

PEMODELAN UML SISTEM PENERIMAAN MAHASISWA BARU BERBASIS WAP (Studi Kasus : Sistem Penerimaan Mahasiswa Baru UPI “YPTK” Padang)

Gushelmi¹, Deded Ramad Kamda²

¹Sistem Informasi Universitas Putra Indonesia YPTK Padang

²Sistem Informasi Universitas Putra Indonesia YPTK Padang

Jl.Raya Lubug Begalung Padang SUMBAR

Email : gushelmi74@gmail.com

ABSTRAK

Sistem Penerimaan Mahasiswa Baru berbasis WAP bukanlah sesuatu yang baru pada masa sekarang ini. Penerapan aplikasi berbasis WAP (*mobile*) telah banyak diterapkan pada semua bidang baik bidang pendidikan maupun bidang perdagangan. Pada masa sekarang ini banyak orang ingin mendapatkan informasi secara cepat dengan melakukan akses ke internet dengan menggunakan komputer dan perangkat *mobile*. Untuk itu perlunya pengembangan aplikasi berbasis Web dengan di imbangi dengan penerapan aplikasi berbasis WAP akan meningkatkan *mobilitas* penggunaannya. Dengan menerapkan aplikasi WAP pada sistem pendaftaran tentu akan memberi kemudahan kepada pendaftar untuk melakukan pendaftaran dan mendapatkan informasi menggunakan perangkat *mobile*. Pada pengembangan ini tentu diperlukan model yang dapat membantu menganalisa sistem dan merancang sistem . UML merupakan konsep pemodelan berbasis *Object Oriented* (OO) yang dapat membantu dalam perancangan dan penganalisaan sistem. Di dalam UML banyak diagram yang dapat digunakan untuk menganalisa sistem dan perancangan sistem. UML mempunyai tiga kategori utama diagram yaitu struktur diagram, behavior diagram dan *interaction* diagram. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

Kata Kunci : WAP, *Mobile*, UML, *Object Oriented*, Penerimaan Mahasiswa Baru

ABSTRACT

Admissions system based on WAP is not something new in the present. Implementation of applications based on WAP (mobile) has been widely applied to all areas of both education and trade. At the present time many people want to get information quickly to make access to the internet using the computer and mobile devices. Therefore the need for the development of Web-based applications with the Balance with the implementation of WAP-based applications will enhance the mobility of users. By implementing a WAP application on the course registration system will provide convenience to the registrar to register and obtain information using mobile devices. This is certainly necessary to the development of models to help analyze the system and designing the system. UML-based modeling is the concept of Object Oriented (OO) that can assist in the design and analysis system. In the UML many diagrams that can be used to analyze the system

and system design. UML diagram has three main categories, namely the structure diagrams, interaction diagrams and behavior diagrams. Where each of these categories has a diagram that describes the system architecture and integrated with each other.

Keywords : WAP, Mobile, UML, Object Oriented, Admission

1. PENDAHULUAN

Pada masa sekarang ini teknik pemodelan perangkat lunak terus berkembang, dimana pemodelan tersebut bertujuan untuk membantu pengembang perangkat lunak dalam melakukan pembuatan program / aplikasi. Banyak model pengembang -an perangkat lunak pada saat ini baik yang bersifat prosedural maupun *object oriented*. Salah satu model pengembangan perangkat lunak adalah UML (*Unified Modeling Language*), UML merupakan alat perancangan sistem yang berorientasi pada objek (*object oriented*). Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standar dan bersifat independen. UML diagram memiliki tujuan utama untuk membantu tim pengembang proyek berkomunikasi, mengeksplorasi potensi disain, dan memvalidasi desain arsitektur perangkat lunak (Haviluddin,2011).

Pada tahun 2011 UPI "YPTK" mengembangkan aplikasi pendaftaran mahasiswa baru yang berbasis WEB sebagai bentuk layanan untuk mempermudah calon mahasiswa mendapatkan informasi pendaftaran mahasiswa baru. Namun, dalam pengembangan aplikasi tersebut masih ada kekurangan yang dirasakan diantaranya :

1. Kurangnya tingkat keamanan informasi karna ada modul-modul program yang mudah diakses dari luar tanpa melakukan validasi *user*
2. Aplikasi masih berbasis WEB menyebabkan orang yang mencari informasi dengan menggunakan perangkat *mobile* kesulitan dalam mendapatkan informasi tentang pendaftaran.

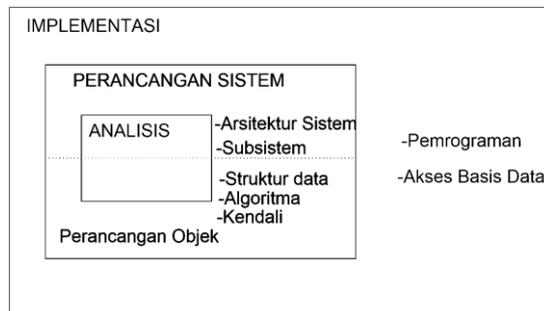
2. DASAR TEORI

Rekayasa perangkat lunak adalah sebuah profesi yang dilakukan oleh seorang perekayasa perangkat lunak yang berkaitan dengan pembuatan dan pemeliharaan aplikasi perangkat lunak dengan menerapkan teknologi dan praktik dari ilmu komputer, manajemen proyek, dan bidang-bidang lainnya (Janner Simarmata,2010:1). Perangkat lunak merupan insruksi langsung komputer untuk melakukan pekerjaan dan dapat ditemukan disetiap aspek kehidupan *modern* dari aplikasi yang kritis untuk hidup (*life-critical*).

2.1 Siklus Hidup Pengembangan Sistem Berorientasi Objek

Siklus hidup pengembangan sistem/ perangkat lunak berorientasi objek mengandung 3 proses *makro* : analisis berorientasi objek, perancangan berorientasi objek, dan implementasi berorientasi objek (Adi Nugroho,2005:130).

Model *use case* dapat dipergunakan sebagai pemandu pada hampir semua aktivitas pengembangan perangkat lunak mulai dari tahapan analisis, perancangan, implementasi, serta pengujian.



