

ANALISIS DAN PERANCANGAN BERORIENTASI OBJEK SISTEM PENJADWALAN KOAS

Pulut Suryati

Program Studi Sistem Informasi, STMIK AKAKOM
Jl. Raya Janti No. 143 Karangjamber Yogyakarta, 0274-486664
e-mail : lut_surya@akakom.ac.id

ABSTRAK

Koas merupakan sebutan untuk calon dokter merupakan kegiatan program kepaniteraan klinik dilaksanakan di Rumah sakit atau pun puskesmas yang ditunjuk. Dengan program ini diharapkan para calon dokter dapat mempunyai pengalaman menangani pasien yang sebenarnya dibawah pengawasan dokter sehingga nantinya menjadi lebih siap untuk menjadi dokter. Penjadwalan kaas yaitu pengaturan dari beberapa kaas yang akan dijadwalkan pada beberapa kegiatan dalam waktu atau periode tertentu. Hal ini bukan merupakan pekerjaan yang mudah. Sehingga perlu di analisis dan dirancang sistem untuk kebutuhan pengolahan data sehingga pada diperoleh pendokumentasian data yang baik. Pada penelitian ini metodologi yang akan digunakan untuk analisis dan perancangan menggunakan pemodelan berorientasi object. Perancangan menggunakan pemodelan UML (Unified Modeling Language). Berdasarkan list requirement dilakukan perancangan meliputi penentuan komponen actor, use case diagram, diagram interaktif dan diagram class. Hasil dari penelitian ini akan diketahui kebutuhan sistem dalam bentuk design sistem berorientasi obyek yang dapat digunakan sebagai acuan untuk pembuatan aplikasi perangkat lunak untuk sistem penjadwalan kaas.

Kata Kunci : penjadwalan, pemodelan, UML, berorientasi obyek

A. PENDAHULUAN

Program kepaniteraan klinik merupakan suatu bagian penting dalam sistem pendidikan kedokteran, program kepaniteraan klinik yaitu suatu periode pendidikan kedokteran yang ditekankan pada penerapan (aplikasi) teori-teori yang sebelumnya sudah di dapat dari periode pra klinik. Praklinik adalah menjalani kuliah-kuliah selama 3,5 tahun-4 tahun. Program kepaniteraan klinik dilaksanakan di Rumah sakit atau pun puskesmas yang ditunjuk, para calon dokter yang sedang melaksanakan program ini sering di sebut dengan istilah kaas (*trainee/ cleckship/ physicians*) (dari kata ko-asisten, artinya sebagai asisten dokter). Dengan program ini diharapkan para calon dokter dapat mempunyai pengalaman menangani pasien yang sebenarnya dibawah

pengawasan dokter sehingga nantinya menjadi lebih siap untuk menjadi dokter.

Program kepaniteraan klinik merupakan suatu bagian penting dalam sistem pendidikan kedokteran, program kepaniteraan klinik yaitu suatu periode pendidikan kedokteran yang ditekankan pada penerapan (aplikasi) teori-teori yang sebelumnya sudah di dapat dari periode pra klinik. Praklinik adalah menjalani kuliah-kuliah selama 3,5 tahun-4 tahun. Program kepaniteraan klinik dilaksanakan di Rumah sakit atau pun puskesmas yang ditunjuk, para calon dokter yang sedang melaksanakan program ini sering di sebut dengan istilah kaas (*trainee/ cleckship/ physicians*) (dari kata ko-asisten, artinya sebagai asisten dokter). Dengan program ini diharapkan para calon dokter dapat mempunyai pengalaman menangani pasien yang sebenarnya dibawah

pengawasan dokter sehingga nantinya menjadi lebih siap untuk menjadi dokter.

Pada program paniterian klinik para koas mengikuti sejumlah *activity* yaitu unit di rumah sakit yang ditunjuk untuk satu periode yang telah ditentukan. Program ini juga menjadi bagian untuk pemenuhan sumber daya di rumah sakit atau klinik (Belien, 2006; Amar, 2011). Karena sejumlah koas akan ditugaskan pada tiap unit bagian (disebut dengan *activity*) dalam suatu periode sehingga dapat memenuhi kebutuhan pegawai/staff, oleh karena itu perlu dijadwalkan dengan baik.

