. Ada beberapa cara untuk mengevaluasi ontologi yang telah dibuat misalnya saja dengan menggunakan consistency checking [14][15]. Consistency checking memanfaatkan inferensi untuk melihat apakah ontology yang dibangun sudah baik dengan inference engine yang ada.

3. HASIL DAN PEMBAHASAN

3.1. System Request Neur-O

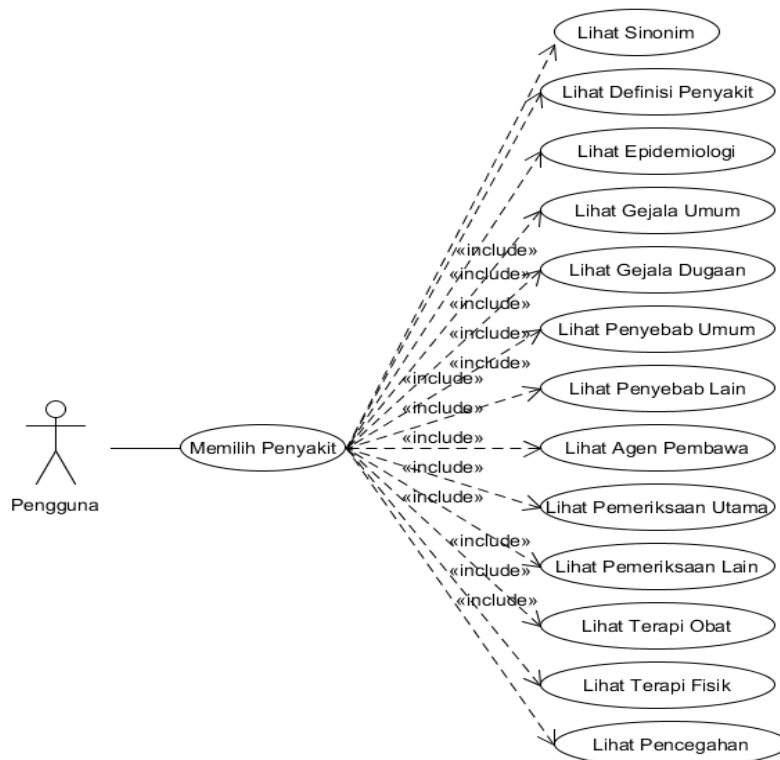
Tabel 1. Tabel system request Neur-O

NO	Item System Request	Ket
1	<i>Project Sponsor</i>	Tim Peneliti Prototype Neur-O
2	<i>Business Need</i>	Memberikan pengetahuan pengguna untuk lebih mengerti tentang penyakit saraf melalui sistem pakar
3	<i>Business Requirement</i>	Ada beberapa fitur yang diperlukan seperti: <ol style="list-style-type: none"> 1. Fitur pencarian berdasarkan penyakit yang ada 2. Adanya fitur memilih kategori berdasarkan hasil yang ditampilkan

4	<i>Business Value</i>	Diharapkan sistem pakar ini dapat meningkatkan pengetahuan dan pemahaman akan penyakit saraf.
5	<i>Special Issue</i>	Sistem pakar Neur-O haru di validasi dari pakar penyakit araf

Pada tabel 1 diperlihatkan rincian *system request* untuk perancangan sistem pakar penyakit saraf. Pada item *project sponsor*, peneliti sendiri yang menjadi sponsor karena penelitian mandiri. Pada *business Need*, peneliti memberikan pengetahuan pengguna tentang penyakit saraf. Sistem pakar ini mampu mengedukasi penggunanya dalam hal penyakit saraf. Pada *business requirement* ada beberapa fitur yang diinginkan seperti fitur pencarian berdasarkan penyakit dan adanya pemilihan kategori sesuai dengan hasil yang dicapai. Tim mengharapkan sistem yang dibuat mampu meningkatkan pemahaman dan pengetahuan penyakit saraf ke pengguna. Pada *special issue*, sistem harus divalidasi oleh pakar di bidang penyakit saraf sehingga sistem menyajikan pengetahuan yang benar agar supaya pengguna tidak salah dalam menerima pengetahuan penyakit saraf.

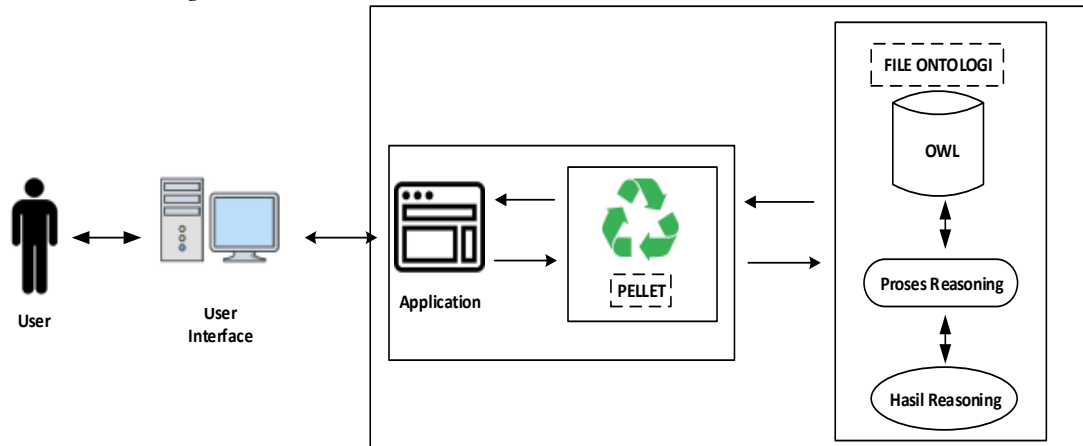
3.2. Use Case Diagram Neur-O



Gambar 4. Tampilan Use-case Diagram

Pada gambar 4 menggambarkan tentang *use-case* penggunaan aplikasi Neur-O. Dalam use-case tersebut, pengguna memilih penyakit yang ingin diketahui. Setelah penyakit telah dipilih maka dari penyakit tersebut, pengguna bisa memilih kategori yang ingin diketahui. Ada beberapa kategori yang bisa dipilih seperti Sinonim, Definisi, Epidemiologi, Gejala baik umum maupun dugaan, Penyebab umum dan pendukung (lain), Agen pembawa, Pemeriksaan utama dan lain, Terapi obat dan fisik serta Pencegahan. Dengan memilih kategori, aplikasi menampilkan penjelasan terkait penyakit sesuai kategori yang dipilih.

3.4. *Block Diagram Sistem*

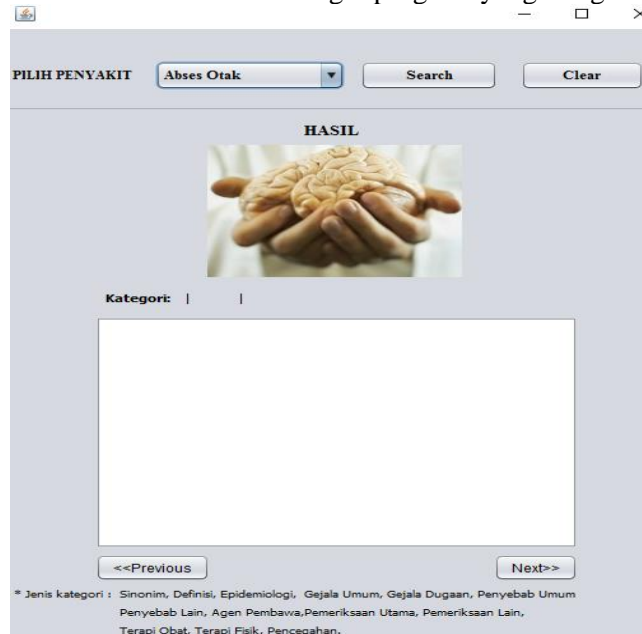


Gambar 6. Rancangan *Block Diagram* Neur-O

Pada Gambar 6 memperlihatkan *block diagram* sistem yang sedang berjalan. User mengakses aplikasi yang dibuat melalui user interface yang telah dibangun. Dari *user interface*, user memberi *input* yang akan diproses. Pada aplikasi java akan meneruskan inputan yang ada lalu akan di *reasoning* oleh Pellet sebagai *reasoner*. Pellet akan memprosesnya ke *file* OWL setelah itu akan mengirimkan hasilnya ke aplikasi untuk ditampilkan ke user. Ontologi yang dipakai dalam sistem ini dibangun menggunakan aplikasi Protégé. Pada proses reasoning, input yang diberikan user akan jadikan instance di ontologi yang dibuat. Setelah itu pellet akan melakukan pengecekan baik *instance*, *properties* maupun *rule* yang telah dibuat di ontologinya dan *rule* yang ada dieksekusi sehingga *instance* yang memiliki relasi akan intance inputan akan muncul. *Instance* tersebut menjadi output yang akan ditampilkan pada user melalui *user interface* aplikasi.

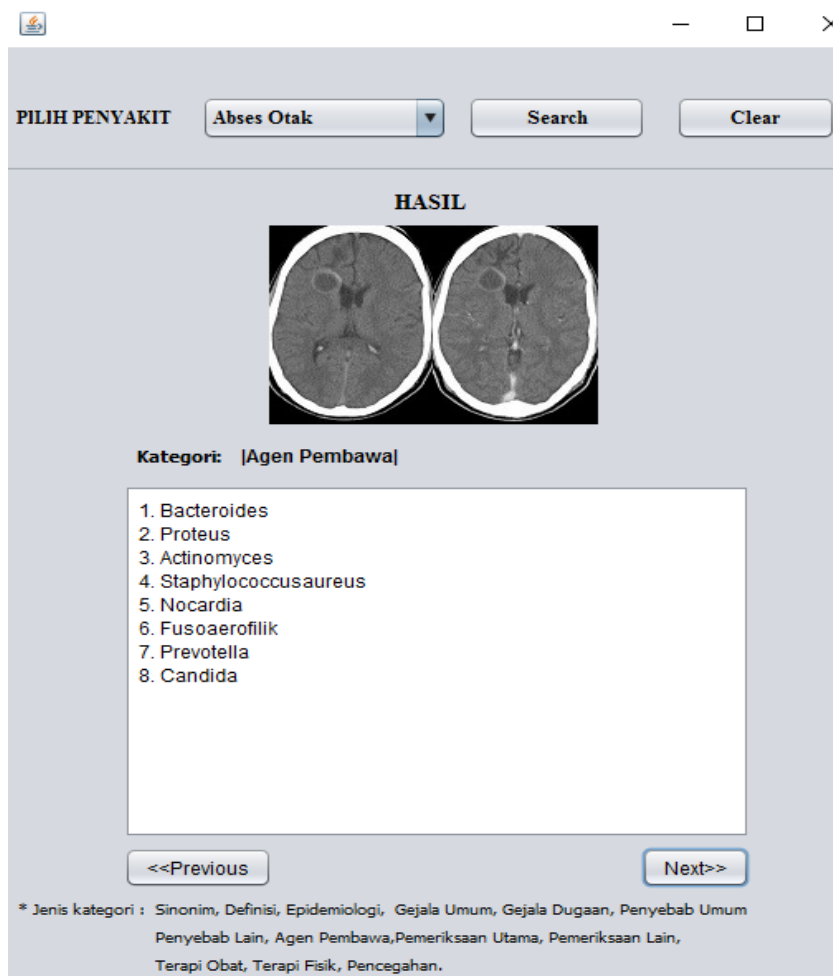
3.5. *Sistem yang terbangun dan Evaluasi*

Setelah dibuat desain sistemnya, sistem dibangun menggunakan code program. Setelah dibuat menggunakan code maka sistem dieksekusi agar program yang diinginkan dapat berjalan.



Gambar 7. Tampilan Neur-O yang berjalan

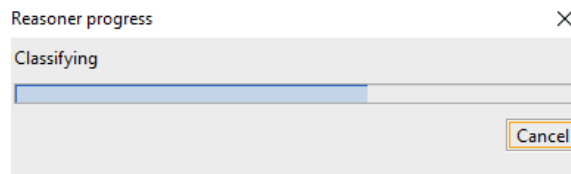
Pada Gambar 7 memperlihatkan tampilan aplikasi yang sudah dibuat. Pada aplikasi, user bisa memilih penyakit. Di aplikasi juga, ada *button Search* yang berguna untuk mencari untuk memproses pilihan penyakit dengan mengambil hasilnya dari ontologi yang telah dibuat dan ditampilkan pada aplikasi berdasarkan kategori yang ada. *Button previous* dan *next* berguna untuk mengganti hasil kategori yang ada. Ada 15 penyakit saraf yang menjadi pilihan dalam Neur-O ini yaitu [10][20][21][22]: Abses Otak, Aneurisma Otak, Bells Palsy, Cerebral Palsy, Epilepsi, Kejang Otot, Neuropati Diabetika, Nyeri Punggung, Stroke Pendarahan, Tumor Otak, Malformasi Arteri Vena Otak, Transient Ischemic Attack, Sindrom Terowongan Karpal dan Stroke Iskemik.



Gambar 8 Tampilan sistem Neur-O berjalan

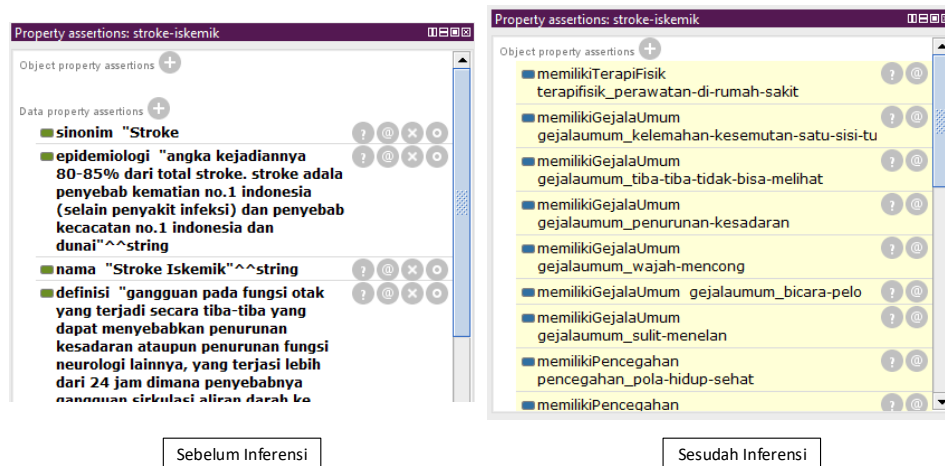
Pada gambar 8 menampilkan hasil dari pilihan penyakit abses otak. Hasilnya ditampilkan pada *text area* output berdasarkan kategori. Pemilihan kategori bisa menggunakan *button previous* dan *next* sesuai dengan urutan kategori.

Setelah sistem telah dibuat maka sistem dievaluasi menggunakan proses *consistency checking* [15] dan proses validasi pakar. Pada *consistency checking*, proses *reasoning* (inferensi) dilakukan pada ontologi yang dibuat untuk melihat hasil *reasoning*-nya. Jika *reasoning* menghasilkan *error* artinya ada ketidakkonsistenan pada ontologi yang dibuat sehingga ontologi perlu ditinjau lagi dan diperbaiki. Pada evaluasi ini, proses menggunakan reasoner Pellet. Proses *reasoning* dapat dilihat pada gambar 8.



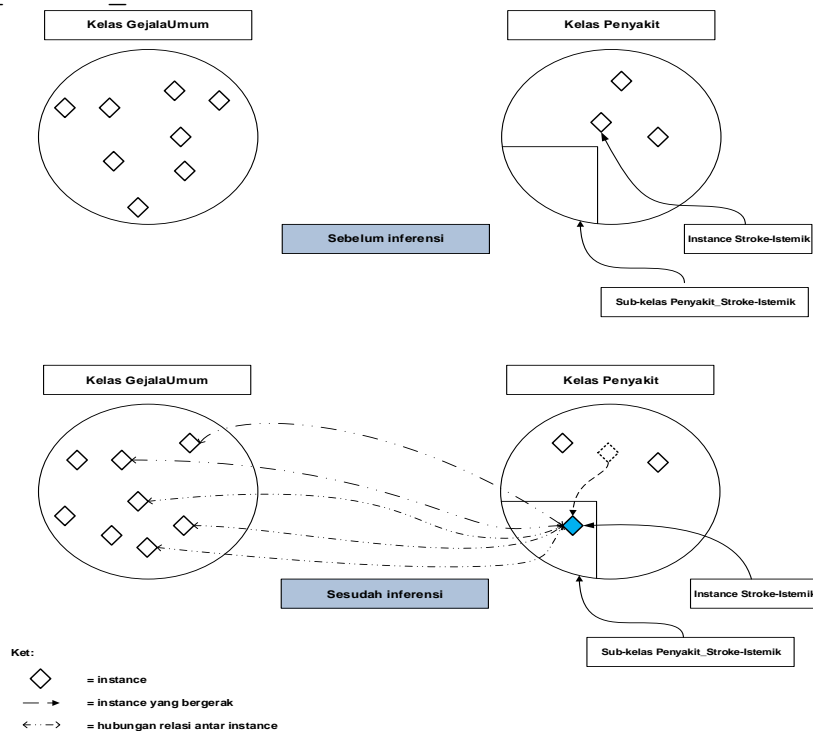
Gambar 9 Tampilan *reasoning process*

Setelah dilakukan proses inferensi, sistem akan menampilkan semua hasil berdasarkan SWRL (*Semantic Web Rule Language*) yang dibuat.



Gambar 10. Tampilan Proses Inferensi

Pada Gambar 10 menunjukkan bahwa sub-kelas `Penyakit_stroke_iskemik` sebelum inferensi tidak memiliki relasi dengan instance yang lain. Setelah dilakukan inferensi, setiap *rule* dan aksiom aktif sehingga membuat tiap instance dan relasi masuk ke sub-kelas `Penyakit_stroke_iskemik`.



Gambar 11. Alur pergerakan *instance* saat Inferensi.

Dapat dilihat pada gambar 11 dimana instance `stroke_iskemik` dan beberapa instance yang berkaitan karena *rule* yang dibuat masih berada pada kelas masing-masing. Tapi setelah dilakukan inferensi maka instance yang berkaitan masuk ke dalam kelas sub-kelas `Stroke-Istemik` baik instance `stroke_iskemik` maupun instance dari kelas `GejalaUmum`.

Proses inferensi memakan waktu 5903 milidetik dan ontologi tidak mengeluarkan hasil error dan dapat dilihat pada gambar 10.

```
0[34m----- Running Reasoner -----0[0;39m
0[34mPre-computing inferences:0[0;39m
0[34m  - class hierarchy0[0;39m
0[34m  - object property hierarchy0[0;39m
0[34m  - data property hierarchy0[0;39m
0[34m  - class assertions0[0;39m
0[34m  - object property assertions0[0;39m
0[34m  - same individuals0[0;39m
0[34mOntologies processed in 5903 ms by Pellet0[0;39m
0[34m0[0;39m
```

Gambar 12. Tampilan Inferensi menggunakan Pellet.

Pada proses validasi untuk sistem pakar, diperlukan ahli untuk melihat konten dari sistem yang dibangun. Proses validasi dilakukan dengan cara wawancara kepada pakar lalu pakar memberikan skor terhadap sistem yang ada. skor bernilai 1-5 dimana 1 sangat buruk sampai 5 adalah sangat baik. Berikut hasil validasi dari pakar:

- a. Masukan yang digunakan sudah baik, hanya akan jauh lebih baik lagi jika masukan atau input sistem bisa menerima data amnesis. Amnesis adalah cara pengambilan data pasien dalam bentuk wawancara. Bentuk Amnesis ialah riwayat penyakit ataupun keluhan pasien. Skala yang diberikan bagian ini adalah 3 yaitu cukup bagus.
- b. Pada output, penilaiannya ialah sudah baik tapi akan lebih baik lagi jika beberapa output yang perlu dilengkapi seperti diagnosa banding dan tingkat stadium yang sebenarnya ada pada penyakit saraf. Untuk output, pakar memberikan skala 4 yaitu baik.
- c. Pada keseluruhan sistem yang sedang berjalan baik dari segi fungsi sistem pakar dan tujuannya. Skala 5 diberikan pakar untuk ini.

Dari skala yang didapatkan, rata-rata nilai adalah 4 sehingga sistem yang dibangun sudah baik.

4. KESIMPULAN

Rancangan sistem pakar Neur-O menggunakan metode Waterfall dimana ada beberapa tahapan yaitu Perencanaan, Analisa, Desain, Implementasi serta menghasilkan sistem. Implementasinya sistem dibangun menggunakan aplikasi IDE NetBeans dan ontologi dibangun menggunakan protégé lalu menggunakan Pellet sebagai *reasoner*. Pada tahap Perencanaan menggunakan *system request*, Analisa menggunakan use-case diagram dan block diagram untuk memudahkan analisis. Di tahap Desain, didisain user interface lalu diimplementasikan dengan bahasa java.

Evaluasi untuk sistem digunakan evaluasi konsistensi dan evaluasi pakar. Evaluasi konsistensi menggunakan *reasoner* Pellet. Dari evaluasi tidak terdapat adanya error pada ontologi dan reasoner memakan waktu 5903 milidetik pada saat inferensi. Evaluasi pada pakar dilakukan berdasarkan wawancara untuk melihat masukan, keluaran serta keseluruhan sistem. Nilai rata-rata dari pakar adalah 4 atau sistem yang dibangun sudah baik. Adapun beberapa hal

yang menjadi cacatan oleh pakar ialah belum ada masukan amnesis dan pada outputnya masih belum dilengkapi diagnosa banding dan tingkat stadium untuk penyakit yang ada.

5. SARAN

Berdasarkan hasil evaluasi dan masukan dari pakar maka protipe sistem pakar penyakit saraf Neur-O kedepan bisa dikembangkan dengan membuat input sistem bisa menerima amnesis seperti riwayat penyakit dan keluhan dari pasien dan Output dilengkapi dengan diagnosa banding serta adanya tingkat stadium penyakit.

UCAPAN TERIMA KASIH

Terima kasih untuk dokter Dito Anurogo M.D (ditoanurogo@gmail.com), yang telah menjadi pakar untuk memvalidasi protipe sistem pakar ini. Terima kasih juga untuk ibu Dr. Sri Suning Kusumawardani, ST, MT (suning@ieee.org) dan Adhistya Erna Permanasari, PhD (adhistya@ugm.ac.id) yang telah membimbing sehingga penelitian ini bisa selesai dengan baik.

DAFTAR PUSTAKA

- [1] W. H. Organization, "Neurological Disorders Public Health Challenges."
- [2] W. H. Organization, "WHO | The top 10 causes of death." [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs310/en/>. [Accessed: 08-Oct-2014].
- [3] Rianto, "Sistem Pakar Diagnosa Permasalahan Pada Pesawat Cessna Grand Caravan 208b Menggunakan Metode Backward Chaining," *J. Angkasa*, vol. VIII, pp. 129–138, 2016.
- [4] S. H. A. Cholil Jamhari, Agus Kiryanto, "Sistem Pakar Diagnosis Kerusakan Sepeda Motor Non Matic," *Semin. Nas. IENACO*, vol. 1, p. 375, 2014.
- [5] M. Burhannudin, Suprpto, and N. Hidayat, "Pemodelan Sistem Pakar Diagnosis Penyakit Tanaman Apel Manalagi Dengan Metode Backward Chaining Menggunakan Certainty Factor," *J. Pengemb. Teknol. Inf. dan ilmu Komput.*, vol. 1, no. 5, pp. 399–404, 2017.
- [6] Juwairiah, Y. Fauziah, and Y. E. Afriliana, "Sistem Pakar Berbasis Web Penentu Pasal Tindak Pidana Narkotika," *Semin. Nas. Inform. UPN Veteran Yogyakarta*, vol. 1, no. 22, p. 36, 2008.
- [7] Ashari and A. Y. Muniar, "Penerapan Sistem Pakar Untuk Mendiagnosa Penyakit Pencernaan Dengan Pengobatan Bahan Alami," *Semin. Nas. Sains dan Teknol.*, no. November, pp. 1–9, 2016.
- [8] N. F. Noy, D. L. McGuinness, and others, *Ontology development 101: A guide to creating your first ontology*. Citeseer, 2001.
- [9] U. Prot *et al.*, "A Practical Guide To Building OWL Ontologies Using Prot' eg' e 4 and CO-ODE Tools Edition 1.3," 2011.
- [10] C. H. Simanjuntak, "Pengembangan Ontologi Domain Medis Untuk Menunjang Sistem Pakar Pada Penyakit Saraf," Universitas Gadjah Mada, Yogyakarta, 2016.
- [11] A. Dennis, B. Wixom, and R. Roth, *System Analysis & Design*, 5th ed. United States: John Wiley & Sons Inc, 2012.
- [12] H. Gomaa, *Software Modelling and Design UML, Use Cases, Patterns an Softwarwe Architecture*, 1st ed. United States of America: Cambrige University Press, 2011.
- [13] M. Arhami, *Konsep Dasar Sistem Pakar*, 2nd ed. Yogyakarta: AndiPublisher, 2006.
- [14] N. StellaN and Agbo Okechuku Chuks, "Expert System: A Catalyst Ineducational Development Innigeria," no. c, pp. 519–524, 1988.
- [15] K. Baclawski, M. Kokar, and R. Waldinger, "Consistency checking of semantic web ontologies," *Semant. Web — ISWC 2002*, pp. 454–459, 2002.

- [16] G. E. Mendel-Gleason, R. Brennan, and K. Feeney, "Ontology Consistency and Instance Checking For Real World Linked Data," *Proc. 2nd Work. Linked Data Qual.*, pp. 1–4, 2015.
- [17] K. Dwi and P. Novianti, "Perancangan Ontologi sebagai Teknologi Penyimpanan Informasi untuk Penelusuran Pustaka pada SIRREF JTETI UGM," vol. 3, pp. 98–103, 2014.
- [18] V. Fortineau, T. Paviot, L. Louis-Sidney, and S. Lamouri, "SWRL as a rule language for ontology-based models in power plant design," *IFIP Adv. Inf. Commun. Technol.*, vol. 388 AICT, pp. 588–597, 2012.
- [19] P. Čech, V. Bureš, K. Antoš, T. Otčenášková, A. Macela, and P. Musilek, "Ontological Models and Expert Systems in Desicion Support of Emergency Situations," *Mil. Med. Sci. Lett.*, vol. 80, pp. 21–27, 2011.
- [20] D. G. S. Ruindungan, "Perancangan Ontologi Prenatal-Nutrition dan Evaluasinya Menggunakan Schema Metric OntoQA," pp. 40–45, 2014.
- [21] C. H. Simanjuntak, S. S. Kusumawardani, A. E. Permanasari, and U. G. Mada, "Perancangan Ontologi Domain Pengetahuan Penyakit Saraf Berbasis SWRL Dengan Metode METHONTOLOGY.," pp. 489–494, 2015.
- [22] C. H. Simanjuntak, S. S. Kusumawardani, A. E. Permanasari, J. Grafika, and N. Yogyakarta, "Evaluasi Ontologi Penyakit Saraf menggunakan Schema Metric Onto-QA," no. 2, 2015.

Rancangan dan Pengujian Perangkat Lunak Menggunakan *Web Service* pada Sistem Informasi Akademik

Software Design and Testing Using Web Service on Academic Information Systems

Ceria Asa Malinda¹ Sandy Kosasi²

^{1,2}STMIK Pontianak; Jl. Merdeka No. 372 Pontianak, Telp. (0561) 735555, Fax. (0561) 737777
Jurusan Teknik Informatika, STMIK Pontianak
e-mail: ceriaasamalinda@gmail.com, Sandykosasi@yahoo.co.id

Abstrak

Penulis melakukan penelitian terhadap kebutuhan untuk mengolah dan mengintegrasikan data di Sekolah Menengah Atas dengan sistem-sistem yang sudah ada agar sistem-sistem tersebut dapat berkomunikasi tanpa terhalang platform dan perbedaan bahasa pemrograman yang digunakan. Oleh karena itu, penulis membuat sebuah perangkat lunak Sistem Informasi Akademik Menggunakan Web Service agar dapat menghubungkan Sistem Informasi Akademik dengan Sistem Informasi Administrasi Keuangan. Bentuk penelitian yang penulis lakukan adalah studi literatur dengan teknik pengumpulan data observasi, wawancara, dan studi dokumentasi. Metode penelitian yang digunakan adalah Research and Development (R&D), sedangkan metode perancangan perangkat lunak menggunakan Extreme Programming (XP) dengan menerapkan 4 cara praktis pengembangan XP, yaitu planning, design, coding, dan testing. Hasil perancangan perangkat lunak ini berupa Sistem Informasi Akademik, Sistem Informasi Administrasi Keuangan, dan Web Service. Perangkat lunak Sistem Informasi Akademik menggunakan aplikasi Sublime Text 2 dengan bahasa pemrograman PHP, perangkat lunak Sistem Informasi Administrasi Keuangan menggunakan platform Visual Studio 2012 dengan bahasa pemrograman Visual Basic.NET, dan web service dibuat menggunakan aplikasi ASP.NET Web Form Application dengan bahasa pemrograman Visual Basic.NET dan di-publish dengan menggunakan Internet Information Services (IIS) versi 7.5.

Kata kunci—Web Service, Sistem Informasi Akademik, Sistem Informasi Administrasi Keuangan, WSDL, Extreme Programming.