Gambar 2.1 Siklus Pengembangan Sistem Berbasis Objek

2.1.1 Analisis Berorientasi Objek

Analisis berorientasi objek (*OOA-Object Oriented Analysis*) adalah tahapan perangkat lunak dengan menentukan spesifikasi sistem (sering orang menyebutnya sebagai *SRS (System Requirement Specification)* dan mengidentifikasi kelas-kelas serta hubungannya satu terhadap yang lain (Adi Nugroho, 2005:131).

Untuk memahami spesifikasi sistem kita perlu mengidentifikasi para pengguna atau yang sering disebut sebagai aktor-aktor. Menurut Ivar Jacobson (Adi Nugroho, 2005:131), memperkenalkan konsep *use case* sebagai skenario untuk menjelaskan interaksi pengguna dengan sistem. Model *use case* menggambarkan pandangan pengguna atau kebutuhan pengguna.

2.1.2 Perancangan Berorientasi Objek

Sasaran dari perancangan berorientasi objek (*OOD-Object Oriented Design*) adalah merancang kelas-kelas yang teridentifikasi selama tahap analisis dan antarmuka pengguna (Adi Nugroho, 2005:132). Selama tahap ini mengidentifikasi dan menambah beberapa objek dan kelas yang mendukung implementasi dan spesifikasi kebutuhan. Perancangan berbasis objek dan analisis berorientasi objek adalah topik-topik yang terpisah namun keduanya saling bekerja sama dengan erat. Aktivitas dan fokus dari analisis berorientasi objek dan perancangan berbasis objek saling bekerja sama, saling melengkapi (Adi Nugroho, 2005:132).

2.2 UML

Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal didunia pengembangan siste yang berorientasi obyek (Munawar, 2005:18). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomuni- kasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembang -kan oleh Booch, *Object Modeling Technique (OMT)* dan *Object Oriented Software Engineering*

(OOSE) . Metode Booch dari Grady Booch sangat terkenal dengan nama Metode *Design Object Oriented*.

Untuk perancangan sebuah aplikasi dengan menggunakan UML, UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Tipe diagram UML dapat dilihat pada table berikut ini.

Tabel 2.1 Tipe Diagram UML

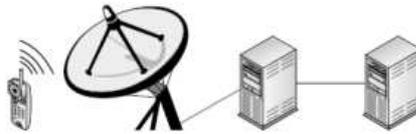
Diagram	Tujuan
<i>Activity</i>	Perilaku prosedural dan paralel
<i>Class</i>	<i>Class</i> , Fitur dan relasinya
<i>Communication</i>	Interaksi diantara obyek. Lebih menekankan ke link
<i>Component</i>	Struktur dan koneksi dari komponen
<i>Composite structure</i>	Dekomposisi sebuah class saat <i>runtime</i>
<i>Deployment</i>	Penyebaran/ instalasi ke klien
<i>Interaction Overview</i>	Gabungan antara <i>activity</i> dan <i>sequence</i> diagram
<i>Object</i>	Contoh konfigurasi <i>instance</i>
<i>Package</i>	Struktur hierarki saat kompilasi
<i>Sequence</i>	Interaksi antara obyek. Lebih menekankan pada urutan
<i>State Machine</i>	Bagaimana <i>event</i> mengubah sebuah obyek
<i>Timing</i>	Interaksi antara obyek. Lebih menekankan pada waktu
<i>Use Case</i>	Bagaimana <i>User</i> berinteraksi dengan sebuah sistem

2.3 WAP

WAP adalah singkatan dari *Wireless Application Protocol*. Definisi kamus untuk masing-masing kata-kata ini (Dale Bulbrook,2001), diantaranya :

1. Aplikasi Sebuah program komputer atau bagian dari perangkat lunak komputer yang dirancang untuk melakukan tugas tertentu.
2. *Nirkabel*, Kurang atau tidak memerlukan kawat atau kabel: yang berkaitan dengan radio transmisi.
3. Protokol satu set aturan teknis tentang bagaimana informasi harus dikirim dan diterima menggunakan komputer.

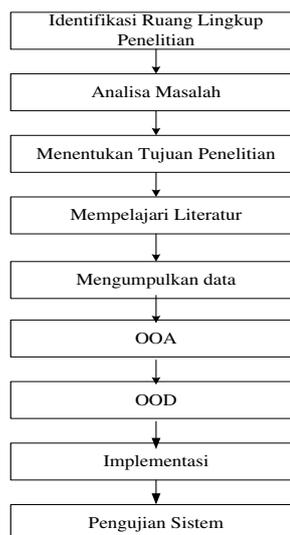
Prinsip kerja dari WAP adalah aliran data dari *phones (client)* atau WAP protokol, akan mengirim *encoded request*, protokol *gateway* akan mentranslasikan *request* dari WAP protokol yang terdiri dari WSP, WTP, WTLS dan WDP tersebut menuju WWW protokol (server, yaitu HTTP, TCP/IP), *encoder* akan menyesuaikan format data dengan server jaringan www yang dapat berupa CGI dan script, kemudian server akan merespon request tersebut menjadi kode-kode yang dimengerti oleh WAP *emulator* dan ponsel.