Penjadwalan koas yaitu pengaturan dari beberapa koas yang akan dijadwalkan pada beberapa *activity* dalam waktu atau periode tertentu. Hal ini bukan merupakan pekerjaan yang mudah. Sehingga perlu di analisisi dan dirancang sistem untuk kebutuhan pengolahan data sehingga pada diperoleh pendokumentasian data yang baik. Pada penelitian ini metodologi yang akan digunakan untuk analisis dan perancangan berorientasi object untuk sistem penjadwalan koas. Perancangan sistem berorientasi obyek akan memudahkan pengembangan dan pemeliharaan sistem karena karakteristik dari sistem beroreientasi obyek yaitu reusable.

Beberapa publikasi yang menggunakan pendekatan berorientasi obyek untuk analisis dan design sistem diantaranya Suruali dan Paillin (2010) menggunakan UML untuk Analisis Dan Desain Sistem Informasi Perpustakaan Migrasi Ke Digital Library. Pada publikasinya perpustakaan memiliki tujuan untuk meningkatkan pengeloalan dan pelayanan melalui

penyediaan berbagai sumber informasi dan multi media serta pemanfaatan teknologi informasi. Perpustakaan yang memiliki sistem informasi dapat mempermudah pengelolaan dan layanan perpustakaan. Metodologi dilakukan dalam tiga tahap. Pada tahap persiapan dilakukan pengumpulan data melalui kajian pustaka dan benchmarking beberapa perpustakaan sejenis. Tahap analisis meliputi analisis sistem yang ada (*excisting system*) serta perpustakaan yang akan dibangun, analisis kesiapan organisasi menggunakan uraian tugas perpustakaan, analisis terhadap prinsip pengembangan digital library menggunakan sepuluh prinsip pengembangan digital library, analisis terhadap tiga kegiatan utama perpustakaan yang terkait sistem, analisis terhadap basis data menggunakan pendekatan pemusatan data dengan database management system (DBMS), dan analisis terhadap aktor. Desain untuk sistem menggunakan bahasa pemodelan unified modelling language (UML).

Selain itu Windarti dan Banowosari (2006) Menggunakan UML untuk merancang dan menganalisis Sistem Informasi Bidang Kemahasiswaan menuliskan faktor utama dari ditemukannya pendekatan berorientasi obyek adalah karena ditemukannya kekurangan-kekurangan pada pendekatan terstruktur seperti biaya pengembangan perangkat lunak berkembang sesuai dengan berkembangnya keinginan atau kebutuhan pengguna, pemeliharaan yang sukar, lamanya penyelesaian suatu proyek, jangka waktu penyelesaian proyek yang selalu terlambat, biaya pengembangan perangkat lunak yang sangat tinggi dan sebagainya. Pendekatan

berorientasi obyek membuat data terbungkus pada setiap fungsi atau prosedur dan melindunginya terhadap perubahan yang tidak dikehendaki dari fungsi yang berada di luar. Model berorientasi obyek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi. Dalam tulisannya digunakan bahasa permodelan UML yang merupakan sistem arsitektur yang bekerja dalam Object Oriented Analysis and Design dengan menggunakan bahasa yang konsisten menentukan, visualisasi, mengkonstruksi, dan mendokumentasikan artifact yang terdapat dalam sistem software.

B. METODE PENELITIAN

Analisis dan desain berorientasi obyek adalah cara baru dalam memikirkan nyata. Dasar pembuatan adalah obyek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian "berorientasi obyek" berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari obyek tertentu yang memiliki struktur data dan perilakunya (Rumbaugh, et all, 1999).

Metodologi pengembangan sistem berorientasi obyek mempunyai tiga karakteristik utama yaitu *encapsulation*, *inheritance* dan *polymorphism*. *Encapsulation* merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu obyek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau obyek lain, kecuali prosedur yang berada dalam obyek itu sendiri. *Inheritance* (pewarisan) adalah

teknik yang menyatakan bahwa anak dari obyek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari obyek dari obyek induk diturunkan kepada anak obyek, demikian seterusnya. *Inheritance* mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya. Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Kemampuan obyek-obyek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Obyek (Booch, et all, 2005).

Adapun tahapan proses dalam penelitian ini pertama studi pustaka, yaitu mencari informasi dari literatur mengenai penjadwalan koas serta analisis dan perancangan berorientasi obyek. Kedua Mengumpulkan informasi yang dibutuhkan sistem seperti wawancara, yaitu mencari informasi dari sumber-sumber terkait mengenai penjadwalan koas. Ketiga melakukan proses analisis kebutuhan terhadap system dan keempat melakukan perancangan untuk menghasilkan design sistem penjadwalan koas.