Abstract

The author conducted a study of the need to process and integrate data at High School with systems that already exist so that these systems can communicate unhindered difference platform and programming language used. Therefore, the author makes an Academic Information System software Using Web Services in order to connect the Academic Information System with the Financial Administration Information System. Forms of research by the author is the study of literature with data collection techniques of observation, interviews, and documentation. The method used is a Research and Development (R & D), whereas software design methods using Extreme Programming (XP) by implementing 4 practical way XP development, i.e, planning, design, coding, and testing. The results of this in the form of software design Academic Information Systems, Information Systems of Financial Administration, and Web Service. Academic Information System software using Sublime Text 2 application with PHP programming language, software Financial Administration Information

System platform using Visual Studio 2012 with Visual Basic.NET programming language and web services created using ASP.NET Web Form Application with the programming language Visual Basic.NET and publish it by using the Internet Information Services (IIS) version 7.5.

Keywords—*Web Service, Academic Information Systems, Information Systems of Financial Administration, WSDL, Extreme Programming.*

1. PENDAHULUAN

Perangkat lunak berbasis teknologi *web service* dapat menyediakan data maupun fungsi tertentu bagi aplikasi lain meskipun berbeda sistem operasi, perangkat keras, maupun bahasa pemrograman yang digunakan untuk membangunnya. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*services*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* dapat memberikan banyak keuntungan bagi sebuah organisasi, yaitu institusi pendidikan Sekolah Menengah Atas. Semakin berkembang Sekolah Menengah Atas, maka akan semakin sulit dalam pengolahan dan integrasi data atau informasi sehingga dibutuhkan sebuah pengembangan sistem informasi akademik yang sudah ada. Jika pengolahan dan integrasi data menjadi lebih kompleks, maka data dan informasi perlu diintegrasikan ke sistem dan aplikasi lainnya untuk menciptakan aktivitas sekolah yang cepat, tepat, dan *up-to-date*. Dalam melakukan pengembangan sistem ini, sekolah dihadapkan pada permasalahan untuk membuat sistem-sistem di dalam sekolah dapat berinteraksi meskipun berbeda *platform* sehingga muncul sebuah gagasan untuk dapat menggunakan kembali data dan informasi lama pada sebuah *interface* sistem baru yang tidak terbatas hanya pada satu *platform* saja. Salah satu contoh sistem yang dapat diintegrasikan di dalam sekolah adalah Sistem Informasi Akademik dan Sistem Informasi Administrasi Keuangan. Maka dari itu, *web service* dapat digunakan untuk menghadapi masalah tersebut.

Sistem informasi akademik yang dapat terintegrasi dengan sistem informasi administrasi keuangan dalam sekolah dengan menggunakan *web service* dapat dimanfaatkan kepala sekolah atau pun guru dalam mengakses pemasukan dan pengeluaran sekolah, serta bagi siswa atau orangtua dan wali siswa dapat mengetahui jumlah pembayaran apa saja yang harus ditunaikan. *Web service* di sini berperan dalam pengintegrasian data jumlah pembayaran siswa, dan pemasukan serta pengeluaran sekolah, beasiswa, dan pencarian data pembayaran siswa dengan memasukkan parameter berupa nama siswa yang diambil dari sistem informasi administrasi keuangan sehingga data-data tersebut dapat diakses lewat sistem informasi akademik.

Pada penelitian yang dilakukan sebelumnya aplikasi dapat digunakan untuk proses peng-*input-an* LIHS (Lembar Isian Hasil Studi) secara *online*[1]. Penelitian lainnya adalah aplikasi berbasis android yang menerapkan SOA menggunakan *web service* untuk mengintegrasikan sistem informasi akademik, sistem informasi perpustakaan, dan sistem informasi kepegawaian[2]. Penelitian dalam penggunaan SOA dan *Web Service* pada sistem informasi akademik Fakultas Teknik UNSRAT mempermudah penggunaan data-data mahasiswa/dosen/matakuliah dan data lainnya untuk digunakan oleh sistem informasi/aplikasi lain dalam organisasi[3].

Berdasarkan ketiga penelitian di atas memiliki persamaan dengan penelitian yang dilakukan yaitu membuat sistem informasi akademik yang menggunakan *web service* sebagai media untuk mempermudah sekolah dalam mengatur aktivitas akademiknya. Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya adalah pada metode perancangan yang digunakan, penambahan fitur penyampaian informasi profil sekolah sebagai ajang promosi sekolah, pengontrolan sistem secara administrasi, dapat mengintegrasikan sistem informasi

akademik dan sistem informasi administrasi keuangan, dan perancangan antarmuka yang digunakan yaitu membuat sistem informasi akademik berbasis *desktop* secara *offline* dengan Microsoft Visual Studio 2012 Express, sedangkan antarmuka untuk *web service* berbasis *web* menggunakan Sublime Text 2.

2. METODE PENELITIAN

Bentuk penelitian yang digunakan dalam penelitian ini adalah studi literatur dan metode *Research and Development* (R&D). Penelitian ini menggunakan metode *Research and Development* (R&D) yang merupakan suatu proses untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada, maksud produk dalam konteks ini adalah tidak selalu berbentuk *hardware* (buku, modul, alat bantu pembelajaran di kelas dan laboratorium), tetapi bisa juga berupa perangkat lunak (*software*) seperti program pengolahan data, pembelajaran di kelas, perpustakaan atau laboratorium, manajemen, dan lain sebagainya[4].

2.1 Metode Pengumpulan Data

Berikut adalah sumber data yang diperoleh dalam penelitian ini:

a. Data Primer

Data primer, yaitu sumber data utama yang diperoleh melalui kata-kata atau tindakan orang-orang yang diamati, berkaitan dengan penyajian informasi pada SMAN 1 Selimbau. Dalam wawancara ini menggunakan wawancara terstruktur yaitu wawancara yang terdiri dari suatu daftar pertanyaan yang telah direncanakan dan telah disusun sebelumnya. Sebagai bentuk pertanyaannya, digunakan wawancara terbuka yaitu terdiri dari pertanyaan-pertanyaan yang sedemikian rupa bentuknya sehingga responden atau informan diberi kebebasan untuk menjawabnya. Observasi dibutuhkan untuk dapat memahami proses terjadinya wawancara dan hasil wawancara dapat dipahami dalam konteksnya. Observasi yang akan dilakukan adalah observasi terhadap objek yang diteliti dan hal-hal yang dianggap relevan sehingga dapat memberikan data tambahan terhadap hasil wawancara.

b. Data Sekunder

Data sekunder yaitu sumber data tertulis data ini dapat ditemukan dengan cepat. Sumber data sekunder dalam penelitian ini adalah *literature*, artikel, dan jurnal yang berkenaan dengan penelitian yang dilakukan serta melakukan dokumentasi yaitu mencari sumber-sumber data yang berupa arsip, visi dan misi, struktur organisasi sekolah serta program kerja kepala sekolah pada SMAN 1 Selimbau.

2.2 Teknik Pengumpulan Data

Untuk memperoleh data dan informasi, digunakan teknik pengumpulan data sebagai berikut:

a. Observasi

Metode observasi adalah metode yang dilakukan dengan cara pengamatan dan pencatatan yang sistematis terhadap gejala-gejala yang diselidiki[5]. Dalam hal ini, peneliti melakukan pengamatan secara langsung pada Sekolah Menengah Atas Negeri 1 Selimbau untuk merekam fenomena yang terjadi berupa situasi dan kondisi yang sedang terjadi.

b. *Interview* (wawancara)

Peneliti dalam hal ini berkedudukan sebagai *interviewer*, mengajukan pertanyaan, menilai jawaban, meminta penjelasan, dan kadang-kadang juga membalas pertanyaan[6]. Metode ini digunakan untuk mendapatkan data tentang sesuatu yang berkaitan dengan penyampaian informasi di Sekolah Menengah Atas Negeri 1

Selimbau. Penelitian ini menggunakan wawancara terstruktur yaitu wawancara yang terdiri dari suatu daftar pertanyaan yang telah direncanakan dan telah disusun sebelumnya. Semua responden yang diwawancarai diajukan pertanyaan-pertanyaan yang sama, dengan kata-kata dan dalam tata urutan secara *uniform*. Sebagai bentuk pertanyaannya, digunakan wawancara terbuka yaitu terdiri dari pertanyaan-pertanyaan yang sedemikian rupa bentuknya sehingga responden atau informan diberi kebebasan untuk menjawabnya.

c. Dokumentasi

Metode dokumentasi adalah metode mencari data mengenai hal-hal atau variabel yang berupa catatan, transkrip, buku, surat kabar, majalah, prasasti, notulen rapat, legger, agenda dan sebagainya[7]. Metode ini digunakan untuk memperoleh dokumen-dokumen atau arsip yang ada di Sekolah Menengah Atas Negeri 1 Selimbau, yang berkaitan dengan sistem informasi yang ada di sekolah tersebut.

2.3 Pengembangan Perangkat Lunak Extreme Programming (XP)

Metode perancangan perangkat lunak yang digunakan dalam penelitian ini adalah *Extreme Programming*. *Extreme Programming* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan[8]. Adapun tahapan pada *Extreme Programming* pada perangkat lunak Sistem Informasi Akademik (SIKAD) dan Sistem Informasi Administrasi Keuangan (SIK):

a. *Planning*/Perencanaan

Pada tahap perencanaan ini dimulai dari analisa kebutuhan sekolah menengah atas dan analisa kebutuhan sistem informasi akademik dan sistem informasi administrasi keuangan, serta merencanakan *web service* yang ingin dibuat untuk menghubungkan kedua perangkat lunak tersebut. Selain itu, pada tahap ini juga mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh perangkat lunak, fungsi dari perangkat lunak yang dikembangkan, dan perencanaan jadwal pelaksanaan pengerjaan pembangunan proyek perangkat lunak sistem informasi akademik, sistem informasi administrasi keuangan, dan *web service*.

b. *Design*/Perancangan

Metode ini menekankan desain aplikasi yang sederhana, untuk mendesain perangkat lunak yang dapat menggunakan *Class-Responsibility-Collaborator (CRC) cards* yang mengidentifikasi dan mengatur *class* pada *object-oriented*. Perangkat lunak yang dibuat berupa Sistem Informasi Akademik (SIKAD) dan Sistem Informasi Administrasi Keuangan (SIK) masing-masing bersifat *Object Oriented Programming* dengan menerapkan teknik CRC, maka dapat mempermudah desainer dalam menghubungkan tanggung jawab tiap *form* dan kelas yang melaksanakan tanggung jawab tersebut.

c. *Coding*/Pengkodean

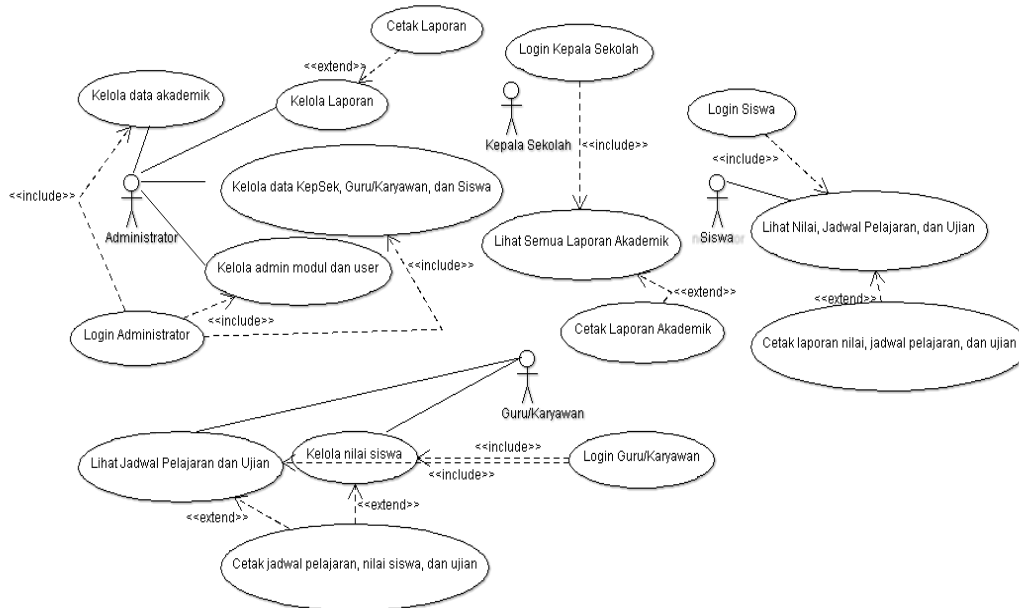
Setelah desain perangkat lunak selesai, maka dibuat *function coding* untuk tiap-tiap *form* dari perangkat lunak SIKAD maupun SIK dengan memanfaatkan *class-class* sehingga *coding* menjadi lebih sederhana dan dapat dimanfaatkan kembali tanpa mengubah desain *interface* perangkat lunaknya.

d. *Testing*/Pengujian

Pada tahapan peneliti menggunakan pengujian Black-Box Testing yang melakukan pengujian pada fitur dan fungsionalitas dari perangkat lunak SIKAD dan SIK serta Web Service. Pengujian ini dilakukan pada tiap-tiap *function* yang baru selesai dibuat sehingga akan meminimalisir kesalahan bila pengujian baru dilakukan setelah semua *coding function* selesai dibuat untuk seluruh *form* desain.

3. HASIL DAN PEMBAHASAN

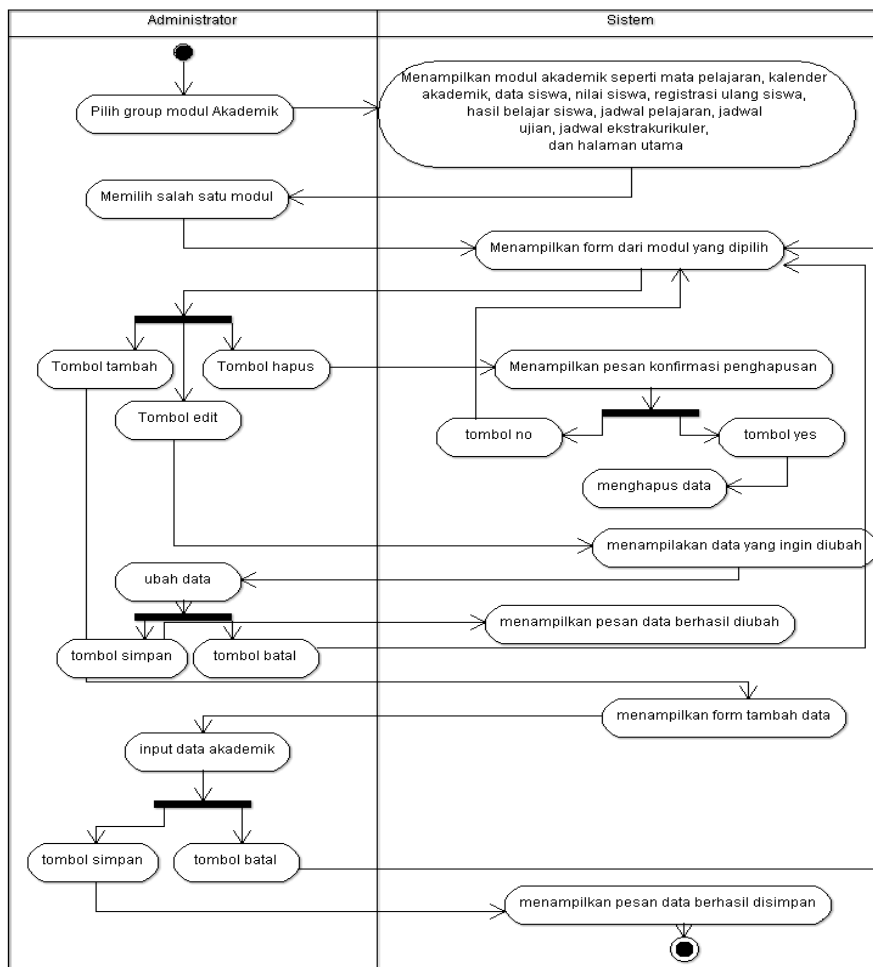
3.1 Use Case Diagram *Sistem Informasi Akademik*



Gambar 1. Use case diagram perangkat lunak Sistem Informasi Akademik SMA

Use case diagram perangkat lunak sistem informasi akademik Sekolah Menengah Atas ditangani oleh empat actor yaitu administrator, kepala sekolah, guru/karyawan, dan siswa.

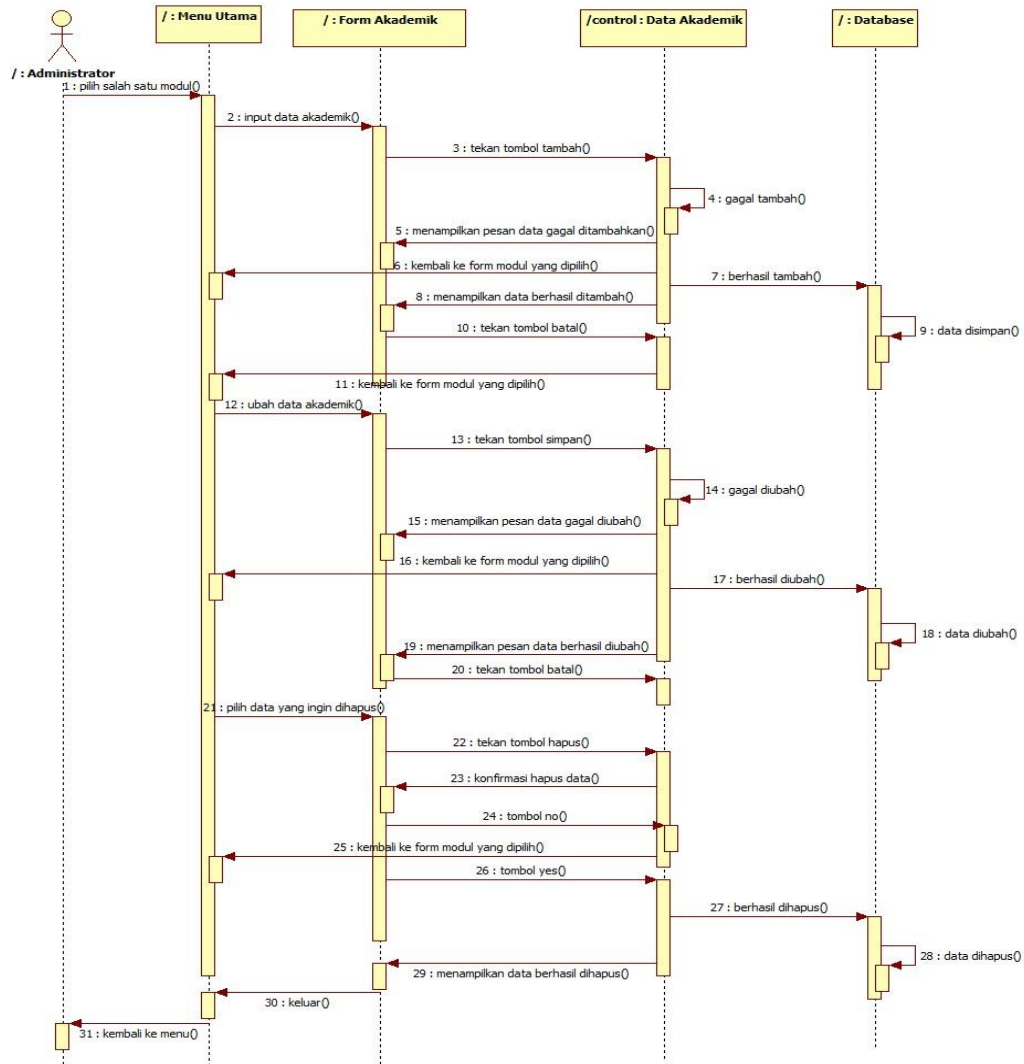
3.2 Activity Diagram *Kelola Data Akademik*



Gambar 2. Activity diagram kelola data akademik

Administrator memilih *group* modul Akademik dan sistem menampilkan modul-modul yaitu data akademik seperti mata pelajaran, kalender akademik, data siswa, nilai siswa, registrasi ulang siswa, hasil belajar siswa, jadwal pelajaran, jadwal ujian, jadwal ekstrakurikuler, dan halaman utama. Administrator memilih salah satu dari modul di dalam *group* modul Akademik. Sistem kemudian menampilkan *form* dari modul yang dipilih. Di dalam *form* tersebut terdapat tombol-tombol yang dapat melakukan penambahan, perubahan, penghapusan, dan pencarian data-data tersebut kecuali pada bagian modul halaman utama administrator tidak dapat mencari atau mencetak data. Jika administrator memilih tombol hapus, maka sistem akan menampilkan pesan konfirmasi hapus data. Bila memilih tombol *no*, maka akan menampilkan kembali *form* dari modul yang dipilih. Dan bila memilih tombol *yes*, maka sistem akan menampilkan pesan data berhasil dihapus. Jika administrator memilih tombol tambah, maka sistem akan menampilkan *form* untuk tambah data baru dan administrator harus mengisi data yang diperlukan. Bila administrator memilih tombol simpan, maka sistem akan menampilkan pesan bahwa data berhasil disimpan. Bila administrator memilih tombol batal, maka akan kembali ke *form* modul yang dipilih. Dan jika administrator memilih tombol edit, maka akan tampil *form* untuk mengubah data yang dipilih dan administrator mengubah data yang diperlukan. Bila administrator menekan tombol batal, maka akan kembali ke *form* modul yang dipilih. Bila tombol yang ditekan simpan, maka sistem akan menampilkan pesan bahwa data berhasil diubah. Selesai.

3.3 Sequence Diagram *Kelola Data Akademik*



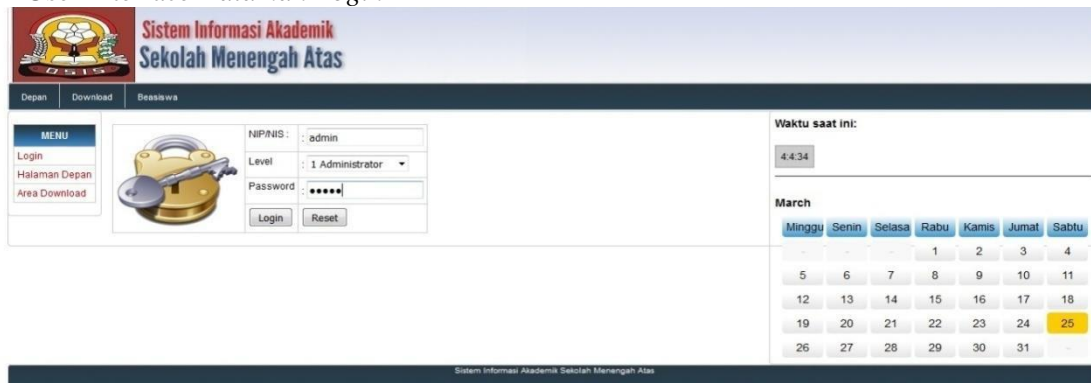
Gambar 3. *Sequence diagram* kelola data akademik

Sequence diagram kelola data akademik di atas merupakan lanjutan dari administrator yang sudah berhasil *login* dan menggambarkan tentang seorang *actor* yaitu administrator yang sedang melakukan kelola data akademik. Alur diagram di atas dapat dijelaskan sebagai berikut:

1. Administrator memilih salah satu modul di menu utama.
2. Setelah masuk ke *form* akademik, administrator mengisikan data akademik tersebut dan menekan tombol tambah.
3. Dalam proses *control* data, bila terjadi kesalahan data dan data gagal ditambahkan, maka akan menampilkan pesan data gagal ditambahkan. Kemudian akan menampilkan kembali *form* akademik.
4. Jika proses *control* data berhasil menambahkan data, maka akan menampilkan pesan data berhasil ditambahkan dan data tersebut masuk ke *database*.
5. Jika menekan tombol batal, maka akan kembali ke *form* modul yang dipilih.
6. Jika administrator menekan salah satu tombol edit pada data akademik, maka administrator dapat mengubah data yang diperlukan.

7. Bila tombol simpan yang ditekan dan terjadi kegagalan dalam pengubahan data, maka akan menampilkan data gagal diubah dan kembali ke *form* yang dipilih.
8. Bila data berhasil diubah, maka data yang sudah diubah tersebut akan masuk ke *database* dan menampilkan pesan bahwa data berhasil diubah.
9. Jika menekan tombol batal, maka akan kembali ke *form* modul yang dipilih.
10. Pada langkah ke 21, administrator memilih data yang akan dihapus.
11. Bila menekan tombol hapus, maka akan menampilkan pesan konfirmasi hapus. Jika menekan tombol *no*, maka akan menampilkan *form* modul yang dipilih. Dan bila menekan tombol *yes*, maka data berhasil dihapus di dalam *database* dan menampilkan pesan bahwa data berhasil dihapus.
12. Jika administrator memilih tombol keluar, maka akan kembali ke menu utama.

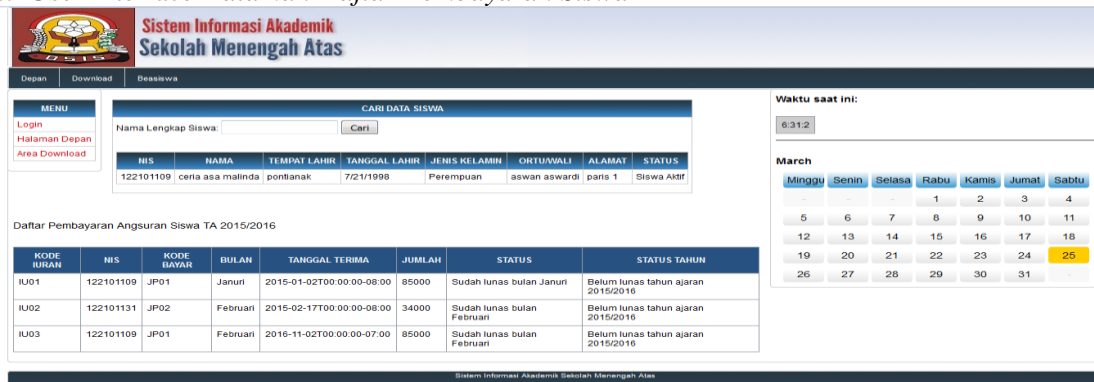
3.4 User Interface Halaman Login



Gambar 4. User Interface Halaman Login

Halaman *login* terdiri dari 4 *level user* yaitu administrator, kepala sekolah, guru atau karyawan, dan siswa. Masing-masing level memiliki hak akses ke modul yang berbeda.

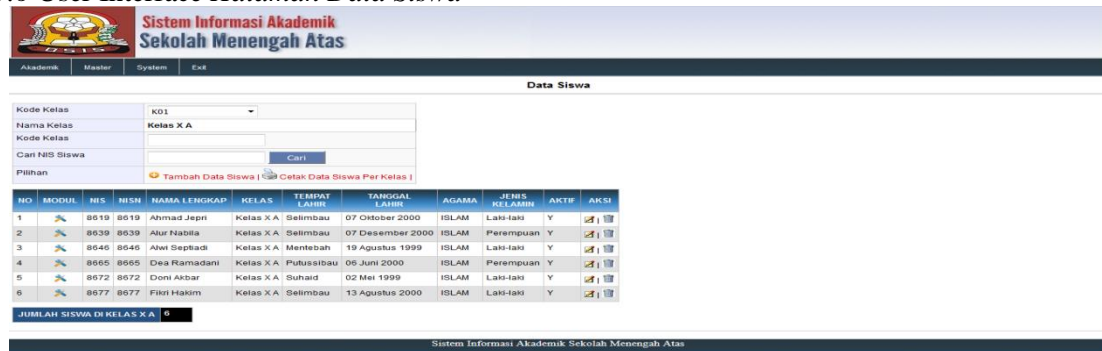
3.5 User Interface Halaman Daftar Pembayaran Siswa



Gambar 5. User Interface Halaman Daftar Pembayaran Siswa

Halaman Daftar Pembayaran Siswa ini berisi data yang terhubung dengan Sistem Informasi Administrasi Keuangan dengan menggunakan *web service*. *Service* tersebut dapat memanggil seluruh daftar iuran siswa maupun mencari data dari daftar iuran tersebut. *Web service* yang digunakan terbatas pada pemanggilan suatu tabel tertentu dan pencarian data dengan satu parameter tanpa bisa memasukkan data dari Sistem Informasi Akademik ke Sistem Informasi Administrasi Keuangan.

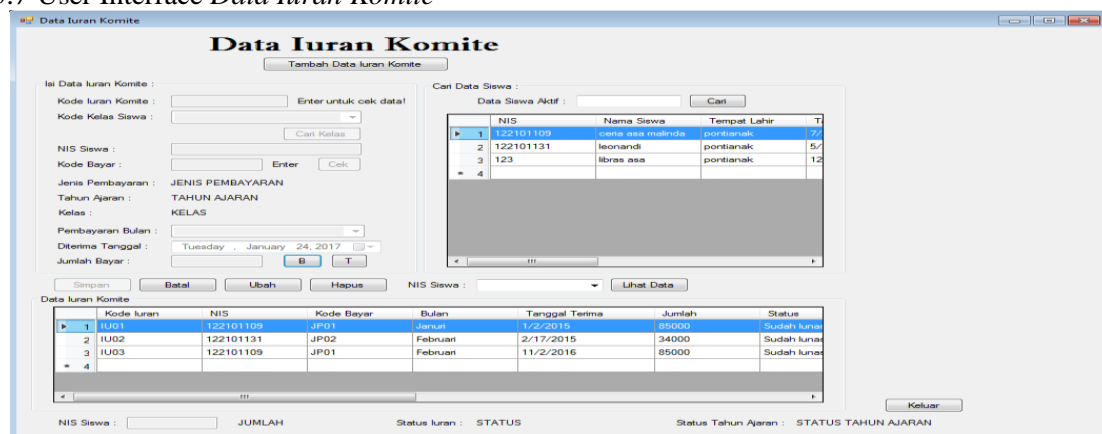
3.6 User Interface Halaman Data Siswa



Gambar 6. User Interface Halaman Data Siswa

Halaman ini berisi form untuk tambah, edit, hapus, cari, dan cetak data siswa ataupun cetak data perkelas. Pada halaman tambah dan edit data siswa, masih belum terdapat validasi untuk setiap input yang dimasukkan.

