

PANDUAN PRAKTIKUM ANALISIS DESAIN SISTEM



Disusun oleh :

Deborah Kurniawati, S.Kom., M.Cs

Sari Iswanti, S.Si., M.Kom

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

AKAKOM

YOGYAKARTA

2019

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan atas perkenan-Nya, kami dapat menyelesaikan panduam praktikum Analisis Desain Sistem. Panduan ini secara garis besar terbagi menjadi 2 (dua) yaitu modul 1-7 berisi materi untuk melakukan analisis dan desain dengan pendekatan terstruktur dan modul 8–14 adalah materi analisis dan desain dengan pendekatan pendekatan berorientasi obyek.

Dalam panduan ini, untuk memodelkan sistem dengan pendekatan terstruktur menggunakan DAD (Diagram Arus Data) dan pendekatan berorientasi obyek menggunakan UML. Perlu diingat bahwa penekanan pada panduan praktikum ini bukan pada penggunaan tools yang mendukung pembuatan model tetapi lebih pada bagaimana memahami dan melakukan proses analisis dan desain serta memodelkannya baik dengan pendekatan terstruktur maupun berorientasi obyek.

Kami mengucapkan terima kasih kepada Ibu Febri Nova Lenti, Bapak Cuk Subiyantoro, Ibu Sri Redjeki, Ibu Endang Wahyuningsih, dan Ibu Pulut Suryati untuk materi-materi dokumen sistemnya. Kami menyadari masih banyak kekurangan dalam panduan paraktikum ini. Masukan dan saran yang bersifat membangun dan menyempurnakan panduan ini sangat Kami harapkan.

Penyusun,

-Debbie & Sari-

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
MODUL 1 SISTEM	1
MODUL 2 ANALIS SISTEM.....	6
MODUL 3 DIAGRAM ARUS DATA, DIAGRAM KONTEKS.....	10
MODUL 4 DIAGRAM ARUS DATA LEVEL 1	15
MODUL 5 DIAGRAM ARUS DATA LEVEL 2.....	18
MODUL 6 STUDI KASUS : ANALISIS SISTEM	22
MODUL 7 STUDI KASUS : PEMODELAN SISTEM	24
MODUL 8 DASAR UNIFIED MODELING LANGUAGE (UML)	26
MODUL 9 USE CASE DIAGRAM	35
MODUL 10 CLASS DIAGRAM.....	43
MODUL 11 SEQUENCE DIAGRAM	47
MODUL 12 ACTIVITY DIAGRAM.....	50
MODUL 13 INTERFACE PROTOTYPE.....	54
MODUL 14 DOKUMENTASI DAN PRESENTASI.....	57

MODUL 1

SISTEM



CAPAIAN PEMBELAJARAN

Mahasiswa mampu :

1. menjelaskan sebuah sistem sesuai klasifikasi sistem
2. Menyebutkan komponen-komponen sebuah sistem



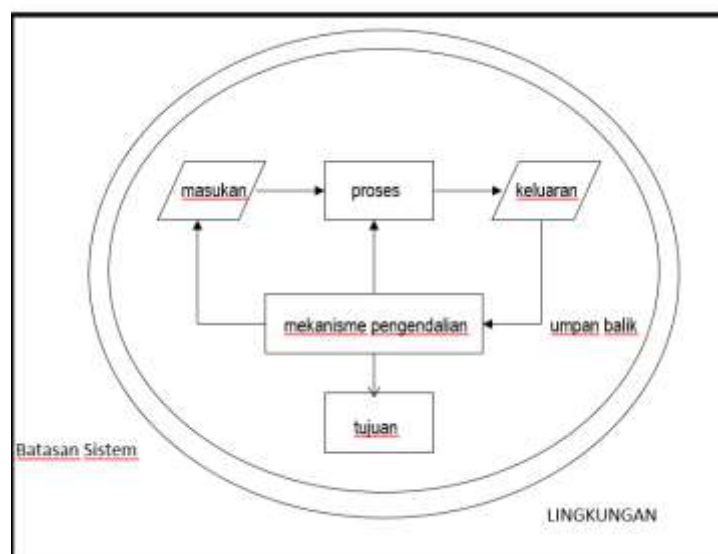
KEBUTUHAN ALAT/BAHAN/SOFTWARE

Tools pengolah kata lengkap



DASAR TEORI

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu untuk mencapai tujuan tertentu. Elemen sistem dapat digambarkan sebagai berikut :



Penjelasan untuk masing-masing elemen :

1. Tujuan : segala sesuatu yang akan diraih atau dituju oleh sebuah sistem
2. Masukan : Segala sesuatu yang masuk ke dalam sistem. Sesuatu yang masuk ke dalam sistem bisa berwujud (bahan mentah) maupun tidak berwujud (data/informasi)
3. Proses : mentransformasikan masukan menjadi keluaran
4. Ouput : hasil pemrosesan (bisa berupa : informasi, saran, laporan, produk tertentu)
5. Mekanisme : mengendalikan masukan dan proses sehingga pengendalian dan sistem dapat berjalan sesuai tujuan umpan balik
6. Batasan (*boundary*) : pemisah antara sistem dengan luar sistem (lingkungan) atau membatasi sub sistem dengan sub sistem yang lain
7. Lingkungan : segala sesuatu yang berada di luar sistem yang (*environment*) mempengaruhi kerja sistem

Sistem dapat diklasifikasikan sebagai berikut :

1. sistem abstrak dan sistem fisik
2. sistem deterministik dan sistem probabilistik
3. sistem tertutup dan sistem terbuka
4. sistem alami dan sistem buatan manusia
5. sistem sederhana dan sistem kompleks

Sistem abstrak : sistem yang berisi gagasan atau konsep tertentu. contoh : sistem teologi (hubungan manusia dan tuhan), ideologi tertentu. Sedangkan **sistem fisik** : tampak secara fisik. **Sistem deterministik** adalah sistem dimana operasi dan hasilnya dapat diprediksi secara tepat, sedangkan **sistem probabilistik** adalah sebuah sistem yang tidak dapat diprediksi secara tepat karena mengandung unsur probabilistik (contoh : sistem persediaan barang, arisan).

Sistem tertutup (relatif tertutup):_ sistem yang tidak menerima Pengaruh dari lingkungan luar memiliki masukan dan keluaran tertentu

Sistem terbuka_: sistem yang dapat menerima pengaruh dari lingkungan luar dan dapat beradaptasi terhadap lingkungan

Sistem alami adalah sebuah sistem yang sudah ada secara alamiah, sedangkan **sistem buatan manusia** adalah sistem yang sengaja dibuat oleh manusia.

Sebuah sistem dapat masuk dalam kategori lebih dari 1(satu) klasifikasi, misalnya Sistem KRS online di STMIK Akakom bisa dikategorikan sebagai sistem terbuka dan sistem buatan manusia.



PRAKTIK

STMIK Akakom merupakan sebuah sistem. Dari sisi klasifikasi termasuk dalam Sistem fisik, deterministik, terbuka, dan sistem buatan manusia.

Penjelasan berdasarkan elemen sistemnya :

NO	ELEMEN	KETERANGAN
1	Masukan	Calon mahasiswa
2	Proses	Perkuliahan
3	Keluaran (output)	Iulusan
4	Tujuan	Menghasilkan lulusan yang berkualitas sesuai kebutuhan pasar dan memiliki etika yang baik.
5	Mekanisme Pengendalian	Mengendalikan input (syarat untuk mendaftar) : lulusan SLTA/SMK dengan rata-rata nilai ujian 6,5 dari jurusan tertentu, dan lain-lain Mengendalikan proses : mata kuliah diajar oleh dosen yang sesuai kompetensi ada syarat tertentu untuk dapat UAS dan lain-lain
6	Umpan Balik	Saran/masukan untuk perbaikan proses sehingga tujuan tercapai : Masukan dari mahasiswa (kuisisioner) Masukan dari dosen, pegawai

NO	ELEMEN	KETERANGAN
		Masukan dari lulusan Masukan dari orang tua/wali mahasiswa Masukan dari industri, masyarakat

- *Boundary System* (batasan sistem) : - perguruan tinggi yang lain
- Lingkungan (*environment*) : pemerintah, pihak industri, masyarakat

Penjelasan berdasarkan klasifikasi sistem :

NO	KLASIFIKASI	KETERANGAN
1	Sistem fisik	STMIK Akakom memiliki gedung/bangunan, dosen, mahasiswa, tenaga kependidikan, dan sarana prasarana lain yang terlihat secara fisik
2	Sistem Terbuka	Dapat menerima pengaruh/masukan dari luar, misalnya regulasi pemerintah, masukan dari mahasiswa, alumni, industri
3	Sistem Buatan manusia	Sistem yang sengaja diciptakan oleh manusia (pendiri) dalam bidang pendidikan pada tahun 1979.