Gambar 2.2 Arsitektur Aplikasi WAP

3. METODOLOGI PENELITIAN

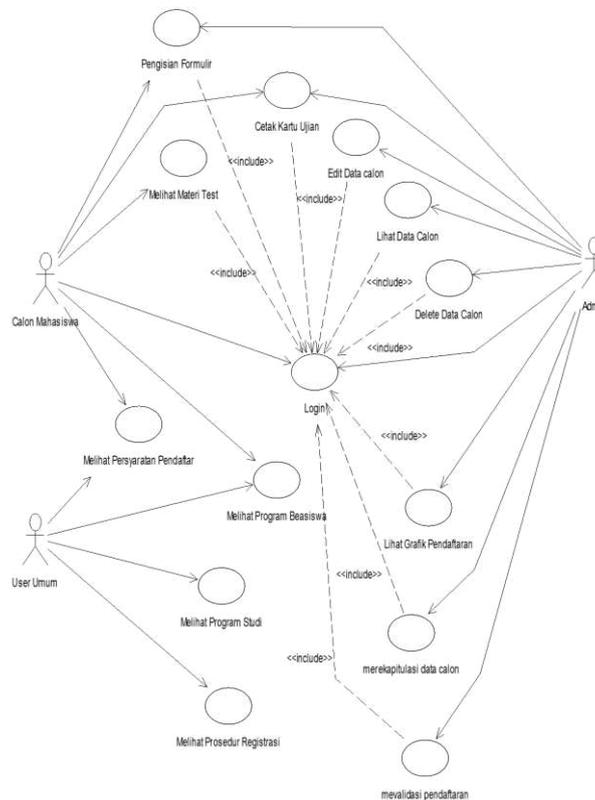
Kerangka kerja merupakan langkah-langkah yang akan di lakukan dalam rangka penyelesaian masalah yang akan dibahas. Sedangkan metodologi yang akan digunakan pada thesis ini adalah metodologi yang berbasis objek yang terdiri dari OOA (*Object Oriented Analysis*) dan OOD (*Object Oriented Design*). Untuk memperjelas kerangka kerja dari tahap awal sampai akhir, metodologi penelitian langsung dihubungkan dengan kerangka kerja (*frame work*) dalam sebuah kerangka penelitian.



Gambar 3.1 Kerangka Penelitian

4. ANALISA DAN PERANCANGAN

Pada tahap analisa akan dilakukan analisis terhadap data dan informasi yang didapat di lapangan. Data yang digunakan pada penelitian ini bersumber dari bagian SISFO Kampus Universitas Putra Indonesia “YPTK” Padang dan hasil pengamatan peneliti di lapangan. Untuk melihat aplikasi atau kegiatan yang ada pada aplikasi SPMB berbasis WEB bisa dilihat pada gambar dibawah ini.



Gambar 4.1 Use Case Diagram SPMB UPI “YPTK” Padang Berbasis Web

Setelah mempelajari sistem yang ada dengan melakukan serangkaian pengujian berdasarkan *use case* di atas dijumpai beberapa kekurangan, diantaranya :

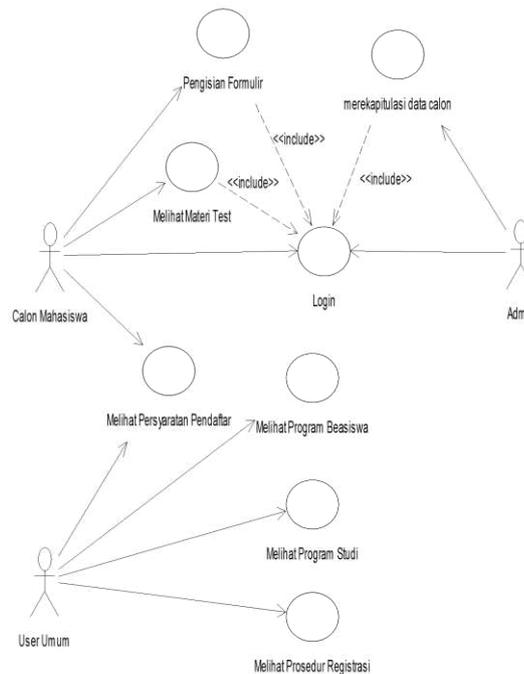
1. Pada pengujian menggunakan perangkat *mobile*, aplikasi kurang mendukung perangkat *mobile* terutama dari segi ukuran halaman web yang ditampilkan.
2. Pada pengujian *link* yang ada pada aplikasi, ditemukan kekurangan dari segi validasi modul untuk halaman admin. Pada halaman admin masih ada celah keamanan karna modul program tidak terlindungi dengan sistem keamanan yang telah dibangun.

Berdasarkan identifikasi masalah yang didapat, maka diusulkan sebuah perbaikan sistem. Usulan sistem diantaranya :

1. Untuk masalah mobilitas dan ukuran, aplikasi akan dikembangkan dengan menggunakan aplikasi yang berbasis *mobile*. Dengan penerapan aplikasi berbasis *mobile* akan memberi kemudahan bagi calon mahasiswa untuk mengakses menggunakan perangkat *mobile*.
2. Dalam mengatasi masalah autentikasi/perlindungan modul juga akan dibuatkan sebuah fungsi untuk pengecekan *session* pada program dan dilakukan pewarisan fungsinya. Dengan pembuatan cek *session* yang *mendirect* link ke halaman menu umum bagi yang tidak memiliki hak akses. Fungsi-fungsi yang dibangun pada aplikasi berbasis WAP ini bisa digunakan untuk memperbaiki kekurangan pada sistem berbasis website dengan memasukkan modul program untuk autentifikasinya.

Dalam pengembangan aplikasi tetap menggunakan aliran sistem sedang berjalan, pengembangan dilakukan dengan membuat aplikasi berbasis WAP menggunakan UML sebagai perancangan. Pada pengembangan ini tidak semua konten/fungsi yang ada pada

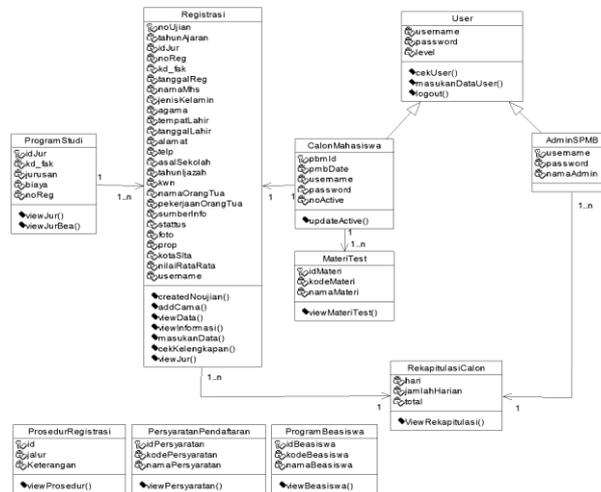
aplikasi berbasis website yang akan diadopsi pada aplikasi berbasis WAP. Beberapa konten akan dihilangkan terutama konten pada admin karena admin tetap menggunakan komputer untuk memasukkan data tetapi admin tetap bisa memantau perkembangan dengan menggunakan perangkat *mobile* untuk melihat perkembangan pendaftaran. Berdasarkan analisa diatas, konten/fungsionalitas yang akan digunakan pada aplikasi berbasis WAP dapat dilihat pada gambar 4.2.



