

## Perancangan Neur-O: Sistem Pakar Penyakit Saraf berbasis Ontologi

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

**Christopel H. Simanjuntak\*<sup>1</sup>, Dirko G. S. Ruindungan<sup>2</sup>**

<sup>1</sup>Prodi D4 Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Manado, Manado

<sup>2</sup>Jurusan Teknik Elektro, Fakultas Teknik Elektro UNSRAT, Manado

e-mail: \*<sup>1</sup> [christopel.simanjuntak@polimdo.ac.id](mailto:christopel.simanjuntak@polimdo.ac.id), <sup>2</sup>[dirko@unsrat.ac.id](mailto:dirko@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

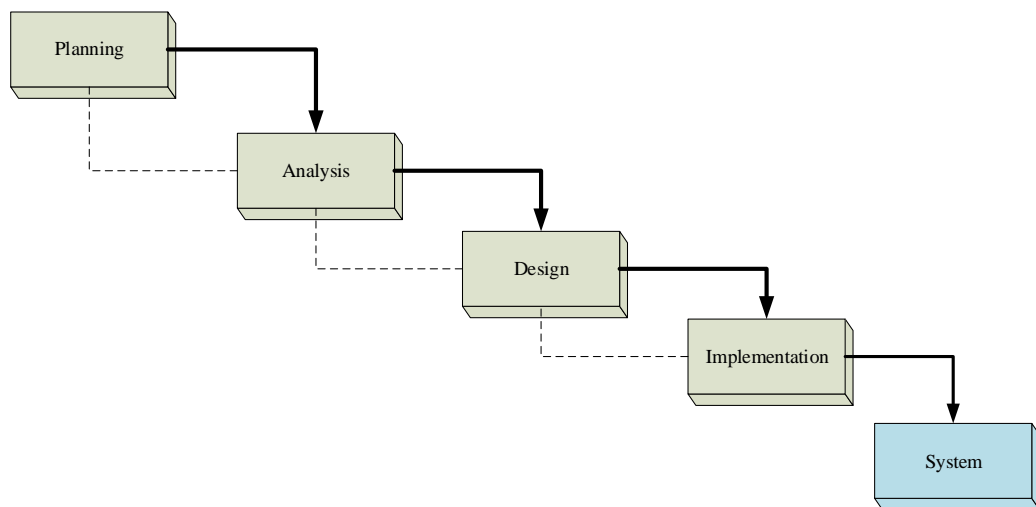
Penyakit saraf adalah penyakit yang bahaya untuk manusia. Menurut WHO, tahun 2005 hanya untuk penderita cerebrovascular disease mencapai 50785000 jiwa. Tahun 2015 penderita mencapai 53815000 jiwa dan diprediksikan mencapai 60864000 jiwa di seluruh dunia [1]. Penyakit saraf memberikan kontribusi sebesar 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

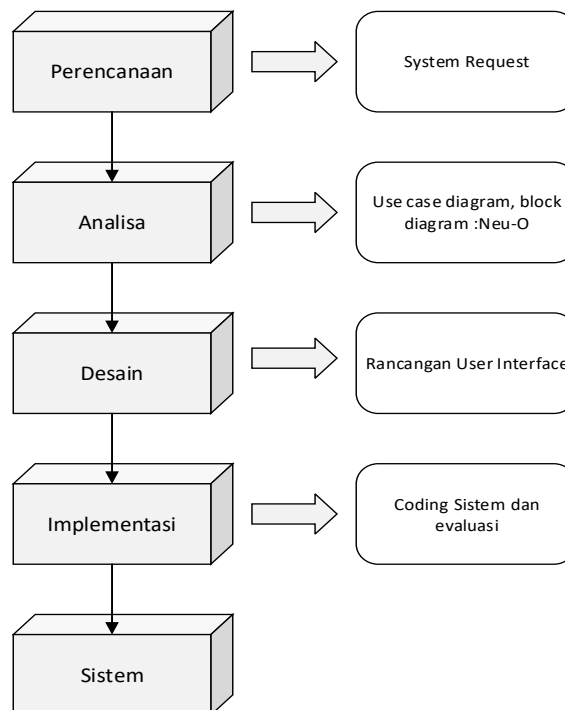
Dalam melakukan penelitian ini, metode mengacu pada metodologi pengembangan sistem *waterfall*[11] untuk membangun sistem pakar ini. Waterfall adalah salah satu metode pengembangan sistem [11]. Pada alur Waterfall ada 4 proses penting sehingga menghasilkan sistem atau protipe yang diinginkan.