LATIHAN

1. Berikan penjelasan mengapa STMIK Akakom dapat diklasifikasikan sebagai sistem yang deterministik ?
2. Dalam Elemen sistem 'STMIK Akakom', apakah dosen dapat dimasukkan ke dalam elemen sistem ? Beri penjelasan !



TUGAS

Cermati Sistem KRS Online yang ada di STMIK Akakom, Buatlah :

1. Penjelasan apabila dilihat dari sudut pandang elemen sistemnya !
2. Penjelasan apabila dilihat dari sudut pandang klasifikasi sistem !



REFERENSI

- Analisis dan Desain : Pendekatan Terstruktur, Jogiyanto
- Semua pustaka yang relevan

MODUL 2

ANALISIS SISTEM



CAPAIAN PEMBELAJARAN

Mahasiswa mampu :

1. menjelaskan proses bisnis yang ada dalam sebuah sistem
2. Membuat analisis sistem sesuai dengan langkah-langkah yang ditetapkan



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Tools pengolah kata lengkap



DASAR TEORI

Analisis Sistem adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan - permasalahan, kesempatan- kesempatan, hambatan- hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan- perbaikannya “. Hasil dari suatu analisis sistem adalah kebutuhan sistem yang akan digunakan untuk melakukan desain sistem.

Langkah dalam analisis sistem :

1. Studi kelayakan, meliputi :
 - a. Penentuan masalah
 - b. Pembentukan sasaran sistem
 - c. Pengidentifikasian pemakai sistem
 - d. Pembentukan lingkup sistem

2. Analisis kebutuhan :

- a. Masukan yang diperlukan sistem
- b. Keluaran yang dibutuhkan
- c. Lingkup proses
- d. Volume data yang ditangani sistem
- e. Kategori pemakai sistem
- f. Kontrol sistem



PRAKTIK

Diberikan deskripsi sebagai berikut :

Sebuah usaha tour “CAHAYA TOUR” yang masih sederhana selalu kewalahan melayani konsumen yang bertanya mengenai info wisata yang bisa dikunjungi dan biayanya. Pemilik “CAHAYA TOUR” sedang berpikir bagaimana memanfaatkan teknologi informasi untuk mendukung usaha tour yang dimiliki sehingga memudahkan para konsumen mendapatkan informasi dan meningkatkan layanan “CAHAYA TOUR” terhadap para konsumen.

Lakukan **analisis sistem** untuk kasus di atas !

LANGKAH 1 : STUDI KELAYAKAN

NO	LANGKAH	KETERANGAN
1	Penentuan masalah	<ul style="list-style-type: none">• Bagaimana memanfaatkan TI untuk membantu mengelola usaha tour
2	Sasaran Sistem	<ul style="list-style-type: none">• Sistem dapat membantu konsumen dalam mencari informasi wisata yang diinginkan• Sistem dapat membantu “CAHAYA TOUR” dalam menyajikan informasi secara cepat dan tepat kepada konsumen
3	Identifikasi pemakai sistem	<ul style="list-style-type: none">• Konsumen• Pemilik usaha tour
4	Lingkup Sistem	<ul style="list-style-type: none">• Dibuat SI berbasis web• Tujuan wisata : DIY dan Jawa Tengah• Menyediakan informasi paket wisata :<ul style="list-style-type: none">- Tujuan wisata + transport- Jemput & antar wisatawan ke bandara/stasiun• Sistem tidak menyediakan fasilitas pembayaran hanya informasi mengenai tata cara pembayaran bagi konsumen



LATIHAN

Kita diminta untuk merancang sistem informasi/aplikasi yang dapat digunakan untuk menyelesaikan masalah diatas.

3. Buatlah deskripsi lengkap mengenai aplikasi yang akan dibangun
4. Tuliskan langkah 2 dari analisis sistem yaitu analisis kebutuhan yang meliputi :
 - a. Masukan yang diperlukan sistem
 - b. Keluaran yang dibutuhkan
 - c. Lingkup proses
 - d. Volume data yang ditangani sistem
 - e. Kategori pemakai sistem
 - f. Kontrol sistem



TUGAS

Akan dibangun sebuah sistem informasi yang dapat membantu pengelolaan pendaftaran asisten laboratorium. Sistem ini direncanakan berbasis web. Asisten praktikum di lab. dibutuhkan untuk membantu dosen dalam pelaksanaan kegiatan praktikum.

Mekanismenya :

Dalam pelaksanaannya, sistem ini akan melibatkan secara langsung calon asisten dan administrasi lab. Hal-hal yang harus dilakukan oleh calon asisten dan administrasi lab berbeda. Secara rinci mekanismenya sebagai berikut :

1. calon asisten :
 - Mengisi form pendaftaran menjadi asisten lab (download)
 - Mengumpulkan form pendaftaran yg sdh diisi disertai

- a. Foto 4x6
 - b. transkrip nilai
 - Calon asisten menunggu pengumuman penerimaan menjadi asisten
2. Administrasi laboratorium :
- Memasukkan data praktikum yang membutuhkan asisten beserta jadwalnya
 - Menyeleksi asisten yang mendaftar
 - Mengumumkan calon asisten yang diterima sebagai asisten

Sistem yang akan dibuat nantinya :

- a. Menyediakan form pendaftaran asisten di web dan dapat di unduh
- b. Menampilkan daftar mata praktikum beserta jadwalnya yang membutuhkan praktikum
- c. Menampilkan pengumuman calon asisten yang diterima menjadi asisten lab.

Dari deskripsi yang ada, buatlah **analisis kebutuhannya**.

Praktikan boleh menambahkan asumsi yang dianggap perlu.



REFERENSI

- Analisis dan Desain : Pendekatan Terstruktur, Jogiyanto
- Semua pustaka yang relevan

MODUL 3

DIAGRAM ARUS DATA : DIAGRAM KONTEKS



CAPAIAN PEMBELAJARAN

Mahasiswa mampu membuat pemodelan sistem dengan menggunakan DAD (Diagram Arus Data)



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Tools pengolah kata lengkap
2. Tools penunjang untuk membuat DAD misalnya DIA



DASAR TEORI

Salah satu cara untuk membuat pemodelan sistem dengan pendekatan terstruktur yang menekankan pada proses-proses yang ada dalam sistem dan menggambarkan aliran data dalam sistem adalah dengan menggunakan DAD (Diagram Arus Data). Dalam DAD terdapat beberapa hal yaitu

- a) proses dpt beroperasi secara paralel
- b) proses looping (perulangan) & branching (percabangan) tidak ditunjukkan secara khusus
- c) tidak menunjukkan urutan proses.