Gambar 4.2 Use Case SPMB UPI “YPTK” Padang Berbasis WAP

Class Diagram

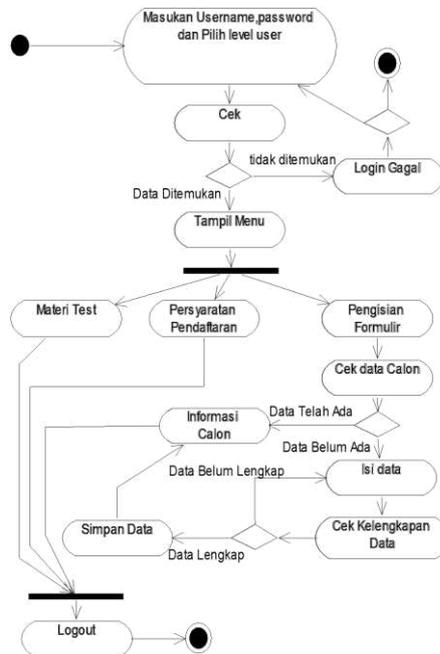
Meskipun dalam pembahasan ini pemodelan *class* dilakukan pemodelan *use case*, sebenarnya pada faktanya kedua aktivitas tersebut dilakukan secara paralel. Kedua model tersebut sebenarnya saling mendukung dalam pemberian informasi. *Class* biasanya digunakan untuk mendefinisikan obyek-obyek bisnis. *Class-class* seperti ini biasanya mendefinisikan model *database* dari suatu aplikasi.



Gambar 4.3 Class Diagram Sistem Penerimaan Mahasiswa Baru

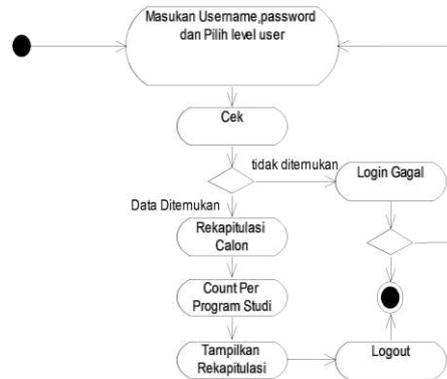
Activity Diagram

Activity diagram merupakan diagram yang nantinya akan menjabarkan sistem berdasarkan Use case Diagram. Dimana dalam pembuatan sebuah activity diagram bisa berasal dari beberapa use case. Berikut ini adalah activity diagram untuk Login, Pengisian Formulir, Melihat Materi Test, Melihat Persyaratan Pendaftaran.

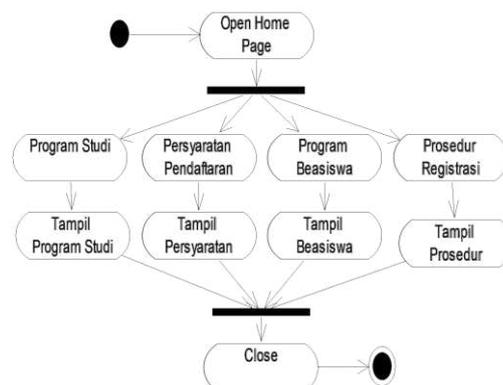


Gambar 4.4 Activity Diagram Pendaftaran Mahasiswa Baru

Untuk *activity* diagram kedua adalah *activity* diagram dari sisi adminnya dimana pada *activity* ini berdasarkan *use case* login dan Merkapitulasi Data Calon .



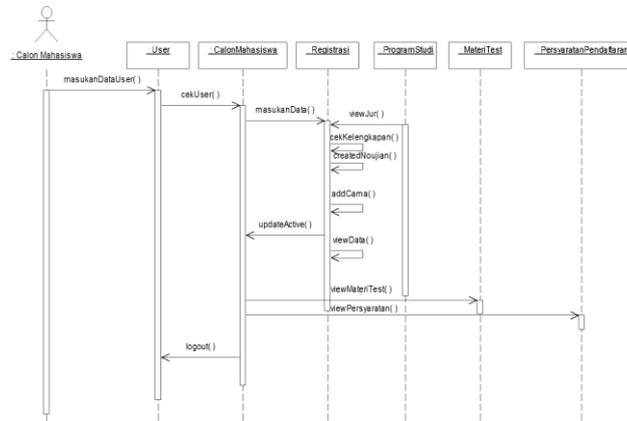
Gambar 4.5 Activity Diagram Informasi Pendaftaran



Gambar 4.6 Activity Diagram Untuk User Umum

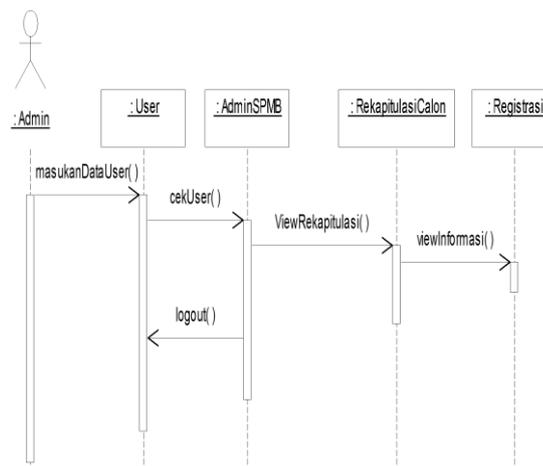
Sequence Diagram

Sequence diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* yang menjalankan sistem.. *Sequence* diagram didasarkan atas *class* diagram yang sudah dibuat. Berikut ini adalah *sequence* diagram pada untuk Sistem Penerimaan Mahasiswa Baru UPI “YPTK” Padang.



