

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
Jurusian 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 diwawancara 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 (SIAKAD) dan Sistem Informasi Administrasi Keuangan (SIAK):

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 (SIAKAD) dan Sistem Informasi Administrasi Keuangan (SIAK) 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

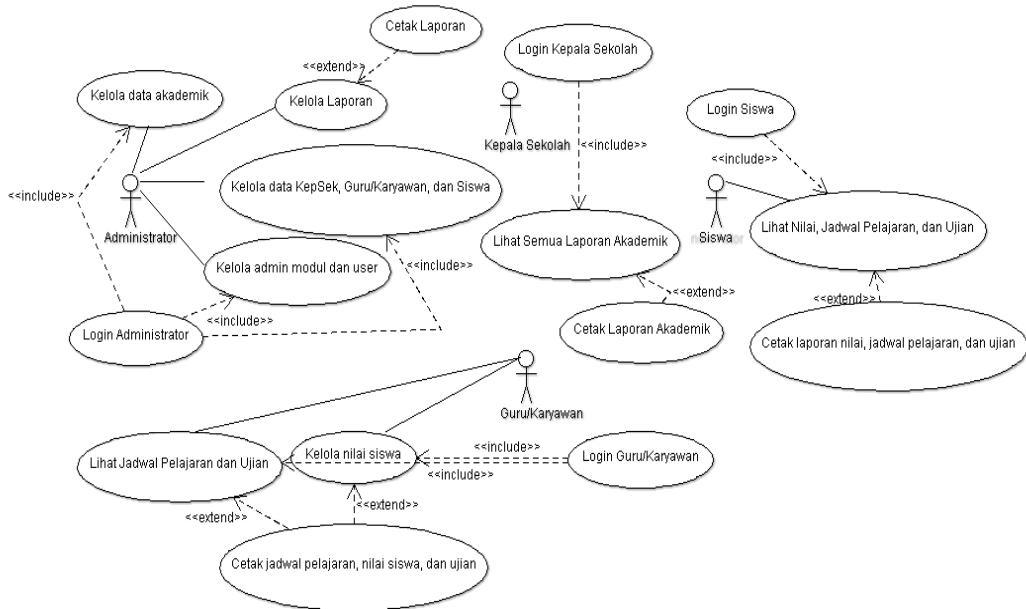
Setelah desain perangkat lunak selesai, maka dibuat *function coding* untuk tiap-tiap *form* dari perangkat lunak SIAKAD maupun SIAK dengan memanfaatkan *class-class* sehingga *coding* menjadi lebih sederhana dan dapat dimanfaatkan kembali tanpa mengubah desain *interface* perangkat lunaknya.

d. *Testing*/Pengujian

Pada tahapan peneliti menggunakan pengujian Black-Box Testing yang melakukan pengujian pada fitur dan fungsionalitas dari perangkat lunak SIAKAD dan SIAK 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

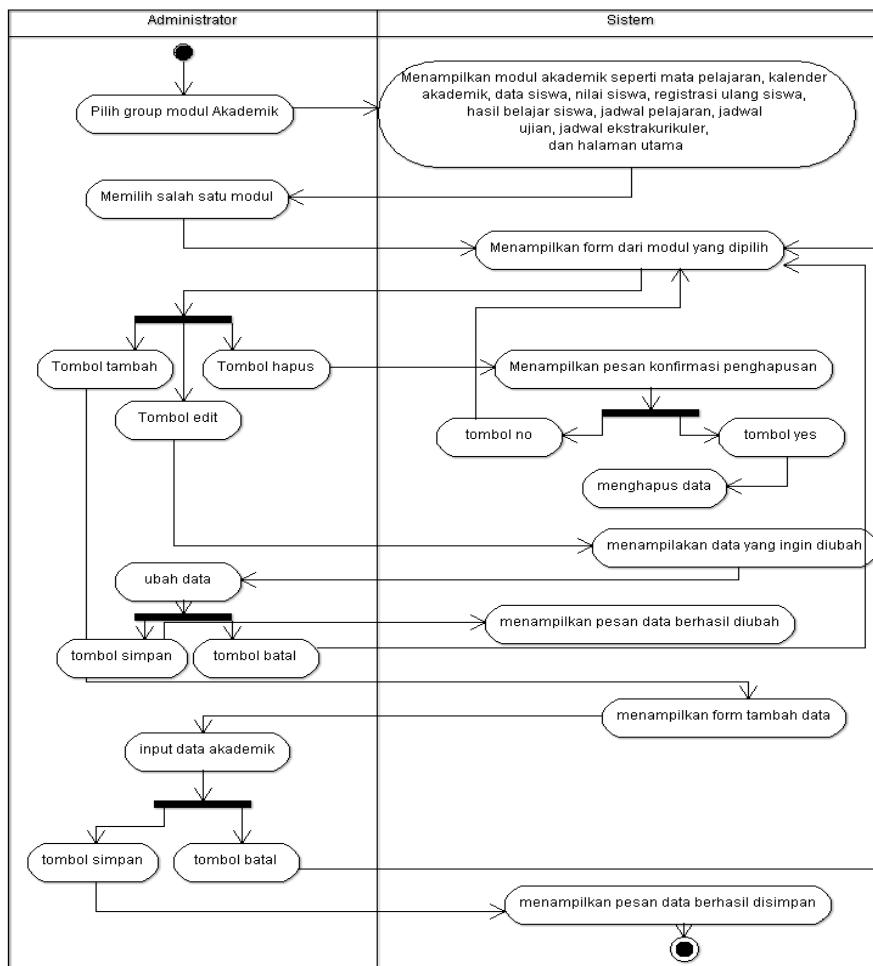
3.1 Use Case Diagram *Sistem Informasi Akademik*



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.