Penggambaran notasi dalam DAD :

MENURUT	NOTASI	PENYIMPAN	EKSTERNAL ENTITAS
ANE & SARSON			
De MARCO/ YOURDON			

❖ ENTITAS LUAR

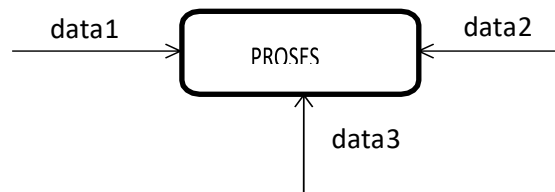
- Merupakan sumber atau tujuan data
- Bagian yang mengirimkan data atau menerima data/informasi dari sistem
- Bisa berupa bagian lain sistem, sebuah perusahaan, orang, atau mesin
- Harus diberi nama yang relevan

❖ PROSES

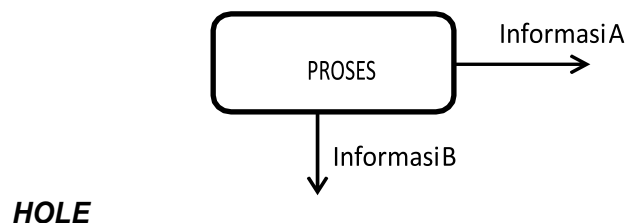
- Data mengalir menuju atau keluar dari “PROSES”



- ADA data masuk HARUS ada data keluar, TIDAK BOLEH ada **BLACKHOLE**



- ADA data KELUAR HARUS ada DATA YG MASUK, TIDAK BOLEH ada **GRAY**

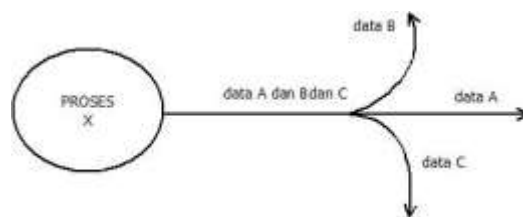


❖ ALIRAN DATA

- Data yang mengalir harus diberi nama

- Data yang mengalir dimungkinkan menyebar (*diverging*) atau mengumpul (*konverging*)
- Data yang menyebar atau mengumpul dapat dihubungkan dengan operator OR atau AND

Contoh aliran data menyebar :



❖ SIMPANAN DATA (*DATA STORE*)

- Tempat untuk menyimpan data/informasi dari sebuah proses
- Tempat untuk mengambil data untuk kemudian di proses oleh suatu pemroses
- Simpanan data **tidak boleh** langsung berhubungan dengan entitas
- Sebaiknya diberi nama sesuai dengan nama data yang disimpan

Dalam pembuatan DAD untuk menggambarkan proses-proses yang ada dalam sistem dibagi menjadi 2 :

1. Diagram Konteks
2. DAD level berikutnya

Diagram konteks :

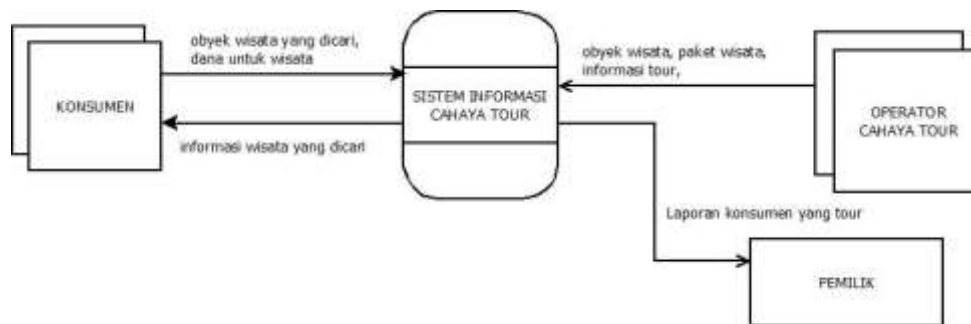
- Tingkatan tertinggi dalam DAD hanya menunjukkan 1 proses yang dapat menggambarkan keseluruhan sistem
- Memuat entitas, data yang mengalir, dan sebuah proses (tidak memuat penyimpanan/data store)

Proses-proses utama yang merupakan penjabaran dari diagram konteks akan ditunjukkan pada DAD level dibawahnya (ada yang menyebut level 0).



PRAKTIK

Dari kasus di modul sebelumnya yaitu SICATUR (Sistem Informasi Cahaya Tour), maka dapat digambarkan Diagram Konteksnya sebagai berikut :



Dari Diagram Konteks yang sudah dibuat terlihat bahwa Sistem Informasi Cahaya Tour, memiliki 3 entitas luar yaitu konsumen, pemilik tour, dan operator Cahaya Tour. Arus data yang masuk ke dalam sistem :

1. Obyek wisata yang dicari dan dana untuk wisata → dari entitas KONSUMEN
2. Obyek wisata, paket wisata, dan informasi tour → dari entitas OPERATOR

Arus data yang keluar dari sistem juga ada 2 :

1. Informasi wisata yang dicari → menuju entitas KONSUMEN
2. Laporan konsumen yang tour → menuju entitas PEMILIK

Arus data dalam Diagram konteks yang menuju sistem maupun keluar dari sistem harus sama dan konsisten dengan arus data yang ada di DAD level .



LATIHAN

1. Buatlah diagram konteks di atas dengan menggunakan *tools* yang mendukung pembuatan DAD !
2. Tentukan proses-proses utama yang dapat dijabarkan dari diagram

konteks Sistem Informasi CAHAYA TOUR !

3. Lakukan identifikasi arus data yang menuju dan keluar dari proses utama yang ditetapkan pada soal nomor 2 !



TUGAS

Buatlah diagram konteks untuk kasus perancangan Sistem Informasi Pengelolaan Pendaftaran Asisten Laboratorium (TUGAS di pertemuan 2) !



REFERENSI

- Analisis dan Desain Sistem : Pendekatan Terstruktur dan praktis, Jogiyanto
- Metode Desain dan Analisis Sistem, Jeffery L.Whitten &Lonnie D.Bentley
- Semua pustaka yang relevan

MODUL 4

DIAGRAM ARUS DATA LEVEL 1



CAPAIAN PEMBELAJARAN

Mahasiswa mampu menguraikan diagram konteks ke dalam DAD level 1



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Tools pengolah kata lengkap
2. Tools penunjang untuk membuat DAD misalnya DIA



DASAR TEORI

Diagram konteks untuk Sistem Informasi Cahaya Tour (SICATUR) yang sudah dibahas pada pertemuan sebelumnya sebagai berikut :

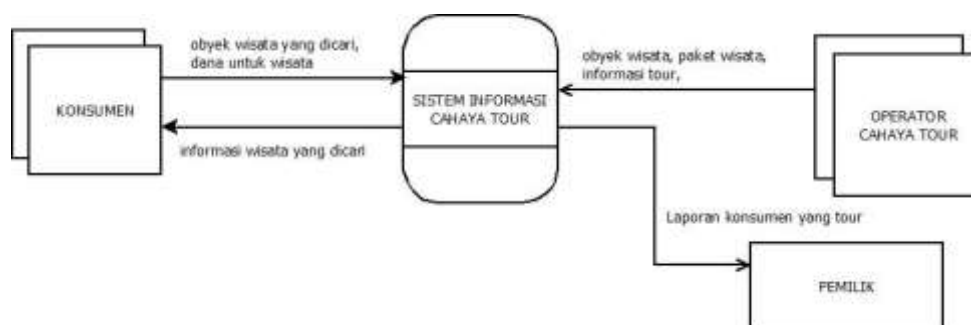


Diagram konteks belum menggambarkan proses-proses utama yang ada dalam sistem. Untuk mengetahui proses-proses utama yang ada dalam SICATUR beserta data yang mengalir dan menyimpan datanya, perlu dibuat DAD level 1.

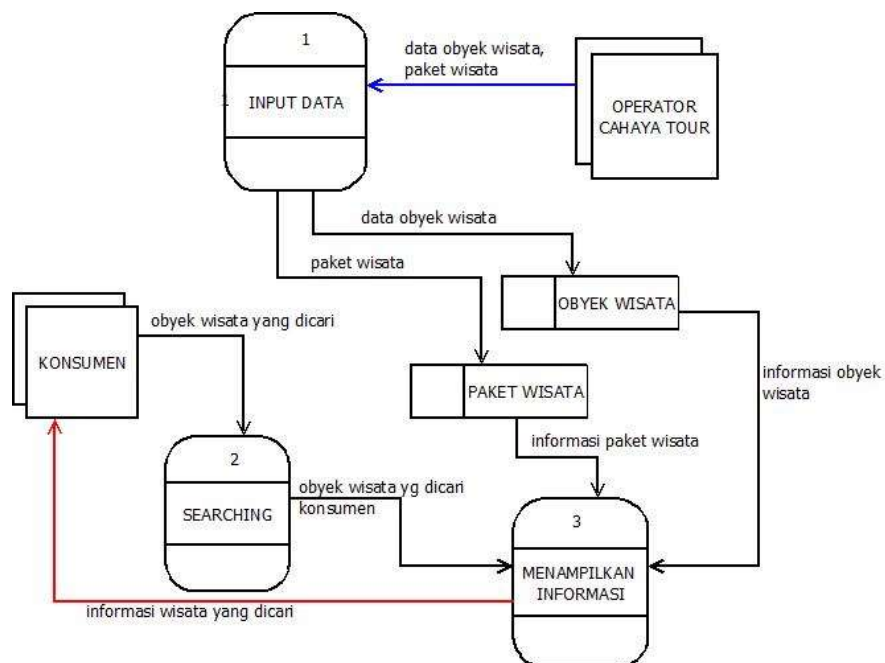
Proses utama yang ada dalam SICATUR, antara lain :

1. Pemasukan Data oleh Operator Cahaya Tour
2. *Searching* Data wisata yang dicari oleh konsumen
3. Proses menampilkan informasi yang dicari oleh konsumen



PRAKTIK

DAD level 1 dari diagram konteks SICATUR yang memuat 3 proses utama, sebagai berikut :



Gambarlah DAD level 1 tersebut menggunakan *tools*.