Analisis sistem dilakukan dengan menyusun daftar kebutuhan pada masalah penjadwalan koas. Tabel 1 menunjukkan *requirement list* dari sistem yang akan dibangun.

Tabel 1 Requirement list masalah penjadwalan koas

o.	Requirements List	Use Case(s)
.	Administrator dapat menginputkan data Bagian	Input Bagian
.	Administrator dapat menginputkan data kelompok.	input kelompok
.	Administrator dapat menginputkan data mahasiswa	Input mahasiswa
.	Adminstrator dapat menginputkan data kapasitas per bagian pada periode penjadwalan	Input kapasitas
.	Adminstrator dapat menginputkan preferensi bagian yaitu apabila pada suatu periode pada bagian tertentu libur	Input preferensi bagian
.	Administrator dapat menginputkan data penjadwalan Aktifitas	Input data jadwal
.	Administrasi dapat melakukan proses Penjadwalan Aktifitas	Penjadwalan
.	Adminstrator dapat melihat jadwal baru maupun jadwal lama	Lihat jadwal
.	Administrator dapat melihat kapasitas	Lihat kapasitas terjadwal

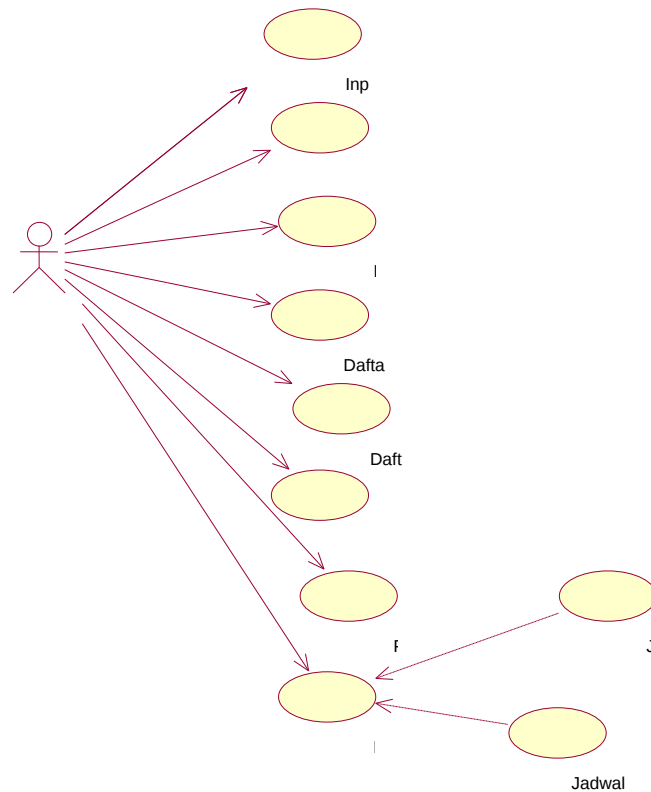
	bagian yang telah terjadwal	
0	Administrator dapat melihat jadwal koas per mahasiswa	Lihat jadwal per mahasiswa

C. HASIL DAN PEMBAHASAN

Berdasarkan diskripsi sistem dan requirement yang telah dibutuhkan, digunakan sebagai dasar untuk analisis dan perancangan sistem berorientasi obyek. Dari daftar kebutuhan sistem, selanjutnya ditentukan komponen aktor dibuat use case diagram. Actor menggambarkan orang, sistem atau external entitas / stakeholder yang menyediakan atau menerima informasi dari sistem. Dalam sistem yang dibuat komponen user (actor) adalah administrator. Administrator yaitu petugas atau staf bagian administrasi yang bertugas memasukkan data bagian, kapasitas, mahasiswa, dan mengelola penjadwalan koas mahasiswa.

Use case diagram

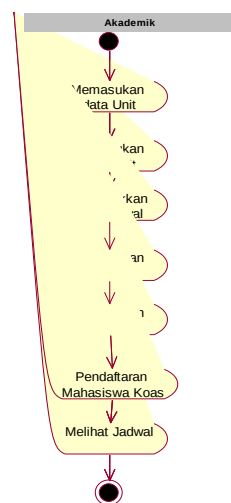
Use case adalah bagain tingkat tinggi dari fungsionalitas yang disediakan oleh sistem. Gambar 1 merupakan diagram use case yaitu menunjukkan beberapa use case dalam sistem, aktor dalam use case dan relasi antar mereka.