Gambar 4.7 Sequence Diagram Registrasi Penerimaan Mahasiswa

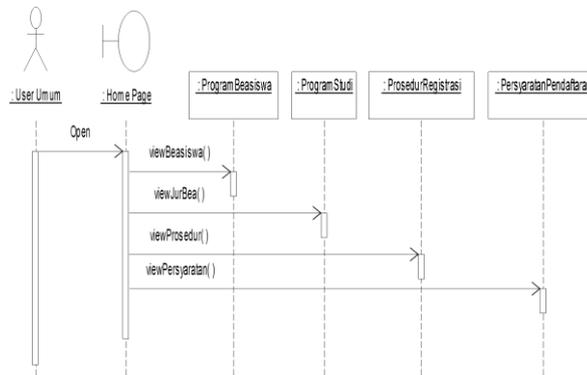
Sequence Diagram Registrasi Penerimaan Mahasiswa menjelaskan urutan kejadian pada saat seorang calon mahasiswa melakukan pendaftaran. Pada tahap pertama seorang aktor memasukkan data *username*, *password* dan pemilihan level berdasarkan *class user*. Setiap user yang memasukkan data ke *form login* maka akan dilakukan validasi data berdasarkan *class CalonMahasiswa*. Setelah *login* seorang calon mahasiswa menginputkan data, dalam penginputan data terjadi pembacaan dari *class* program studi dan menghitung jurusan untuk pemberian nomor ujian. Saat melakukan penyimpanan data calon mahasiswa pada saat itu data PMB *user* akan di *update* menjadi *active*. Selain menginputkan data, seorang calon mahasiswa juga dapat melihat informasi tentang materi *test* dan persyaratan pendaftaran dan jika calon mahasiswa ingin keluar sistem tinggal menekan tombol *logout/keluar*. Semua aksi pada garis panah diatas dibuat berdasarkan operation yang ada pada *class diagram* masing-masing objek.



Gambar 4.8 Sequence Diagram Admin Penerimaan Mahasiswa Baru

Gambar diatas menjelaskan urutan kejadian dan interaksi antara objek, dimana *event* itu dimulai dari seorang admin memasukkan data dan dilakukan pengecekan data. Jika datanya sesuai dengan yang ada pada *database* maka seorang admin dapat melihat hasil

rekapitulasi calon mahasiswa yang mana hasil dari rekapitulasi diambil dari operasi `viewInformasi()` yang ada pada *class* Registrasi.

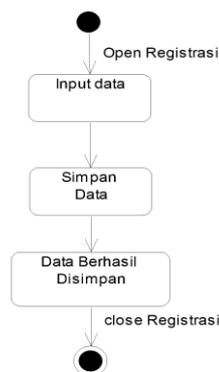


Gambar 4.9 Sequence Diagram Informasi Penerimaan Mahasiswa

Pada gambar diatas melihatkan *class* yang dapat dijalankan oleh seorang user umum. Semua *class* yang disediakan untuk *user* umum dapat diakses melalui halaman utama pendaftaran mahasiswa baru.

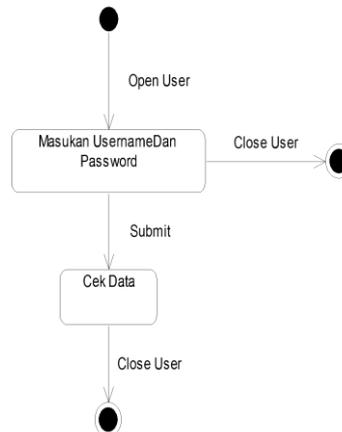
Statechart Diagram

Jika pada pemodelan interaksi menyiapkan detail spesifikasi dari *use case*, pada *statechart* akan diberikan detail deskripsi dari *class* yaitu perubahan *state* dari *class* menjadi lebih tepat. Perubahan dinamis inilah yang akan menjadi perilaku dari suatu objek. Biasanya *statechart* ini memodelkan aturan main suatu proses bisnis. *State* diagram khususnya digunakan untuk memodelkan taraf-taraf diskrit suatu siklus objek. *State* memodelkan objek dari semenjak dibuat sampai selesai. Pada kondisi ini tidak semua *class* akan mempunyai *state*. Maka *statechart* diagram untuk penerimaan mahasiswa baru dapat dilihat pada gambar dibawah ini.



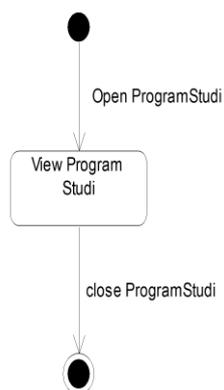
Gambar 4.10 Statechart Diagram Registrasi

Pada gambar 4.10 menggambarkan interaksi yang terjadi dan setelah seorang user masuk ke sistem dan melakukan registrasi datanya. *State* melihat aksi dari satu kejadian ke kejadian selanjutnya. Di *state* diagram akan tergambar proses yang digunakan pada aplikasi saat dirunning. Pada *state* ini dimulai dengan mengisikan membuka form pengisian data kemudian dilanjutkan ke tahap berikutnya untuk melakukan pengecekan kelengkapan data. Jika data telah lengkap diisi baru dilakukan pembuatan no ujian dan melakukan penyimpanan data. Pada *state* ini di akhiri dengan keluar dari *class* register.



Gambar 4.11 Statechart Diagram User

Gambar 4.11 menggambarkan interaksi yang terjadi dan sewaktu seorang *user* admin masuk ke sistem. *State* melihat aksi dari satu kejadian ke kejadian selanjutnya. Pada *state* ini dimulai dengan mengisikan data *username*, *password*, pemilihan level *user* dan dilanjutkan dengan melakukan pengecekan data login. Pada *state* ini di akhiri dengan keluar dari *close user*.