3.7 User Interface Data Iuran Komite



Gambar 7. User Interface Data Iuran Komite

Form ini berisi tentang data iuran komite yang harus dibayarkan siswa tiap bulan sesuai dengan jenis pembayarannya. Data ini yang nantinya akan diintegrasikan dengan sistem informasi akademik SMA.

3.8 Coding Client Web Service GetIuran

```

65     echo "Daftar Pembayaran Angsuran Siswa TA 2015/2016";
66     $client = new SoapClient("http://localhost:13925/Service1.asmx?WSDL", array('cache_wsdl' => WSDL_CACHE_NONE));
67     ;
68     $iuran = $client->GetIuran();
69     $objju = simplexml_load_string($iuran->GetIuranResult->any);
70     echo "<br><br>";
71     if (is_object($objju)){
72         echo "<table border=1>";
73         echo "<tr><th>Kode Iuran</th><th>NIS</th><th>Kode Bayar</th><th>Bulan</th><th>Tanggal Terima</th><th>Jumlah</th><th>Status</th><th>Status Tahun</th></tr>";
74         foreach ($objju->NewDataSet->Table as $data) {
75             echo "<tr>";
76                 <td>$data->kodeiuran</td>
77                 <td>$data->nis</td>
78                 <td>$data->kodebyr</td>
79                 <td>$data->bulan</td>
80                 <td>$data->tglterima</td>
81                 <td>$data->jumlah</td>
82                 <td>$data->status</td>
83                 <td>$data->statustahun</td>
84             </tr>";
85         }
    
```

Gambar 8. Coding Client Web Service GetIuran

Coding web service pada gambar di atas merupakan coding untuk memanggil web service di server dengan menggunakan SoapClient. Service yang dipanggil adalah function GetIuran dan data yang dihasilkan berupa xml yang ditampilkan di dalam tabel.

3.9 Coding Client Web Service GetListSiswa

```
19 $client = new SoapClient("http://localhost:13925/Service1.asmx?WSDL",array('cache_wsdl' => WSDL_CACHE_NONE));
20 $params = array('uname' => $file);
21 $resul = $client->GetListSiswa($params)->GetListSiswaResult;
22 function obj2array($obj) {
23     $out = array();
24     foreach ($obj as $key => $val) {
25         switch(true) {
26             case is_object($val):
27                 $out[$key] = obj2array($val);
28                 break;
29             case is_array($val):
30                 $out[$key] = obj2array($val);
31                 break;
32             default:
33                 $out[$key] = $val;
34         }
35     }
36     return $out;
37 }
38
39 $resul = obj2array($resul);
40 $cari = $client->GetListSiswa($params);
```

Gambar 9. Coding Client Web Service GetListSiswa

Coding web service pada gambar di atas merupakan coding untuk memanggil web service di server dengan menggunakan SoapClient. Service yang dipanggil adalah function GetListSiswa dan data yang dihasilkan berupa array dan menampilkan data yang dicari berdasarkan parameter berupa nama yang dimasukkan di interface client-nya.

3.10 Arsitektur Web Service



Gambar 10. Arsitektur Web Service

Keterangan arsitektur web service pada gambar di atas dapat dijelaskan sebagai berikut:

- Service provider berupa platform .NET dengan aplikasi ASP.NET untuk mengolah method/function web service. Method web service sendiri dibuat menggunakan bahasa pemrograman VB.NET.
- Service requestor berupa client yaitu perangkat lunak Sistem Informasi Akademik yang memanggil web service yang berada di service provider.
- Service registry berupa IIS dengan versi 7.5 yang memungkinkan user untuk menemukan web services yang tersedia.
- Service description pada web service ditetapkan dengan menggunakan Web Service Description Language (WSDL).
- Service binding pada arsitektur web service di atas menggunakan Simple Access Object Protocol (SOAP).

Penjelasan proses dari arsitektur web service pada gambar di atas adalah sebagai berikut:

- a. ASP.NET *Web Application* pada diagram aplikasi di dalam *platform* Visual Studio 2012 merupakan aplikasi yang digunakan untuk membuat *web service* dengan menggunakan bahasa pemrograman Visual Basic.NET.
 - b. Visual Studio 2012 sendiri secara otomatis menghasilkan dokumen/*file Web Service Definition Language* (WSDL) pada aplikasi tersebut. Saat mengetikkan alamat URL *web service* dengan menambahkan parameter "...?WSDL" di *web browser/client*, aplikasi ASP.NET akan menampilkan *file* WSDL yang berisi deskripsi layanan WSDL mengikat/WSDL *binding* untuk *web service*.
 - c. Pada aplikasi ASP.NET, *file* WSDL dapat dipublikasikan dengan aplikasi IIS (Internet Information Services) versi 7.5 dengan menjadikan *file web service* di ASP.NET sebagai *file* aplikasi, kemudian di dalam IIS dibuat *virtual directory* sesuai dengan lokasi *file* aplikasi *web service* berada. *Web service* di-publish di IIS gunanya agar dua buah atau lebih komputer dapat memanggil *web service* secara local menggunakan IP Address dari IP komputer *web service* berada dengan mengetikkan: **http://localhost:ipaddresswebservice/Service1.asmx** di *web browser*. Atau bisa juga memanggil *web service* untuk menguji *coding web service* dengan komputer sendiri/tidak terhubung dengan komputer lain dengan mengetikkan: **http://localhost:13925/Service1.asmx** di *web browser*.
 - d. *Web service* dipublikasikan agar WSDL dapat ditemukan menggunakan WSDL *contract* pada sisi *client* yaitu Sistem Informasi Akademik. WSDL *contract* sendiri merupakan *file* WSDL yang mendeskripsikan *web service* dan berperan sebagai *contract* antara *web service server* dan *client*. Untuk menghasilkan WSDL *contract*, diperlukan: **http://localhost:ipaddresswebservice/Service1.asmx?WSDL** untuk memanggil *web service* secara IP *local*, ataupun **http://localhost:13925/Service1.asmx?WSDL** untuk pengujian *coding web service* di komputer sendiri.
 - e. Di dalam *layer web service* pesan-pesan SOAP berbasis XML digunakan untuk proses *request* dan *response*.
 - f. Pesan SOAP dikirim sebagai *request* dari *client* yaitu Sistem Informasi Akademik menggunakan variabel \$client yang berisi *class soapclient*. Kemudian pesan tersebut ditransfer melalui HTTP.
 - g. Pesan SOAP dengan parameter *request* bila sesuai dengan standar SOAP maka penyedia layanan yaitu ASP.NET dapat menanggapi permintaan tersebut dan mengirimkan *response* sesuai dengan parameter yang diberikan. *Response* dengan menggunakan *soapclient* membuat bentuk *web service* yang akan muncul berupa *object* yang berisi nama-nama parameter dan tipe data masing-masing. Inilah bagaimana *web service* dapat terjadi pertukaran data antara penyedia layanan (ASP.NET) dengan peminta layanan (Sistem Informasi Akademik berbasis *web*).
 - h. Pada *client web service* yaitu Sistem Informasi Akademik terdapat dua *request web service*. *Web service* pada sisi *client* ini digunakan untuk memanggil *method* GetIuran yang berisi tabel Data Iuran. *Web service* kedua yang dipanggil oleh *client* adalah *method* GetListIuran yang berisi tabel Data Siswa yang dilengkapi dengan pencarian data berdasarkan parameter nama yang dimasukkan. Masing-masing dari *method* di atas merupakan *database* dari Sistem Informasi Administrasi Keuangan yang menggunakan *platform* dan bahasa pemrograman yang berbeda dari Sistem Informasi Akademik. Dengan membuat *web service* berdasarkan *database* yang diambil dari perangkat lunak Sistem Informasi Administrasi Keuangan, diharapkan *web service* tersebut dapat dipakai kembali (*reusable*) pada perangkat lunak baru yang dibuat yang membutuhkan *method* di dalam *web service* tersebut meskipun berbeda *platform* dan bahasa pemrogramannya.
 - i. *Web service* untuk *method* GetIuran dan *method* GetListIuran berada pada menu Daftar Pembayaran Siswa di halaman pertama Sistem Informasi Akademik.
-

- j. Pada *client web service* yaitu Sistem Informasi Akademik untuk memanggil *web service method* GetIuran, maka *client* menggunakan variabel \$client yang berisi *class* soapclient yang mendefinisikan WSDL *contract* sehingga dapat terjadi proses *binding*. Proses *binding* ini diperlukan agar *file* WSDL di *server* dapat dihubungi oleh pihak *client* karena *service description file* WSDL di *server* terdapat elemen *binding* menggunakan *servicesoap* dan lokasi di mana WSDL *file* dapat dihubungi *service requestor*, dalam hal ini *service requestor* adalah Sistem Informasi Akademik. Setelah *binding* dibuat, variabel \$iuran kemudian dibuat dan digunakan untuk memanggil *method* GetIuran() dan *response* yang diberikan oleh *web service server* berupa data tabel Data Iuran dalam bentuk XML menggunakan perintah `simplexml_load_string` agar data dapat diolah dan ditampilkan menjadi data tunggal tanpa tipe data dari masing-masing kolom. Kemudian data-data tersebut ditampilkan di dalam tabel yang sudah dibuat di halaman Daftar Pembayaran Siswa. *Method* GetIuran dari *web service server* sendiri digunakan untuk menampilkan data-data iuran yang telah dibayarkan siswa dengan kategori yang SUDAH LUNAS perbulan dalam periode Tahun Ajaran aktif tanpa harus memanggil langsung *database* dari perangkat lunak Sistem Informasi Administrasi Keuangan karena perbedaan *platform* dan bahasa pemrograman yang digunakan.
- k. *Web service method* kedua yang berada di dalam Daftar Pembayaran Siswa adalah *method* GetListSiswa. Pada *method* GetListSiswa ini diperlukan masukan parameter nama untuk dapat mengakses *web service* GetListSiswa di *web service server*. Maka dari itu dibuat variabel untuk POST parameter nama siswa. Sama seperti *web service* untuk pemanggilan *method* GetIuran, *binding* dibuat terlebih dahulu. Setelah itu dibuat lagi variabel \$params untuk membuat variabel POST yang telah dibuat menjadi data *array* karena pada *web service method* GetListIuran di *web service server* menggunakan *List* untuk mengembalikan nilai *array*. Kemudian dibuat variabel \$cari untuk memanggil *method* GetListSiswa berdasarkan data *array* dari variabel POST yang sudah dibuat yaitu masukan parameter nama siswa. Selanjutnya *client* mendapat *response* dari *web service server* berupa data dari tabel Data Siswa dalam bentuk *array* dan dimasukkan ke dalam tabel berupa data tunggal tanpa tipe data. *Web service method* GetListIuran pada *client* Sistem Informasi Akademik digunakan untuk menampilkan data siswa berdasarkan nama siswa yang dimasukkan tanpa harus memanggil *database* dari Sistem Informasi Administrasi Keuangan secara langsung karena perbedaan *platform* dan bahasa pemrograman yang digunakan.

3.11 Pengujian Web Service pada Pencarian Daftar Pembayaran Siswa

Tabel 1. Black-Box Pencarian Daftar Pembayaran Siswa

Nama kasus uji	Pencarian Daftar Pembayaran Siswa
Tujuan	Menguji <i>function web service</i> yang dilakukan oleh pengunjung dengan hak akses manapun dapat melihat daftar pembayaran siswa dan melakukan pencarian data siswa
Prosedur pengujian	<ol style="list-style-type: none"> 1. Pengunjung memasukkan parameter nama siswa dan menekan tombol cari. 2. Data siswa dengan nama tersebut ditemukan, maka; 3. Tabel menampilkan data siswa yang dicari.
Hasil yang diharapkan	Sistem dapat menampilkan data siswa yang dicari dengan memanfaatkan <i>web service</i> yang ada.
Output yang dihasilkan	Data siswa ditampilkan di tabel berdasarkan nama yang dicari.

4. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan beberapa hal mengenai perancangan dan pengujian perangkat lunak menggunakan *web service* pada sistem informasi akademik, yaitu:

1. Sistem Informasi Akademik menggunakan aplikasi Sublime Text 2 dengan bahasa pemrograman PHP.
2. Sistem Informasi Administrasi Keuangan menggunakan *platform* Visual Studio 2012 dengan bahasa pemrograman VB.NET.
3. *Web service* dibuat menggunakan aplikasi ASP.NET Web Form Application dengan bahasa pemrograman VB.NET dan di-*publish* dengan IIS (*Internet Information Services*) versi 7.5.
4. *Database* dirancang menggunakan XAMPP dengan versi *server* 5.0.45-community-nt.
5. Bentuk penelitian yang digunakan yaitu studi literatur. Sedangkan teknik pengumpulan datanya dilakukan dengan observasi, wawancara, dan studi dokumentasi.
6. Metode penelitian yang digunakan yaitu *Research and Development* (R&D). Sedangkan metode perancangan perangkat lunaknya menggunakan metode *Extreme Programming*.
7. Model penelitian yang digunakan adalah *Unified Modelling Language* (UML) berupa *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.
8. Pengujian perangkat lunak menggunakan metode *Black-box Testing*.
9. Sistem Informasi Akademik (SIKAD) Sekolah Menengah Atas dapat mengelola data Kepala Sekolah, guru atau karyawan, dan siswa, nilai siswa, mata pelajaran, jadwal ujian dan ekstrakurikuler, serta mencetak data-data tersebut sesuai dengan kategori yang dipilih.
10. SIKAD ini dapat diakses dengan empat level yaitu level Administrator, Kepala Sekolah, Guru/Karyawan, dan Siswa dengan masing-masing hak levelnya. Khusus administrator dapat mengatur hak akses masing-masing *user* dengan modul yang sesuai.
11. SIKAD ini dapat digunakan untuk Sekolah Menengah Atas yang mempunyai kegiatan ekstrakurikuler maupun tidak ada.
12. SIKAD berbasis *web* ini dapat dijadikan ajang promosi sekolah.
13. Sistem Informasi Administrasi Keuangan (SIK) Sekolah Menengah Atas dapat mengolah data keuangan sekolah berupa pembayaran iuran siswa, jenis pembayaran dan rinciannya, pemasukan, serta pengeluaran sekolah dan dapat mencetak data-data tersebut.
14. SIKAD dapat memanggil fungsi dari SIK yang berbasis *desktop* menggunakan *web service*.

5. SARAN

Setelah kesimpulan di atas dipaparkan, maka diharapkan untuk pengembangan perangkat lunak Sistem Informasi Akademik Sekolah Menengah Atas ini selanjutnya dapat ditambahkan beberapa hal yaitu:

1. Fitur-fitur baru seperti pemanfaatan *web service* dalam pertukaran data antar *platform web* dengan VB.NET ataupun *platform web* dengan Android.
 2. SIKAD SMA untuk selanjutnya dapat dibuat dalam versi Android ataupun IOS.
 3. Dapat ditambahkan modul daftar alumni pada SIKAD SMA.
 4. Dapat ditambahkan validasi untuk setiap *input*-an data pada menu tambah dan ubah data.
 5. Dapat ditambahkan judul-judul berita atau informasi dari sekolah di menu *sidebar* maupun fitur pencarian judul berita atau informasi tersebut sebagai jalan pintas untuk membuka informasi tersebut tanpa harus bolak balik halaman *web*.
-

6. Sistem Informasi Administrasi Keuangan untuk ke depannya dapat dibuat secara *online* dengan meningkat sistem keamanan pada perangkat lunak tersebut.
- 7.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan YME, Ketua STMIK Pontianak, Ketua Jurusan Teknik Informatika STMIK Pontianak, Pembimbing Skripsi dan Jurnal, orangtua tercinta, teman dekat dan kerabat yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Nuari, Novi, 2013, Perancangan Aplikasi Layanan Mobile Informasi Administrasi Akademik Berbasis Android Menggunakan Web Service (Studi Kasus Reg. B Universitas Tanjungpura), *Skripsi*, Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Tanjungpura.
- [2] Christanto, Ari Tunggul Sri, 2015, Penerapan Service Oriented Architecture Menggunakan Web Service pada Aplikasi Perpustakaan Berbasis Android, *Skripsi*, Kantor Sistem Informasi, Universitas Atma Jaya Yogyakarta.
- [3] Kapojos, F, 2011, Implementasi Service Oriented Architecture dengan Web Service untuk aplikasi Informasi Akademik, *Skripsi*, Jurusan Teknik Elektro dan Komputer, Universitas Sam Ratulangi.
- [4] Sujadi, 2003, *Metodologi Penelitian Pendidikan*, Rineka Cipta, Jakarta.
- [5] Usman, Hudaeni dan P.S Akbar, 2003, *Metodologi Penelitian Sosial*, Bumi Aksara, Jakarta.
- [6] Hadi, Sutrisno, 2004, *Metodologi Research*, Andi Offset, Yogyakarta.
- [7] Moleong, Lexy, 2004, *Metode Penelitian Kualitatif*, Remaja Rosdakarya, Bandung.
- [8] Pressman, R.S, 2012, *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7: Buku 1*, Andi Offset, Yogyakarta.
- [9] Shalahuddin dan Rosa, 2013, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika Bandung.
- [10] Munawar, 2005, *Pemodelan Visual dengan UML*, Andi Offset, Yogyakarta.
- [11] Shalahuddin dan Rosa, 2011, *Modul Pembelajaran: Rekayasa Perangkat Lunak*, Modula, Bandung.
- [12] Indrajani, 2011, *Perancangan Basis Data dalam All in 1*, PT. Elex Media Komputindo, Jakarta.

Penerapan Jaringan Syaraf Tiruan *Backpropagation* Dalam Sistem Pakar Diagnosa Virus TORCH

Application of Backpropagation Artificial Neural Networks in the TORCH Virus Diagnosis Expert System

Erwin Gunawan Walker¹, David²

Teknik Informatika STMIK Pontianak: Jl. Merdeka No. 372, Pontianak 78111

Jurusan Teknik Informatika, STMIK Pontianak

e-mail: erwin_g_walker@yahoo.com, davidliauw@gmail.com

Abstrak

Virus TORCH merupakan singkatan dari virus Toxoplasma, Rubella, Cytomegalo Virus, dan Herpes. Virus TORCH menyebabkan banyak anak yang terlahir cacat. Tujuan dari penelitian ini adalah untuk menghasilkan dan membangun sebuah perangkat lunak sistem pakar yang dapat membantu penderita dalam melakukan diagnosa awal terhadap virus TORCH, agar dapat mengurangi resiko anak yang terlahir cacat yang disebabkan oleh virus ini. Bentuk metode penelitian adalah studi kasus dan wawancara dalam mencari tahu tentang virus TORCH. Pembuatan sistem pakar ini menggunakan Visual Basic .NET dalam pembuatan aplikasinya. Sistem pakar yang digunakan adalah forward chaining yang dipadankan dengan jaringan syaraf tiruan Backpropagation sebagai pengambilan keputusan. Sistem pakar ini menampilkan pilihan gejala yang dapat dipilih user, dimana setiap pilihan gejala akan membawa user kepada pilihan gejala selanjutnya sampai mendapatkan hasil akhir. Sedangkan database yang digunakan adalah menggunakan MySql. Model perancangan yang digunakan adalah UML (Unified Modeling Language) dan flowchart. Metode perancangannya adalah prototype dan pengujian perangkat lunak menggunakan black box untuk menguji sistem yang telah dibangun. Hasil penelitian ini adalah dapat membantu masyarakat agar dapat lebih memahami dan mengetahui virus TORCH yang diderita sehingga dapat segera diobati untuk menghindari anak yang terlahir cacat. Saran kedepannya adalah agar sistem pakar ini dapat bersifat online.

Kata kunci : Sistem Pakar, Backpropagation, Forward Chaining, Virus TORCH

Abstract

Virus TORCH is an acronym of the virus Toxoplasma, Rubella, Cytomegalo virus, and Herpes. TORCH virus causes many children born with disabilities. The purpose of this study is to generate and build an expert system software that can help the patient in making an early diagnosis of the TORCH virus, in order to reduce the risk of children born with disabilities caused by this virus. Method research design is a case study and interviews in finding out about the TORCH virus. This expert system making use of Visual Basic .NET in the making of the application. An expert system is paired with a forward chaining neural network Backpropagation as decision-making. This expert system featuring a large selection of symptoms that can be selected user, where each choice of symptoms will bring the user to the selection of the next symptom to get the final result. While the database used is MySQL. Design model used is UML (Unified Modeling Language) and flowchart. Its design method is a prototype and testing software using black box testing systems have been built. The results of this study is to help the public to better understand and know TORCH suffered virus so that it

can be treated promptly to avoid the children who are born with disabilities. Suggestions in the future is that this expert system can be online

Keywords : *Expert System, Backpropagation, Forward Chaining, TORCH Virus*

1. PENDAHULUAN

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli [4]. Sistem pakar terdiri dari dua bagian yang harus dimiliki yaitu *knowledge base* dan *inference engine*. *Knowledge base* adalah sebuah database yang menyimpan informasi pengetahuan tertentu dan aturan-aturan tentang subjek tertentu, sedangkan *inference engine* adalah bagian dari sistem pakar yang mencoba menggunakan informasi yang diberikan untuk menemukan objek yang sesuai.

Komponen sistem pakar yaitu antar muka pengguna (*user interface*), basis pengetahuan (*knowledge base*), mesin inferensi (*inference machine*), memori kerja (*working memory*), *workplace*, fasilitas penjelasan dan perbaikan pengetahuan. Salah satu implementasi yang diterapkan sistem pakar dalam bidang kesehatan yaitu sistem pakar untuk diagnosa virus TORCH. Virus TORCH adalah singkatan dari *Toxoplasma, Rubella, Cytomegalo Virus, dan Herpes*. Virus TORCH merupakan virus yang dapat menyebabkan anak terlahir cacat bila ibu yang sedang hamil terinfeksi virus ini. Gejala virus TORCH pada awalnya seperti demam, batuk pilek, dan lain-lain. Sehingga pada ibu hamil dapat mewaspadaai pada gejala yang terjadi pada mereka. Agar tidak ada kesalahan diagnosa dan untuk mempermudah masyarakat atau penderita mengetahui sejak dini penyakit yang diderita sehingga tidak terlambat mendapatkan pengobatan. Penggunaan jaringan syaraf tiruan *Backpropagation* digunakan dalam sistem pakar ini.

Penelitian sebelumnya mengusulkan sebuah sistem pakar dengan menggunakan jaringan syaraf tiruan *Backpropagation* untuk mendiagnosa penyakit kulit. Sistem pakar ini mendiagnosa dengan mengambil gambar kulit seseorang yang akan di diagnosa dan kemudian di bandingkan dengan database. Berdasarkan gambar tersebut dapat diketahui apakah penyakit kulit yang diderita orang tersebut [1]. Penelitian lainnya adalah mendiagnosa penyakit gigi. Pengujian dilakukan dengan mengambil 50 sampel data yang belum pernah dilatih hasil menunjukkan bahwa 45 dari 50 (90 %) data yang diuji dapat dikenali oleh sistem [2]. Berikutnya adalah diteksi pengeroposan tulang dengan membandingkan hasil gambar *x-ray*. Sistem kerja di dalam perangkat lunak ini meliputi 3 pemrosesan penting, yaitu proses pengolahan citra dasar, proses reduksi piksel, dan proses jaringan syaraf tiruan dalam mendiagnosa [3].

Penelitian ini menggunakan metode pencarian menggunakan *forward chaining*. *Forward chaining* merupakan teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Sistem ini akan menampilkan pilihan gejala yang dapat dipilih oleh user, dimana setiap pilihan gejala akan membawa user kepada pilihan gejala selanjutnya sampai mendapatkan hasil akhir. Penelitian ini juga menggunakan metode jaringan syaraf tiruan *backpropagation* sebagai penalaran pencarian solusi.

2. METODE PENELITIAN

Penelitian ini berbentuk survey, yaitu mencatat gejala dan solusi virus TORCH dari pakar. Pengumpulan data dilakukan dengan melakukan studi dokumentasi dan observasi. Hasil dari studi dokumentasi dan observasi diagnose virus TORCH dikumpulkan dan menjadi sampel yang digunakan sebagai bahan inferensi jaringan syaraf tiruan. Metode perancangan perangkat

lunak menggunakan prototype dengan pendekatan UML (*Unified Modeling Language*) dan flowchart.

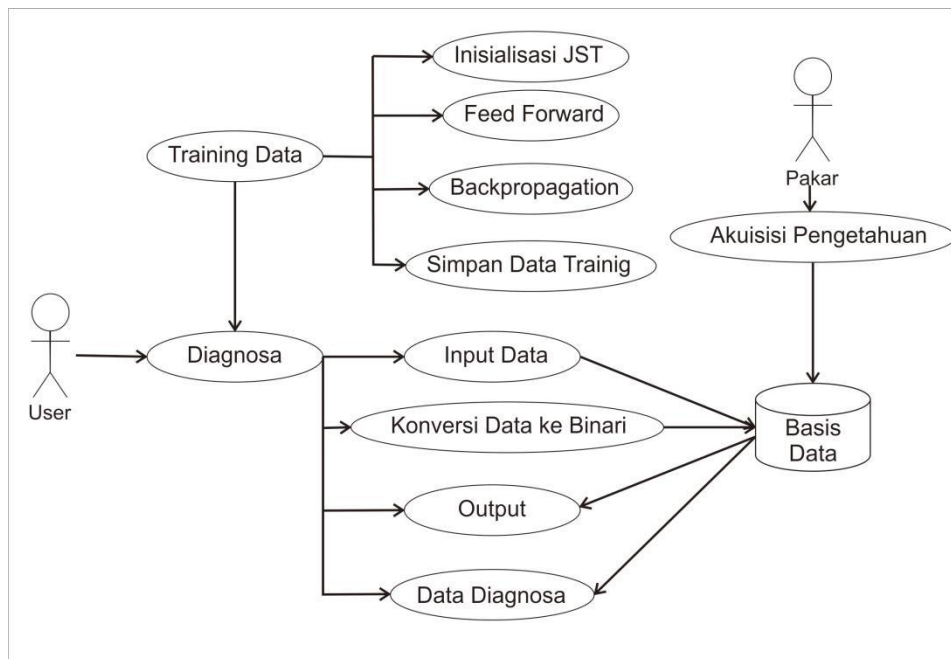
Untuk metode pengujiannya adalah menggunakan pengujian *black-box* yaitu pengujian memungkinkan perekayasa perangkat lunak mendapatkan serangkaian input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pembuatan aplikasi ini menggunakan Visual Basic .net dengan basis data MySQL. Analisis dan perancangan aplikasi dalam penelitian ini menggunakan model UML dan *flowchart*, untuk tahap analisis, metode ini memeriksa kebutuhan yang harus dipenuhi sebuah sistem dan menganalisis data-data yang diperlukan dalam sistem yang dirancang, sedangkan untuk tahap desain, metode ini memfokuskan pada desain pembuatan program perangkat lunak termasuk struktur data, dan arsitektur perangkat lunak [7]. Metode pengujian menggunakan pengujian blackbox yaitu pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan[7].

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, dilakukan indentifikasi kebutuhan yang di perlukan dalam mengembangkan sistem pakar diagnosa virus TORCH. Agar sistem pakar yang di rancang dapat lebih terarah dalam mendiagnosa seorang ibu hamil, maka perlu dilakukan indentifikasi kebutuhan. Identifikasi kebutuhan fungsional (*functional requirement*) adalah kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang mampu dikerjakan oleh sistem pakar. Dimana sistem pakar diagnosa virus TORCH dapat dapat memberikan beberapa fungsi seperti:

- a Sistem pakarmampu menampilkan *form* yang akan digunakan dalam mendiagnosa seseorang.
- b Sistem pakarmampu menampilkan pertanyaan yang akan digunakan dalam mendiagnosa.
- c Sistem pakar mampu melakukan inisialisasi struktur dan bobot jaringan saraf tiruan (JST) dan melakukan pelatihan jaringan syaraf tiruan (JST) untuk mengenal pola gejala virus TORCH.
- d Sistem pakar mampu memberikansolusi yang didapat dari jawaban *user*, yang telah diproses oleh sistem pakar.

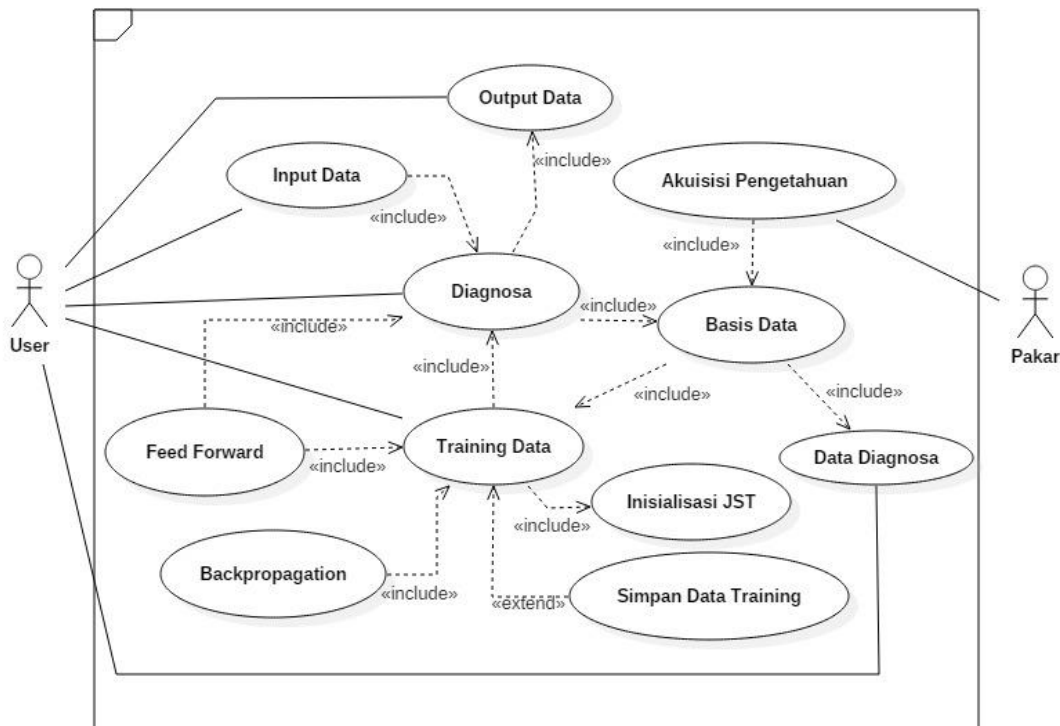
Terdapat beberapa bagian pada arsitektur sistem pakar yang dirancang pada sistem ini diantaranya adalah basis data, basis pengetahuan, akuisisi pengetahuan, mesin inferensi, jaringan syaraf tiruan *Backpropagation*, pengguna dan pakar(gambar 1).