Gambar 1 Use Case Diagram Sistem Penjadwalan Koas

Diagram Aktivitas

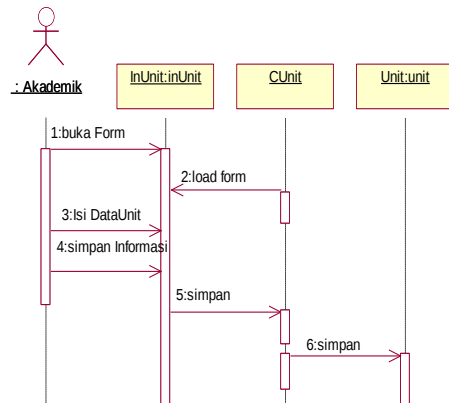
Diagram aktivitas adalah salah satu cara memodelkan aliran kejadian atau aliran kerja yang dalam proses bisnis. Aliran kejadian menyatakan rancangan skenario apa saja yang terjadi antara aktor dan sistem dalam use case. Gambar 2. Menunjukkan aliran kejadian proses penjadwalan Koas



Gambar 2 Aktivitas diagram Proses Penjadwalan Koas

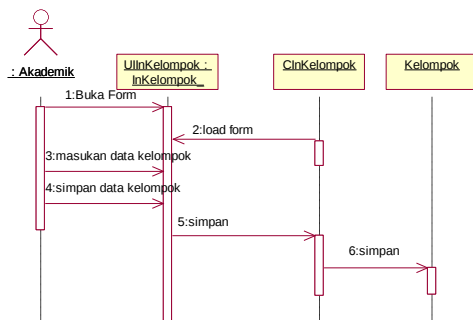
Diagram Sekuensial

Diagram sekuensial merupakan salah satu diagram interaksi. Diagram interaksi menunjukkan langkah-langkah kerja sama objek-objek di dalam use case. Objek apa saja yang dibutuhkan untuk aliran, pesan apa saja yang obyek kirimkan ke obyek lainnya dan urutan pesan-pesan yang dikirimkan. Diagram sekuensial diurutkan berdasarkan waktu. Diagram sekuensial untuk sistem penjadwal koas dari masing-masing use case input Unit ditunjukkan pada Gambar 3 sampai dengan Gambar 10. Gambar 3 menunjukkan diagram sekuensial untuk use case “input unit”. Tahapan ini meliputi proses membuka form, mengisikan dan dan menyimpan data.



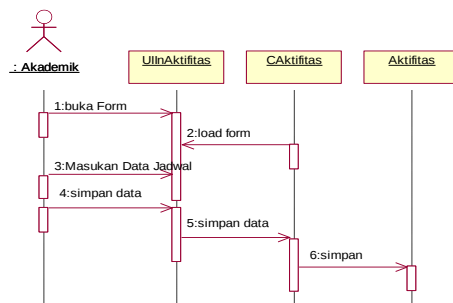
Gambar 3 Diagram sekuensial untuk use case “input Unit”

Gambar 4 menunjukkan diagram sekuensial untuk use case “input kelompok”. Tahapan ini meliputi proses membuka form, mengisi data dan menyimpan data.



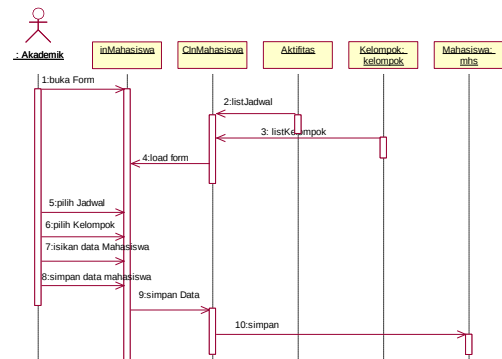
Gambar 4 Diagram sekuensial untuk use case “input Kelompok”

Gambar 5 menunjukkan diagram sekuensial untuk use case “input jadwal”. Tahapan ini meliputi proses membuka form, mengisi data dan menyimpan data.