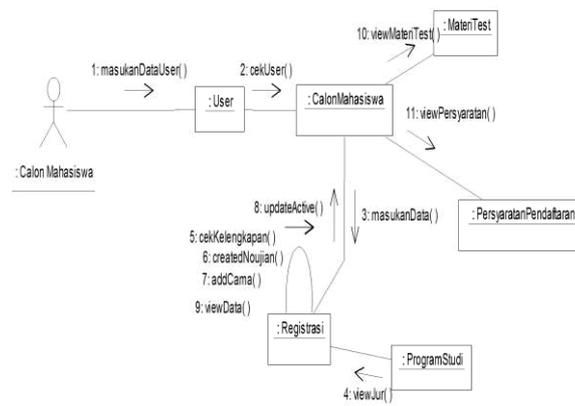
Gambar 4.12 Statechart Diagram ProgramStudi

Statechart Diagram untuk menampilkan informasi-informasi tentang pendaftaran dapat dilihat pada *state* diatas. *State* pada gambar 4.12 menjelaskan untuk menampilkan sebuah informasi tentang program studi hanya terdapat satu kali perpindaahan *state*. *State*

seperti diatas dapat digunakan untuk menjelaskan perilaku dari *class* Persyaratan Pendaftaran, Program Beasiswa dan Prosedur Registrasi. Setiap statechart diagram diakhiri dengan *open class* dan diakhiri *close class*.

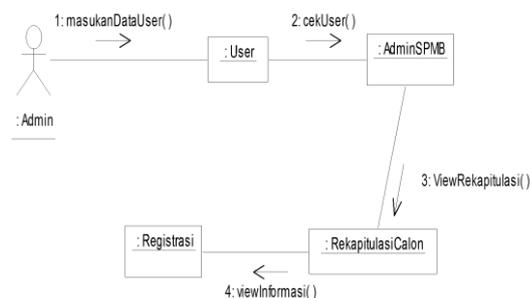
Collaboration Diagram

Pada pembuatan *collaboration* diagram berdasarkan *sequence* diagram yang telah dibuat pada tahap sebelumnya. *Collaboration* diagram dibuat agar setiap proses yang terjadi akan terlihat lebih jelas setiap alur pendaftaran maupun alur pengontrolan oleh admin. Dengan demikian bisa terlihat bagaimana setiap *participant/object* saling terkait satu sama lainnya. Secara lebih detail, *collaboration* diagram dapat dilihat pada gambar berikut ini.



Gambar 4.13 Collaboration Diagram Pendaftaran Calon Mahasiswa

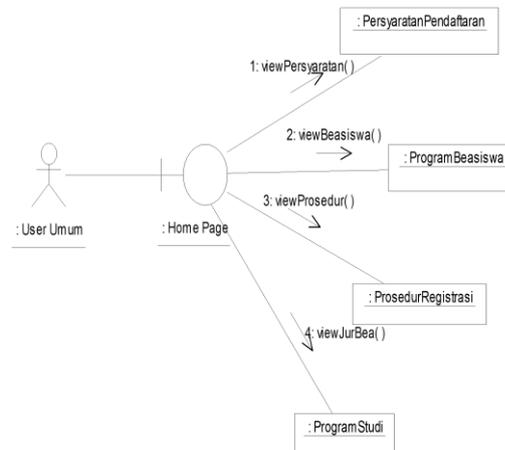
Collaboration Diagram Pendaftaran Calon Mahasiswa merupakan penjabaran dari *sequence* diagram gambar 4.7. Pada *collaboration* diagram akan dilakukan pengurutan dengan menggunakan sistem penomoran yang diawali dari no 1:masukanDataUser sampai dengan 11: viewPersyaratan.



Gambar 4.14 Collaboration Diagram Admin Pendaftaran

Collaboration Diagram Admin Pendaftaran merupakan penjabaran dari *sequence* diagram gambar 4.8. Pada *collaboration* diagram akan dilakukan pengurutan dengan

menggunakan sistem penomoran yang diawali dari no 1:masukanDataUser sampai dengan 4: viewInformasi.



Gambar 4.15 Colaboration Diagram Informasi Pendaftaran

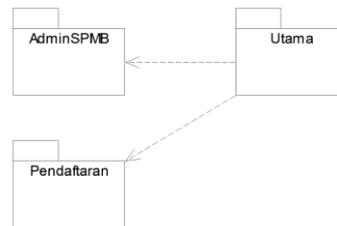
Colaboration Diagram Informasi Pendaftaran merupakan penjabaran dari *sequence* diagram gambar 4.9. Pada *collaboration* diagram akan dilakukan pengurutan dengan menggunakan sistem penomoran yang diawali dari no 1:viewPersyarat sampai dengan 4: viewJur.

Package Diagram

Package diagram bisa digunakan untuk mengelompokkan *use case* dan *class* diagram. Dari *use case* gambar 4.2 dapat dikelompokkan ke dalam 3 (tiga) *package*. Dalam hal ini *use case* dikelompokkan atas beberapa *package* diantaranya :

1. *Package* Utama
 Package ini terdiri dari *use case* Login, Melihat Persyaratan Pendaftar, Melihat Program Beasiswa, Melihat Program Studi, Melihat Prosedur Registrasi.
2. *Package* AdminSPMB
 Package ini terdiri dari *use case* Merekapitulasi Data Calon.
3. *Package* Pendaftaran
 Package ini terdiri dari *use case* Pengisian Formulir, Melihat Materi Test.

Berdasarkan pengelompokan *package* diatas maka dapat dilakukan pembuatan *package* diagram untuk *use case*. Hubungan antara *package* dari masing-masing *use case* bisa dilihat pada gambar berikut ini.



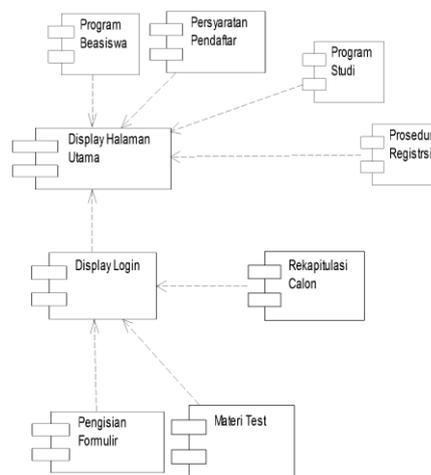
Gambar 4.16 Package Use Case

Component Diagram

Component merupakan bagian fisik dari sebuah sistem, karena perancangan *component* tidak bisa lepas dari *platform* implementasi. SPMB Online merupakan aplikasi berbasis WAP dengan *database server*. Pada sistem berbasis WAP, logika bisnis bisa diletakkan di *server* atau di *client*.