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

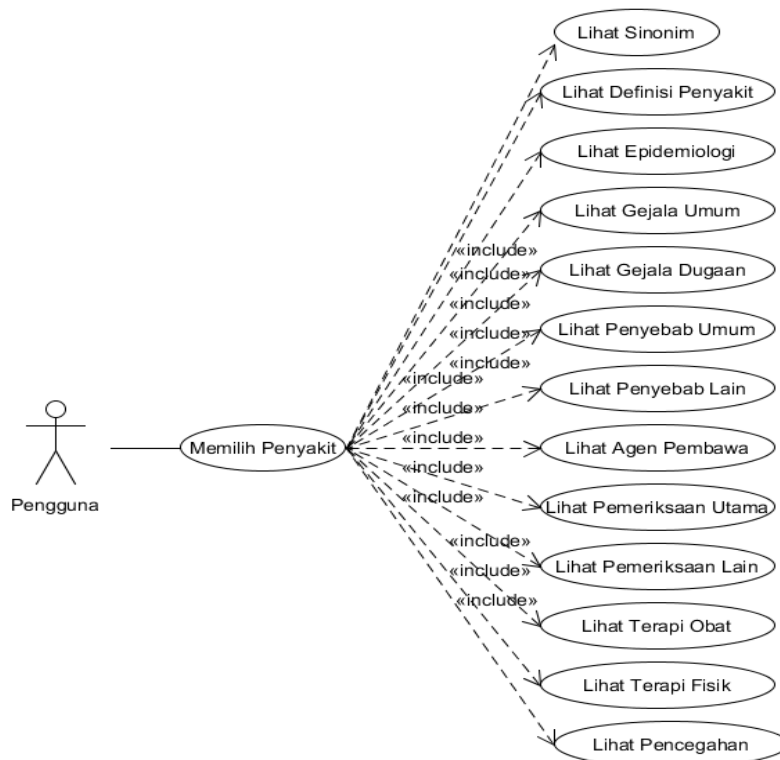
Tabel 1. Tabel system request Neur-O

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"> <li>1. Fitur pencarian berdasarkan penyakit yang ada</li> <li>2. Adanya fitur memilih kategori berdasarkan hasil yang ditampilkan</li> </ol>

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.

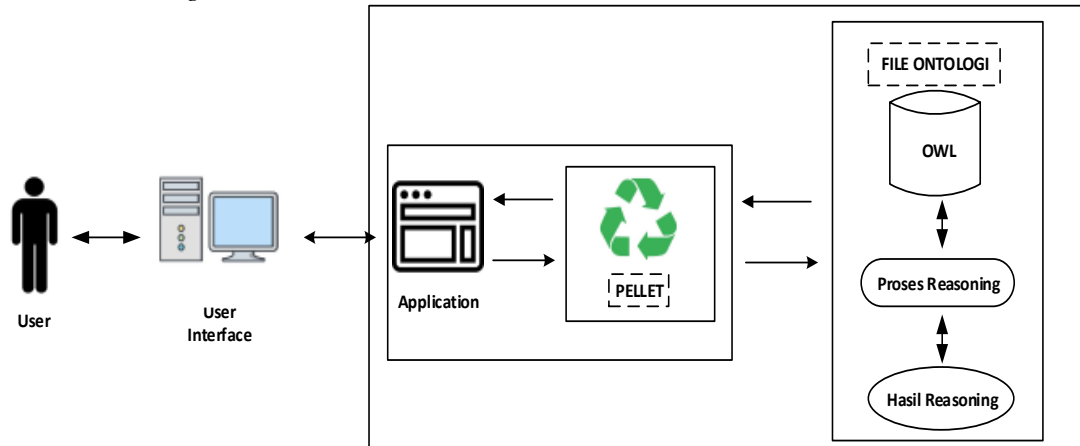
### 3.2. Use Case Diagram Neur-O



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, Epidermiologi, 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*