Gambar 1. Arsitektur Sistem Pakar

Dari gambar diatas (Gambar 2) memperlihatkan gambaran umum dari sistem pakar diagnosa virus TORCH. Gambar arsitektur di atas menunjukansistem pakar akanmelakukan *training* data. Di dalam *training* data terdapat beberapa proses, yaitu *backpropagation*, *feedforward*, inisialisasi jaringan syaraf tiruan, dan simpan data *training*. Setelah itu, *user* akan memulai diagnosa. Dimana sistem pakar akan memberikan beberapa pertanyaan. Dari pertanyaan tersebut, *user* akan memberikan *input* yang sesuai dengan pertanyaan dari sistem pakar. Dari *input* yang diberikan *user*, sistem pakar mulai mendiagnosa dan selajutnya data di teruskan ke basis data. Dimana basis data berisikan akuisisi pengetahuan yang telah didapat dari seorang pakar atau dokter kandungan untuk mendiagnosa dan memberikan solusi. Solusi tersebut akan menjadi *output* yang akan ditampilkan kepada *user*. Pada saat diagnosa selesai, maka sistem pakar akan menyimpan data diagnosa dan data tersebut dapat dilihat kembali oleh *user*.

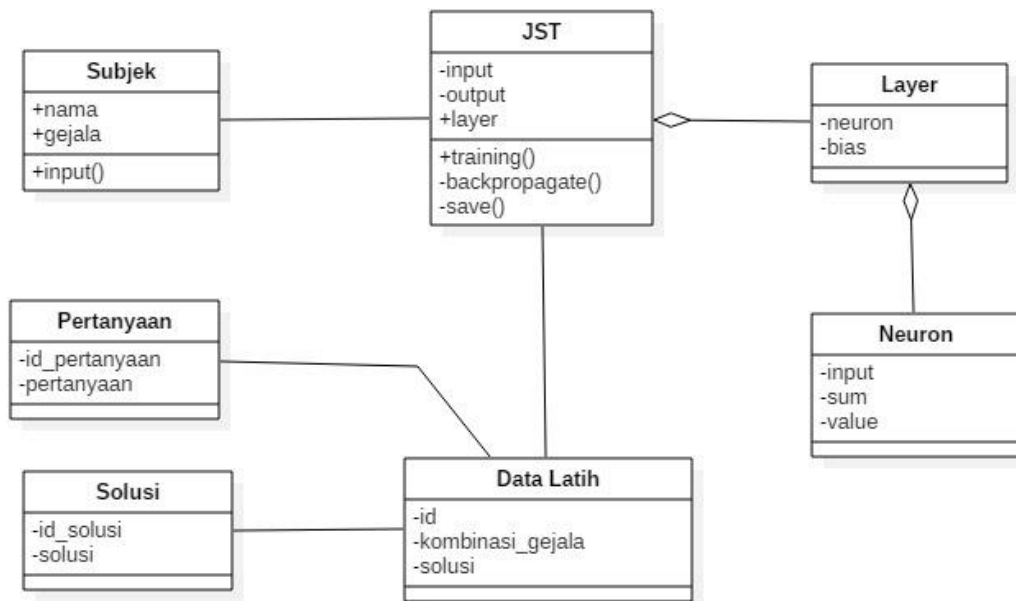
Untuk melihat garis besar dari sistem pakar diagnosa virus TORCH secara fungsional, maka dibuatlah sebuah diagram *use case*. *Use case* didokumentasikan menggunakan *use case diagram* tingkat tinggi, Himpunan *use case* mewakili semua interaksi yang mungkin akan dijelaskan dalam kebutuhan sistem. Aktor dalam proses, yang mungkin manusia atau sistem lainnya direpresentasikan sebagai *stick figure*. Setiap kelas interaksi direpresentasikan sebagai clips yang diberi nama. Garis menghubungkan aktor dengan interaksi. Secara fakultatif, panah dapat ditambahkan ke garis untuk menunjukkan bagaimana interaksi dimulai.



Gambar 2. Diagram Use case Sistem Pakar Diagnosa Virus TORCH

Pada gambar diagram *use case* di atas menunjukkan cara kerja sistem pakar diagnosa virus TORCH. Saat *user* menggunakan sistem pakar diagnosa virus TORCH, *input* data yang diberikan oleh *user* akan diproses menuju diagnosa, dan pada bagian *training* data akan memproses pola-pola yang pernah terjadi sebelumnya. Setelah itu semua data diteruskan ke basis data untuk memperoleh hasil. Dimana hasil tersebut berasal dari perbandingan dengan data akuisisi pengetahuan yang di dapat dari seorang pakar. Kemudian dari hasil tersebut akan menjadi *output* yang akan ditamirkan kepada *user*, dan data tersebut akan disimpan di dalam data diagnosa supaya *user* dapat melihat kembalidata yang terdahulu.

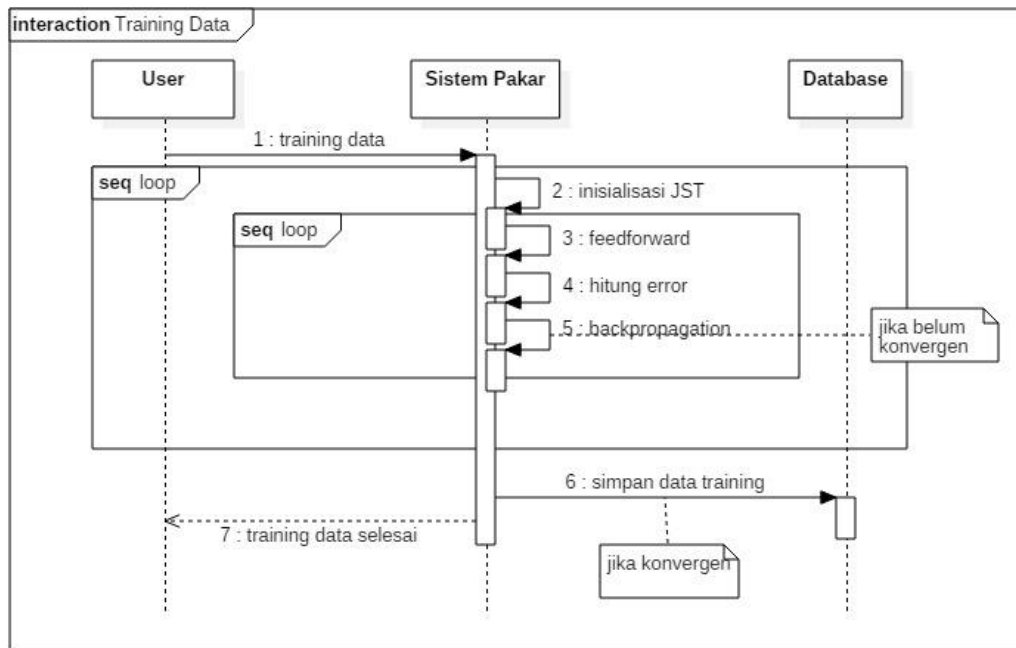
Untuk menggambarkan arsitektur sistem pakar diagnosa virus TORCH, maka dibuatlah *class diagram*. Gambar berikut merupakan *class diagram* dari sistem pakar diagnosa virus TORCH. *Class diagram* dapat dianggap sebagai definisi umum dari satu jenis objek sistem. Sebuah asosiasi adalah hubungan antara kelas-kelas yang menunjukkan hubungan antara kelas-kelas ini. Oleh karena itu, setiap kelas mungkin harus memiliki beberapa pengetahuan dari kelas yang terkait. *Class diagram* dalam UML dapat diekspresikan pada tingkat detail yang berbeda. Ketika mengembangkan model, langkah pertama biasanya untuk melihat dunia, mengidentifikasi objek yang penting dan menyajikannya sebagai sebuah kelas, Cara termudah untuk menulis ini adalah menulis nama kelas dalam sebuah kotak. Dapat juga dengan hanya mencatat keberadaan sebuah asosiasi dengan menggambar garis antara kelas.



Gambar 3. Class Diagram Sistem Pakar Diagnosa Virus TORCH

Class diagram diatas menunjukkan gambaran dari sistem pakar yang sedang dikembangkan. Dalam sistem pakar ini, kelas utamanya adalah kelas jaringan syaraf tiruan yang digunakan untuk mendiagnosa. Kelas jaringan syaraf tiruan tersusun dari layer, *input*, dan *output*. layer merupakan kumpulan dari objek neuron. Sedangkan *input* dan *output* berasal dari data latih, dimana data latih berisikan kombinasi gejala yang di dapat dari kelas pertanyaan dan solusi yang di dapat dari kelas solusi.

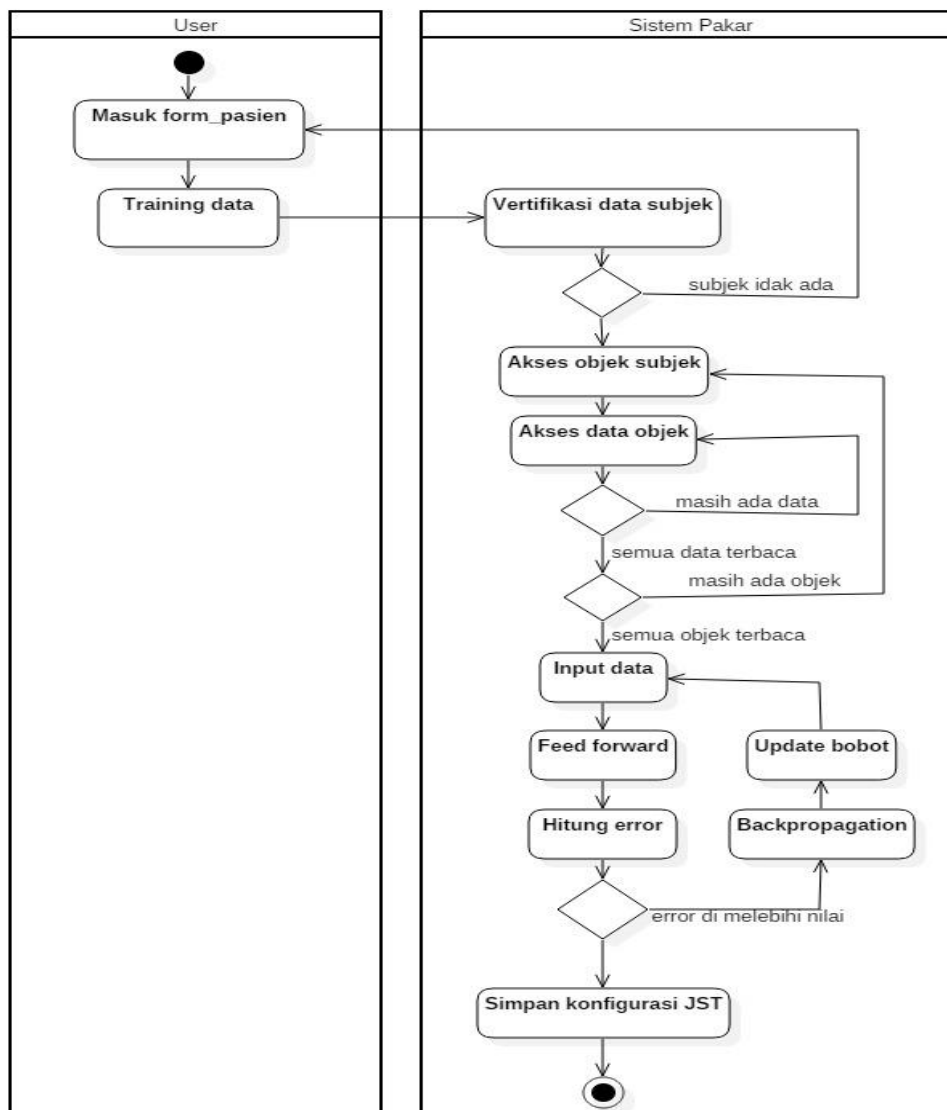
Setelah melakukan class digram, diagram berikutnya adalah *sequence* diagram. untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu, Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.



Gambar 4. *Sequence Diagram Trainig Data*

Dari *sequence diagram trainig data* dapat dilihat, *user* akan melakukan *training data* sebelum melakukan *diagnosa*. Proses *training data* dilakukan dengan melakukan inisialisasi jaringan syaraf tiruan (JST) terlebih dahulu. Proses inisialisasi jaringan syaraf tiruan(JST) dilakukan dengan menginisialisasi beberapa jumlah *layer* dan *neuron* yang telah ditentukan bobot awal secara acak. Data akan dipropagasi maju dalam jaringan, kemudian melakukan perhitungan kesalahan yang terjadi. Jika kesalahan diatas ambang toleransi (belum konvergen) sistem akan mengseksekusi *backpropagation* untuk perbaikan bobot. Jika sudah konvergen, konfigurasi jaringan syaraf tiruan(JST) dan bobot akan di simpan komputer dalam format .NETAK.

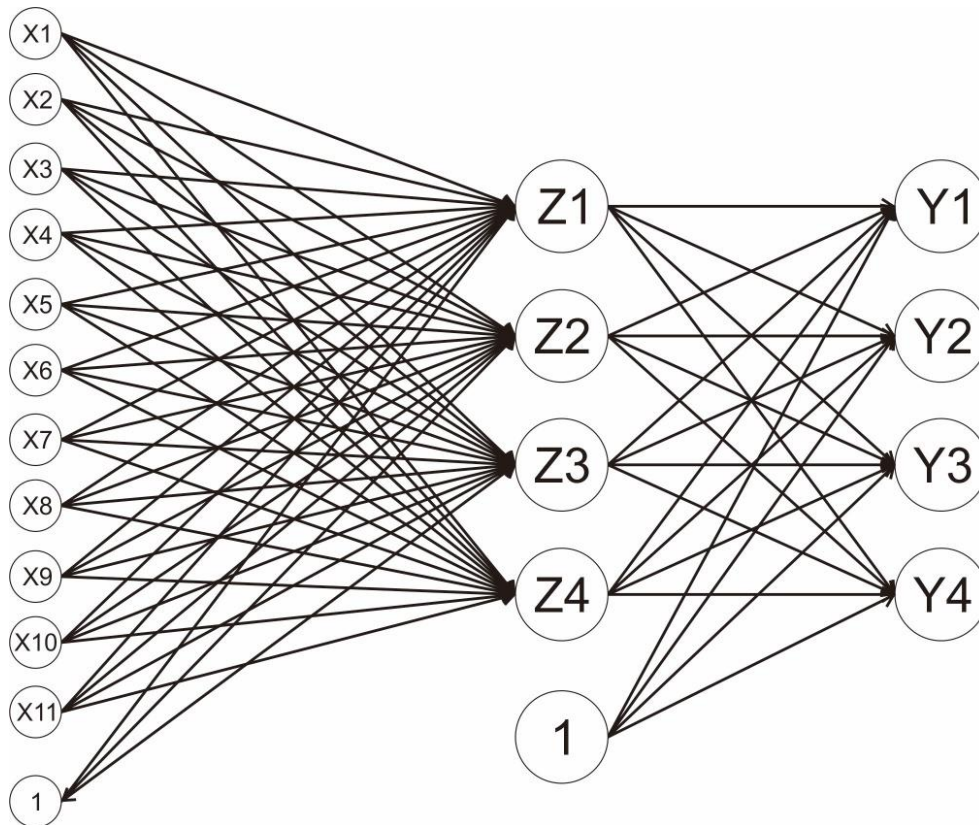
Berikutnya adalah proses dari *activity diagram*. Awal proses ditunjukkan dengan lingkaran tersisi dan di akhir ditunjukkan dengan lingkaran terisi di dalam lingkaran lain. Persegi panjang dengan sudut yang membulat mewakili kegiatan, yaitu sub-proses yang spesifik yang harus dilakukan. UML *activity diagram* , panah mewakili aliran kerja dari satu aktivitas ke aktivitas lainnya. Panah dapat dijelaskan dengan *guards* yang menunjukkan kondisi ketika aliran diambil.



Gambar 5. Activity Diagram Training Data

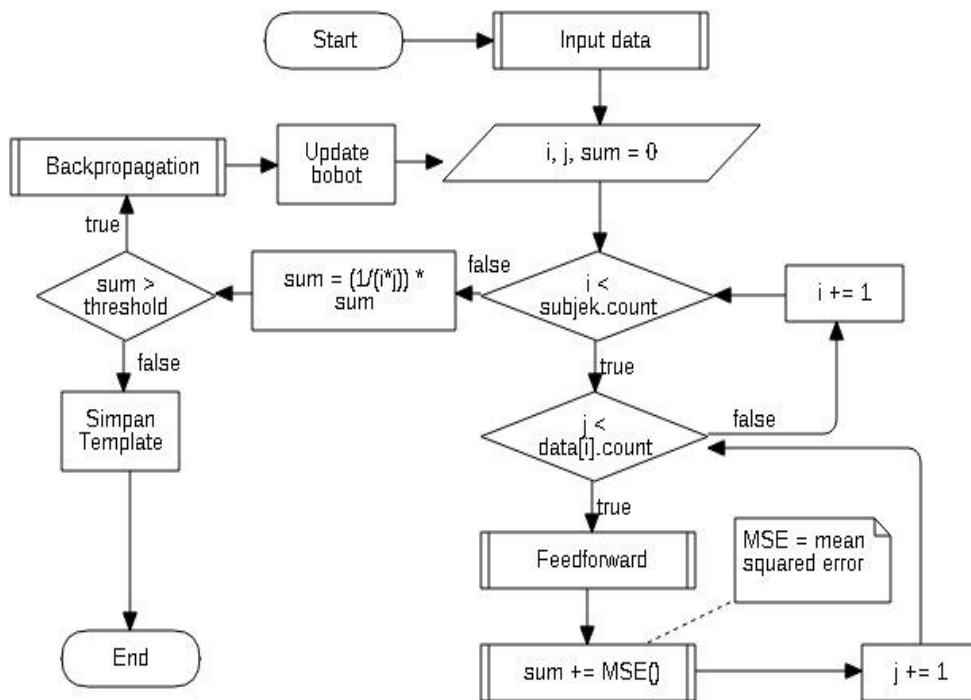
Gambar diatas merupakan *activity diagram* dari *training data*. Saat *user* akan melakukan *training data*, sistem pakar akan memverifikasi keberadaan objek gejala. Setelah itu sistem pakar akan melanjutkan ke dalam jaringan syaraf tiruan (JST) untuk dilatih. Selanjutnya untuk setiap data dan setiap objek akan dimasukkan ke dalam tahap *feedforward*, *backpropagation* dan perbaikan bobot secara berurutan. Proses ini berlangsung sampai tercapainya titik konvergen (JST berhasil dilatih).

Backpropagation adalah salah satu metode dari jaringan syaraf tiruan yang dapat diaplikasikan dengan baik dalam bidang peramalan (*forecasting*). *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama pelatihan. Untuk arsitektur jaringan syaraf tiruan yang digunakan adalah arsitektur jaringan syaraf tiruan *backpropagation*. Arsitektur tersebut dapat dilihat pada gambar dibawah ini (Gambar 2) [8].



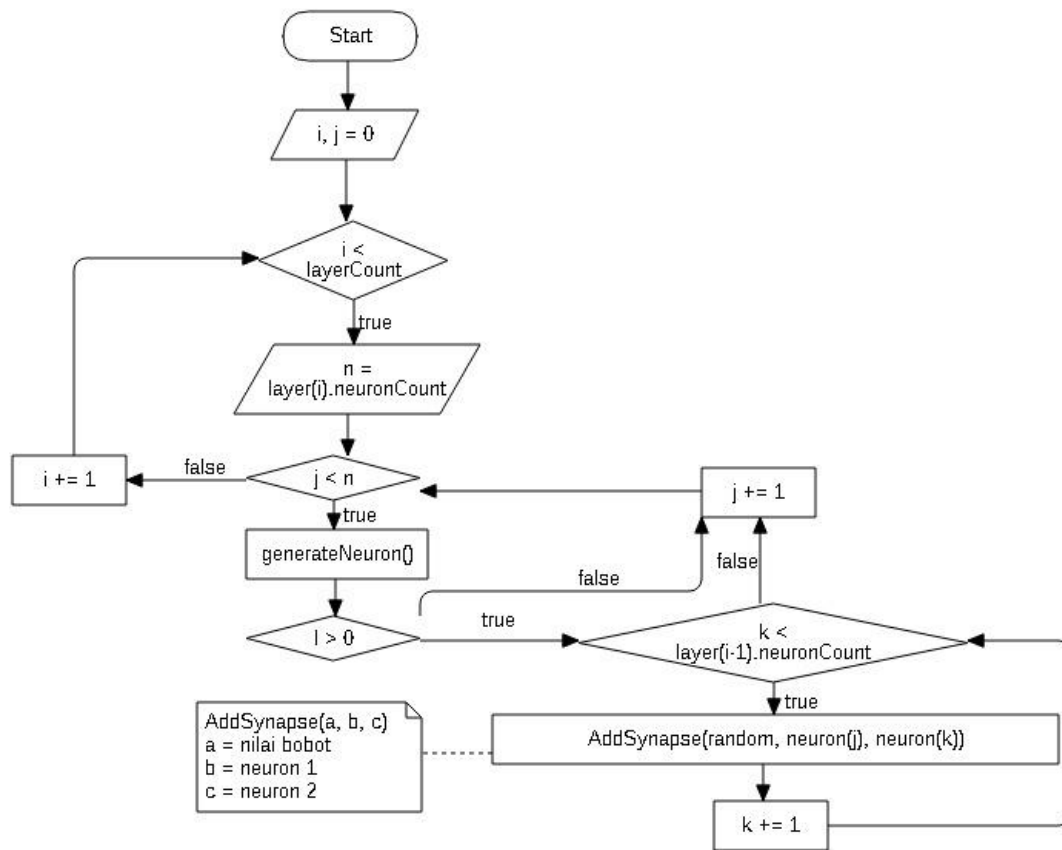
Gambar 6. Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Gambar diatas menunjukkan X adalah *input layer*, Z adalah *hidden layer*, dan Y adalah *output layer*. Saat sistem pakar menerima data masukan, maka data tersebut akan di masukan kedalam bagian X. Kemudian data tersebut akan dihitung dan dilanjutkan ke *hidden layer*. Setelah dihitung, maka data tersebut akan masuk ke *output layer* dan kemudian ditampilkan. Setelah itu, proses *backpropagation* akan dilakukan. Dimana proses *backpropagation* adalah proses menghitung kesalahan dari setiap *layer*. Sehingga proses *backpropagation* akan di mulai dari *output layer*, kemudian *hidden layer*, dan terakhir *input layer*. Setelah proses *backpropagation* selesai, maka sistem pakar akan merubah bobot tersebut sesuai perhitungan kesalahan *backpropagation*.



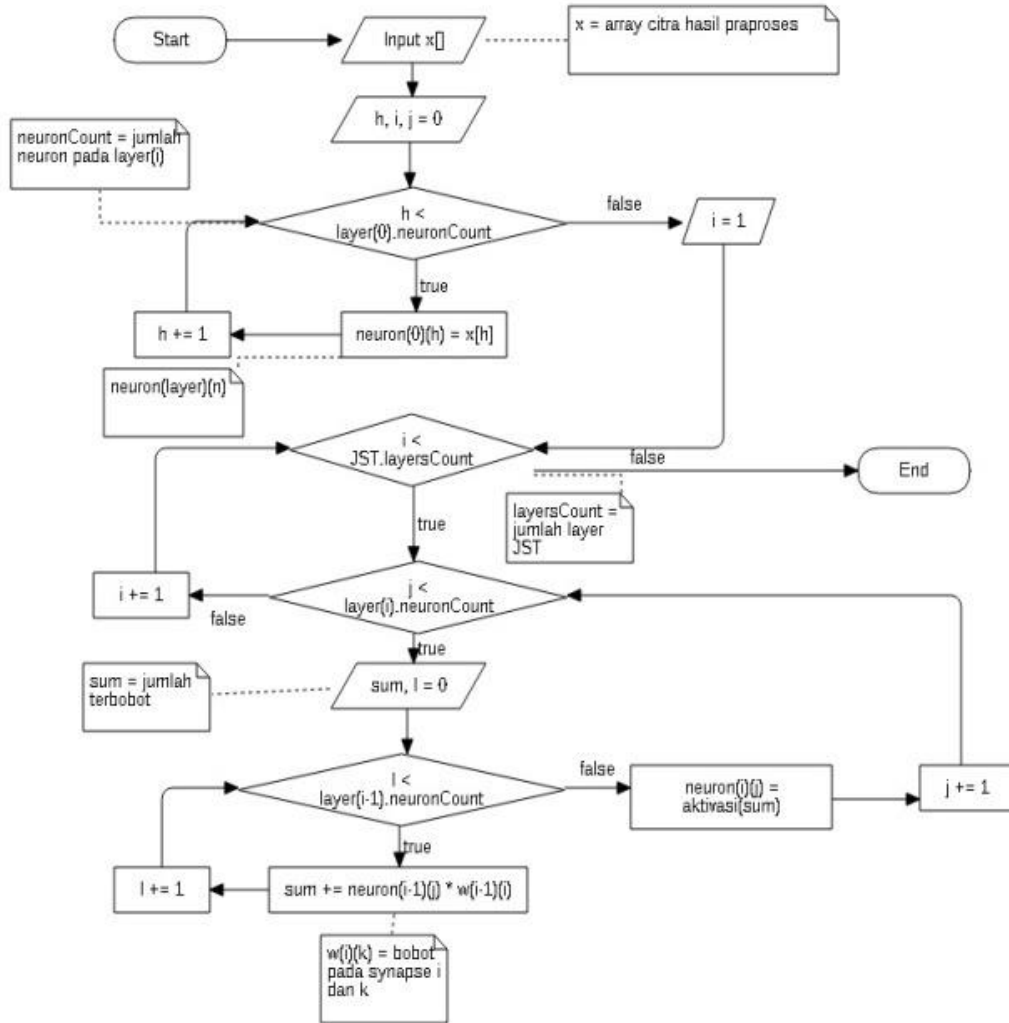
Gambar 7. Flowchart Training Data

Pada gambar *flowcharttrainin* (Gambar 3) datadimulai dengan memasukan pola data gejala. Setelah pola data gejala diterima, akan dilakukan propagasi maju dan perhitungan kesalahan. Kemudian dilakukan pemeriksaan nilai kesalahan dari jaringan yang akan dilatih. Jika nilai kesalahan diatas nilai yang telah ditentukan maka akan dilanjutkan ke tahap propagasi mundur. Perhitungan kesalahan yang di lakukan berasal dari masing – masing sinapsis yang menghubungkan antar *neuron* pada jaringa syaraf tiruan (JST) yang dilatih. Perhitungan ini akan menghasilkan nilai perubahan bobot yang perlu dilakukan pada setiap bobot yang ada dan dapat dilanjutkan dengan proses perbaruan nilai bobot. Jika nilai kesalahan berada dibawah nilai maka konfigurasi jaringan akan disimpan.