Aplikasi berbasis WAP bisa mencakup aplikasi *server* untuk mengatur logika aplikasi untuk memonitor *state* dari aplikasi. Pada aplikasi, monitoring *state* adalah aktivitas yang sangat penting dalam rangka melihat *track* dari aksi *online* para pengakses.

Teknik umum yang dipakai untuk monitoring *state* adalah menaruh *cookie* di *browser*. Biasanya dalam aplikasi akan dilakukan pembatasan waktu aktivitas *user* dengan menggunakan *session time*. Selain itu tentu ada hal lain yang harus diperhatikan yaitu tentang pembatasan dan kejelasan hak akses atas sebuah modul.



Gambar 4.17 Component Diagram

Deployment Diagram

Untuk perancangan aplikasi berbasis WAP, paling tidak ada 4 hal yang perlu diperhatikan saat pembuatan arsitektur *deployment*-nya. tiga hal tersebut adalah :

1. *Browser* pada *client*

Browser pada sistem *client* bisa digunakan untuk menampilkan halaman web dalam ukuran sesuai dengan *mobile* yang digunakan. Aplikasi yang dibangun akan menggunakan PHP, jadi fungsi-fungsi yang akan digunakan berada pada sisi *server*.

2. WAP Gateway

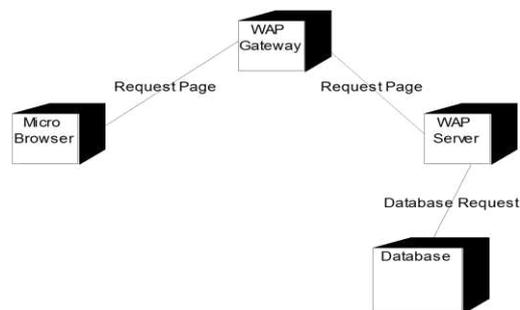
Penggunaan WAP gateway merupakan hal yang digunakan bagi pengguna internet yang menggunakan perangkat *mobile* dengan menggunakan kartu yang meghubungkan dengan provider tertentu. WAP gateway merupakan penghubung antara perangkat *mobile* dengan WAP Server.

3. WAP Server

WAP server-lah yang menangani permintaan/request dari micro browser dan secara dinamis men-generate halaman WAP dan code program untuk dijalankan dan ditampilkan di client. WAP server jugalah yang melakukan penyesuaian dan pemberian parameter untuk sesi kepada pengguna.

4. Database server

Sama sepertihanya Web Server, *database* yang digunakan tetap menggunakan database yang lama sehingga aplikasi ini bisa dimanfaatkan oleh pengunjung.



Gambar 4.18 Deployment Diagram

5. IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada tahap implementasi dan pengujian sistem akan dilakukan pembuatan program dan pengujian keamanan modul yang dibuat dengan menggunakan pengujian *Black box*. Perangkat yang digunakan dalam pengujian diantaranya :

1. *Micro browser*

Micro browser yang digunakan adalah *Opera Mobile*, *Opera Mini* dan pengujian sistem dilakukan secara *online*.

2. Perangkat *mobile*

Perangkat *mobile* yang digunakan dalah *handphone* Sonny Ericson WT19i

5.1 Implementasi

Dalam implementasi *class* aplikasi dilakukan berdasarkan *class- class* yang diperoleh pada tahap perancangan. Implementasi dilakukan dengan menggunakan bahasa pemrograman PHP-WML.

5.1.1 Implementasi Halaman User

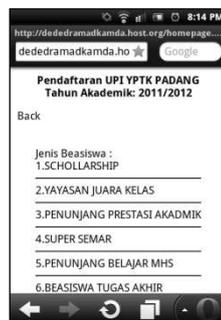
Pada pengujian aplikasi yang telah diimplementasikan ke dalam bentuk *code* program. Maka akan dilakukan pengujian aplikasi dengan menjalankan semua halaman WAP yang telah dirancang. Pada halaman *user* umum akan terlihat menu yang disediakan, diantaranya:

1. Program beasiswa.
2. Program studi.
3. Persyaratan pendaftaran.
4. Prosedur registrasi.

Sedangkan menu *login* ditujukan untuk calon mahasiswa baru dan admin. Dimana hasil *running* untuk halaman *user* umum dapat dilihat pada gambar 5.1



Gambar 5.1 Menu Halaman Utama



Gambar 5.2 Halaman Program Beasiswa

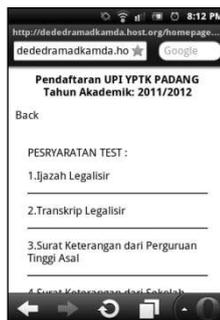
Gambar 5.2 merupakan hasil dari pemilihan menu Program Beasiswa. Jenis beasiswa yang ditampilkan merupakan hasil implementasi *class* Program Beasiswa.



Gambar 5.3 Halaman Program Studi

Gambar 5.3 merupakan tampilan halaman program studi berdasarkan pemilihan pada menu utama. Halaman ini memberikan informasi tentang jurusan yang ada dan rincian biaya perkuliahannya. Halaman program studi merupakan hasil implementasi dari *class* Program Studi.

Pada gambar 5.4 juga merupakan hasil dari pemilihan menu utama yang merupakan hasil dari pemilihan persyaratan pendaftaran. Halaman ini menjelaskan tentang persyaratan tes yang harus dipenuhi sewaktu pendaftaran. Halaman ini merupakan hasil implementasi dari *class* PersyaratanPendaftaran.