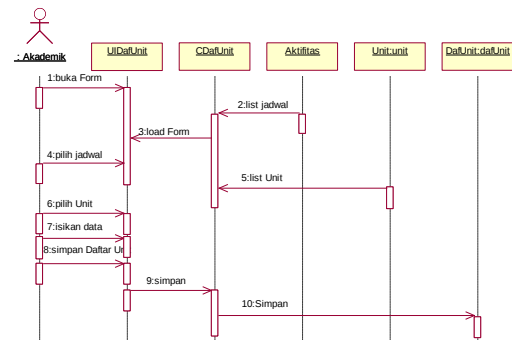
Gambar 5 Diagram sekuensial untuk use case “input Jadwal”

Gambar 6 menunjukkan diagram sekuensial untuk use case “input daftar mahasiswa”. Tahapan ini meliputi proses membuka form, memilih jadwal aktivitas, memilih kelompok, mengisi data mahasiswa dan menyimpan data.



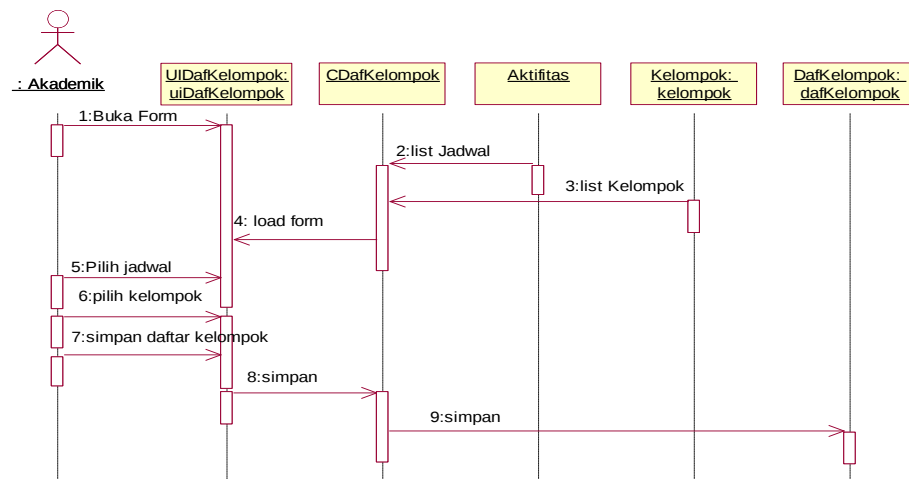
Gambar 6 Diagram sekuensial untuk use case “Input Daftar Mahasiswa”

Gambar 7 menunjukkan diagram sekuensial untuk use case “daftar unit”. Tahapan ini meliputi proses membuka form, memilih jadwal aktifitas, dan menyimpan data daftar unit.



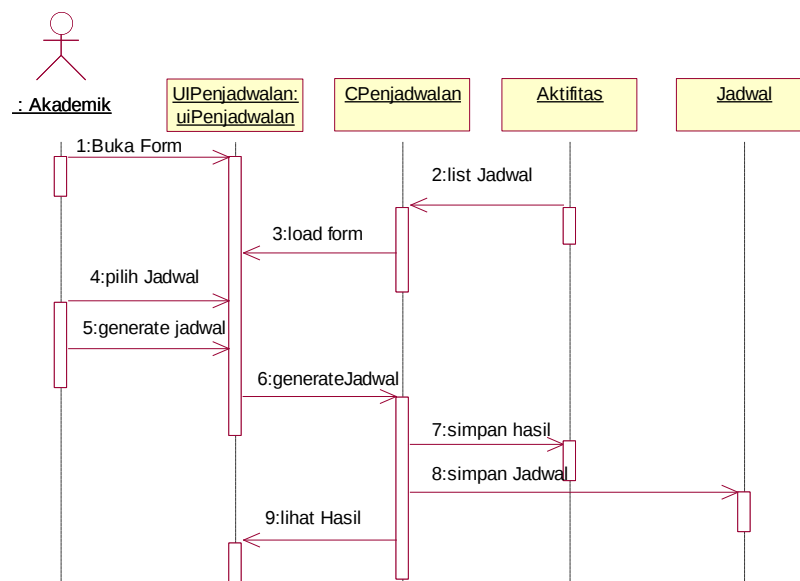
Gambar 7 Diagram sekuensial untuk use case “Input Daftar Unit”

Gambar 8 menunjukkan diagram sekuensial untuk use case “input daftar kelompok”. Tahapan ini meliputi proses membuka form, memilih jadwal aktifitas, memilih kelompok, dan menyimpan data daftar unit.