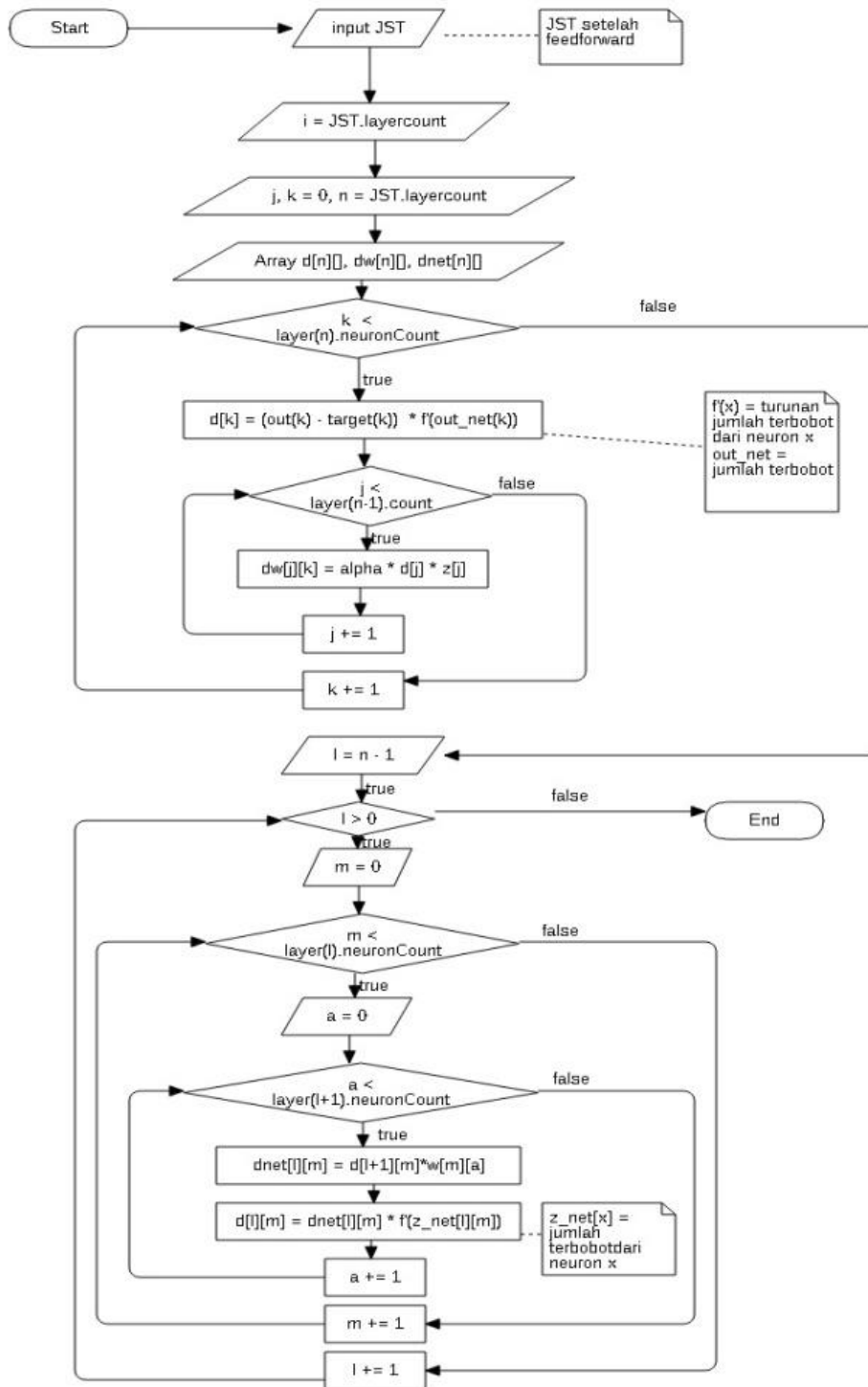
Gambar 8. Flowchart Inisialisasi JST

Pada gambar diatas merupakan flowchart inisialisasi jaringa syaraf tiruan (JST) (Gambar 4). Dalam inisialisasi jaringa syaraf tiruan (JST), sistem pakar akan membagun tiga layer, yaitu input, hidden, dan output. Pertama setiap neuron akan di mulai dari hidden layer yang akan dihubungkan dengan masing-masingneuron pada layer sebelumnya. Neuron tersebut dihubungkan dengan sinapsis yang disertakan dengan nilai bobot awal yang bersifat acak.



Gambar 5. Flowchart Feedforward

Gambar diatas merupakan *flowchart feedforward*. Feedforward (Gambar 5) merupakan tahap propagasi maju, dimana setiap *neuron* masukan akan dilakukan penjumlahan terbobot terhadap masing – masing bobot pada setiap sinapsis yang menghubungkan *neuron* yang bersangkutan ke *layer* selanjutnya. Kemudian dari penjumlahan bobot tersebut akan di masukan ke dalam sebuah fungsi aktivasi yang akan menjadi nilai input untuk *layer* selanjutnya. Proses ini dilakukan untuk setiap *layer* dan setiap *neuron* pada jaringan syaraf tiruan (JST) yang sudah dibentuk.



Gambar 9. Flowchart Backpropagation

Gambar diatas merupakan *flowchart backpropagation* (Gambar 6). *Backpropagation* merupakan tahap propagasi mundur. *Backpropagation* dimulai dengan perhitungan kesalahan pada *layer* terakhir atau *layer output*. Setelah itu akan dicari tanggung jawab dari masing-masing sinapsis yang terhubung atas kesalahan yang terjadi dengan menghitung perubahan bobot yang ada. Proses ini dilakukan secara iteratif terhadap setiap *neuron* pada masing – masing *layer* dari *layer output* ke *layer input*. Hasil tersebut akan digunakan untuk perbaruan nilai bobot pada tahap selanjutnya.

Secara rinci algoritma pelatihan jaringan backpropagation dapat diuraikan sebagai berikut [9]:

- Langkah 0 : Inialisasi bobot (setiap bobot diberi nilai acak antara 0 - 1).
- Langkah 1 : Selama kondisi berhenti belum terpenuhi lakukan langkah 2-3.
- Langkah 2 : Lakukan Langkah 3-13 sebanyak jumlah pelatihan yang diinginkan.
- Langkah 3 : Untuk *output layer* dan setiap *hidden layer* lakukan langkah 4-13.
- Langkah 4 : Hitung *input* setiap *node* pada *hidden layer*
- Langkah 5 : Hitung *output* setiap *node* pada *hidden layer* dengan fungsi aktivasi.
- Langkah 6 : Hitung *input* setiap *node* pada *output layer*.
- Langkah 7 : Hitung output pada setiap *node* pada *output layer*.
- Langkah 8 : Hitung *error* setiap *node* pada *output layer* dengan fungsi deaktivasi.
- Langkah 9 : Hitung perubahan bobot pada setiap *node* pada setiap *hidden layer*.
- Langkah 10 : Hitung *error* setiap *node* pada *output layer* dengan fungsi deaktivasi.
- Langkah 11 : Hitung perubahan bobot pada setiap *node* pada setiap *hidden layer*.
- Langkah 12 : *Update* bobot pada setiap *node* pada *output layer*.
- Langkah 13 : *Update* bobot pada setiap *node* pada setiap *hidden layer*.

Setelah pelatihan, jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut (Kusrini, 2008):

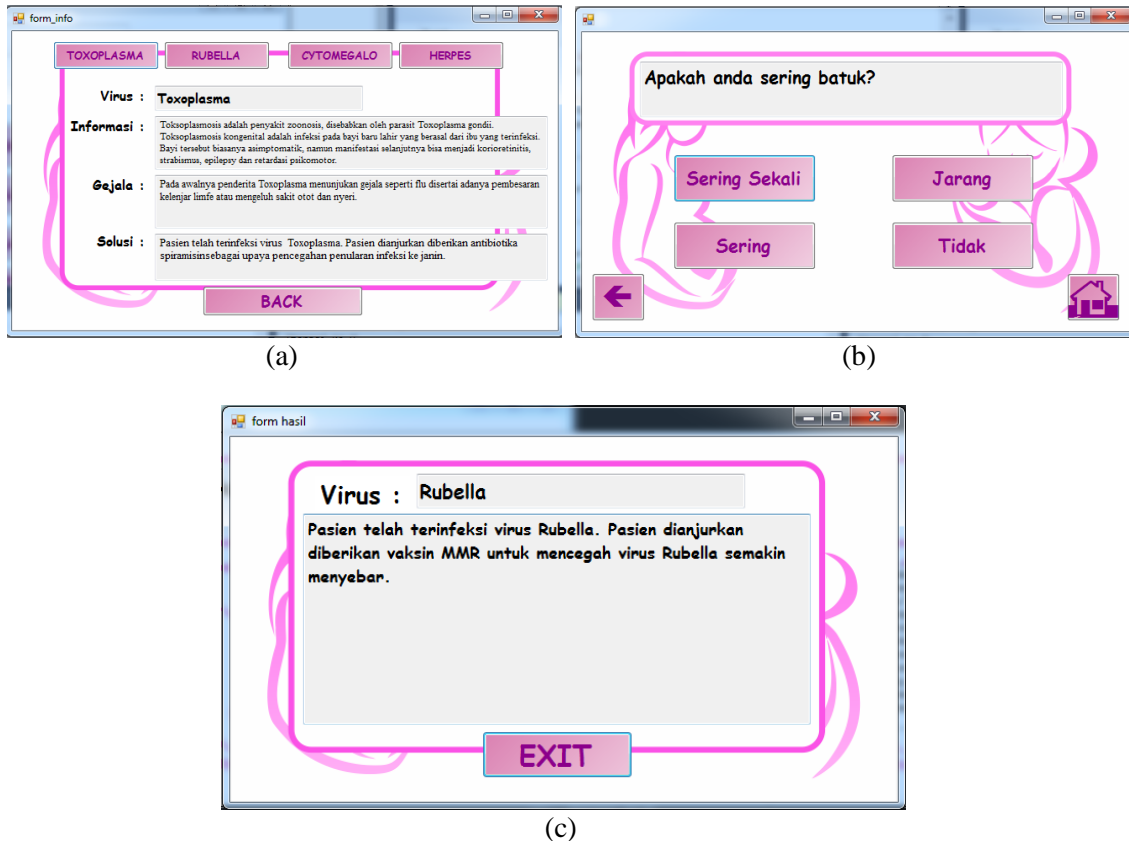
- Langkah 0 : Inialisasi bobot (dari algoritma pelatihan).
- Langkah 1 : Untuk tiap vector masukan, lakukan langkah 2-4.
- Langkah 2 : for $i = 1, \dots,$
- Langkah 3 : for $j = 1, \dots, p :$
- Langkah 4 : for $k = 1, \dots, m :$
- Langkah 5: Jika $y_k \geq 0,5$ maka $y_k = 1$, else $y_k = 0..$

Dari studi kasus dan observasi di dapatkan kesimpulan sebagai berikut (Tabel 1):

Tabel 1. Tabel Pola Gejala Virus TORCH

	Toxoplasma	Rubella	Cyto Megalo Virus	Herpes
Batuk	-	✓	-	-
Demam	✓	✓	✓	-
Flu	✓	✓	-	-
Nyeri	✓	-	-	-
Mata Gatal	-	✓	-	-
Iritasi Kulit	-	✓	-	-
Tenggorokan Sakit	✓	✓	-	-
Lemah	✓	-	✓	-
Lambung Nyeri	-	-	✓	-
Pembengkakan	-	-	✓	-
Cacar Aie	-	-	-	✓

Supaya sistem pakar diagnosa virus TORCH dapat mudah digunakan, maka tampilan sistem pakar ini dibuat sesederhana mungkin. *User interface* adalah bagian dari perangkat lunak yang dapat dirasakan oleh panca indra pengguna, baik penglihatan maupun indra yang lain.



Gambar 7. Tampilan Halaman (a) form_info;(b) form_diagnosa;(c) form_hasil

Dari gambar di atas dapat dilihat, hasil aplikasi yang dibangun akan tampil *user-friendly* agar mudah digunakan user. Penerapan sistem pakar dalam mendiagnosa virus TORCH ini dapat membantu penderita dalam mengetahui awal virus TORCH untuk dapat mengetahui bahaya dari TORCH, agar dapat mencegah atau mengurangi resiko anak yang terlahir cacat yang di timbulkan akibat terlambatnya penanganan virus TORCH. Selain dapat melakukan diagnosa dan memberikan nilai kepercayaan serta solusi yang berhubungan dengan virus TORCH, sistem ini juga dirancang untuk dapat menampilkan informasi mengenai virus TORCH serta dapat melakukan penambahan dan pengubahan data mengenai virus TORCH. Penerapan metode forward chaining dan jaringan syaraf tiruan *Backpropagation* pada aplikasi diagnosa virus TORCH ini dapat mempermudah dan memberikan penyelesaian seberapa pasti user menderita virus TORCH.

untuk pengujian dengan kasus, terdapat 3 ibu hamil yang dijadikan sebagai contoh kasus. Ibu yang pertama adalah ibu Yuli. Ibu Yuli memberitahukan bahwa dia merasa panas demam disertai pilek, nyeri di bagian tangan, sakit tenggorokan, dan mulai merasa lemah. ibu kedua yaitu ibu Desi mengeluhkan bahwa dia merasa demam disertai batuk pilek, matanya mulai gatal tetapi tidak ada debu yang masuk ke mata, kemudian kulitnya mulai gatal seperti terjadi iritasi kulit dan adanya pembengkakan, dan sering sakit di bagian perut di sertai badan

yang mulai lemah. Ibu yang terakhir adalah ibu Yeli. Ibu yeli menatakan bahwa di seperti terserang cacar air pada saat dia hamil. Dari data diatas, dokter mengatakan bahwa ibu Yuli hanya sakit biasa, sedangkan ibu Desi terinfeksi virus *rubella*, dan ibu Yeli terinfeksi virus Herpes. Data diatas dapat dilihat pada tabel dibawah (Tabel 2).

Tabel 2. Tabel Data Gejala

Nama	Pola Gejala											Virus
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	
Yuli	0	1	1	1	0	0	1	1	0	0	0	Tidak Sakit
Desi	1	1	1	0	1	1	1	0	0	0	0	<i>Rubella</i>
Yeli	0	0	0	0	0	0	0	0	0	0	1	<i>Herpes</i>

Data di atas adalah data berdasarkan berdasarkan diagnosa dari dokter. setelah data tersebut di ujikan kepada sistem pakar, hasil yang keluar hampir sama dengan diagnosa dokter. data hasil uji sistem pakar dapat dilihat pada tabel di bawah (tabel 3).

Tabel 3. Tabel Data Gejala UserSetelah Diproses

Nama	Pola Gejala											Virus
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	
Yuli	3,6%	7,5%	8,2%	9,3%	1,2%	4,3%	7,9%	8,1%	1,5%	3,2%	4,3%	<i>Toxoplasma</i>
Desi	8,3%	9,4%	7,9%	5,1%	7,5%	8,2%	7,2%	3,1%	2,1%	1,6%	2,5%	<i>Rubella</i>
Yeli	2,6%	0,7%	3,5%	1,5%	2,2%	3,5%	2,1%	3,1%	2,2%	4,1%	9,4%	<i>Herpes</i>

Tabel diatas adalah hasil dari proses diagnosa setelah jawaban user diproses. Pola gejala diambil dari nilai dari paling besar dan telah dibandingkan dengan data di database. Setelah itu, data akan ditampilkan. Dari hasil di atas di dapatkan, bahwa terjadi kesalahan pada ibu Yuli yang tidak sakit tetapi dikatakan sakit oleh sistem pakar. Sedangkan ibu yang lain mendapatkan hasil yang sama dengan diagnosa dokter.

4. KESIMPULAN

Hasil sistem pakar diagnosa virus TORCH ini menghasilkan kesimpulan sebagai berikut:

- Metode jaringan syaraf tiruan *backpropagation* dapat diimplementasikan pada sistem pakar yang dibangun menggunakan visual basic.net.
- Sistem pakar ini mampu mendiagnosa virus TORCH berdasarkan gejala-gejalanya.

Sistem pakar ini dapat digunakan *user* untuk melakukan diagnosa virus TORCH tanpa harus bertemu dengan seorang pakar.

5. SARAN

Ada beberapa saran yang dapat diberikan berkaitan dengan pengembangan asistem pakar ini. Saran untuk penelitian yang ingin melanjutkan pengembangan sistem pakar ini adalah:

- Pengembangan sistem pakar ini di kemudian hari adalah dapat membuat pilihan untuk menambahkan gejala.

- b. Sistem pakar ini berjalan secara offline, sehingga diharapkan penelitian berikutnya dapat membuat sistem pakar ini berjalan secara *online*.

Dapat mengembangkan sistem pakar ini dengan menggunakan metode lain yang lebih efektif dan efisien.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua dan keluarga yang telah memberikan dukungan baik moral maupun materil kepada penulis untuk selalu berusaha mencapai hasil yang baik. Dan rekan-rekan seperjuangan khususnya di Teknik Informatika angkatan 2013/2014 konsentrasi kecerdasan buatan yang telah memberikan banyak bantuan, dukungan, saran, dan nasehat yang sangat berarti bagi penulis.

DAFTAR PUSTAKA

- [1] Indrawaty, A. Y., Hermana, A. N., & Ramadhan, A. (2012). Implementasi Model Backpropagation Dalam Mengenali Pola Gambar Untuk Mendiagnose Penyakit Kulit. *J. Inform*, 3(1), 1-7.
- [2] Aryasa, K. (2012). Expert System Diagnosa Jenis Penyakit Gigi Menggunakan Jaringan Saraf Tiruan Backpropagation. *CSRID*, 4.
- [3] David, D., & Kosasi, S. (2013). Penerapan Algoritma Jaringan Saraf Tiruan Backpropagation Untuk Pengenalan Pola Tandatangan. *Jurnal Teknologi*, 6(2), 139-146.
- [4] Mardianto, I., & Pratiwi, D. (2008). Sistem deteksi penyakit pengeroposan tulang dengan metode jaringan syaraf tiruan backpropagation dan representasi ciri dalam ruang eigen. *CommIT (Communication and Information Technology) Journal*, 2(1), 69-80.
- [5] David, D. (2015). Pengenalan Pola Plat Nomor Kendaraan Menggunakan Jaringan Syaraf Tiruan Backpropagation. *Sisfotenika*, 3(1), 71-80.
- [6] David, D. (2017). Penerapan Jaringan Syaraf Tiruan Perceptron Dalam Sistem Pakar Diagnosa Penyakit Pada Kambing. *E-Proceedings KNS&I STIKOM Bali*, 85-90.

Pemilihan Paket Wedding Menggunakan Metode Electre

Selection of Wedding Packages Using the Electre Method

Endah Sri Subrata, Susanti Margaretha Kuway, I Dewa Ayu Eka Yuliani

STMIK Pontianak; Jl. Merdeka Barat no. 372, (0561) 735555

Jurusan Teknik Informatika, STMIK Pontianak, Pontianak

e-mail: een363@gmail.com, kuwayshanty@gmail.com, dewaayu.ekayuliani@gmail.com

Abstrak

Seiring dengan perkembangan teknologi yang semakin pesat dan tidak bisa dipungkiri bahwa teknologi sebagai kebutuhan sekunder bagi para penggunanya. Sejauh ini teknologi juga telah diterapkan pada beberapa bidang, salah satu contohnya kehadiran teknologi di sektor pendidikan yang memudahkan para penggunanya untuk melakukan proses pemilihan paket wedding. Sistem pendukung keputusan ini untuk menentukan kriteria-kriteria dalam mengatasi masalah yang dihadapi suatu pernikahan dalam memberikan informasi yang tepat waktu, akurat dan efisien untuk mereka yang membutuhkan informasi tersebut, dimana informasi yang terkandung dalam media penyimpanan data yang khusus disiapkan. Penelitian menggunakan metode ELECTRE untuk proses perhitungannya dan aplikasi berbasis web dengan bahasa pemrograman PHP yang bisa di akses setiap pengguna dimanapun pengguna berada berdasarkan kriteria hasil tersebut pengguna dapat menentukan kriteria mana yang akan diambil dan sesuai dengan budget.

Kata kunci— ELECTRE, Wedding, Sistem pendukung Keputusan, Extreme Programming

Abstrak

Along with the development of technology that is increasingly rapidly and it cannot be denied that the technology as secondary to the needs of its users. So far this technology has also been applied to several fields, one example the presence of technology in the education sector which allows its users to do the wedding package selection process. This decision support system for determining criteria in addressing the issues facing a marriage in providing information that is timely, accurate and efficient for those who need the information, where the information contained in the data storage medium that is specially prepared. Research using ELECTRE method for calculation process and web-based applications with PHP programming language that can access each user regardless of where the user is based on the criteria of the results users can determine which criteria will be taken and according to the budget.

Kata kunci— ELECTRE, Wedding,, decision support system, Extreme Programming

1. PENDAHULUAN

Pernikahan merupakan bentuk hubungan antar manusia yang paling sacral dan utama. Sehingga setiap pasangan akan melakukan persiapan yang total untuk menghadapi pernikahan. Persiapan itu antara lain kelengkapan terselenggaranya sebuah pernikahan. Karena sekali seumur hidup, tentunya kemasan sebuah pernikahan akan dibuat harus sebaik mungkin. Hal-hal seperti ini yang sering kali menyulitkan dan membingungkan para calon pengantin ketika minimnya informasi yang dapat dijadikan acuan dalam pengambilan keputusan tentang pemilihan paket wedding.

Penelitian yang dilakukan oleh Anhar dalam penelitian ini Perancangan Aplikasi Pengambilan Keputusan Penentuan Paket Tujuan Wisata Menggunakan Metode TOPSIS (Studi Kasus PT. Vina Tour dan Travel Medan), penelitian ini termasuk dalam *Multi Atribut Decision Making* (MADM). Metode yang digunakan metode TOPSIS (*Technique for Order Preference by Similiarty to Ideal Solution*). Tujuan dari penelitian ini yaitu untuk membantu konsumen menentukan paket wedding dengan cara merancang sebuah aplikasi pendukung keputusan penentuan paket wisata berdasarkan kriteria yang diinputkan oleh konsumen untuk mendapatkan informasi paket wisata yang sesuai dengan keinginan dan kebutuhan konsumen.[1]

Menurut Setiawan, Indriani, Muliadi melakukan penelitian tentang “Implementasi Metode ELECTRE Pada Sistem Pendukung Keputusan SNMPTN Jalur Undangan”. Pada penelitian ini dibangun Sistem Pendukung Keputusan berbasis web dengan menggunakan metode analisis pengambilan keputusan multikriteria yaitu ELECTRE. Menggunakan metode ELECTRE karena permasalahan ini sesuai dan cocok pada konsep perankingan berdasarkan alternatif dan kriteria yang telah ditetapkan. Metode ELECTRE ini dapat digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi dan alternatif yang sesuai dapat dihasilkan. Keluaran dari sistem ini berupa laporan hasil perankingan calon mahasiswa yang lebih layak masuk di Universitas Lambung Mangkurat sesuai program studi yang diminatinya.[2]

Menurut Setiwati, Hartati, Maulana melakukan penelitian tentang “Sistem Pendukung Keputusan Pembelian Barang Menggunakan Metode ELECTRE”. Dalam melakukan pembelian barang, ada beberapa kriteria yang digunakan oleh Denco Variasi. Kriteria tersebut antara lain, jumlah penjualan barang pada periode sebelumnya dan laba yang didapat dari penjualan tersebut, ketersediaan stok barang di gudang, besarnya modal yang dialokasikan untuk melakukan pembelian. Setelah melakukan analisa, merancang sistem dan membuat aplikasi sistem pendukung keputusan pembelian barang menggunakan metode ELECTRE, serta keputusan ini telah berjalan dengan baik dan sesuai dengan yang diharapkan. Data alternatif yang digunakan didapat dari data penjualan berdasarkan periode yang dipilih, data barang dengan stok minimum dan kategori yang ditentukan. Berikut merupakan proses pembobotan alternatif untuk tiap kriteria yang digunakan.[3]

Penelitian ini menggunakan metode ELECTRE sebagai metode dalam pendukung keputusan ini karena ELECTRE dapat digunakan dalam menyelesaikan berbagai jenis kasus pengambilan keputusan melalui perhitungan yang dapat memberikan nilai preferensi untuk alternatif, dan mampu memilih alternatif terbaik dari alternatif lain, alternative yang dimaksud dalam penelitian ini adalah paket wedding di Pontianak.

2. METODE PENELITIAN

Penelitian ini menggunakan bentuk penelitian berupa studi kasus yaitu bentuk penelitian yang memusatkan perhatian pada suatu kasus tertentu dengan menggunakan individu atau kelompok sebagai bahan studinya. Metode yang digunakan dalam penelitian ini adalah metode penelitian *Research and Development* (R&D).

Teknik perancangan perangkat lunak menggunakan UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. *Use case diagram* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* adalah penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakan antara objek-objek ini didalam *use case*. *Activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram kelas merupakan diagram paling umum dipakai di semua pemodelan berorientasi objek.

Metode *black box*, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian *black box* memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. *Simpul* merepresentasikan suatu objek (seperti modul atau koleksi *statemen*), *link* merepresentasikan hubungan antar objek, *node weight* menggambarkan properti simpul (*atribut*). *link weight* menggambarkan beberapa karakteristik *link*.

3. HASIL DAN PEMBAHASAN

Penelitian yang peneliti lakukan dalam mengembangkan aplikasi system pendukung keputusan ini menggunakan metode *Extreme Programming* memiliki tahapan pengembangan yang dinamis, serta memiliki tahapan pengembangan diantaranya tahap *planning*, *design*, *coding*, dan *tester*. Dengan menggunakan metode *Extreme Programming* ini penulis dapat memangakas kriteria yang sesuai dengan kebutuhan konsumen. kepuasan pengguna dalam memilih paket sesuai kebutuhan dan budget atas aplikasi yang dihasilkan.

Tahap *planning* merupakan tahap dimana peneliti mengidentifikasi setiap kebutuhan yang di perlukan dalam pengembangan aplikasi. Pada tahap ini juga peneliti juga mengumpulkan data-data yang dibutuhkan untuk pengujian aplikasi. Penelitian melakukan beberapa cara dalam mengumpulkan data diantaranya dengan melakukan wawancara di FAMOUS Bridal untuk mendapatkan informasi secara lisan dengan tujuan untuk memperoleh data yang dapat menjelaskan ataupun menjawab suatu permasalahan dalam penelitian kemudian melakukan dokumentasi terhadap apa yang diperlukan oleh aplikasi dengan mencari data berupa, tulisan dan dokumentasi lainnya yang mengenai hal-hal yang berkaitan dengan aplikasi atau variabel yang diperlukan. Selanjutnya melakukan observasi, yaitu mengamati langsung sistem yang sedang berjalan pada bridal-bridal di Pontianak agar bisa dipelajari dan diterapkan terhadap aplikasi yang peneliti rancang. Sementara itu aplikasi yang dihasilkan dengan menggunakan PHP dan memanfaatkan data MySQL sebagai tempat penyimpanan data

Analisis kebutuhan sistem merupakan proses menganalisis serta pemodelan untuk mencari kebutuhan dari seluruh sistem yang sesuai dengan antar muka yang akan dibangun. Aplikasi Sistem Pendukung Keputusan ini menggunakan metode pengembangan *Waterfall model* yang merupakan metodologi untuk pengembangan sistem secara cepat serta memodelkan sistem kedalam UML (*Unified Modelling Language*). Yang terdiri dari *Class Diagram* yang menggambarkan struktur sistem yang mendefinisikan kelas-kelas yang akan dibuat untuk membangun sistem, kemudian *Use Case Diagram* yang mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi, selanjutnya adalah *Activity Diagram* yang

menggambarkan alur kerja dari sebuah sistem yang ada, serta terakhir adalah *Sequence Diagram* yang menggambarkan tujuan dari setiap komponen sistem. Sistem yang digunakan dalam implementasi aplikasi ini terbagi menjadi dua perangkat. Perangkat tersebut yaitu perangkat lunak dan perangkat keras.

Perangkat Keras

Berikut adalah spesifikasi perangkat keras yang dibutuhkan oleh sistem untuk melakukan pembuatan aplikasi Sistem Pendukung Keputusan:

- a. Processor intel core i3
- b. Memori 2 GB
- c. Monitor
- d. Keyboard

Berikut adalah spesifikasi perangkat lunak yang dibutuhkan oleh sistem untuk melakukan pembuatan aplikasi Sistem Pendukung Keputusan:

- a. PHP
- b. Xampp
- c. Crystal Report
- d. Sistem operasi Windows 7

Interface Design perancangan perangkat lunak Sistem Pendukung Keputusan, mendefinisikan *interface-interface* diantara komponen sistem dan spesifikasi *interface* tidak boleh ambigu. Merancang antarmuka merupakan bagian yang paling penting dari merancang sistem. Biasanya hal tersebut juga merupakan bagian yang paling sulit karena dalam merancang antarmuka harus memenuhi tiga persyaratan: sebuah antarmuka harus sederhana (*user friendly*), sebuah antarmuka harus lengkap (*complete interface*), dan sebuah antarmuka harus memiliki kinerja yang cepat (*fast performace*). Alasan utama mengapa antarmuka sulit untuk dirancang adalah karena setiap antarmuka adalah sebuah bahasa pemrograman yang kecil antarmuka menjelaskan sekumpulan objek-objek dan operasi-operasi yang bisa digunakan untuk memanipulasi objek.

Pada gambar 1 Merupakan perancangan interface pada login, dari terdapat dua *textbox* untuk mengisi username dan password admin. Dan satu *button* login, untuk memproses hasil inputan dari username dan password.

Sistem Pendukung Keputusan Untuk Menentukan Paket Wedding menggunakan Metode Electre	
Home Electre Analisa Login	
Login	
Username	<input type="text" value="admin"/>
Password	<input type="password" value="•••"/>
	<input type="button" value="Login"/>
Design By. Endah STMIK Pontianak	
Kontak About	

Gambar.1 Desain form login

Pada gambar 1 merupakan desain form login, terdapat dua *textbox* untuk mengisi username dan password admin. Ketika username dan password sudah diisi klik login untuk memproses hasil inputan kemudian akan masuk ke form utama.

Perancangan struktur menu ini digunakan untuk mempermudah pengguna dalam mengoperasikan aplikasi yang dibangun atau ataupun dikembangkan. Dengan adanya struktur

menu sistem tentunya akan membantu pengguna dalam menjalankan aplikasi sesuai dengan menunya. Adapun rancangan struktur menu sistem yang peneliti rancang adanya sebagai berikut:

Gambar 2 Desain form utama

Form utama adalah form yang muncul setelah memilih form login berhasil form utama terdiri dari beberapa menu, diantaranya Home, Electre, Analisis, Login.

ID Alternatif	Nama Alternatif Produk	Deskripsi	Add
1	Paket Wedding 1	- Biaya Rp. 100.000.000	Edit Del
2	Paket Wedding 2	- Biaya Rp. 150.000.000	Edit Del
3	Paket Wedding 3	- Biaya > Rp. 150.000.000	Edit Del

Gambar 3. Desain Form Alternatif/Produk

Merupakan form yang didesain untuk menginputkan daftar paket wedding terdapat menu tambah, ubah dan hapus data seperti penginputan nama Alternatif Produk, ID Alternatif, Deskripsi.