LATIHAN

1. Lengkapilah DAD level 1 di atas apabila :
 - a. konsumen dapat memasukkan dana wisata yang dimiliki untuk mendapatkan paket wisata sesuai dengan dana yang dimiliki
 - b. konsumen dapat memilih paket wisata

2. Gambarkan DAD nya secara lengkap, sehingga pemilik CAHAYA TOUR dapat menerima laporan sesuai dengan yang digambarkan dalam diagram konteks
(lengkapilah dengan menambahkan proses dan arus data yang dianggap perlu)



TUGAS

Gambarkan DAD level 1 untuk sistem pengelolaan pendaftaran asisten laboratorium (mengacu tugas pertemuan 3)



REFERENSI

- Analisis dan Desain Sistem : Pendekatan Terstruktur dan praktis, Jogiyanto
- Metode Desain dan Analisis Sistem, Jeffery L.Whitten &Lonnie D.Bentley
- Semua pustaka yang relevan

MODUL 5

DIAGRAM ARUS DATA LEVEL 2



CAPAIAN PEMBELAJARAN

Mahasiswa mampu menjabarkan DAD level 2 ke dalam level yang berikutnya untuk menunjukkan proses yang lebih rinci.



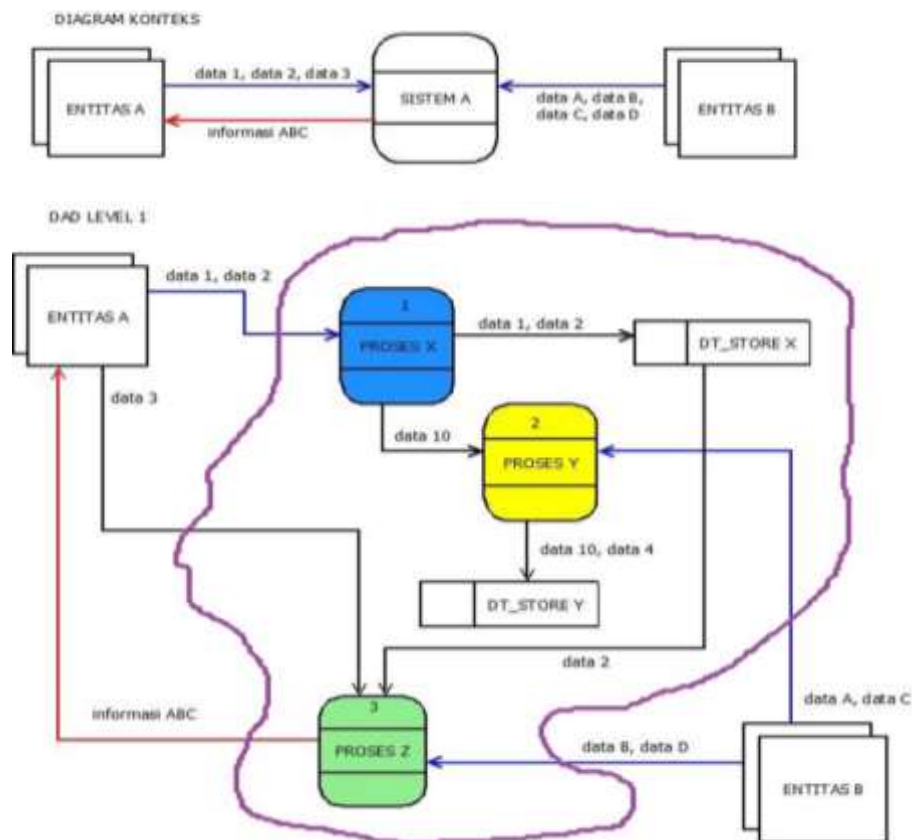
KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Tools pengolah kata lengkap
2. Tools penunjang untuk membuat DAD misalnya DIA



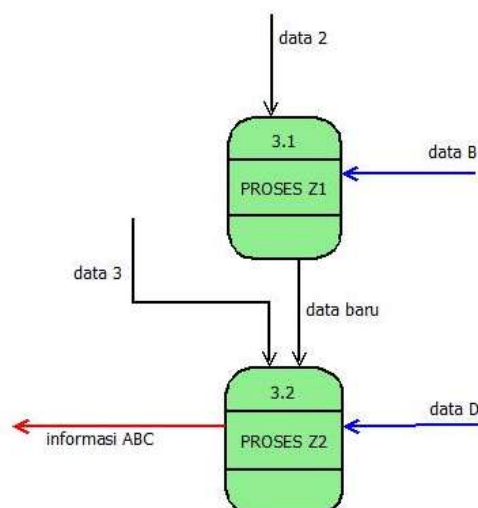
DASAR TEORI

Suatu DAD yang sudah terbentuk, prosesnya dapat diuraikan menjadi lebih rinci apabila diperlukan. Prinsipnya sama dengan penguraian diagram konteks ke dalam DAD level 1 untuk melihat proses-proses utama yang ada dalam sebuah sistem yang sedang dibuat rancangan DAD nya. Ilustrasinya dapat dilihat pada Gambar 5.1



Gambar 5.1 ilustrasi penguraian diagram konteks ke dalam DAD level 1

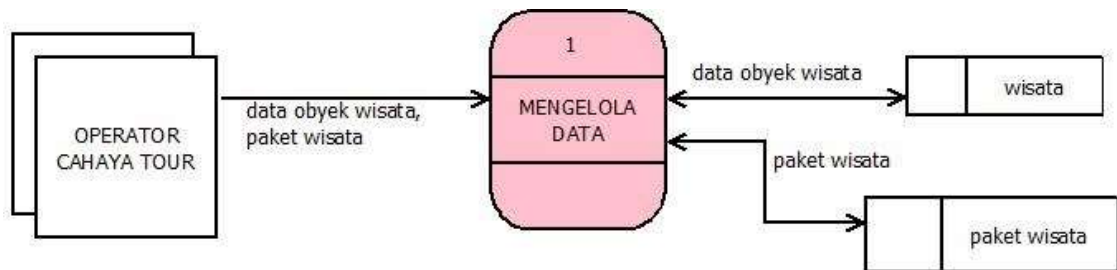
Mengacu gambar 5.1, apabila sebuah proses akan dijabarkan menjadi proses yang lebih rinci maka akan menjadi DAD level 2 proses n (n adalah proses yang dijabarkan). Misalnya proses 3 akan dijabarkan, maka namanya menjadi DAD level 2 proses 3. Hasilnya sebagai berikut :