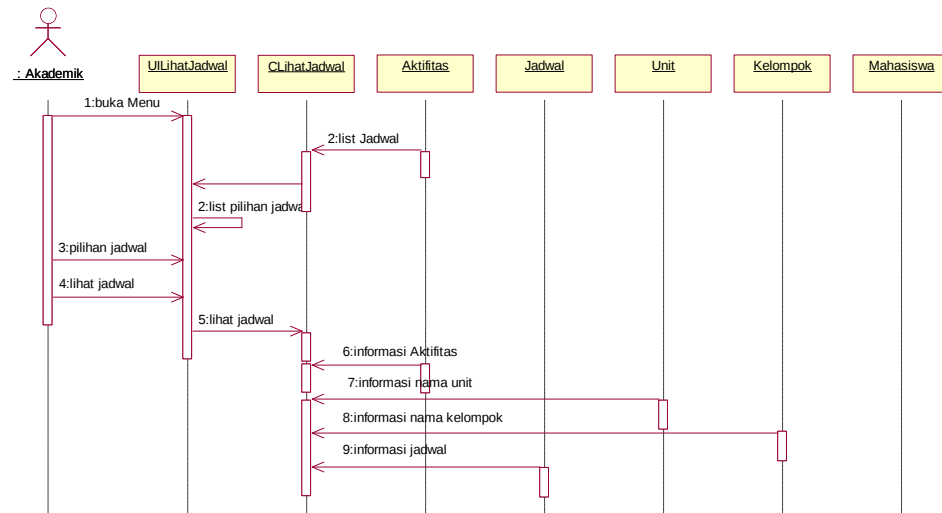
Gambar 8 Diagram sekuensial untuk use case “Input Daftar Kelompok”

Gambar 9 menunjukkan diagram sekuensial untuk use case “Generate Jadwal”. Tahapan ini meliputi proses membuka form, memilih jadwal aktifitas dan menyusun jadwal.



Gambar 9 Diagram sekuensial untuk use case “Generate Jadwal”

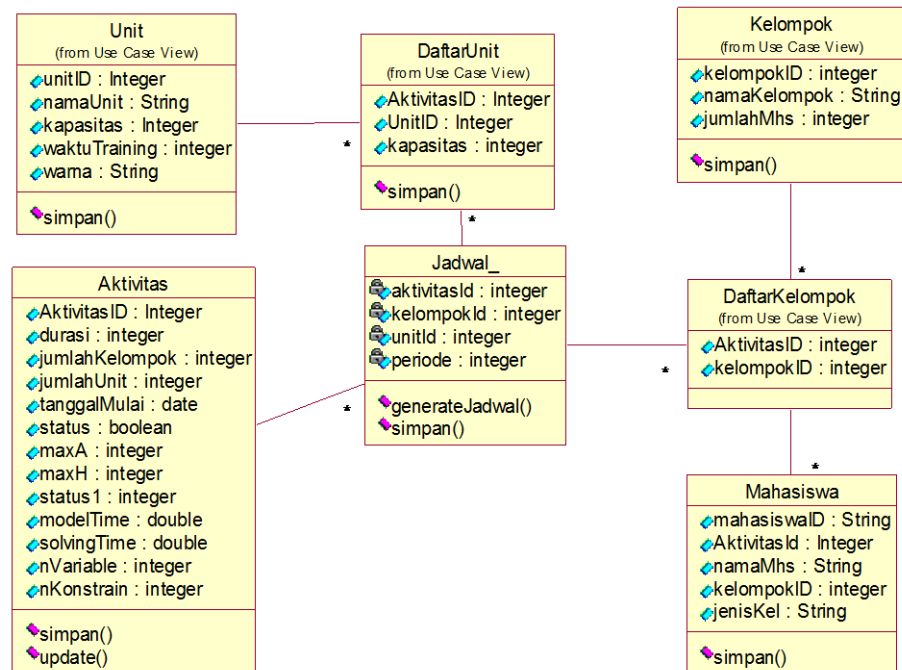
Gambar 10 menunjukkan diagram sekuensial untuk use case “Lihat Jadwal”. Tahapan ini meliputi proses membuka form, memilih jadwal, memilih kelompok, dan menyimpan data daftar unit.