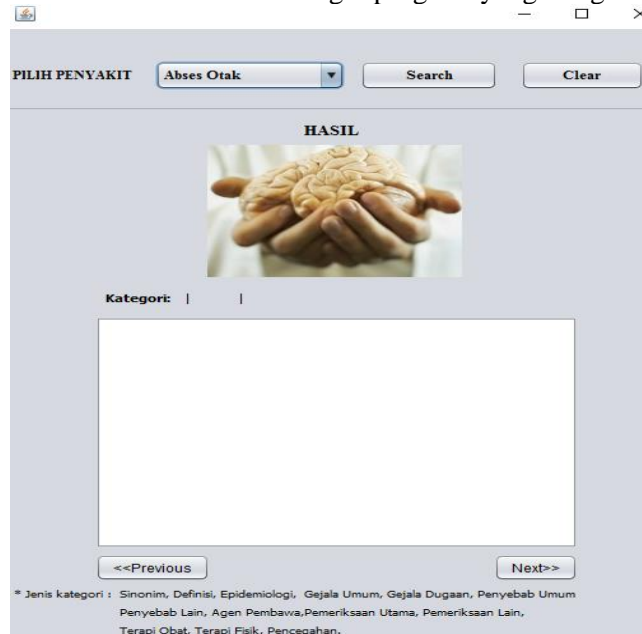


Gambar 6. Rancangan *Block Diagram* Neur-O

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.

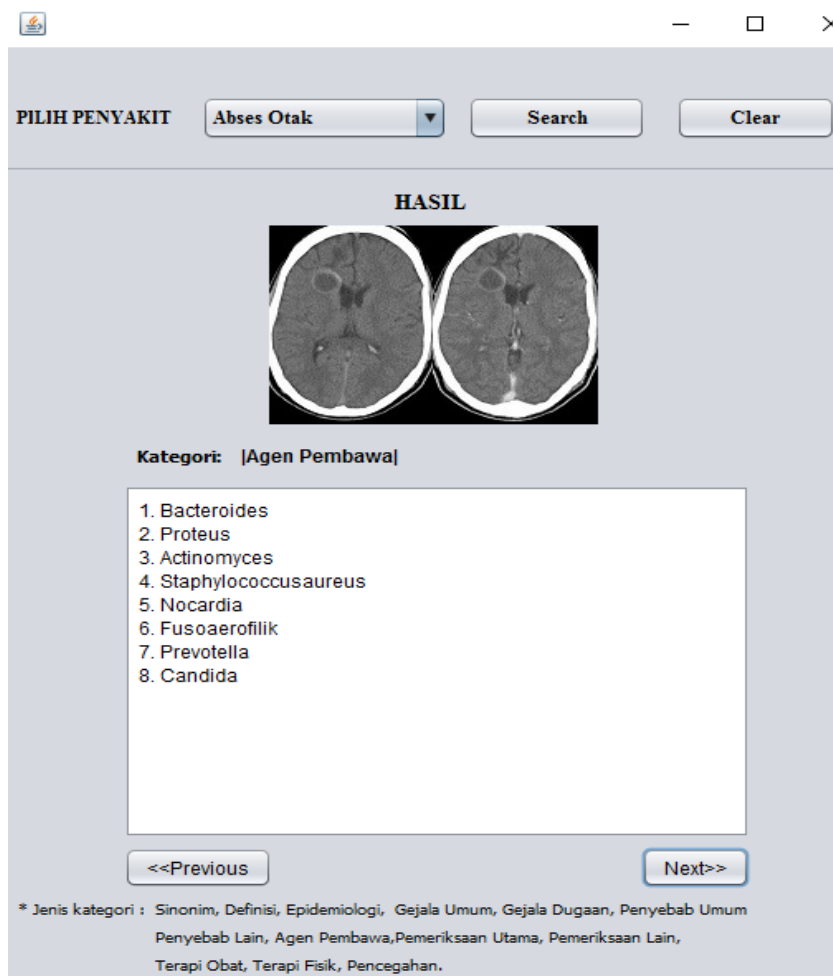
3.5. *Sistem yang terbangun dan Evaluasi*