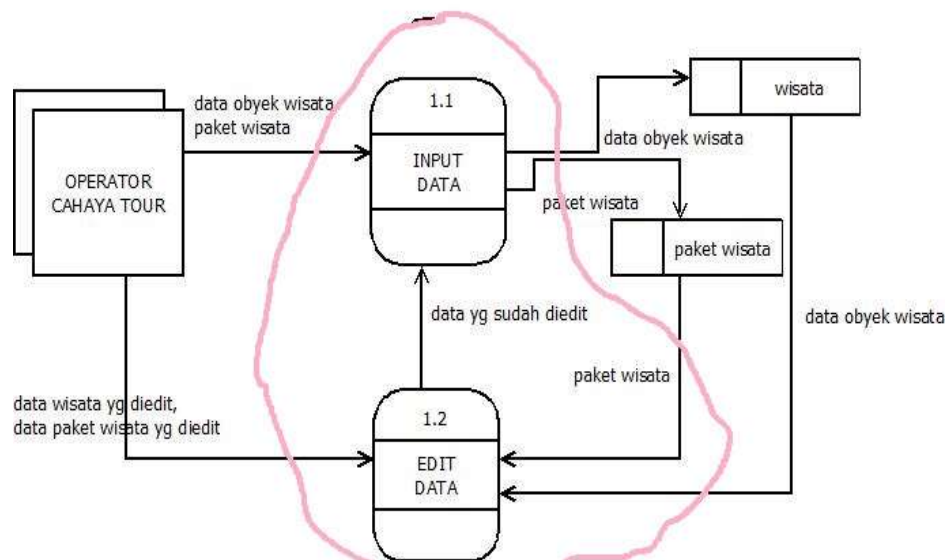


PRAKTIK

Mengacu dari kasus Sistem Informasi Cahaya Tour (SICATUR), diberikan sebuah proses 1 : MENGELOLA DATA yang dilakukan oleh entitas luar operator cahaya tour. Proses mengelola data yang dimaksud adalah proses melakukan pemasukan data dan edit data dengan catatan bahwa data yang diedit akan disimpan kembali ke penyimpanan data yang sama dengan data awal sebelum diedit. Proses Mengelola Data :



DAD Level 1 Proses 1 nya menjadi :



LATIHAN

1. Buatlah menggunakan *tools* DAD level 2 proses 1 di atas

2. Lengkapilah DAD yang sudah Saudara buat pada pertemuan-pertemuan sebelumnya dengan menambahkan **proses edit data**
3. Cermati DAD yang sudah dibuat sebelumnya, buatlah DAD level 2 - nya.

catatan :

- a. tidak semua proses harus dibuat rinci, sesuai kebutuhan saja
- b. Bisa dilakukan perubahan di DAD level 1 apabila dianggap perlu



TUGAS

Gambarkan DAD yang lengkap untuk sistem pengelolaan pendaftaran asisten laboratorium sampai pada DAD LEVEL 2.



REFERENSI

- Analisis dan Desain Sistem : Pendekatan Terstruktur dan praktis, Jogiyanto
- Metode Desain dan Analisis Sistem, Jeffery L.Whitten &Lonnie D.Bentley
- Semua pustaka yang relevan

MODUL 6

STUDI KASUS : ANALISIS SISTEM



CAPAIAN PEMBELAJARAN

Mahasiswa mampu : melakukan analisis sistem dari sebuah sistem yang proses bisnisnya sudah ditentukan.



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Tools pengolah kata lengkap



DESKRIPSI

Sebuah kampus merasa kesulitan untuk mendata lulusannya yang baru saja wisuda. Dari data yang diperoleh setiap kali wisuda rata-rata jumlah yang diwisuda 200 wisudawan. Dalam 1 tahun terdapat 2x wisuda. Daftar lulusan tentu saja tersedia dalam bentuk buku wisuda. Kesulitan muncul saat pihak kampus ingin menghubungi lulusan tersebut untuk memberitahukan ada permintaan pegawai dari suatu perusahaan dan menginginkan segera dilakukan tes. Terkadang jarak antara permintaan dengan waktu tes hanya berkisar 7 hari. Pendataan lulusan apakah sudah bekerja atau belum dan posisi saat ini berada di mana bukan menjadi persoalan yang sederhana. Dihubungi dengan telpon bisa saja dilakukan tetapi tidak akan efisien menghubungi lulusan yang jumlahnya ratusan.

Keinginan pihak kampus tentu saja adalah bagaimana data/informasi segera diperoleh jika dibutuhkan. Misalnya bagaimana pihak kampus segera mendapatkan data siapa saja yang belum bekerja sehingga apabila ada permintaan pegawai dari suatu perusahaan segera dapat dihubungi.



PRAKTIK

Dari kasus yang deskripsinya sudah disampaikan di atas, **lakukan proses Analisis Sistem berupa :**

1. studi kelayakan, meliputi :
 - g. Penentuan masalah
 - h. Pembentukan sasaran sistem
 - i. Pengidentifikasian pemakai sistem
 - j. Pembentukan lingkup sistem
2. analisis kebutuhan :
 - k. Masukan yang diperlukan sistem
 - l. Keluaran yang dibutuhkan
 - m. Lingkup proses
 - n. Volume data yang ditangani sistem
 - o. Kategori pemakai sistem
 - p. Kontrol sistem



TUGAS

-



REFERENSI

- Analisis dan Desain Sistem : Pendekatan Terstruktur dan praktis, Jogiyanto
- Metode Desain dan Analisis Sistem, Jeffery L.Whitten &Lonnie D.Bentley
- Semua pustaka yang relevan

MODUL 7

STUDI KASUS : PEMODELAN SISTEM



CAPAIAN PEMBELAJARAN

Mahasiswa mampu membuat pemodelan sistem dari sebuah sistem yang proses bisnisnya sudah ditentukan menggunakan DAD dengan mengacu hasil analisis sistem



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Tools pengolah kata lengkap
2. Tools penunjang untuk membuat DAD misalnya DIA



DESKRIPSI (SAMA DENGAN PERTEMUAN 6)

Sebuah kampus merasa kesulitan untuk mendata lulusannya yang baru saja wisuda. Dari data yang diperoleh setiap kali wisuda rata-rata jumlah yang diwiuda 200 wisudawan. Dalam 1 tahun terdapat 2x wisuda. Daftar lulusan tentu saja tersedia dalam bentuk buku wisuda. Kesulitan muncul saat pihak kampus ingin menghubungi lulusan tersebut untuk memberitahukan ada permintaan pegawai dari suatu perusahaan dan menginginkan segera dilakukan tes. Terkadang jarak antara permintaan dengan waktu tes hanya berkisar 7 hari. Pendataan lulusan apakah sudah bekerja atau belum dan posisi saat ini berada di mana bukan menjadi persoalan yang sederhana. Dihubungi dengan telpon bisa saja dilakukan tetapi tidak akan efisien menghubungi lulusan yang jumlahnya ratusan.

Keinginan pihak kampus tentu saja adalah bagaimana data/informasi segera diperoleh jika dibutuhkan. Misalnya bagaimana pihak kampus segera mendapatkan data siapa saja yang belum bekerja sehingga apabila ada permintaan pegawai dari suatu perusahaan segera dapat dihubungi.



PRAKTIK

Dari kasus yang deskripsinya sudah disampaikan di atas dan mengacu hasil analisis sistem yang sudah dilakukan pada pertemuan 6, **buatlah** :

1. Diagram Konteks
2. DAD Level 1



TUGAS



REFERENSI

- Analisis dan Desain Sistem : Pendekatan Terstruktur dan praktis, Jogiyanto
- Metode Desain dan Analisis Sistem, Jeffery L.Whitten &Lonnie D.Bentley
- Semua pustaka yang relevan

MODUL 8

DASAR UNIFIED MODELING LANGUAGE (UML)



CAPAIAN PEMBELAJARAN

Mahasiswa dapat menggunakan UML untuk desain sistem



Rational rose

KEBUTUHAN ALAT/BAHAN/SOFTWARE



DASAR TEORI

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya [HAN98]. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. UML dibuat oleh *Grady Booch*, *James Rumbaugh*, dan *Ivar Jacobson* di bawah bendera *Rational Software Corp* [HAN98]. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan..

Bagian-bagian Dari UML

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism*.

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view*, *logical view*, *component view*, *concurrency view*, dan *deployment view*.

◆ Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya. *View* ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

◆ Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

◆ Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

◆ Concurrency view

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

◆ Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam

deployment diagrams dan digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

2. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

◆ Use Case Diagram

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

◆ Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

◆ State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

◆ Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

◆ Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

◆ **Activity Diagram**

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

◆ **Component Diagram**

Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa *source code*, komponent biner, atau *executable component*. Sebuah komponent berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

◆ **Deployment Diagram**

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

Tujuan Penggunaan UML

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Komponen-komponen yang terlibat dalam use case diagram :

1. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. *Actor* digambarkan dengan *stick man*

(Gambar 8.1). *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.