Gambar 10 Diagram sekuensial untuk use case “Lihat Jadwal” Kelas Diagram

Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sistem dan relasi antar mereka. Diagram kelas memberikan gambaran sistem secara statis. Stereotype adalah sebuah mekanisme yang digunakan untuk mengkategorikan kelas-kelas. Stereotype juga membantu untuk proses pembangkitan kode. Ketika proses pembangkitan kode, stereotype kelas menentukan tipe kelas yang akan dibawa kedalam bahasa pemrograman.

Gambar 11. Menunjukkan kelas diagram sistem untuk stereotype entitas. Pada Gambar 11 diagram kelas stereotype Entitas yang memuat 7 kelas entitas. Kelas adalah sebuah kategori yang membungkus informasi dan perilaku. Salah satu perbedaan antara pendekatan terstruktur dengan pendekatan berorientasi obyek adalah bahwa pada berorientasi obyek terjadi pengabungan informasi dan perilaku pengolahan serta menyembunyikan kedua-duanya ke dalam sebuah kategori yang disebut kelas.



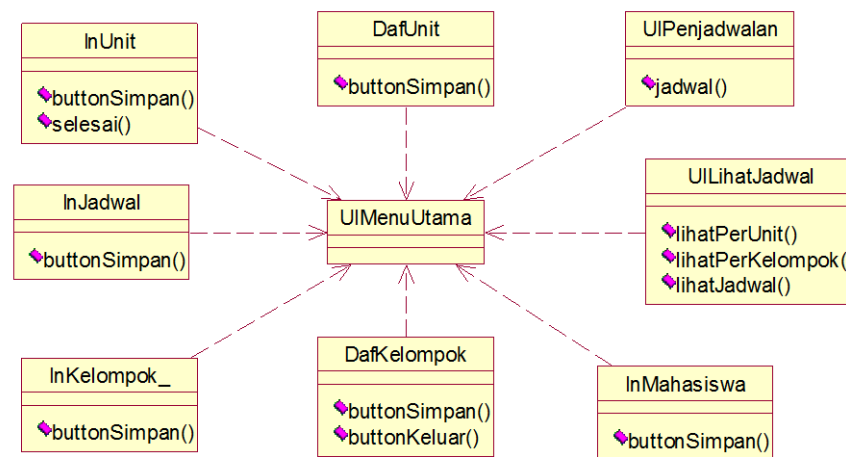
Gambar 11 Diagram Kelas

Untuk menangani informasi yang mungkin disimpan secara permanen digunakan kelas entitas, misalnya kelas Unit, kelas Kelompok dan sebagainya. Selain itu kelas entitas juga digunakan untuk membangun struktur basis data. Rancangan basis data dapat ditelusuri dari atribut datau field terhadap kebutuhan sistem.

Kebutuhan sistem menentukan aliran kejadian (*flow if event*) dan aliran kejadian menentukan obyek-obyek kelas-kelas dan atribut-atribut dalam kelas. Masing-masing atribut dalam kelas entitas mungkin akan menjadi field atau atribut dalam basis data. Selain itu juga terdapat relasi yang menggambarkan jumlah instand yang mungkin dapat suatu kelas. Multipicity yaitu memberikan gambaran sejumlah instan

yang akan ditampung dalam kelas. Dalam kelas Unit memungkinkan beberapa instan pada kelas DaftarUnit.

Gambar 12. menunjukan class diagram untuk stereotype boundary. kelas boundary adalah apa saja yang memungkinkan aktor berinteraksi dengan sistem. Pada diagram kelas juga ditentukan pilihan visibilitas yaitu menentukan dapat tidaknya sebuah kelas diliha dari luar paket. Ada empat pilihan visibilitas untuk sebuah kelas yaitu public, private, protected dan package. *Boundary Class* memungkinkan aktor untuk berinteraksi dengan sistem/perangkat lunak yang dikembangkan. Hasil input data disimpan dalam kelas entitas Jadwal.



Gambar 12. Class Diagram Stereotype Boundary

Boundary Class memungkinkan aktor untuk berinteraksi dengan sistem/perangkat lunak yang dikembangkan.

Perancangan Boundary Class

Boundary Class memungkinkan aktor untuk berinteraksi dengan sistem/perangkat lunak yang dikembangkan. Gambar 13 sampai dengan Gambar 19. Menunjukkan rancangan prototype untuk *Boundary Class* pada sistem .