ID Kriteria	Nama Kriteria	Bobot	Add
1	Wedding Organizer	5	Edit Del
2	Decoration	3	Edit Del
3	Wedding Photography	4	Edit Del
4	Wedding Videography	4	Edit Del
5	Catering	2	Edit Del
6	Wedding Card	8	Edit Del
7	Photobooth	7	Edit Del

Gambar 4 Desain form kriteria paket wedding

Desain form kriteria yang terdapat dalam paket wedding yang didesain untuk melakukan proses penginputan kriteria terdapat menu tambah data seperti nama kriteria, id

kriteria, nilai bobot. Kemudian dari data yang sudah diinputkan bisa diubah dan dihapus setelah penginputan selesai. terdapat dua menu button yaitu simpan data dan batal.

SPK Metode Electre

Home | Alternatif | Kriteria | Alternatif Kriteria | Ganti Password | Logout | Anda Login Sebagai : admin

Data Kriteria

ID Alternatif Kriteria	Nama Alternatif	Nama Kriteria	Nilai	Add
16	Paket Wedding 1	Wedding Organizer	7	Edit Del
17	Paket Wedding 1	Decoration	8	Edit Del
3	Paket Wedding 1	Wedding Photography	5	Edit Del
2	Paket Wedding 1	Wedding Videography	8	Edit Del
5	Paket Wedding 1	Catering	3	Edit Del
6	Paket Wedding 2	Wedding Organizer	3	Edit Del
7	Paket Wedding 2	Decoration	3	Edit Del
8	Paket Wedding 2	Wedding Photography	4	Edit Del
9	Paket Wedding 2	Wedding Videography	2	Edit Del
10	Paket Wedding 2	Catering	3	Edit Del
11	Paket Wedding 3	Wedding Organizer	5	Edit Del
12	Paket Wedding 3	Decoration	4	Edit Del
13	Paket Wedding 3	Wedding Photography	2	Edit Del
14	Paket Wedding 3	Wedding Videography	2	Edit Del
15	Paket Wedding 3	Catering	2	Edit Del

Gambar 5 Desain form Alternatif kriteria

Pada desain form Alternatif kriteria merupakan form yang didesain untuk melakukan proses penginputan pilihan paket wedding beserta kriteria . kemudian dari data yang sudah diinputkan bisa diubah dan dihapus setelah penginputan selesai terdapat dua menu button yaitu simpan data dan batal.

Langkah-langkah seleksi (perhitungan) pada metode ELECTRE adalah sebagai berikut:

1. Normalisasi matrik keputusan

$$r_{25} = \frac{x_{25}}{\sqrt{\sum_i^m = 1x_{i5}^2}} = \frac{3}{\sqrt{3^2 + 3^2 + 2^2}} = \frac{3}{4,690} = 0,396$$

$$r_{31} = \frac{x_{31}}{\sqrt{\sum_i^m = 1x_{i1}^2}} = \frac{5}{\sqrt{4^2 + 3^2 + 5^2}} = \frac{5}{7,071} = 0,7071$$

$$r_{32} = \frac{x_{32}}{\sqrt{\sum_i^m = 1x_{i2}^2}} = \frac{4}{\sqrt{4^2 + 3^2 + 4^2}} = \frac{4}{6,403} = 0,6247$$

$$r_{33} = \frac{x_{33}}{\sqrt{\sum_i^m = 1x_{i3}^2}} = \frac{2}{\sqrt{5^2 + 4^2 + 2^2}} = \frac{2}{6,7082} = 0,2981$$

$$r_{34} = \frac{x_{34}}{\sqrt{\sum_i^m = 1x_{i4}^2}} = \frac{2}{\sqrt{3^2 + 3^2 + 2^2}} = \frac{2}{4,690} = 0,4264$$

Dari perhitungan di atas diperoleh hasil :

0,5657	0,6247	0,7454	0,7276	0,6396
0,4243	0,4685	0,5963	0,4851	0,6396
0,7071	0,6247	0,2981	0,4851	0,4264

2. Pembobotan pada matrik yang ternormalisasi

0,5657	0,6247	0,7454	0,7276	0,6396
0,4243	0,4685	0,5963	0,4851	0,6396
0,7071	0,6247	0,2981	0,4851	0,4264
5	0	0	0	0

$$0 \quad 3 \quad 0 \quad 0 \quad 0$$

$$0 \quad 0 \quad 4 \quad 0 \quad 0$$

$$0 \quad 0 \quad 0 \quad 4 \quad 0$$

$$0 \quad 0 \quad 0 \quad 0 \quad 2$$

2,8285	1,8741	2,9816	2,9104	1,2792
2,1215	1,4055	2,3853	1,9404	1,2792
3,5355	1,8751	1,1924	1,9404	0,8528

3. Menentukan himpunan concordance dan discordance indek

a. Concordance

Sebuah Kriteria dalam alternatif termasuk concordance

$$C_{kl} = \{j, V_{kj} \geq V_{klj}\}, \text{ Untuk } j = 1, 2, 3, \dots, n$$

$$C_{12} = \{j, V_{1j} \geq V_{2j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{1, 2, 3, 4, 5\}$$

$$C_{13} = \{j, V_{1j} \geq V_{3j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{2, 3, 4, 5\}$$

$$C_{13} = \{j, V_{1j} \geq V_{3j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{2, 3, 4, 5\}$$

$$C_{21} = \{j, V_{2j} \geq V_{1j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{5\}$$

$$C_{23} = \{j, V_{2j} \geq V_{3j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{3, 4, 5\}$$

$$C_{31} = \{j, V_{3j} \geq V_{1j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{1, 2\}$$

$$C_{32} = \{j, V_{3j} \geq V_{2j}\}, \text{ Untuk } j = 1, 2, 3, \dots, 5 \\ = \{1, 2, 4\}$$

b. Discordance

Sebuah kriteria dalam suatu alternatif termasuk discordance jika :

$$D_{kl} = \{j, V_{kl} \geq V_{lj}\}, \text{ Untuk } j = 1, 2, 3, \dots, n$$

$$D_{12} = \{j, V_{1j} \geq V_{2j}\}, \text{ Untuk } j = 1, 2, \dots, 5 \\ = \{ \}$$

$$D_{13} = \{j, V_{1j} \geq V_{3j}\}, \text{ Untuk } j = 1, 2, \dots, 5 \\ = \{1\}$$

$$D_{21} = \{j, V_{2j} \geq V_{1j}\}, \text{ Untuk } j = 1, 2, \dots, 5 \\ = \{1, 2, 3, 4\}$$

$$D_{23} = \{j, V_{2j} \geq V_{3j}\}, \text{Untuk } j = 1, 2, \dots, 5 \\ = \{1, 2\}$$

$$D_{31} = \{j, V_{3j} \geq V_{1j}\}, \text{Untuk } j = 1, 2, \dots, 5 \\ = \{3, 4, 5\}$$

$$D_{23} = \{j, V_{3j} \geq V_{2j}\}, \text{Untuk } j = 1, 2, \dots, 5 \\ = \{3, 5\}$$

4. Menghitung matriks concordance dan discordance

$$C_{kl} = \sum_{j \in C_{kl}} w_j$$

$$C_{12} = w_1 + w_2 + w_3 + w_4 + w_5 \\ = 5 + 3 + 4 + 4 + 2 = 18$$

$$C_{13} = w_2 + w_3 + w_4 + w_5 \\ = 3 + 4 + 4 + 2 = 13$$

$$C_{21} = w_5 \\ = 2$$

$$C_{23} = w_3 + w_4 + w_5 \\ = 4 + 4 + 2 = 10$$

$$C_{31} = w_1 + w_2 \\ = 5 + 3 = 8$$

$$C_{32} = w_1 + w_2 + w_4 \\ = 5 + 3 + 4 = 12$$

Jadi Matriks *Concordance* adalah :

$$\Gamma \begin{matrix} - & 18 & 13 & \uparrow \\ & 2 & - & 10 \end{matrix}$$

$$L \begin{matrix} 8 & 12 & - & \uparrow \end{matrix}$$

B . Menghitung Matriks discordance

$$D_{kl} = \frac{\max\{[V_{kj} - V_{lj}]\}}{\max\{[V_{kj} - V_{lj}]\} \vee j} j \in D_{kl}$$

$$= \frac{\max\{0\}}{\max\{[2,8285 - 2,1215]; [1,8741 - 1,4055]; [2,9816 - 2,3852]; [2,9104 - 1,9404]; [1,2792 - 1,2792]\}} \\ = \frac{\max\{0\}}{\max\{0,7070; 0,4686; 0,5964; 0,9700; 0\}}$$

$$= \frac{0}{0,9700}$$

$$= 0$$

$$D_{13} = \frac{\max\{[V_{1j} - V_{3j}]\}}{\max\{[V_{1j} - V_{3j}]\} \vee j} j \in D_{13}$$

$$= \frac{\max\{[2,8285 - 3,5355]\}}{\max\{[2,8285 - 3,5355]; [1,8741 - 1,8741]; [2,9816 - 1,1924]; [2,1904 - 1,9404]; [1,2792 - 0,8528]\}} \\ = \frac{\max\{0,7070\}}{\max\{0,7070\}}$$

$$= \frac{\max\{0,7070; 0; 1,7892; 0,9701; 0,4264\}}{0,7070}$$

$$= \frac{0,7070}{1,7893}$$

$$= 0,3951$$

$$D_{21} = \frac{\max\{[V_{2j} - V_{1j}]\}}{\max\{[V_{2j} - V_{1j}]\} \vee j} j \in D_{21}$$

$$= \frac{\max\{[2,1215 - 2,8285]; [1,4055 - 1,8741]; [2,3852 - 2,9816]; [2,9104 - 1,9404]\}}{\max\{[2,1215 - 2,8285]; [1,4055 - 1,8741]; [2,3852 - 2,9816]; [2,9104 - 1,9404]; [1,2792 - 1,2792]\}}$$

$$\begin{aligned}
 & \frac{\max \{0,7070; 0,4648; 0,5964; 0,9700\}}{\max \{0,7070; 0; 0,4648; 0,5964; 0,9700; 0\}} \\
 &= \frac{0,9700}{0,9700} \\
 &= 1 \\
 D_{23} &= \frac{\max\{[V_{2j} - V_{3j}]\}}{\max\{[V_{2j} - V_{3j}]\} \forall j \in D_{23}} \\
 &= \frac{\max \{[2,1215-3,5355];[1,4055-1,8741]\}}{\max \{[2,1215-3,5355];[1,4055-1,8741]\}} \\
 & \quad \frac{\max \{[0,7070; 0,4686; 0,5964; 0,9700]\}}{\max \{0,7070; 0,4686; 0,5964; 0,9700; 0\}} \\
 &= 1 \\
 D_{31} &= \frac{\max\{[V_{3j} - V_{1j}]\}}{\max\{[V_{3j} - V_{1j}]\} \forall j \in D_{31}} \\
 &= \frac{\max \{[1,1924-2,9816];[1,9404-2,9104];[0,8528-1,2792]\}}{\max \{[3,5355-2,8285];[1,4055-1,8741];[1,1924-2,9816];[1,9404-2,9104];[0,8528-1,2792]\}} \\
 &= \frac{\max \{[1,7892; 0,9700; 0,4264]\}}{\max \{[0,707; 0; 1,7892; 0,9700; 0,4264]\}} \\
 &= \frac{1,7892}{1,7892} \\
 &= 1 \\
 D_{32} &= \frac{\max\{[V_{3j} - V_{2j}]\}}{\max\{[V_{3j} - V_{2j}]\} \forall j \in D_{32}} \\
 &= \frac{\max \{[1,1924-2,3852];[0,5828-1,2792]\}}{\max \{[3,5355-2,1215];[1,8741-1,4055];[1,1924-2,3852];[1,9404-2,9104];[0,8528-1,2792]\}} \\
 &= \frac{\max \{[1,1928; 0,4264]\}}{\max \{[1,4141; 0,4686; 1,1928; 0; 0,4264]\}} \\
 &= \frac{1,1928}{1,1928} \\
 &= 1
 \end{aligned}$$

Jadi matrik discordance adalah : $\begin{vmatrix} - & 0 & 0,3951 \\ 1 & - & 1 \\ 1 & 1 & - \end{vmatrix}$

5. Menentukan matrik dominan concordance dan discordance

a. Menghitung matriks dominan concordance

Nilai threshold © adalah :

$$\begin{aligned}
 \frac{c}{m} &= \frac{\sum_{k=1}^m \sum_{l=1}^m C_{kl}}{M(m-1)} \\
 &= \frac{18+13+2+10+8+12}{3(3-1)} \\
 &= \frac{63}{6} = 10,5
 \end{aligned}$$

Element matriks F ditemukan sebagai berikut :

$$\begin{aligned}
 F_{kl} &= \{1, \text{jika } C_{kl} \geq \frac{c}{m}\} \\
 & \quad \{0, \text{jika } C_{kl} < \frac{c}{m}\}
 \end{aligned}$$

Sehingga matriks dominan concordance adalah :

$$\begin{aligned}
 & \begin{matrix} - & 1 & 1 \\ F = & 0 & 1 & 0 \\ & 0 & 1 & 0 \end{matrix}
 \end{aligned}$$

b. Menghitung matriks dominan discordance

Nilai threshold © adalah :

$$\begin{aligned} \frac{d}{6} &= \frac{\sum_{k=1}^m \sum_{l=1}^m d_{kl}}{m(m-1)} \\ &= \frac{0 + 0,3951 + 1 + 1 + 1 + 1}{3(3-1)} \\ &= \frac{4,9351}{6} = 0,7325 \end{aligned}$$

Dan elemen matriks G ditentukan sebagai berikut :

$$G_{kl} = \begin{cases} 1, & \text{jika } d_{kl} \geq \frac{d}{6} \\ 0, & \text{jika } d_{kl} < \frac{d}{6} \end{cases}$$

Sehingga matriks dominan discordance adalah :

$$G = \begin{matrix} & - & 0 & 0 \\ 1 & - & 1 & \\ 1 & 1 & - & \end{matrix}$$

6. Menentukan aggregate dominance matrix

Rumus umum untuk anggota matriks aggregate dominance adalah :

$$\begin{aligned} e_{kl} &= f_{kl} \times g_{kl} \\ e_{12} &= f_{12} \times g_{12} = 1 \times 0 = 0 \end{aligned}$$

$$\begin{aligned} e_{13} &= f_{13} \times g_{13} = 1 \times 0 = 0 \\ e_{21} &= f_{21} \times g_{21} = 0 \times 1 = 0 \\ e_{23} &= f_{23} \times g_{23} = 0 \times 1 = 0 \\ e_{31} &= f_{31} \times g_{31} = 1 \times 0 = 0 \\ e_{32} &= f_{32} \times g_{32} = 1 \times 1 = 1 \end{aligned}$$

Sehingga matriks aggregate dominan adalah :

$$E = \begin{matrix} & - & 0 & 0 \\ 0 & - & 0 & \\ 0 & 1 & - & \end{matrix}$$

Hasil akhir

Alternatif Produk Terbaik = Paket Wedding yang keluar dengan Jumlah Nilai Terbanyak .

SPK Metode Electre

Home | Electre | Analisa | Login

Analisa Menggunakan SPK Metode Electre

Perhitungan

Ranking	Alternatif	Nilai
1	Paket Weding 2	1
2	Paket Weding 1	0
3	Paket Weding 3	0
4	Paket Wedding 4	0
5	wedding 3	0

Alternatif Produk Terbaik = Paket Weding 2 dengan Jumlah Nilai 1 Terbanyak = 1

© 2014 ContohProgram.com Kontak | About

Gambar 6 Hasil Akhir

Dengan pengujian *black-box*, data yang dimasukkan lalu diproses oleh program aplikasi yang dibuat. Pengujian ini dilakukan agar dapat diketahui apakah fungsi dari program aplikasi menghasilkan *output* yang benar dan sesuai dengan spesifikasi rancangan. Jika pada waktu pengujian program, *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, berarti masih terdapat kesalahan pada program aplikasi tersebut, dan selanjutnya akan dilakukan perbaikan (*debugging*) untuk memperbaiki kesalahan yang terjadi setelah proses pengujian program, proses kerja ini juga dapat disebut dengan *trial and Error*.

Pada pengujian perangkat lunak Spk ini, penulis menggunakan *blackbox* Metode pengujian ini tidak memperhatikan struktur internal atau sifat dari sebuah program atau modul. *blackbox* menggunakan strategi dengan melakukan pengujian data secara menyeluruh.

Dengan pengujian *blackbox*, data yang dimasukkan lalu diproses oleh program aplikasi menghasilkan *output* yang benar dan sesuai dengan spesifikasi rancangan. Jika pada waktu pengujian program, *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, berarti masih terdapat kesalahan pada program aplikasi tersebut, dan selanjutnya akan dilakukan perbaikan (*debugging*) untuk memperbaiki kesalahan yang terjadi setelah proses pengujian program.

4.KESIMPULAN

Sistem Pendukung Keputusan ini dibangun dengan tujuan untuk memberikan rekomendasi tema skripsi yang akan diambil oleh peneliti.

Sistem pendukung keputusan ini dibangun dengan kriteria pemilihan paket wedding yang berbasis web,

Sistem Pendukung Keputusan ini menggunakan metode ELECTRE dimana hasil akhir dari proses pendukung penentuan tema skripsi ini adalah menghasilkan paket wedding sesuai perhitungan metode ELECTRE. Alternatif dengan bobot tertinggillah yang menjadi hasil akhir dalam pemilihan paket wedding ini.

5. SARAN

Dalam pengembangan system selanjutnya disarankan untuk menggunakan kriteria penentuan berbasis yang berbeda dengan yang dipakai sekarang. Misalkan berbasis mobile untuk mengambil keputusan untuk kriteria penentuan. Dengan system yang telah dibangun ini mempunyai sifat dinamis, dan diharapkan pada saat melakukan perubahan kriteria penentuan system ini akan lebih mudah dibandingkan dengan Sistem Pendukung keputusan yang bersifat statis. Berdasarkan hasil pembahasan dan kesimpulan, maka dapat diambil beberapa saran yaitu pengembangan lebih lanjut diperlukan agar bisa disesuaikan dengan kebutuhan Paket Wedding Pontianak serta pengoperasian secara maksimal dari aplikasi Sistem Pendukung Keputusan perlu adanya perangkat keras dan perangkat lunak maksimal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing, orang tua serta sahabat yang telah memberi dukungan finansial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Marlinda, L. (2016). Sistem pendukung keputusan pemilihan tempat wisata yogyakarta menggunakan metode ELimination Et Choix Traduisant La RealitA (ELECTRE). *Prosiding Semnastek*.

- [2] Kosasi, S., & David, D. (2013). Sistem Penunjang Keputusan Pembelian Mobil Menggunakan Adaptive Neuro Fuzzy Inference System. *Prosiding SeNAIK*, 74-80.
- [3] Setiawan, F., Indriani, F., & Aziz, M. (2016). Implementasi Metode Electre Pada Sistem Pendukung Keputusan SNMPTN Jalur Undangan. *Klik-Kumpulan Jurnal Ilmu Komputer*, 2(2), 197-120.
- [4] Setiwati, Hartati, Maulana melakukan penelitian tentang “Sistem Pendukung Keputusan Pembelian Barang Menggunakan Metode ELECTRE”. Dalam melakukan pembelian barang, ada beberapa kriteria yang digunakan oleh Denco Variasi.
- [5] Sundari, S., Wanto, A., & Gunawan, I. (2017). Sistem Pendukung Keputusan Dengan Menggunakan Metode Electre Dalam Merekomendasikan Dosen Berprestasi Bidang Ilmu Komputer (Study Kasus di AMIK & STIKOM Tunas Bangsa).

Perancangan Perangkat Lunak Enkripsi SMS Menggunakan Algoritma RC6 Dan Rijndael Pada Smartphone

SMS Encryption Software Design Using RC6 and Rijndael Algorithms on Smartphones

Benny Djoede Kristianto, Gat, Gusti Syarifudin

STMIK Pontianak; Jln. Merdeka Barat No. 372 Pontianak, (0561)73555/(0561)737777
Jurusan Teknik Informatika, STMIK Pontianak

e-mail: bennydjoede87@gmail.com, gutsy1802@gmail.com, gus_wet@yahoo.com

Abstrak

Proses pengiriman pesan teks yang dilakukan pada perangkat smartphone pada dasarnya pengiriman pesan teks tersebut tanpa ada melakukan pengamanan terhadap isi pesan yang dikirim, sehingga ketika dilakukan penyadapan terhadap alur pengirimannya maka pesan teks yang disadap dapat langsung dibaca oleh penyadap. Untuk itu dibutuhkan perangkat lunak sebagai penunjang metode penyandian tertentu sehingga pesan terkirim tersebut menjadi lebih aman. Penelitian ini menggunakan bentuk penelitian studi literature dengan metode penelitian menggunakan Research & Development (R&D). Adapun teknik pengumpulan data menggunakan studi dokumentasi dan observasi untuk memperoleh teori algoritma rc6 dan algoritma rijndael. Sedangkan metode perancangan perangkat lunak menggunakan Rapid Application Development (RAD). Perancangan perangkat lunak menggunakan bahasa pemrograman java . Hasil perancangan ini menghasilkan sebuah perangkat lunak yang diberi nama "Enkripsi Sms" . Perangkat lunak enkripsi sms menggunakan algoritma rc6 dan rijndael telah dijalankan dan sesuai dengan yang diharapkan pengguna. Dengan adanya perangkat lunak ini, kerahasiaan dan keaslian informasi berupa pesan teks akan lebih terjaga.

Kata Kunci : Enkripsi Sms, Algoritma RC6, Algoritma Rijndael, Java, RAD.

Abstract

Sending text messages is done on a smartphone device is basically the text message delivery without doing a safeguard against the content of the messages sent, so that when it is done tapping against the Groove pengirimannya then intercepted text messages can be directly read by the tappers. For that it needs the software as a specific encoding method support so that the message sent is secure. This research uses the form of research studies, literature research method using Research & Development (R & D). As for the technique of data collection using the study documentation and observation to acquire the theory and algorithm rijndael rc6 algorithm. While the method of software design using Rapid Application Development (RAD). The design of software using the java programming language. The results of this design generates a software named "Sms Encryption". Encryption software sms using the rc6 algorithm and rijndael has been executed and in accordance with the expected user. With the

software, the confidentiality and authenticity of the information in the form of a text message will be more awake.

Keywords: Sms Encryption, RC6 Algorithms, Rijndael Algorithm, RC6, Java, RAD.

1. PENDAHULUAN

Perkembangan teknologi telekomunikasi yang ada pada saat ini mampu menciptakan berbagai macam perangkat keras yang dapat digunakan untuk mengirim atau menerima informasi dengan cepat dan mudah. Penggunaan *smartphone* sebagai device akses informasi telah berkembang pesat pada era ini. Terlebih lagi, banyak fitur-fitur aplikasi yang disediakan oleh android sebagai system operasi ponsel. Dari sekian banyak fitur yang dimiliki oleh android, salah satunya yang masih banyak digunakan yaitu SMS. Namun, seringkali pengguna kurang memperhatikan system keamanan data yang ada pada ponsel tersebut khususnya keamanan informasi yang ada di pesan singkat[1].

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang hanya boleh diketahui oleh pihak yang berhak saja. Layanan SMS yang menggunakan aplikasi SMS bawaan ponsel bukan merupakan jalur yang aman dalam pertukaran informasi. Pesan yang dikirim menggunakan aplikasi SMS bawaan ponsel masih berupa teks terbuka yang belum terproteksi selain itu pengiriman SMS yang dilakukan tidak sampai secara langsung ke penerima, akan tetapi pengiriman SMS harus melewati *Short Message Service Center (SMSC)* yang berfungsi mencatat komunikasi yang terjadi antara pengirim dan penerima. Tersimpannya SMS pada SMSC, maka seorang operator mungkin pihak lain tidak berhak mengetahui informasi tersebut dapat memperoleh informasi atau membaca SMS didalam SMSC. Tersimpannya SMS pada SMSC, maka seorang operator mungkin pihak lain tidak berhak mengetahui informasi tersebut dapat memperoleh informasi atau membaca SMS didalam SMSC [2].

Ada beberapa metode yang bisa digunakan dalam keamanan informasi adalah kriptografi. Salah satu ilmu kriptografi yaitu algoritma simetris atau sering disebut algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi. Supaya pertukaran kunci simetrik aman pada jalur public maka dibutuhkan suatu protocol untuk pertukaran kunci. Supaya pertukaran kunci simetrik aman pada jalur public maka dibutuhkan suatu protocol untuk pertukaran kunci [3].

Pada penelitian ini menggunakan dua algoritma yang berbeda yaitu RC6 dan Rijndael yang digunakan dalam melakukan proses enkripsi maupun dekripsi serta ditambahkan informasi karakter yang terdapat pada pesan tersebut dengan format heksadesimal agar informasi lebih terbaca, perancangan *user interface* yang menarik serta penggunaan aplikasinya bisa diterapkan kedalam perangkat lunak *smartphone* dengan android versi 4.4 kitkat, fitur android kitkat yang cukup banyak dan telah diperbaharui dari versi sebelumnya membuat android versi ini banyak digunakan, namun spesifikasi android kitkat dengan android lainnya tidak jauh beda, hanya saja ada beberapa bug atau celah yang telah ditambal dan mendapat berbagai update baru, baik dari segi tampilan android kitkat itu sendiri hingga fitur-fitur dan fungsinya[4].

Penggunaan dua algoritma RC6 dan Rijndael ini dilakukan demi memperkuat keamanan pesan teks, kedua algoritma RC6 dan Rijndael termasuk kedalam algoritma blok cipher. Pada RC6 kode yang sangat pendek merupakan sebuah kemampuan tersendiri dari algoritma ini sangat sepadan apabila diimplementasikan kedalam

lingkungan smart card. Sedangkan pada Rijndael memiliki kemampuan untuk bekerja sangat baik untuk *platform* apapun. Ditambah dengan operasi yang menggunakan *table lookup* dan operasi XOR membuat prosesnya menjadi tidak terlalu rumit[5].

2. METODE PENELITIAN

Dalam penelitian ini penulis menggunakan metode penelitian *Research & Development* (R&D). R&D adalah “Metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut”. Produk tersebut tidak selalu berbentuk benda atau perangkat keras (*hardware*), seperti buku, alat tulis, dan alat pembelajaran lainnya. Akan tetapi dapat pula dalam bentuk perangkat lunak (*software*) seperti program pengolah data, pembelajaran di kelas, perpustakaan atau laboratorium, atau model – model pendidikan pembelajaran pelatihan, bimbingan, evaluasi, manajemen dan lain sebagainya [6].

2.1 Metode Pengumpulan Data

Adapun jenis data yang digunakan penulis dalam penelitian ini yaitu, sebagai berikut:

a. Data primer

merupakan sumber data yang diperoleh secara langsung dari sumber asli atau pihak pertama. Data primer secara khusus dikumpulkan oleh peneliti untuk menjawab pertanyaan riset atau penelitian. Data primer dapat berupa pendapat subjek riset (orang) baik secara individu maupun kelompok, hasil observasi terhadap suatu benda (fisik), kejadian, atau kegiatan, dan hasil pengujian.

b. Data sekunder

Data sekunder adalah data yang tidak didapatkan secara langsung dari objek penelitian, melainkan sumber data yang diperoleh peneliti secara tidak langsung melalui media perantara. Data sekunder pada umumnya berupa bukti, catatan, atau laporan historis yang telah tersusun dalam arsip, baik yang dipublikasikan dan yang tidak dipublikasikan. Data sekunder antaralain disajikan dalam bentuk tabel-tabel, diagram-diagram, atau mengenai topik penelitian.