Gambar 8.1 Notasi UML untuk actor

2. Use Case

Use case ini merupakan bentuk fungsionalitas dari suatu sistem. Use case juga merupakan dialog antara actor dan sistem. Notasi use case pada UML dapat dilihat pada Gambar 8.2.



Gambar 8.2 Notasi UML untuk Use Case

Cara menentukan use case pada sistem:

- Pola perilaku perangkat lunak aplikasi
- Gambaran tugas dari sebuah actor
- Sistem atau benda yang memberikan sesuatu yang bernilai kepada actor
- Apa yang dikerjakan oleh suatu perangkat lunak

Beberapa relasi yang terdapat pada use case diagram :

1. Association, menghubungkan link antar elemen
2. Generalization, disebut juga inheritance(pewarisan) artinya sebuah elemen merupakan spesialisasi dari elemen lain.
3. Dependency, elemen bergantung dalam beberapa cara pada elemen lain
4. Aggregation, merupakan bentuk asosiasi, dimana elemen dapat berisi elemen lain

Tipe relasi/stereotype antara lain:

1. <<include>> yaitu perilaku yang harus terpenuhi agar sebuah event dapat terjadi, pada kondisi ini use case menjadi bagian dari use case lainnya

2. <<extends>> yaitu perilaku yang hanya berjalan pada kondisi tertentu, misal : jam alarm
3. <<communicates>> biasanya ditambahkan untuk asosiasi yang menunjukkan asosiasinya yaitu *communicates association*.

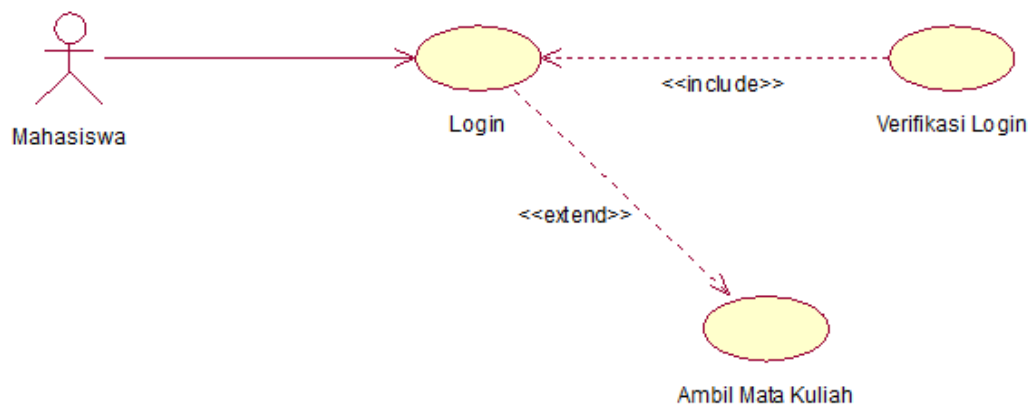


PRAKTIK

Membuat Diagram Use Case

1. Klik pada use case view package di browser
2. Pilih New → Use Case Diagram, lalu diberi nama : **Latihan**
3. Double Click pada nama Latihan (pastikan Anda sudah berada di jendela Use Case Diagram).


Pada praktik kali ini akan dibuat diagram seperti yang ada pada Gambar 8.3




Gambar 8.3 Hubungan Use Case dengan Use case lainnya.

Langkah-langkah yang dapat dilakukan adalah sebagai berikut.


Membuat actor pada Rational Rose

1. Klik gambar Actor  pada toolbar, lalu letakkan pada documentation window
2. Klik gambar Actor tersebut, dan namakan Mahasiswa.
3. Ulangi langkah 2 untuk menambah actor yang diperlukan pada diagram.



Membuat Use Case

1. Klik gambar Use Case , pada toolbar, lalu letakkan pada documentation window
2. Klik gambar Use Case tersebut, dan namakan Login
3. Ulangi langkah 2 untuk menambah use case pada diagram

Membuat Relasi

1. Klik gambar , untuk membuat relasi dependency dari actor ke use case.
2. Ulangi no. 1 untuk menambahkan dependency dari actor ke use case lainnya.

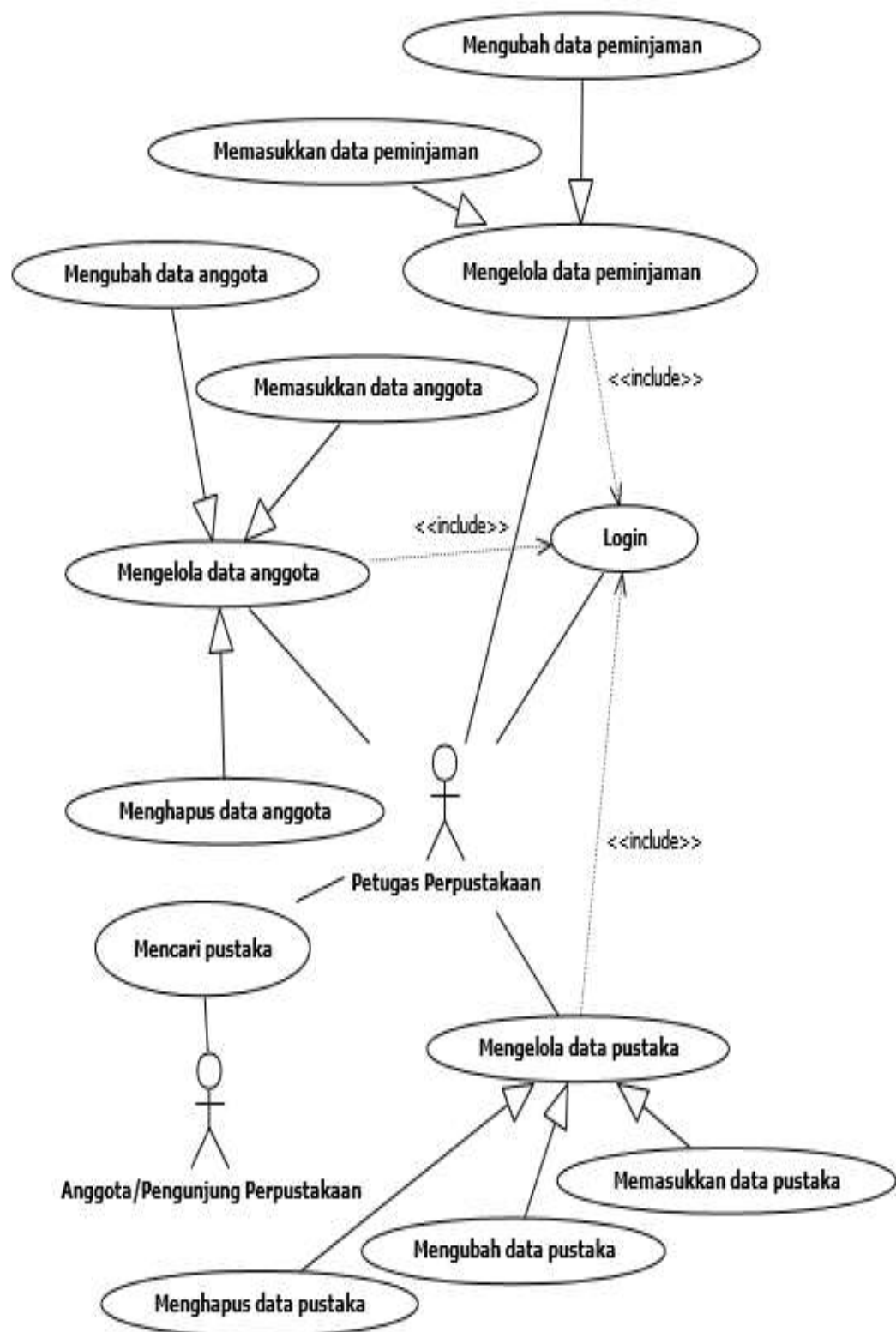
Membuat Relasi Include dan Extend

1. Klik gambar Use Case  kembali pada toolbar, lalu letakkan pada documentation window (Namakan Verifikasi Login)
2. Ulangi langkah 1 (Namakan Ambil Mata Kuliah).
3. Klik gambar , untuk membuat hubungan dari use case ke use case.
4. Ulangi no. 3 untuk menambahkan dependency dari use case ke use case lainnya.
5. Double click pada relasi dari Use Case : Verifikasi Login ke Use Case : Login, lalu pada Stereotype, pilih <<include>>.
6. Ulangi langkah 5, untuk membuat relasi <<extend>>. dari Login ke Ambil Mata Kuliah, pada Stereotype, pilih <<extend>>.