Gambar 13 User Interface untuk Input data Unit

Gambar 13 menunjukkan user interface untuk input data unit, merupakan kelas boundary dengan nama kelas InUnit.

Gambar 14 User Interface untuk Input data Kelompok

Gambar 14 menunjukkan user interface untuk input data Kelompok, merupakan kelas boundary dengan nama kelas InKelompok. Hasil input data disimpan dalam kelas entitas Kelompok.

Gambar 15 User Interface untuk Input data Jadwal

Gambar 15 menunjukkan user interface untuk input data Jadwal, merupakan

kelas boundary dengan nama kelas InJadwal. Hasil input data disimpan dalam kelas entitas Jadwal.

Gambar 16 User Interface untuk Input Pendaftaran Mahasiswa

Gambar 16 menunjukkan user interface untuk pendaftaran mahasiswa, merupakan kelas boundary dengan nama kelas DaftarMhs. Hasil input data disimpan dalam kelas entitas Mahasiswa.

Gambar 17 User Interface untuk Penjadwal Koas

Gambar 17 menunjukkan user interface untuk input data Jadwal, merupakan kelas boundary dengan nama kelas Penjadwalan. Hasil proses penjadwalan data disimpan dalam kelas entitas DetailJadwal.

Gambar 18 User Interface untuk Lihat Jadwal Per Unit

Gambar 18 menunjukkan user interface untuk Output tabel jadwal per Unit, merupakan kelas boundary dengan nama kelas LihatPerUnit.

Gambar 19 User Interface untuk Lihat Jadwal Per Kelompok

Gambar 19 menunjukkan user interface untuk Output tabel jadwal per Kelompok, merupakan kelas boundary dengan nama kelas LihatPerKelompok.

D. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari analisis dan design sistem penjadwal Koas adalah sebagai berikut :

1. Analisis dan design sistem penjadwalan koas dilakukan dengan pendekatan obyek oriented menggunakan notasi pemodelan Unified Modeling Language (UML).
2. Diagram yang digunakan dalam pemodelan meliputi use case diagram berikut diskripsi use case, activity diagram, sequen diagram dan class diagram
3. Basis data yang dibutuhkan dalam ditunjukkan pada class diagram untuk stereotype entitas yaitu sebanyak 7 model.
4. Bahasa permodelan visual yang digunakan UML yang bekerja

dalam Object Oriented Analysis and Design dengan tool pemodelan visual berupa Rasional Rose yang mendukung round-trip engineering yang dapat men-generate model kedalam kode (visual basic, java, C++ dan sebagainya) dan melakukan reverse engineering untuk menampilkan arsitektur software dari kode yang ada.

5.2. Saran

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu pertama hasil analisis dan design pada tahap selanjutnya akan dikembangkan perangkat lunak Sistem Informasi Penjadwalan koas. Kedua proses penjadwalan koas dapat dioptimalkan dengan pemodelan matematika programming yaitu dalam bentuk penyelesaian masalah *integer linear programming*, sehingga jadwal yang dihasilkan dapat lebih memberikan kepuasan untuk berbagai pihak.

E. DAFTAR PUSTAKA

- Booch Grady, et all, 2005, "The Unified Modeling Language User Guide SECOND EDITION".
- Suruali, N dan Paillin, D.B., Analisis Dan Desain Sistim Informasi Perpustakaan Migrasi Ke Digital Library, ARIKA, Vol. 04, No. 2, Agustus 2010,
- Windarti, I, Banowosari, L.Y., 2006, Sistem Informasi Bidang Kemahasiswaan Dengan Metode Berorientasi Obyek Menggunakan Unified Modeling Language (UML) Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2006) Auditorium Universitas Gunadarma, Depok, 23-24 Agustus 2006 ISSN : 1411-6286
- Belien,J., 2006, Exact and Heuristic Methologies for Scheduling in Hospitals: Problem, Formulation and Algorithms, Tesis, Faculteit Economische En Toegepaste Economische Wetenschappen, Katholieke Universiteit Leuven
- Amar, S., Dharma, I.G.B.B., 2011, Trainee Scheduling At Hospital: A Paper Review, SeNTI-UGM 2011:Synergy For Sustainability, B-071,
- Rumbaugh James, et all, 1999, "The Unified Modeling Language Reference Manual".