Gambar 5.4 Halaman Persyaratan Pendaftaran



Gambar 5.5 Halaman Prosedur Pendaftaran

Untuk gambar 5.1, gambar 5.2, gambar 5.3, gambar 5.4 dan gambar 5.5 merupakan aksi dari pemilihan menu sesuai dengan informasi yang diinginkan oleh *user*.

Semua halaman difasilitasi dengan tombol kembali ke menu utama untuk memberi kemudahan dalam mengoperasikannya.

Pada gambar 5.5 merupakan hal yang berbeda dengan sistem berbasis website pada desktop. Pada aplikasi *desktop*, informasinya dibuatkan dalam bentuk gambar sedangkan pada aplikasi WAP dilakukan dengan membuatnya dalam bentuk narasi. Pada aplikasi WAP, prosedur registrasi merupakan hasil implementasi *class* ProsedurRegistrasi

5.1.2 Implementasi Halaman Login

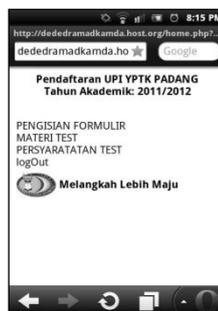
Halaman *login* merupakan implementasi dari *use case* atau pada *class* diagram, dimana *user* login memiliki 2 generalisasi. Dimana *user* terhubung dengan Calon Mahasiswa dan AdminSPMB. berfungsi untuk mengarahkan halaman *user* yang masuk ke sistem yang terdiri dari Calon Mahasiswa Baru dan admin. Hasil dari implementasi pemrograman dapat dilihat pada gambar 5.6.



Gambar 5.6 Menu Login

5.1.3 Implementasi Halaman Calon Mahasiswa

Untuk *login* akan dilakukan verifikasi berdasarkan level *user* yang dipilih, dimana level tersebut menentukan menu yang akan dipakai oleh *user* yang *login* dan menentukan *class* yang akan dipakai. Berikut ini hasil tampilan hasil *login* seorang Calon Mahasiswa masuk ke sistem. Untuk menu calon mahasiswa dapat dilihat pada gambar 5.7.



Gambar 5.7 Halaman Menu Calon Mahasiswa Baru

5.2 Pengujian Sistem

Tahap pengujian sistem dilakukan untuk melakukan validasi kesesuaian aplikasi yang telah dibuat dengan model yang dirancang. Pada tahap ini pengujian dilakukan dengan mengunakan *use case* diagram. Dimana setiap *link* pada setiap halaman pada WAP mewakili objek atau *use case* yang ada pada *use case* diagram.

Pengujian tahap pertama adalah pengujian menggunakan *micro browser* dan *browser* desktop. Pada pengujian ini menggunakan *google chrome* dan *opera mobile*. Langkah pengujiannya dengan memasukan address homepage WAP ke *browser*.



Gambar 5.14 Halaman SPMB Berbasis Web

Dari hasil pengujian dengan memasukan *address homepage WAP* pada *browser google chrome* maka halaman yang dijujut akan dialihkan ke halaman website berbasis *desktop*. Hasil dari pengujian dapat dilihat pada gambar 5.15 sedangkan jika menggunakan *opera mobile* akan tampil seperti gambar 5.1.

Pengujian tahap kedua ditujukan kepada semua modul pemrograman yang menggunakan algoritma terutama untuk keamanan untuk sisi admin dan calon mahasiswa yang mana dalam *use case* tergambar *include* ke *login*. Daftar *Use case* yang akan diuji.

1. Pengisian Formulir
2. Melihat Materi *Test*
3. Melihat Persyaratan Pendaftaran
4. Merkapitulasi Data Calon

6 KESIMPULAN

Kesimpulan dan saran didapat setelah melakukan pengembangan aplikasi pendaftaran mahasiswa baru berbasis *mobile*. Dari pengembangan aplikasi berbasis *mobile* menggunakan UML dapat diambil kesimpulan, diantaranya:

1. Pengembangan aplikasi berbasis *mobile* dapat meningkatkan tingkat *mobilitas* pengaksesnya karna informasi dapat diakses dimana saja dan kapan saja dengan menggunakan perangkat *mobile* tanpa harus menggunakan komputer.
2. Dokumentasi pemodelan sangat dibutuhkan karna dapat membantu dalam melakukan pengujian sistem sehingga sistem dapat berjalan dengan baik.
3. Penggunaan UML dalam perancangan sangat membantu dalam melakukan manajemen modul program saat dilakukan pengkodean.

DAFTAR PUSTAKA

- Al-Masri, Eyhab dan Qusay H.Mahmoud. *Investigating Web Services on the World Wide Web*. <http://www2008.org/papers/pdf/p795-almasriA.pdf>. Di Akses Pada Tanggal 9/02/2012. Pukul 08 :21 WIB.
- Bulbrook, Dale. 2001. *WAP A Beginner's Guide*. Osborne/McGraw-Hill. New York.
- Deshpande, Dkk. 2002. *Web Engineering*. Journal of Web Engineering Vol 1 No. 1 (2002) 003-017. Copy right Rinto Press.
- Haviluddin. 2011. *Memahami Penggunaan UML*. Unified Modelling Language. Jurnal Informatika Mulawarman. Vol 6 No.1 Februari 2011. Samarinda.
- Murugesan, Deshpande, dan Steve. *Web Engineering : A new Discipline For Development Of Web-Based Systems*. <http://www-itec.uni-klu.ac.at/~harald/proseminar/web11.pdf>. Di Akses Pada Tanggal 10/01/2012. Pukul 20:21 WIB.
- Nugroho, Adi. 2005. *Analisa dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Informatika: Bandung.
- Simarmata, Janner. 2010. *Rekayasa Perangkat Lunak*. Andi Offset: Yogyakarta.
- Sommerville, Ian. 2001. *Software Engineering*. Edisi 6. Addison-Wesey.
- Wijaya, Mulyanto, dan Mustakim. 2010. *Sistem Informasi Penerimaan Mahasiswa Baru Berbasis Web dan Wap*. Seminar Nasional VI SDM Teknologi Nuklir. 18 November 2010.ISSN 1978-0176. Yogyakarta.