LATIHAN

Gambarkan diagram usecase berikut menggunakan tools UML





TUGAS

Diberikan oleh dosen pengampu



REFERENSI

Adi Nugroho, 2005, Rational Rose untuk Pemodelan Berorientasi Objek,
Informatika Bandung

MODUL 9

USE CASE DIAGRAM



CAPAIAN PEMBELAJARAN

Mahasiswa dapat merancang use case diagram sesuai kebutuhan sistem dan menggambarannya dengan menggunakan tools desain.



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Rational rose



DASAR TEORI

Use Case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut. Di dalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *Use Case Specification*, yang terdiri atas:

a. **Nama Use Case**

Penamaan use case sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas. Contoh : Menerima Pendaftaran Mahasiswa Baru

b. **Deskripsi singkat (*Brief Description*)**

Menjelaskan secara singkat dalam 1 atau 2 kalimat mengenai tujuan *use case*.

Contoh : - *Use Case* ini untuk mencatat data mahasiswa baru.

- *Use case* ini dimulai saat mahasiswa datang untuk mendaftarkan diri.

c. Aliran Normal (*Basic Flow*)

Aliran normal menjelaskan interaksi antara *actor* dan sistem dalam keadaan normal, yaitu segala sesuatu berjalan dengan lancar, tiada halangan dalam mencapai tujuan dari *use case* ini.

Contoh :

- Petugas mengisi data mahasiswa baru.
- Setelah selesai mengisi semua data yang diperlukan, petugas mengkonfirmasi untuk menyimpan data tersebut.
- Sistem menuliskan data mahasiswa ke dalam *database*.
- Sistem memunculkan pesan bahwa proses pendaftaran sukses dilakukan.

d. Aliran Alternatif (*Alternate Flow*)

Alternate flow menjelaskan apa yang akan dilakukan bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung, terutama berhubungan dengan *error* yang mungkin terjadi.

Contoh :

- Bila proses penulisan ke database gagal : sistem memunculkan pesan bahwa proses pendaftaran gagal dilakukan,
- Bila ada data yang diperlukan belum terisi : sistem meminta petugas untuk mengecek isian kembali

e. *Special Requirement*

Special requirement berisi kebutuhan lain yang belum tercakup dalam aliran normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case*, sementara *special requirement* yang tidak berhubungan dengan kebutuhan fungsional.

Misalnya : kecepatan transaksi maksimum berapa cepat dan berapa lama, serta kapasitas akses yaitu jumlah *user* yang akan mengakses dalam waktu bersamaan.

f. Pre-Condition :

Menjelaskan persyaratan yang harus dipenuhi sebelum *use-case* bisa dimulai.

g. Post-Condition :

Menjelaskan kondisi yang berubah atau terjadi saat *use case* selesai dieksekusi.



PRAKTIK

Kasus berikut akan digunakan untuk pertemuan saat ini dan berikutnya.

Sebelum menyelesaikan studinya, setiap mahasiswa diwajibkan untuk menyelesaikan Tugas Akhir (TA). Pada hakekatnya Tugas Akhir merupakan kegiatan akademik yang dirancang untuk melatih kemandirian dan tanggung jawab ilmiah mahasiswa sebagai calon ilmuwan, mulai dari pemilihan topik dan penyusunan rencana penelitian, pelaksanaan penelitian, evaluasi hasil penelitian, hingga penulisan laporan tugas akhir.

Pada Kampus Sumber Ilmu, Tugas Akhir ditulis dalam bentuk Laporan Tugas Akhir, yang akan dievaluasi oleh tim penguji yang dibentuk oleh pengelola program studi. Ada tiga tahap yang harus dilakukan oleh mahasiswa tugas akhir, yaitu ujian proposal, ujian awal, dan ujian akhir. Untuk setiap tahap tersebut mahasiswa melakukan pembimbingan secara berkelanjutan kepada dosen pembimbing. Apabila pembimbing menyatakan telah menyetujui tugas akhir, maka pembimbing akan mengisi formulir check list format penulisan dan form check list isi naskah. Dokumen tugas akhir dan checklist yang sudah ditandatangani mahasiswa dan pembimbing diserahkan kepada administrator. Administrator akan menyusun jadwal ujian serta menentukan dosen penguji. Setelah jadwal ujian dan dosen penguji ditentukan maka mahasiswa menjalankan ujian sesuai jadwal. Pada saat ujian dosen penguji memberikan penilaian sesuai dengan borang evaluasi ujian Tugas Akhir. Hasil penilaian akan diserahkan kepada pihak administrator untuk di dokumentasikan.

Pada sistem yang dijelaskan sebelumnya, semua pekerjaan terpusat pada Administrator. Hal tersebut mengakibatkan beban kerja administrator lebih tinggi dan pekerjaan dokumentasi akan menjadi lebih lama. Dengan melihat permasalahan tersebut maka dikembangkan sistem informasi penilaian tugas akhir.

Berikut adalah ketentuan Tugas akhir.

a. Syarat ujian Proposal

- Naskah proposal, penulisan tabel, gambar dan daftar pustaka sudah memenuhi panduan penulisan proposal.
- Pengumpulan naskah paling lambat seminggu sebelum tanggal ujian, untuk menentukan dosen penguji.
- Bilamana tidak memenuhi persyaratan tersebut diatas, ujian dilaksanakan pada bulan berikutnya.
- Keterangan :

- Kualitas proposal dievaluasi secara kuantitatif
- Kemungkinan hasil evaluasi : diterima, perbaikan tanpa sidang lagi, perbaikan dengan sidang.

b. Syarat ujian awal

- Naskah tugas akhir sudah memenuhi check list format
- Naskah tugas akhir sudah memenuhi check list content /isi naskah
- Mahasiswa menandatangani pernyataan sudah memenuhi check list format dan isi, dan diketahui oleh dosen pembimbing
- Pengumpulan naskah paling lambat seminggu sebelum tanggal ujian, untuk menentukan dosen penguji
- Ujian dijadwalkan setiap bulan, pada minggu pertama
- Bilamana tidak memenuhi persyaratan tersebut diatas, ujian dilaksanakan pada bulan berikutnya.

c. Syarat ujian Akhir

- Naskah sudah memenuhi check list format
- Naskah tesis sudah memenuhi check list content/isi naskah
- Mahasiswa menandatangani pernyataan sudah memenuhi check list format dan isi, dan diketahui oleh dosen pembimbing
- Menyerahkan naskah publikasi sesuai dengan format jurnal yang dipilih
- Memenuhi semua usulan perbaikan yang disampaikan oleh penguji pada saat ujian awal
- Pengumpulan naskah paling lambat seminggu sebelum tanggal ujian, untuk menentukan dosen penguji
- Bilamana tidak memenuhi persyaratan tersebut diatas, ujian dilaksanakan pada bulan berikutnya.

Keterangan:

- Ujian akhir dievaluasi secara kuantitatif, dari sisi kualitas penulisan, kualitas kemampuan dan penyajian, serta kualitas penugasan dan penalaran.
- Kemungkinan hasil evaluasi adalah A, B, C, D, dengan persyaratan penyelesaian tugas akhir tidak lebih dari 1 tahun.
- Penyelesaian tugas akhir yang >1 tahun tidak dimungkinkan mendapatkan nilai A
- D: tidak lulus, C: operasional mengulang
- Form check list format dan check list isi tugas akhir ada pada administrasi prodi.

Pelaksanaan ujian awal

Waktu ujian masimum 120 menit yang dibagi dengan alokasi waktu sebagai berikut,

1. Mahasiswa presentasi maksimum 20 menit
2. Tanya jawab untuk setiap penguji maksimum 30 menit
3. Rekapitulasi hasil dan penentuan hasil.