2.2 Teknik Pengumpulan Data

Teknik pengumpulan data adalah cara-cara yang dilakukan untuk mencari, mengumpulkan dan memperoleh data untuk digunakan dalam melakukan penelitian, baik itu data yang diperoleh dengan survei langsung maupun dengan penggalian informasi. Teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian ini adalah mendapatkan data. Untuk memperoleh data dan informasi dalam penelitian ini, penulis menggunakan teknik pengambilan data sebagai berikut :

a. Studi Dokumentasi

Teknik dokumentasi berupa studi keputusan dan kajian dari buku-buku, jurnal-jurnal pendukung (*hardcopy dan software*), literatur dari internet dan sejumlah dokumen mengenai data variabel yang diperlukan.

b. Observasi

Pada penelitian ini observasi yang dilakukan dengan pengamatan langsung mengumpulkan data mengenai dokumentasi yang mengacu pada instrumen pengamatan yang berisi definisi-definisi dari item-item data. Melakukan kajian letretur yang berkaitan dengan penelitian yang dilakukan, pengumpulan data yang diperoleh dari sumber tertulis seperti: literatur artikel, berbagai websait, dan tulisan ilmiah yang dianggap terkait dan relevan dengan topik penelitian

2.3 Pengembangan Perangkat Lunak RAD

Penulis menggunakan metode perancangan RAD (*Rapid Application Development*) karena proses perkembangan perangkat lunak ini menekankan pada siklus perkembangan yang singkat dan pemanfaatan fungsi yang telah ada sebelumnya. Adapun langkah-langkah yang dilakukan penulis yaitu[7]:

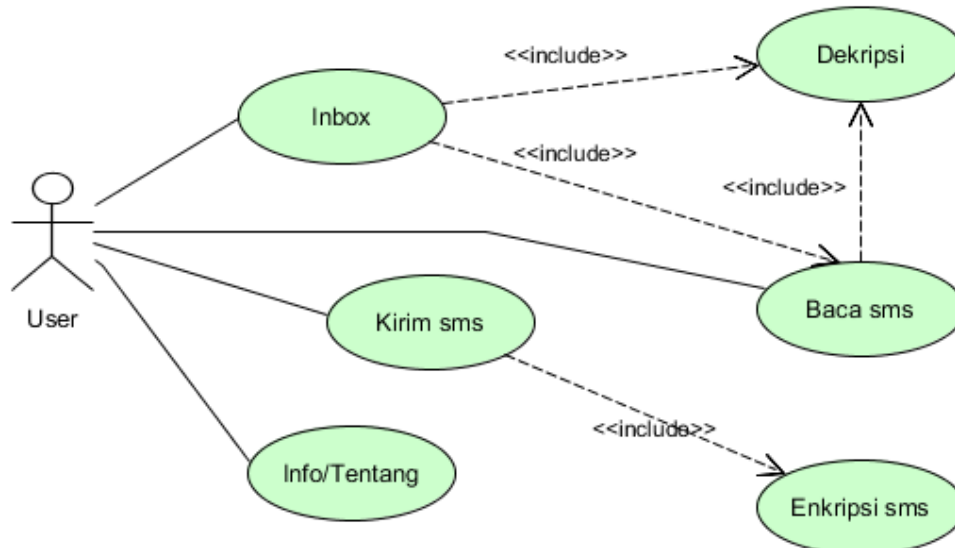
- a. *Business Modeling*. Pada tahap ini, penulis mendaftarkan dan mendefinisikan fungsi-fungsi yang akan dipakai dalam pembuatan aplikasi.
- b. *Data modeling*. Penulis menggunakan informasi yang didapat dalam tahap diatas untuk menentukan banyaknya modul dan form yang akan digunakan dalam program tersebut. Jumlah komponen yang terdapat dalam setiap modul dan form akan ditentukan juga.
- c. *Process Modeling*. Form dan modul yang sudah didefinisikan sebelumnya beserta komponennya disatukan untuk menentukan suatu program untuk. Hubungan antara modul dengan form juga didefinisikan oleh penulis.
- d. *Application Generation*. Penulis membangun aplikasi enkripsi sms pengaman pesan teks dengan menggunakan algoritma RC6 dan Rijndael ini menggunakan program *Eclipse*.
- e. *Testing and turnover*. Setelah modul dirancang ke dalam program tersebut penulis melakukan testing pada form yang membuat modul tersebut. Setelah setiap modul dan form terbentuk dan diuji, semua modul dan form tersebut kemudian disatukan dan dilakukan pengujian kembali akan integritasnya, termasuk didalamnya pengujian validasi input tiap form.

2.4. Unified Modeling Language (UML)

Unified Modeling Language (UML) sebagai media untuk menampilkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak[8].

2.4.1 Use Case Diagram

Use case diagram enkripsi SMS menceritakan tentang user yang menggunakan aplikasi ini secara optional bisa memilih untuk mengakses menu sesuai dengan yang diinginkan.

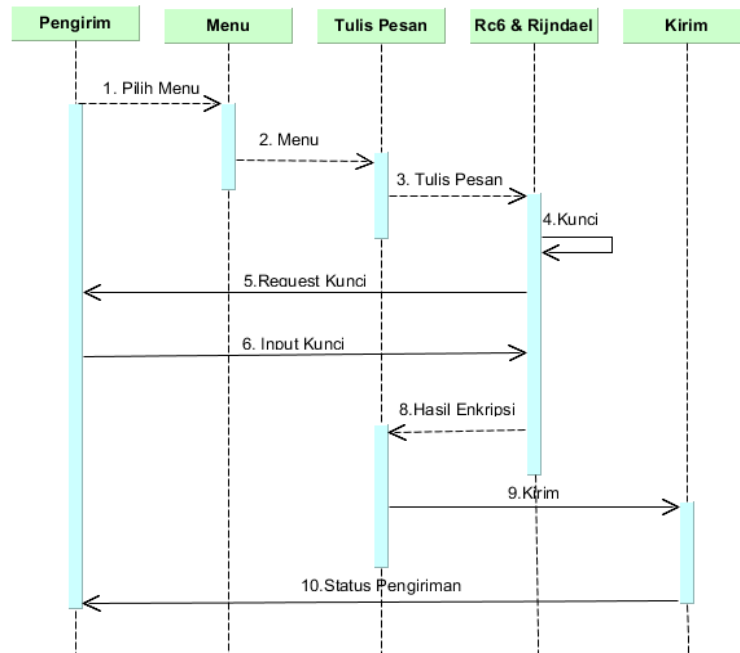


Gambar 1. Use Case Diagram

2.4.2 Sequence Diagram

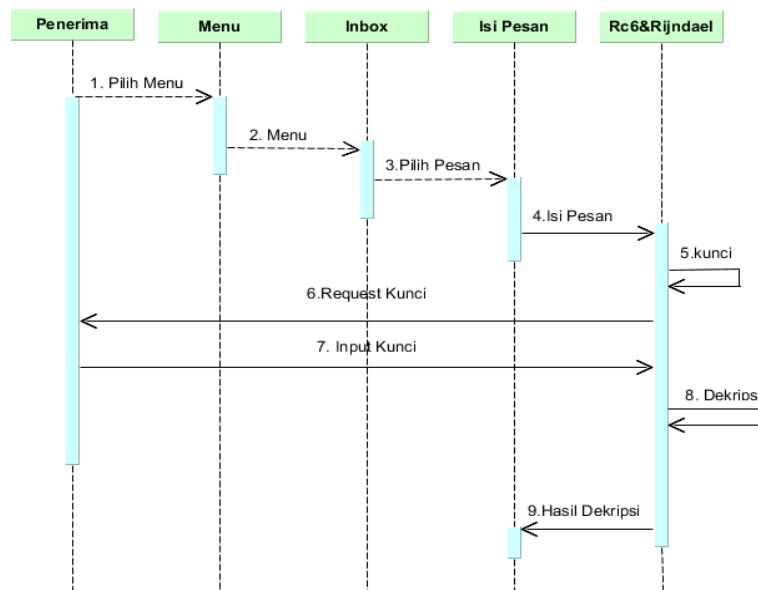
Diagram *Sequence* menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk *user*, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri antar dimensi vertical (waktu) dan dimensi horizontal (objek-objek terkait).

Gambar berikut ini terlihat pengirim memilih menu tulis pesan. Setelah menulis pesan dipilih sistem kemudian memanggil fungsi tulis pesan. Setelah pesan ditulis, proses selanjutnya adalah *user* diminta memasuki kunci enkripsi pada fungsi Rc6 dan Rijndael maka hasil enkripsi akan tampil pada fungsi tulis pesan pengiriman.



Gambar 2. Sequence Diagram Tulis Pesan

Sequence diagram baca sms di bawah ini penerima memilih menu inbox yang berisi pesan, kemudian dari inbox yang berisi pesan dibuka maka sistem meminta untuk *user* memasukan kunci yang sama dengan kunci enkripsi pada fungsi Rc6 dan Rijndael, setelah *user* memasukan kunci maka sistem melakukan proses dekripsi yang kemudian hasil pesan akan tampil pada fungsi isi pesan yang dapat dibaca oleh penerima.



Gambar 3. Sequence Diagram Baca Pesan

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Form Menu

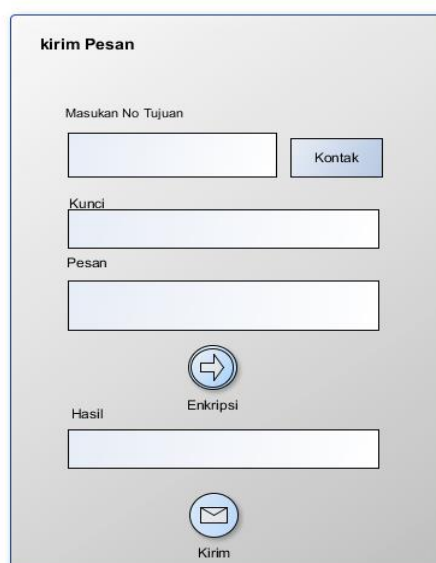
Pada form menu ini terdapat tiga button yang memiliki fungsi masing-masing yaitu, button Tulis pesan berfungsi untuk menuju activity tulis pesan. button kotak masuk berfungsi menuju activity Inbox. Sedangkan button info berfungsi menuju activity info



Gambar 4. Perancangan Form Menu

3.2 Perancangan Form Kirim Pesan

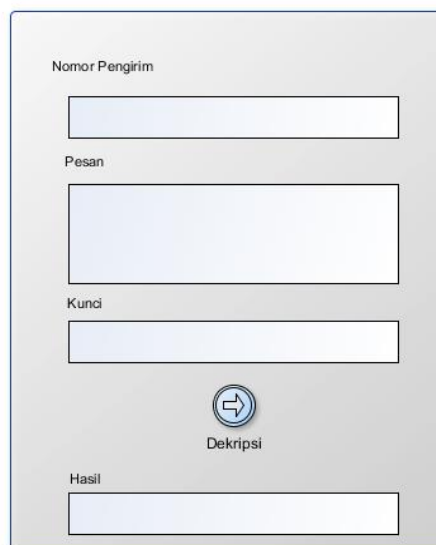
Rancangan interface yang ada pada form kirim pesan terdiri dari beberapa button yang memiliki fungsi melakukan import berdasarkan kontak yang ada pada smartphone yang digunakan, serta melakukan proses pengiriman pesan yang telah di enkripsi.



Gambar 5. Perancangan Form Kirim Pesan

3.3 Perancangan Form Pesan Masuk

Rancangan interface yang ada pada form pesan masuk terdiri dari button dekripsi yang memiliki fungsi melakukan proses membuka pesan yang di enkripsi serta menampilkan hasil dari enkripsi pesan tersebut.



The image shows a wireframe of an incoming message decryption form. It consists of several input fields and a central button. At the top, there is a label 'Nomor Pengirim' followed by a rectangular input box. Below that is a label 'Pesan' followed by a larger rectangular input box. Underneath is a label 'Kunci' followed by another rectangular input box. In the center, there is a circular button with a right-pointing arrow and the text 'Dekripsi' below it. At the bottom, there is a label 'Hasil' followed by a rectangular output box.

Gambar 6. Perancangan Form Pesan Masuk

3.5 Tampilan Menu Utama

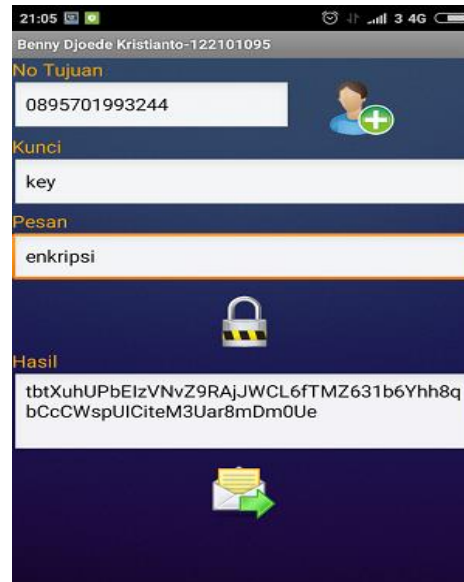
Dari tampilan menu utama dari perangkat lunak Enkripsi SMS ini terdapat beberapa button untuk masuk ke form lain. Menu Utama akan tampil pada saat pertama aplikasi dijalankan. Berikut merupakan tampilan dari form beranda



Gambar 7. Tampilan Menu Utama

3.6 Tampilan Menu Tulis Pesan/Enkripsi

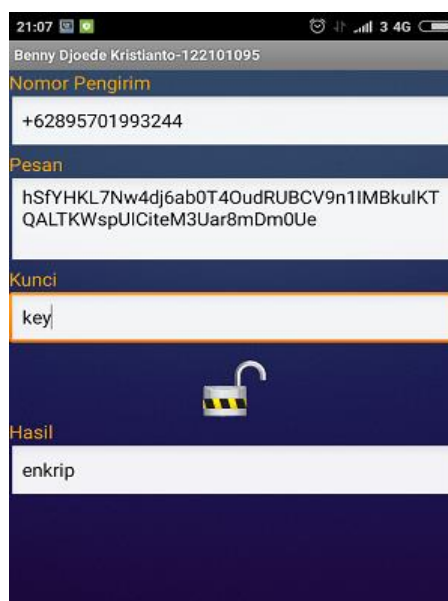
Tampilan menu Tulis Pesan akan muncul, setelah user menekan button Tulis Pesan pada form Menu Utama, form Tulis Pesan berfungsi sebagai form untuk melakukan proses enkripsi sms serta melakukan pengiriman pesan. Berikut merupakan tampilan dari form Tulis Pesan.



Gambar 8. Tampilan Menu Tulis Pesan/Enkripsi

3.7 Tampilan Menu Baca Pesan/Dekripsi

Tampilan menu Baca Pesan akan tampil ketika user menekan button Baca Pesan pada form Menu Utama, form ini berfungsi untuk membaca pesan masuk dan melakukan proses dekripsi pesan. Berikut merupakan tampilan dari form Baca Pesan



Gambar 9. Tampilan Menu Baca Pesan/Dekripsi

3.8 Pengujian Pengujian Keamanan Berdasarkan *Avalanche Effect*

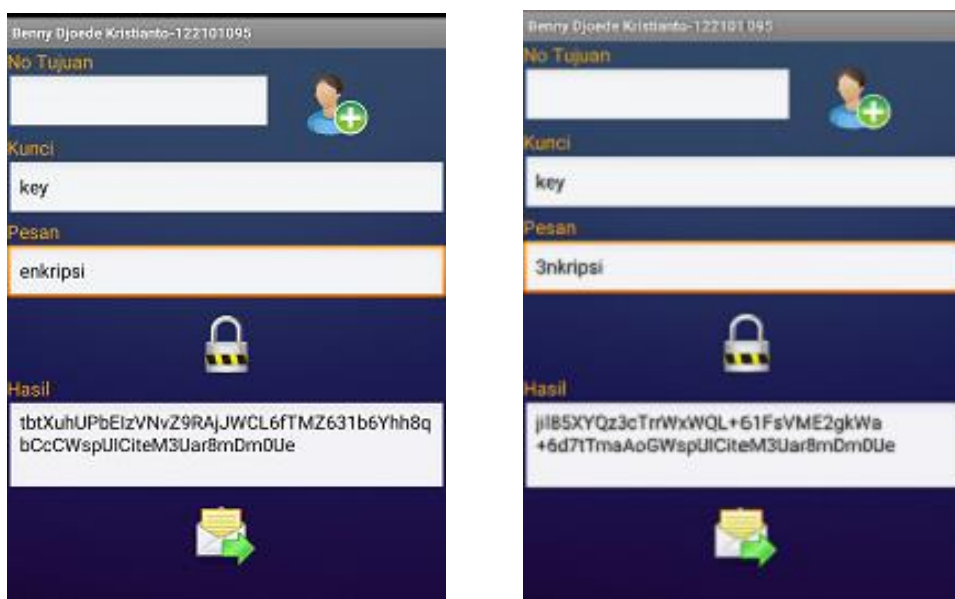
Salah satu karakteristik untuk menentukan baik atau tidaknya suatu algoritma kriptografi adalah dengan melihat avalanche effect-nya. Perubahan yang kecil pada *plaintext* maupun *key* akan menyebabkan perubahan yang signifikan terhadap cipherteks yang dihasilkan. Atau dengan kata lain, perubahan satu bit pada *plaintext* maupun *key* akan menghasilkan perubahan banyak bit pada cipherteks.

Suatu avalanche effect dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45-60% (sekitar separuhnya, 50 % adalah hasil yang sangat baik). Hal ini dikarenakan perubahan tersebut berarti membuat perbedaan yang cukup sulit untuk kriptanalis melakukan serangan. Avalanche effect dihitung dengan rumus :

$$\text{Avalanche effect} = \frac{\text{Jumlah bit yang berubah}(\text{Chipertext})}{\text{Jumlah Bit}(\text{Chipertext})} \times 100\%$$

Pengujian dilakukan dengan evaluasi performa ke-1:

Pada langkah pertama dilakukan pengujian dengan menguji perubahan 1 bit pada *plaintext* yang digunakan adalah “enkripsi” dan “3nkripsi” dengan kunci sama yaitu “key” seperti tampak pada Gambar 3.5 berikut ini :



Gamara 10. Enkripsi Pengujian Evaluasi Pertama

Plaintext 1 : enkripsi
 Plaintext 2 : 3nkripsi
 Kunci : key

Ciphertext 1 :

tbtXuhUPbEIzVNvZ9RAjJWCL6fTMZ631b6Yhh8qbCcCWspUICiteM3Uar8mDm0Ue

Dalam biner adalah sebagai berikut :

1110100 1100010 1110100 1011000 1110101 1101000 1010101 1010000 100010 1000101
 1101100 1111010 1010110 1001110 1110110 1011010 0111001 1010010 1100001 1101010

1001010 1010111 1000011 1001100 0110110 1110010 1010100 1001101 1011010 0110110
0110011 0110001 1000010 0110110 1011001 1101000 1101000 0111000 1010001 1000010
1000011 1100011 1000011 1010111 1110011 1110000 1010101 1101100 1000011 1101001
1110100 1100101 1001101 0110011 1010101 1100001 1110010 0111000 1101101 1100100
1101101 0110000 1010101 1100101

Ciphertext 2 :

jil85XYQz3cTrrWxWQL+61FsVME2gkWa+6d7tTmaAoGWspUICiteM3Uar8mDm0Ue.

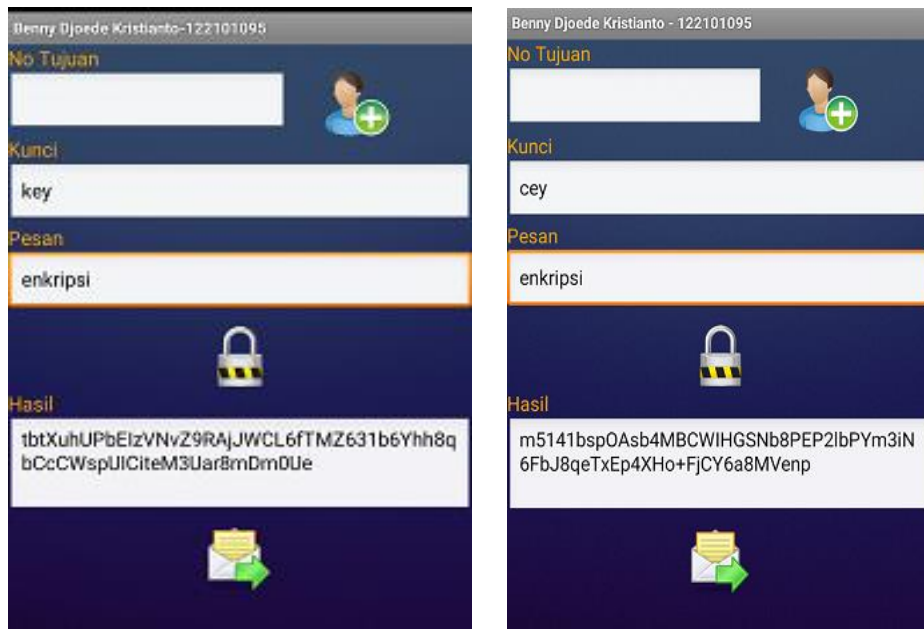
Dalam biner dalah sebagai berikut :

1101010 1101001 1101100 0111000 0110101 1011000 1011001 1010001 1111010 0110011
1100011 1010100 1110010 1110010 1010111 1111000 1010111 1010001 1001100
01010110110110 0110001 1110010 1110011 1010110 1001101 1000101 0110010 1100111
1101011 1010111 1100001 0101011 0110110 1100100 0110111 1110100 1010100 1101101
1100001 1000001 1101111 1000111 1010111 1110011 1110000 1010101 1101001 1000011
1101001 1110100 1100101 1001101 0110011 1010101 1100001 1110010 0111000 1101101
1100100 1101101 0110000 1010101 1100101

$$140/448 * 100\% = 31.25 \%$$

Dari hasil penujian diatas tampak bahwa perbedaan kunci satu bit pada plaintext menghasilkan perbedaan bit sebesar 140 dari total 448 bit atau sekitar 31.25 %.
Pengujian dilakukan dengan evaluasi performa ke-2:

Pada langkah Kedua dilakukan pengujian dengan menguji perubahan 1 bit pada kunci pesan yang digunakan adalah plaintext “enkripsi” dengan kunci berbeda yaitu “key” dan “cey” seperti tampak pada Gambar 3.6 berikut ini.



Gambar 11. Enkripsi Pengujian Evaluasi ke Dua

Plaintext 1 : enkripsi
Kunci 1: key
Plaintext 2 : enkripsi

Kunci 2: cey

Ciphertext 1 :

tbtXuhUPbEIzVNvZ9RAjJWCL6fTMZ631b6Yhh8qbCcCWspUICiteM3Uar8mDm0Ue

Dalam biner adalah sebagai berikut:

```
1110100 1100010 1110100 1011000 1110101 1101000 1010101 1010000 100010 1000101
1101100 1111010 1010110 1001110 1110110 1011010 0111001 1010010 1100001 1101010
1001010 1010111 1000011 1001100 0110110 1110010 1010100 1001101 1011010 0110110
0110011 0110001 1000010 0110110 1011001 1101000 1101000 0111000 1010001 1000010
1000011 1100011 1000011 1010111 1110011 1110000 1010101 1101100 1000011 1101001
1110100 1100101 1001101 0110011 1010101 1100001 1110010 0111000 1101101 1100100
1101101 0110000 1010101 1100101
```

Ciphertext 2 :

m5141bspOAsb4MBCWIHGSNb8PEP2lbPYm3iN6FbJ8qeTxEp4XH0+FjCY6a8MVenp

Dalam biner adalah sebagai berikut :

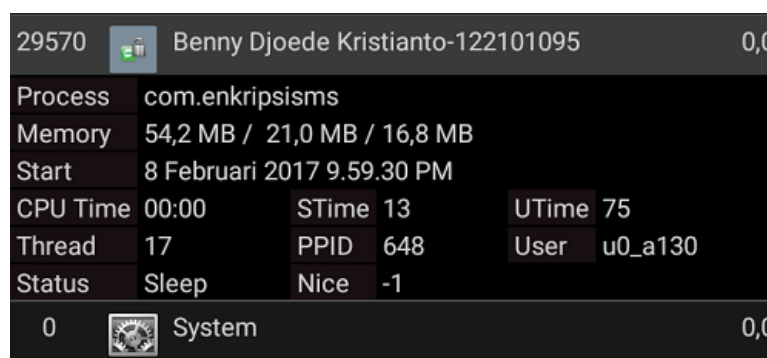
```
1101101 0110101 0110001 0110100 0110001 1100010 1110011 1110000 1001111 1000001
1110011 1100010 0110100 1001101 1000010 1000011 1010111 1010111 1001000 1000111
1010011 1001110 1100010 0111000 1010000 1000101 1010000 0110010 1101100 1100010
1010000 1011001 1101101 0110011 1101001 1001110 0110110 1000110 1100010 1001010
0111000 1010001 1100101 1010100 1111000 1000101 1010000 0110100 1011000 1001000
1001111 0101011 1000110 1101010 1000011 1011001 0110110 1100001 0111000 1001101
1010110 1100101 1101110 1110000
```

$$232/448 * 100\% = 51.78\%$$

Dari hasil penujian diatas tampak bahwa perbedaan kunci satu bit pada kunci menghasilkan perbedaan bit sebesar 232 dari total 448 bit atau sekitar 51.78 %.

3.6 Pengujian *Heap Memory*

Heap memory merupakan porsi dari *memory* yang dialokasikan secara dinamis. Untuk menyediakan pengalaman user yang stabil, penting untuk aplikasi tidak begitu banyak mengambil porsi *memory* [7].



Gambar 13. Pengujian *Heap Memory* dengan *ADV*

4 KESIMPULAN

Setelah melalui proses penyelesaian skripsi yang berjudul “Perancangan dan Pengujian Perangkat Lunak Enkripsi SMS Menggunakan Algoritma RC6 Dan Rijndael Pada Smartphone Android”, penulis menarik kesimpulan sebagai berikut :

- a. Penggunaan kedua algoritma RC6 dan Rijndael sebagai algoritma enkripsi memperlihatkan sebuah avalanche effect yang baik. Hasil yang ditunjukkan ini sesuai dengan parameter yang ditetapkan yaitu 50% dari besar blok penyandian.
- b. Hasil enkripsi selalu sama dengan dekripsi walaupun teks diinput dengan berbagai kombinasi karakter (huruf, angka, simbol).
- c. Koneksi jaringan dibutuhkan pada perangkat lunak ini untuk dapat menjalankan fungsi pengiriman pesan yang sudah di enkripsi.
- d. Penerapan algoritma kunci privat untuk enkripsi SMS pada smartphone dapat meningkatkan keamanan. Pesan yang terenkripsi tidak akan dapat dibaca jika tidak didekripsi dengan menggunakan kunci yang benar.

5 SARAN

Berdasarkan hasil pembahasan dan kesimpulan, maka dapat diambil beberapa saran sebagai berikut:

- a. Meningkatkan nilai *avalanche effect* bisa dikembangkan lagi dengan menggunakan algoritma yang berbeda sehingga diperoleh nilai *avalanche effect* yang lebih tinggi lagi.
- b. Perangkat lunak saat ini hanya mampu mengirimkan sms dengan sim *default* saja belum bisa digunakan untuk mengirimkan sms dengan menggunakan sim dua (pada *smartphone* dengan dua sim card)
- c. Perangkat lunak belum dapat mengirimkan langsung kunci enkripsi dan kunci dekripsinya bersama pesan teksnya, diharapkan pada pengembangan selanjutnya perangkat dapat mengirimkan kunci bersamaan dengan pesan teksnya.
- d. Dalam implementasi algoritma Rijndael ini hanya dalam cakupan kecil dan sangat mendasar dengan menggunakan bahasa pemrograman Java. Penyusun berharap agar kedepan dapat dikembangkan.

6 DAFTAR PUSTAKA

- [1] Satriawan, I. W. D., Sasmita, I. G. M. A., & Bayupati, I. P. A. (2014). Aplikasi Enkripsi SMS dengan Metode RSA pada Smartphone Berbasis Android. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*.
- [2] Defni, D., & Rahmayuni, I. (2014). Enkripsi SMS (Short Message Service) pada telepon selular berbasis Android dengan metode RC6. *Jurnal Momentum*, 16(1).
- [3] Ariyus, D. (2008). *Pengantar ilmu kriptografi: teori analisis & implementasi*. Penerbit Andi.
- [4] Irawan, R., & Ilhamsyah, Y. B. Aplikasi Enkripsi Dan Dekripsi Pesan Singkat Menggunakan Algoritma Knapsack Berbasis Android. *Coding Jurnal Komputer dan Aplikasi*, 3(3).
- [5] Azhar, R., & Kurniawan, K. (2016). Aplikasi Keamanan Sms Menggunakan Algoritma Rijndael. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 16(1), 105-112.

Pedoman Penulisan Makalah SISFOTENIKA

1. Topik yang akan dipublikasikan oleh Jurnal – Jurnal SISFOTENIKA berhubungan dengan teknologi informasi, komunikasi dan komputer yang berbentuk kumpulan/akumulasi pengetahuan baru, pengamatan empirik atau hasil penelitian, dan pengembangan gagasan atau usulan baru
2. Naskah ditulis dalam bahasa Indonesia baku atau bahasa Inggris, belum pernah dipublikasikan serta bebas dari unsur plagiat. Naskah dilengkapi surat pernyataan tertulis dikirim melalui OJS masing – masing Jurnal anggota CORIS (Cooperation Computer Research Inter University).
3. Redaksi berhak menolak naskah yang tidak memenuhi kriteria/persyaratan teknis, mengadakan perubahan susunan naskah, memperbaiki bahasa dan berkonsultasi dengan penulis sebelum naskah dimuat.
4. Naskah diketik dengan komputer menggunakan Microsoft Word, di atas kertas ukuran 21 cm x 29,7 cm (A4), margin atas bawah kanan kiri 3 cm, spasi 1, huruf Times New Roman ukuran font untuk judul artikel adalah 18 point, dan font pada isi makalah 11 point. Naskah ditulis dengan layout 1 kolom.
5. Jumlah halaman berkisar antara 10 sampai 14 halaman, dan jumlah gambar tidak boleh melebihi 30% dari seluruh tulisan
6. Judul makalah maksimal 12 kata dalam bahasa Indonesia atau 10 kata dalam Bahasa Inggris. Judul harus mencerminkan dengan tepat masalah yang dibahas di makalah, dengan menggunakan kata-kata yang ringkas, lugas, tepat, jelas dan mengandung unsur-unsur yang akan dibahas.
7. Nama penulis ditulis di bawah judul sebelum abstrak tanpa disertai gelar akademik atau gelar lain apapun. Instansi penulisa dituliskan Program Studi, Jurusan, Fakultas, dan nama Perguruan Tinggi penulis bernaung dan alamat *email* untuk korespondensi dengan ukuran 11 point bold.
8. Sistematika penulisan naskah, terdiri dari:

- a. Abstrak dan kata kunci

Abstrak ditulis dalam bahasa Indonesia dan bahasa Inggris dengan panjang masing-masing 150 - 200 kata dan dicetak miring dengan Times New Roman 11 point, diketik dengan jarak 1 spasi. Abstrak disusun dengan kalimat-kalimat ringkas, jelas, runtut, sistematis, dapat menggambarkan apa serta mengapa penelitian dikerjakan, bagaimana dikerjakan, dan apa hasil penting yang dicapai dari penelitian.