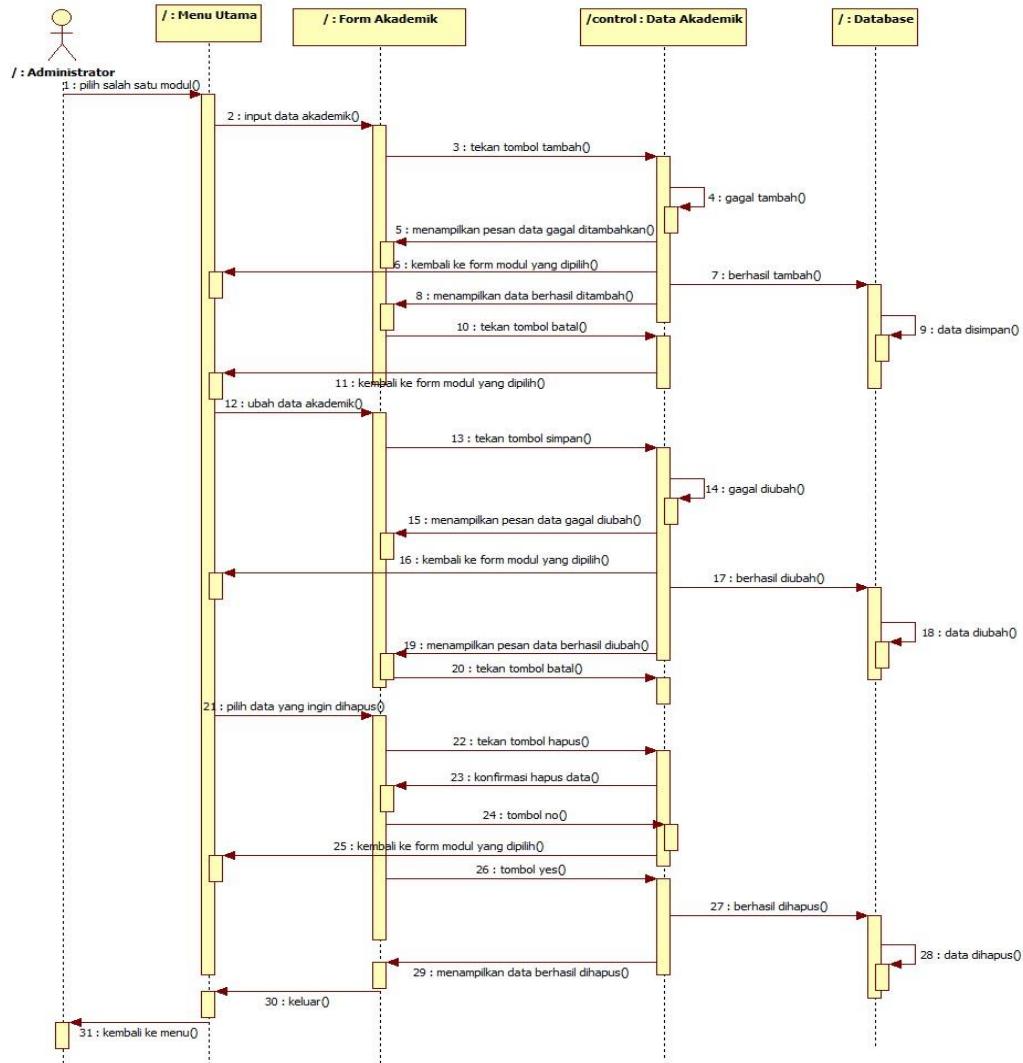
3.2 Activity Diagram *Kelola Data Akademik*



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     $iuran = $client->GetIuran();
68     $objiu = simplexml_load_string($iuran->GetIuranResult->any);
69
70     echo"<br><br>";
71     if (is_object($objiu)){
72         echo"<table border=1>";
73         echo"<tr><th>Kode Iuran</th><th>NIS</th><th>Kode Bayar</th><th>Bulan</th><th>Tanggal Terima</th><th>Jumlah</th><th>Status</th><th>Status Tahun</th></tr>";
74         foreach ($objiu->NewDataSet->Table as $data) {
75             echo "<tr>
76                 <td>$data->kodeiuran</td>
77                 <td>$data->nis</td>
78                 <td>$data->kodebayr</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' => $sile);
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 memanggil *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"> Pengunjung memasukkan parameter nama siswa dan menekan tombol cari. Data siswa dengan nama tersebut ditemukan, maka; 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, and Class Diagram*.
8. Pengujian perangkat lunak menggunakan metode *Black-box Testing*.
9. Sistem Informasi Akademik (SIAKAD) 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. SIAKAD 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. SIAKAD ini dapat digunakan untuk Sekolah Menengah Atas yang mempunyai kegiatan ekstrakurikuler maupun tidak ada.
12. SIAKAD berbasis *web* ini dapat dijadikan ajang promosi sekolah.
13. Sistem Informasi Administrasi Keuangan (SIAK) 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. SIAKAD dapat memanggil fungsi dari SIAK 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. SIAKAD SMA untuk selanjutnya dapat dibuat dalam versi Android ataupun IOS.
3. Dapat ditambahkan modul daftar alumni pada SIAKAD 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.