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

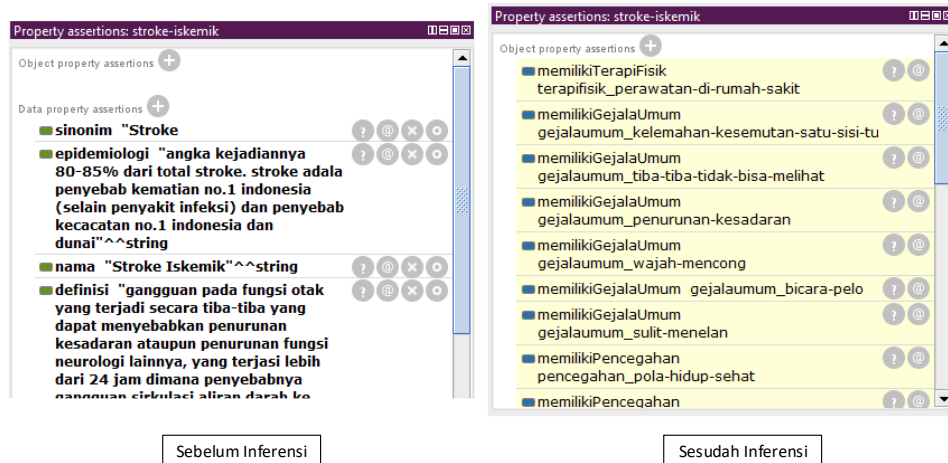
Pada gambar 8 menampilkan hasil dari pilihan penyakit abses otak. Hasilnya ditampilkan pada *text area* output berdasarkan kategori. Pemilihan kategori bisa menggunakan *button previous* dan *next* sesuai dengan urutan kategori.

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.



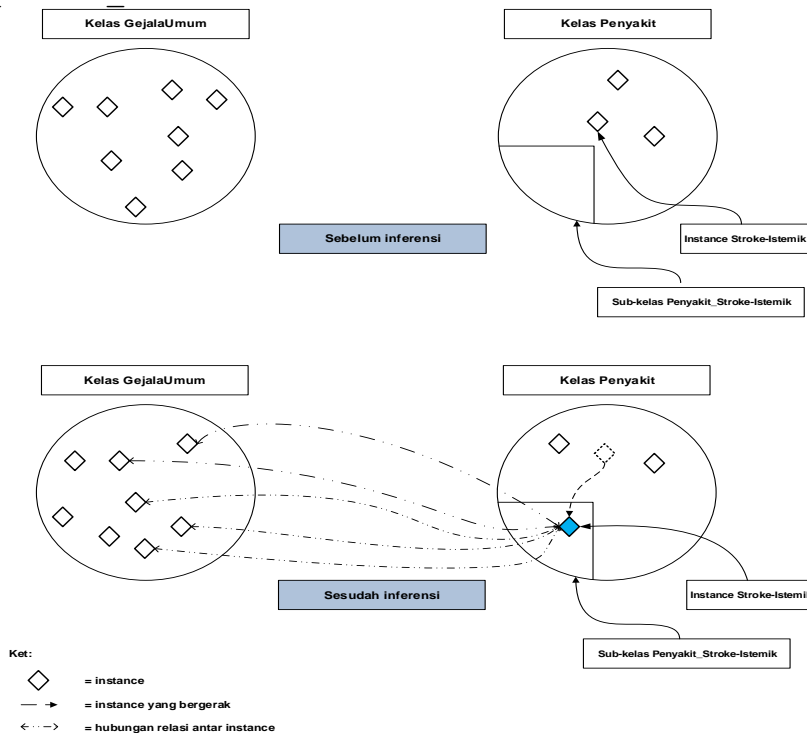
Gambar 9 Tampilan *reasoning process*

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.