- b. Pendahuluan

Pendahuluan ditulis dengan Times New Roman 11 point. Pendahuluan menguraikan:

- 1) latar belakang permasalahan yang diselesaikan, dan isu-isu yang terkait dengan masalah yang diselesaikan.
- 2) tinjauan pustaka yang memuat uraian sistematis tentang informasi hasil-hasil penelitian yang pernah dilakukan oleh peneliti-peneliti sebelumnya yang relevan dengan penelitian yang akan dilakukan. Bagian ini memuat kelebihan dan kelemahan yang mungkin ada pada penelitian-penelitian sebelumnya yang dapat dijadikan argumen bahwa penelitian yang akan dikerjakan ini bersifat menyempurnakan atau mengembangkan penelitian terdahulu.
- 3) landasan teori berupa rangkuman teori-teori yang diambil dari pustaka yang mendukung penelitian, serta memuat penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk pemecahan permasalahan. Landasan teori dapat berbentuk uraian kualitatif, model matematis, atau tools yang langsung berkaitan dengan permasalahan yang diteliti.

c. Metode Penelitian

Bagian ini memuat penjelasan secara lengkap dan terinci tentang langkah-langkah yang dilakukan dalam penelitian ini. Selain itu, langkah penelitian juga perlu ditunjukkan dalam bentuk diagram alir langkah penelitian atau framework secara lengkap dan terinci termasuk di dalamnya tercermin algoritma, *rule*, pemodelan-pemodelan, desain, dan lain-lain yang terkait dengan aspek perancangan sistem.

d. Hasil dan Pembahasan

Bagian Hasil dan Pembahasan merupakan bagian yang memuat semua temuan ilmiah yang diperoleh sebagai data hasil penelitian. Bagian ini diharapkan memberikan penjelasan ilmiah yang secara logis dapat menerangkan alasan diperolehnya hasil-hasil tersebut yang dideskripsikan secara jelas, lengkap, terinci, terpadu, sistematis, serta berkesinambungan.

Pemakalah menyusun secara sistematis disertai argumentasi yang rasional tentang informasi ilmiah yang diperoleh dalam penelitian, terutama informasi yang relevan dengan masalah penelitian. Pembahasan terhadap hasil penelitian yang diperoleh dapat disajikan dalam bentuk uraian teoritik, baik secara kualitatif maupun kuantitatif. Dalam pelaksanaannya, bagian ini dapat digunakan untuk memperbandingkan hasil-hasil penelitian yang diperoleh dalam penelitian yang sedang dilakukan terhadap hasil-hasil penelitian yang dilaporkan oleh peneliti terdahulu yang diacu pada penelitian ini. Secara ilmiah, hasil penelitian yang diperoleh dalam penelitian dapat berupa temuan baru atau perbaikan, penegasan, atau penolakan interpretasi suatu fenomena ilmiah dari peneliti sebelumnya

Untuk memperjelas penyajian, hasil penelitian disajikan secara cermat agar mudah dipahami, misalnya dapat ditunjukkan dalam bentuk tabel, kurva, grafik, gambar, foto, atau bentuk lainnya sesuai keperluan secara lengkap dan jelas. Perlu diusahakan agar saat membaca hasil penelitian dalam format tersebut, pembaca tidak perlu mencari informasi terkait dari uraian dalam pembahasan. Akhir dari bagian ini memuat keterangan tentang kelebihan dan kelemahan sistem, yang dideskripsikan secara terinci.

Tabel dan gambar harus diberi identitas yang berupa nomor urut dan judul tabel atau gambar yang sesuai dengan isi tabel atau gambar, serta dilengkapi dengan sumber kutipan.

Judul tabel ditulis dalam Times New Roman 11 point, ditempatkan di atas tabel, tanpa diakhiri tanda titik. Tabel tidak boleh dipenggal, kecuali kalau tidak mungkin diketik dalam satu halaman. Pada halaman lanjutan tabel dicantumkan nomor tabel dan ditulis kata Lanjutan tanpa judul. Bagan, grafik, peta, foto, semuanya disebut gambar. Judul gambar dalam Times New Roman 11 point, tepat di bawah gambar, tanpa diakhiri oleh tanda titik. Keterangan gambar dituliskan pada tempat yang kosong pada halaman yang sama. Skala dan satuan pada grafik harus dibuat se jelas mungkin. Setiap tabel dan gambar harus dirujuk dalam makalah.

Persamaan harus diberi nomor urut pada bagian sebelah kanan.

e. Kesimpulan dan Saran

1) Kesimpulan

Saran merupakan pernyataan singkat, jelas, dan tepat tentang apa yang diperoleh, memuat keunggulan dan kelemahan, dapat dibuktikan, serta terkait langsung dengan tujuan penelitian. Uraian pada bagian ini harus merupakan pernyataan yang pernah dianalisis/dibahas pada bagian sebelumnya, bukan pernyataan yang sama sekali baru dan tidak pernah dibahas pada bagian sebelumnya, serta merupakan jawaban atas permasalahan yang dirumuskan. Bagian ini tidak perlu ada uraian penjelasan lagi.

2) Saran

Saran memuat berbagai usulan atau pendapat yang sebaiknya dikaitkan oleh penelitian sejenis. Saran dibuat berdasarkan kelemahan, pengalaman, kesulitan, kesalahan, temuan baru yang belum diteliti dan berbagai kemungkinan arah penelitian selanjutnya.

f. Daftar Pustaka

Pustaka Buku yang digunakan harus maksimal 10 tahun terakhir dari waktu penyusunan artikel dan untuk pustaka Jurnal/Proceeding maksimal 5 tahun terakhir. Setiap penulis wajib merujuk 1 artikel yang telah dipublish oleh anggota CORIS. Adapun anggota CORIS adalah:

No	Nama PTS	Nama Jurnal
1	Univ Potensi Utama Medan	CSRID
2	Univ Klabat Manado	CogITo Smart Journal
3	Univ Dian Nuswantoro Semarang	JAIS dan TECHNO.COM
4	STMIK Pontianak	Sisfotenika
5	STMIK Dipanegara	Jusiti
6	STMIK Tasikmalaya	Voice of Informatics
7	STIKOM Bali	Eksplora Informatika
8	STMIK Raharja	CCIT
9	Univ. AMIKOM Yogyakarta	CITEC Journal

Daftar pustaka disusun menurut urutan kemunculan rujukan. Urutan dimulai dengan penulisan nama penulis, tahun, judul, penerbit, dan kota terbit. Penulisan nama penulis adalah nama keluarga (nama belakang) diikuti nama kecil (nama depan). Untuk kutipan dari internet berisi nama penulis, judul artikel, alamat website, dan tanggal akses. Daftar Pustaka hanya memuat pustaka yang benar-benar diacu dalam makalah ditulis Times New Roman 11 point, dan disusun sbb:

1. Urutan Daftar Pustaka berdasarkan berdasarkan urutan kemunculan rujukan.
2. Daftar Pustaka hanya memuat pustaka yang benar-benar diacu dalam makalah ditulis Times New Roman 11 point, dan disusun sbb:
 - 1) Urutan Daftar Pustaka berdasarkan berdasarkan urutan kemunculan rujukan.
 - 2) Tulisan untuk suatu sumber pustaka diketik satu spasi. Jarak di antara sumber pustaka tetap dua spasi.
 - 3) Sumber referensi dari Internet harus berasal dari artikel ilmiah-resmi.
 - 4) Setiap pustaka ditulis menurut:
 - a) **Buku:** nama pengarang, tahun penerbitan, *judul*, edisi (jika perlu), jilid (jika perlu), nama penerbit, kota penerbit
 - b) **Majalah/Jurnal Ilmiah/Prosiding:** nama penulis, tahun penerbitan, *judul*, nama majalah/jurnal ilmiah/prosiding, edisi (jika perlu), nama penerbit, kota penerbit
 - c) **Laporan Penelitian:** nama peneliti, tahun, judul, jenis penelitian, nama lembaga, kota
 - d) **Internet:** nama penulis, tanggal akses, *judul artikel*, alamat URL secara lengkap. Publikasi di web **selain** e-book, e-journal, dan e-proceeding tidak diperbolehkan untuk dijadikan rujukan penelitian ilmiah.

g. Biodata Penulis

Pada bagian akhir paper memuat biodata penulis yang mencakup nama lengkap, tempat tanggal lahir, alamat koresponden (rumah/kantor dan email), tahun lulus dan bidang ilmu untuk S1, S2 atau S3, spesialisasi dan minat keilmuan serta hal-hal lain yang dianggap perlu dicantumkan.

Judul Naskah Publikasi Maksimum 12 Kata dlm bhs.Ind

(Center, Times New Roman 18, maks 12 kata Bhs. Ind. or 10 words in English)

Tuliskan Judul Dalam Bahasa Inggris Max. 10 kata

(Center, Times New Roman 18, maks 10 words in English)

Penulis pertama*¹, Penulis kedua², Penulis ketiga³

^{1,2,3}Instansi Penulis meliputi Program Studi Jurusan Fakultas Nama Perguruan Tinggi

E-mail: *¹xxxx@xxxx.xxx, ²xxxx@xxxx.xxx, ³xxxx@xxxx.xxx

Abstrak

Abstrak terdiri dari 150-200 kata berbahasa Indonesia dicetak miring dengan Times New Roman 11point. Abstrak harus jelas, deskriptif dan harus memberikan gambaran singkat masalah yang diteliti. Abstrak meliputi alasan pemilihan topik atau pentingnya topik penelitian, metode penelitian dan ringkasan hasil. Abstrak harus diakhiri dengan komentar tentang pentingnya hasil atau kesimpulan singkat.

Kata Kunci—3-5 kata kunci dalam bahasa Indonesia

Abstract

Abstract should contain at least 150 - 200 words, written in English in italics with Times New Roman 11 point. Abstract should be clear, descriptive, and should provide a brief overview of the problem studied. Abstract topics include reasons for the selection or the importance of research topics, research methods and a summary of the results. Abstract should end with a comment about the importance of the results or conclusions brief.

Keywords—3-5 kata kunci dalam bahasa Inggris

1. PENDAHULUAN

Pendahuluan menguraikan latar belakang permasalahan yang diselesaikan, isu-isu yang terkait dengan masalah yang diselesaikan, ulasan penelitan yang pernah dilakukan sebelumnya oleh peneliti lain yang relevan dengan penelitian yang dilakukan.

2. METODE PENELITIAN

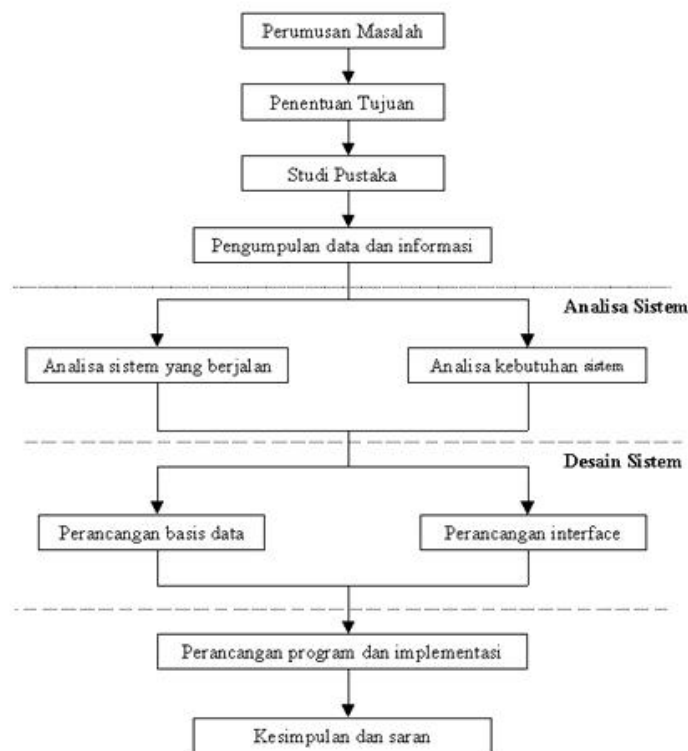
Metode Penelitian (bisa meliputi analisa, arsitektur, metode yang dipakai untuk menyelesaikan masalah, implementasi), dalam bahasan ini penulis bisa menguraikan bagaimana penelitian tersebut akan dilakukan.

2.1. Tahapan Review

Harap mengirimkan naskah anda secara elektronik untuk direview sebagai attachment e-mail. Ketika anda mengirimkan dokumen naskah versi awal dalam format *word.doc* satu kolom, termasuk gambar dan tabel.

2.1.1. Gambar dan Tabel

Semua tabel dan gambar yang Anda masukkan dalam dokumen harus disesuaikan dengan urutan 1 kolom atau ukuran penuh satu kertas, agar memudahkan bagi reviewer untuk mencermati makna gambar. Gambar dan tabel yang dimuat harus dirujuk dan diberikan penjelasannya dalam naskah.



Gambar 1. Alur Penelitian

2.2. Formulir Copyright

Formulir copyright harus disertakan pada pengiriman naskah akhir. Anda bisa meminta versi .pdf, atau .doc via email ke indoceiss@gmail.com

2.2.1. Rumus Matematika

Jika anda menggunakan *Ms. Word*, gunakan persamaan *Microsoft Equation Editor* atau *MathType*, ditulis di tengah, dan diberi nomor persamaan mulai dari (1), (2) dst.

$$p(x,y) = (0 \leq x \leq M - 1, 0 \leq y \leq N - 1) \quad (1)$$

2.2.2. Pengacuan Pustaka

Pengacuan pustaka dilakukan dengan menggunakan penomoran sesuai urutan munculnya pustaka tersebut, misal sitasi buku [1], sitasi jurnal ilmiah [2]. Sitasi kepustakaan harus ada dalam Daftar Pustaka dan Daftar Pustaka harus ada sitasinya dalam naskah. Pustaka yang disitasi pertama kali pada naskah, harus ada pada daftar pustaka nomor satu, pustaka yang disitasi kedua yang muncul dalam naskah muncul sebagai daftar pustaka urutan kedua, berikut seterusnya.

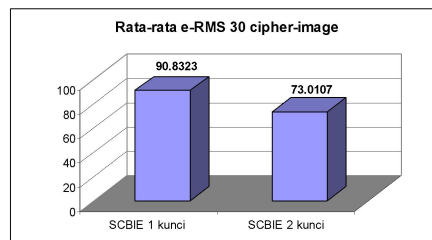
Setiap Penulis wajib merujuk 1 artikel yang telah dipublish jurnal-jurnal anggota CORIS. Berikut daftar anggota CORIS:

No	Nama PTS	Nama Jurnal
1	Univ Potensi Utama Medan	CSRID
2	Univ Klabat Manado	CogITo Smart Journal
3	Univ Dian Nuswantoro Semarang	JAIS dan TECHNO.COM
4	STMIK Pontianak	Sisfotenika
5	STMIK Dipanegara	Jusiti
6	STMIK Tasikmalaya	Voice of Informatics
7	STMIK STIKOM Bali	Eksplora Informatika
8	STMIK Raharja	CCIT
9	Univ. AMIKOM Yogyakarta	CITEC Journal

Pustaka Buku yang digunakan harus maksimal 10 tahun terakhir dari penyusunan artikel dan untuk pustaka Jurnal/Proceeding maksimal 5 tahun terakhir.

3. HASIL DAN PEMBAHASAN

Pembahasan terhadap hasil penelitian dan pengujian yang diperoleh disajikan dalam bentuk uraian teoritik, baik secara kualitatif maupun kuantitatif. Hasil percobaan sebaiknya ditampilkan dalam berupa grafik ataupun tabel. Untuk grafik dapat mengikuti format untuk diagram dan gambar..



Gambar 2. Grafik perbandingan

Tabel 1. Perbandingan Algoritma A dan Algoritma B

Memori	Ketelitian	Waktu Proses	Algoritma
200 KB	98 %	120 ms	A
415 KB	95 %	105 ms	B

4. KESIMPULAN

Kesimpulan harus mengindikasikan secara jelas hasil-hasil yang diperoleh, kelebihan dan kekurangannya, serta kemungkinan pengembangan selanjutnya.

Kesimpulan dapat berupa paragraf, namun sebaiknya berbentuk point-point dengan menggunakan numbering.

5. SARAN

Dalam bahasan ini memuat saran untuk menutup kekurangan penelitian. Tidak memuat saran-saran selain untuk penelitian yang lebih lanjut.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada xxx yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

● **Buku** dengan urutan penulisan: Penulis, tahun, *judul buku* (harus ditulis miring) volume (jika ada), edisi (jika ada), nama penerbit dan kota penerbit.

[1] Castleman, K. R., 2004, *Digital Image Processing*, Vol. 1, Ed.2, Prentice Hall, New Jersey.

● **Buku Terjemahan** dengan urutan penulisan: Penulis asli (nama depan, tengah. (disingkat), belakang. (disingkat)), tahun buku terjemahan, *judul bukuterjemahan* (harus ditulis miring), volume (jika ada), edisi (jika ada), (diterjemahkan oleh: nama penerjemah), nama penerbit terjemahan dan kota penerbit terjemahan.

[2] Gonzales, R., P. 2004, *Digital Image Processing (Pemrosesan Citra Digital)*, Vol. 1, Ed.2, diterjemahkan oleh Handayani, S., Andri Offset, Yogyakarta.

● **Artikel dalam Buku** dengan urutan penulisan: Penulis artikel, tahun, *judul artikel* (harus ditulis miring), nama editor, *judul buku* (harus ditulis miring), volume (jika ada), edisi (jika ada), nama penerbit dan kota penerbit.

[3] Wyatt, J. C, dan Spiegelhalter, D., 1991, *Field Trials of Medical Decision-Aids: Potential Problems and Solutions*, Clayton, P. (ed.): *Proc. 15th Symposium on Computer Applications in Medical Care*, Vol 1, Ed. 2, McGraw Hill Inc, New York.

● **Pustaka dalam bentuk artikel dalam majalah ilmiah:**

Urutan penulisan: Penulis, tahun, judul artikel, *nama majalah* (harus ditulis miring sebagai singkatan resminya), nomor, volume dan halaman.

- [4] Yusoff, M, Rahman, S., A., Mutalib, S., and Mohammed, A., 2006, Diagnosing Application Development for Skin Disease Using Backpropagation Neural Network Technique, *Journal of Information Technology*, vol 18, hal 152-159.

● **Pustaka dalam bentuk artikel dalam seminar ilmiah:**

Artikel dalam prosiding seminar dengan urutan penulisan: Penulis, tahun, judul artikel, *Judul prosiding Seminar* (harus ditulis miring), kota seminar, tanggal seminar.

- [5] Wyatt, J. C, Spiegelhalter, D, 2008, Field Trials of Medical Decision-Aids: Potential Problems and Solutions, *Proceeding of 15th Symposium on Computer Applications in Medical Care*, Washington, May 3.

● **Pustaka dalam bentuk Skripsi/Tesis/Disertasi** dengan urutan penulisan: Penulis, tahun, judul skripsi, *Skripsi/Tesis/Disertasi* (harus ditulis miring), nama fakultas/ program pasca sarjana, universitas, dan kota.

- [6] Prasetya, E., 2006, Case Based Reasoning untuk mengidentifikasi kerusakan bangunan, *Tesis*, Program Pasca Sarjana Ilmu Komputer, Univ. Gadjah Mada, Yogyakarta.

● **Pustaka dalam bentuk Laporan Penelitian:**

Urutan penulisan: Peneliti, tahun, judul laporan penelitian, *nama laporan penelitian* (harus ditulis miring), nama proyek penelitian, nama institusi, dan kota.

- [7] Ivan, A.H., 2005, Desain target optimal, *Laporan Penelitian Hibah Bersaing*, Proyek Multitahun, Dikti, Jakarta.

Pustaka dalam bentuk artikel dalam internet (tidak diperkenankan melakukan sitasi artikel dari internet yang tidak ada nama penulisnya):

● **Artikel majalah ilmiah versi cetakan** dengan urutan penulisan: Penulis, tahun, judul artikel, *nama majalah* (harus ditulis miring sebagai singkatan resminya), nomor, volume dan halaman.

- [8] Wallace, V. P., Bamber, J. C. dan Crawford, D. C. 2000. Classification of reflectance spectra from pigmented skin lesions, a comparison of multivariate discriminate analysis and artificial neural network. *Journal Physical Medical Biology*, No.45, Vol.3, 2859-2871.

● **Artikel majalah ilmiah versi online** dengan urutan penulisan: Penulis, tahun, judul artikel, *nama majalah* ((harus ditulis miring sebagai singkatan resminya), nomor, volume, halaman dan alamat website.

- [9] Xavier Pi-Sunyer, F., Becker, C., Bouchard, R.A., Carleton, G. A., Colditz, W., Dietz, J., Foreyt, R. Garrison, S., Grundy, B. C., 1998, Clinical Guidelines on the identification, evaluation, and treatment of overweight and obesity in adults, *Journal of National Institutes of Health*, No.3, Vol.4, 123-130, http://journals.lww.com/acsm-msse/Abstract/1998/11001/paper_treatment_of_obesity.pdf, diakses tanggal 23 Februari 2016.

● **Artikel umum** dengan urutan penulisan: Penulis, tahun, judul artikel, *alamat website* (harus ditulis miring), diakses tanggal ...

- [10] Borglet, C, 2003, Finding Association Rules with Apriori Algorithm, <http://www.fuzzy.cs.uniagdeburgde/~borglet/apriori.pdf>, diakses tgl 23 Februari 2007.

Daftar Pustaka hanya memuat semua pustaka yang diacu pada naskah tulisan, bukan sekedar pustaka yang dibaca. Pustaka ditulisurut kemunculan pengacuan di naskah, bukan urut abjad penulis.

- [1] Castleman, Kenneth R., 2004, *Digital Image Processing*, Vol. 1, Ed.2, Prentice Hall, New Jersey.
- [2] Gonzales, R., P. 2004, *Digital Image Processing (Pemrosesan Citra Digital)*, Vol. 1, Ed.2, diterjemahkan oleh Handayani, S., Andri Offset, Yogyakarta.
- [3] Wyatt, J. C, dan Spiegelhalter, D., 1991, *Field Trials of Medical Decision-Aids: Potential Problems and Solutions*, Clayton, P. (ed.): *Proc. 15th Symposium on Computer Applications in Medical Care*, Vol 1, Ed. 2, McGraw Hill Inc, New York.
- [4] Yusoff, M, Rahman, S., A., Mutalib, S., and Mohammed, A., 2006, Diagnosing Application Development for Skin Disease Using Backpropagation Neural Network Technique, *Journal of Information Technology*, vol 18, hal 152-159.
- [5] Wyatt, J. C, Spiegelhalter, D, 2008, *Field Trials of Medical Decision-Aids: Potential Problems and Solutions*, *Proceeding of 15th Symposium on Computer Applications in Medical Care*, Washington, May 3.
- [6] Prasetya, E., 2006, Case Based Reasoning untuk mengidentifikasi kerusakan bangunan, *Tesis*, Program Pasca Sarjana Ilmu Komputer, Univ. Gadjah Mada, Yogyakarta.
- [7] Ivan, A.H., 2005, Desain target optimal, *Laporan Penelitian Hibah Bersaing*, Proyek Multitahun, Dikti, Jakarta.
- [8] Wallace, V. P., Bamber, J. C. dan Crawford, D. C. 2000. Classification of reflectance spectra from pigmented skin lesions, a comparison of multivariate discriminate analysis and artificial neural network. *Journal Physical Medical Biology*, No.45, Vol.3, 2859-2871.
- [9] Xavier Pi-Sunyer, F., Becker, C., Bouchard, R.A., Carleton, G. A., Colditz, W., Dietz, J., Foreyt, R. Garrison, S., Grundy, B. C., 1998, Clinical Guidelines on the identification, evaluation, and treatment of overweight and obesity in adults, *Journal of National Institutes of Health*, No.3, Vol.4, 123-130, http://journals.lww.com/acsm-msse/Abstract/1998/11001/paper_treatment_of_obesity.pdf.
- [10] Borglet, C, 2003, Finding Association Rules with Apriori Algorithm, <http://www.fuzzy.cs.uniagdeburgde/~borglet/apriori.pdf>, diakses tgl 23 Februari 2007.

FORM PENILAIAN REVIEWER SISFOTENIKA

Judul :

Penulis :

NO	UNSUR	KETERANGAN	MAKS	NILAI	KETERANGAN	MASUKAN
1	Keefektifan Judul Artikel	Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris	2		a. Tidak lugas dan tidak ringkas (0)	
					b. Kurang lugas dan kurang ringkas (1)	
					c. Ringkas dan lugas (2)	
2	Pencantuman Nama Penulis dan Lembaga Penulis		1		a. Tidak lengkap dan tidak konsisten (0)	
					b. Lengkap tetapi tidak konsisten (0,5)	
					c. Lengkap dan konsisten (1)	
3	Abstrak	Dalam Bahasa Indonesia dan Bahasa Inggris yang baik, jumlah 150-200 kata. Isi terdiri dari latar belakang, metode, hasil, dan kesimpulan. Isi tertuang dengan kalimat yang jelas.	2		a. Tidak dalam Bahasa Indonesia dan Bahasa Inggris (0)	
					b. Abstrak kurang jelas dan ringkas, atau hanya dalam Bahasa Inggris, atau dalam Bahasa Indonesia saja (1)	
					c. Abstrak yang jelas dan ringkas dalam Bahasa Indonesia dan Bahasa Inggris (2)	
4	Kata Kunci	Maksimal 5 kata kunci terpenting dalam paper	1		a. Tidak ada (0)	
					b. Ada tetapi kurang mencerminkan konsep penting dalam artikel (0,5)	
					c. Ada dan mencerminkan konsep penting dalam artikel (1)	
5	Sistematika Pembaban	Terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, hasil dan pembahasan, kesimpulan dan saran, daftar pustaka	1		a. Tidak lengkap (0)	
					b. Lengkap tetapi tidak sesuai sisetm (0,5)	
					c. Lengkap dan bersistem (1)	
6	Pemanfaatan	Pemanfaatan Instrumen	1		a. Tak termanfaatkan (0)	

	Instrumen Pendukung	Pendukung seperti gambar dan tabel			b. Kurang informatif atau komplementer (0,5)	
					c. Informatif dan komplementer (1)	
7	Cara Pengacuan dan Pengutipan		1		a. Tidak baku (0)	
					b. Kurang baku (0,5)	
					c. Baku (1)	
8	Penyusunan Daftar Pustaka	Penyusunan Daftar Pustaka	1		a. Tidak baku (0)	
					b. Kurang baku (0,5)	
					c. Baku (1)	
9	Peristilahan dan Kebahasaan		2		a. Buruk (0)	
					b. Baik (1)	
					c. Cukup (2)	
10	Makna Sumbangan bagi Kemajuan		4		a. Tidak ada (0)	
					b. Kurang (1)	
					c. Sedang (2)	
					d. Cukup (3)	
					e. Tinggi (4)	
11	Dampak Ilmiah		7		a. Tidak ada (0)	
					b. Kurang (1)	
					c. Sedang (3)	
					d. Cukup (5)	
					e. Besar (7)	
12	Nisbah Sumber Acuan Primer berbanding Sumber lainnya	Sumber acuan yang langsung merujuk pada bidang ilmiah tertentu, sesuai topik penelitian dan sudah teruji. Sumber acuan primer dapat berupa: tulisan dalam makalah ilmiah dalam jurnal internasional maupun nasional terakreditasi, hasil penelitian di dalam disertasi, tesis, maupun skripsi	3		a. < 40% (1)	
					b. 40-80% (2)	
					c. > 80% (3)	
13	Derajat Kemutakhiran Pustaka Acuan	Derajat Kemutakhiran Pustaka Acuan	3		a. < 40% (1)	
					b. 40-80% (2)	
					c. > 80% (3)	
14	Analisis dan Sintesis	Analisis dan Sintesis	4		a. Sedang (2)	
					b. Cukup (3)	
					c. Baik (4)	
15	Penyimpulan	Sangat jelas relevasinya	3		a. Kurang (1)	

		dengan latar belakang dan pembahasan, dirumuskan dengan singkat			b. Cukup (2)	
					c. Baik (3)	
16	Unsur Plagiat		0		a. Tidak mengandung plagiat (0)	
					b. Terdapat bagian-bagian yang merupakan plagiat (-5)	
					c. Keseluruhannya merupakan plagiat (-20)	
TOTAL			36		Catatan : Nilai minimal untuk diterima 25	

Catatan:

Plagiat :

Reviewer,

(Nama Reviewer)

**FORMULIR BERLANGGANAN
JURNAL SISFOTENIKA
STMIK PONTIANAK**

Nama :
Unversitas/Instansi :
Alamat Pengiriman :
Telp. /Fax/Email :

Telah mengirimkan uang sebesar Rp.

Rincian : Rp. [Sisfotenika Edisi]
Rp. [Biaya Kirim]



Pilihan Berlangganan :

2 [dua] edisi – Akademisi Rp. 100.000,-^{*)}

2 [dua] edisi – Umum (Non Akademisi) Rp. 120.000,-^{*)}

^{*)} Biaya berlangganan belum termasuk biaya kirim sebesar Rp. 20.000,- / 2 edisi

Untuk berlangganan, kirimkan formulir ini beserta bukti transfer pembayaran:

a. Via surat ke alamat:

Redaksi Jurnal SISFOTENIKA

STMIK Pontianak

Jl. Merdeka No. 372 Pontianak, Kalimantan Barat

Telp. 0561-735555

b. Via fax ke **0561-737777**

c. Via Email ke alamat : **Sisfotenika@stmikpontianak.ac.id**

Untuk informasi lebih lanjut dapat menghubungi telepon 0561-735555 atau mengirimkan email ke sisfotenika@stmikpontianak.ac.id atau sisfotenika@gmail.com

Pembayaran dapat dilakukan melalui transfer rekening:

Bank

No. Rek.

a/n.

Berita :Sisfotenika (Biaya Berlangganan)