Ketentuan ujian awal dan ujian akhir :

1. Ketua penguji :

- Memimpin pelaksanaan ujian dan menguji
- Mengatur perwaktuan pelaksanaan ujian agar ujian tepat 120 menit
- Menguji moderator ujian, mengalihkan waktu tanya jawab pada penguji lain, bila melebihi batas maksimal 30 menit
- Merangkap hasil ujian

2. Pembimbing

- Mencatat saran/komentar penguji pada buku notulen dan ujian

3. Penguji

- Melakukan tanya jawab, selama maksimum 30 menit
- Memberi nilai angka pada form evaluasi ujian

Pelaksanaan ujian awal/akhir

1. Dilaksanakan pada hari Jumat dan Sabtu setiap minggu pertama setiap bulam
2. Jumlah penguji sebanyak 3 orang dosen penguji, paling sedikit 1 orang sesuai bidang dan 1 orang dosen pembimbing .
3. Ketua penguji ditentukan oleh ka prodi
4. Pembimbing berperan sebagai notulis
5. Jika pembimbing tidak hadir maka ujian tetap berjalan. Pembimbing akan diganti dengan dosen lain dengan bobot penilaian 1 bukan 2
6. Bilamana dosen penguji berhalangan supaya memberitahui ke prodi paling lambat 2 hari setelah penetapan jadwal.
7. Peserta ujian harus menyerahkan naskah ujian dan naskah publikasi yang sudah memenuhi syarat paling lambat 1 minggu setelah pelaksanaan ujian.
8. Pengumuman pendjadwalan ujian paling lambat 5 hari sebelum pelaksanaan ujian.

Rancangan sistem

1. Requirement List

Berdasarkan permasalahan yang telah diuraikan, berikut adalah requirement list yang dapat didefinisikan.

No	Requirement	Use case
1	Administrator dapat mengentry data dosen	Entry Data Dosen
2	Administrator dapat mengentry data mahasiswa	Entry Data Mahasiswa
3	Administrator dapat mengentry data tugas akhir mahasiswa berupa proposal, naskah ujian awal, dan naskah ujian akhir	Entry TA
4	Administrator dapat mengentry borang penilaian ujian proposal, ujian awal, dan ujian akhir	Entry Borang
5	Administrator dapat mengentry rincian borang penilaian ujian proposal, ujian awal, dan ujian akhir	Entry Detail Borang
6	Administrator dapat menambah jadwal ujian dengan mengecek check list terlebih dahulu untuk memastikan bahwa naskah ujian sudah disetujui oleh pembimbing. Penambahan jadwal sekaligus menentukan dosen penguji, waktu dan tempat ujian	Entry Jadwal
7	Penguji dan administrator dapat melihat jadwal ujian	Lihat Jadwal
8	Pembimbing dan penguji bisa memberikan penilaian dan revisi terhadap hasil ujian tugas akhir mahasiswa dengan mengisi borang evaluasi	Penilaian
9	Penguji dan administrator dapat melihat nilai	Lihat Nilai

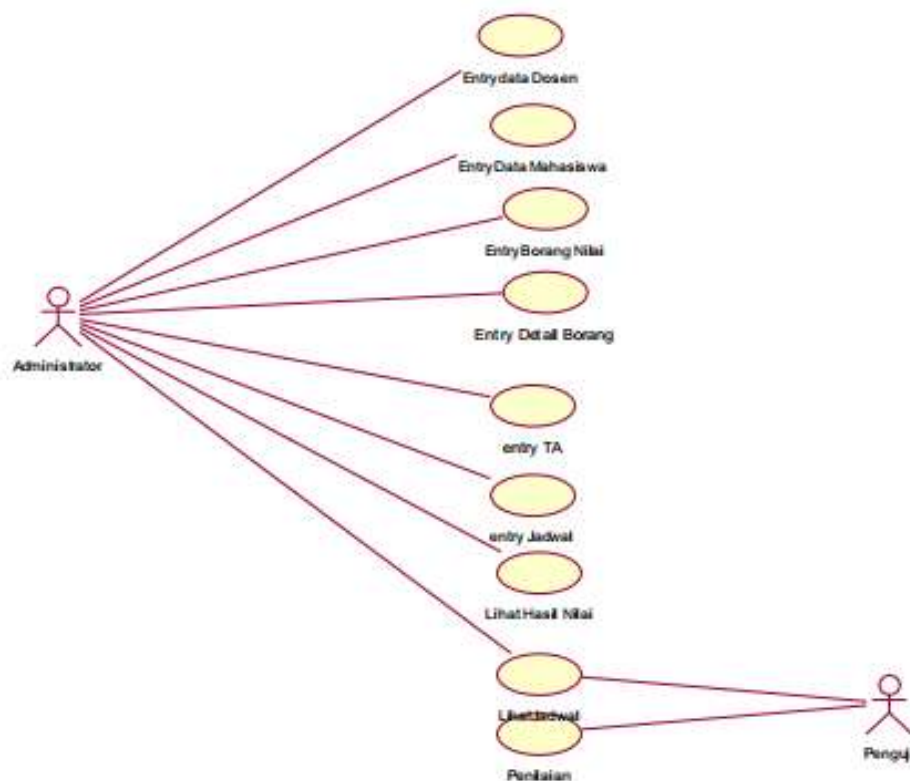
2. Komponen User (Actor)

Actor menggambarkan orang, system atau external entitas / stakeholder yang menyediakan atau menerima informasi dari sistem. Dalam sistem yang dibuat komponen user (actor), meliputi :

- a. Administrator : yaitu petugas atau staf bagian administrasi yang bertugas memasukkan data mahasiswa, dosen, TA dan mengelola tugas akhir mahasiswa seperti penjadwalan ujian, pengumuman hasil ujian
- b. Penguji : yaitu dosen yang ditunjuk untuk menguji TA mahasiswa termasuk juga dosen pembimbing. Penguji bisa melihat jadwal dari mahasiswa yang akan diuji dan bisa melakukan penilaian dengan mengisi borang penilaian.

3. Use Case Diagram

Dari actor dan requirement yang telah didefinisikan, dapat digambarkan use case diagram yang dirancang seperti yang ada pada Gambar 9.1



Gambar 9.1 Use case diagram system ujian tugas akhir

4. Deskripsi Use case diagram

Berikut beberapa deskripsi dari use case yang ada pada Gambar 9.1.

i. Entry Data Dosen

Nama use case : entry data dosen

Actor : administrator
 Tujuan : use case ini bertujuan untuk menggambarkan proses masukan data dosen yang merupakan pembimbing dan penguji tugas akhir mahasiswa
 Deskripsi : proses ini adalah sebuah kegiatan untuk mendaftarkan dosen baru ataupun mengubah data dosen yang sudah ada
 Precondition : 1. Administrator sudah harus memiliki ID agar bisa masuk di form data dosen
 2. Administrator sudah harus berada di menu memasukkan data dosen
 Post-condition : data dosen baru sudah tersimpan

Skenario

Actor Action	System Respon
1. Menekan tombol Entry Data Dosen 3. Mengisi data dosen 5. Menekan tombol simpan	2. Menampilkan form entry Data Dosen 4. Menampilkan isian data dosen 6. Menyimpan data dosen dan menampilkan data dosen yang telah serta mengkosongkan isian data dosen untuk data entry data dosen lag
Alur alternative No 6 Jika data dosen yang dimasukkan tidak valid, maka akan menampilkan pesan “tidak sukses disimpan”	

ii. Penilaian

Nama use case : Penilaian
 Actor : Penguji
 Tujuan : use case ini bertujuan untuk menampilkan form untuk melakukan penilaian terhadap proposal, ujian awal dan ujian akhir
 Deskripsi : proses ini adalah sebuah kegiatan untuk memberikan penilaian terhadap hasil ujian mahasiswa
 Precondition : 1. Penguji sudah harus memiliki ID agar bisa masuk di form penilaian
 2. Penguji sudah harus berada di menu penilaian
 Post-condition : Data penilaian mahasiswa tersimpan

Skenario

Actor Action	System Respon
1. Menekan tombol penilaian 3. Memilih nim yang akan dinilai 5. Memilih form penilaian 7. Mengisi borang penilaian 9. Menekan tombol simpan setelah seluruh borang penilaian sudah diisi	2. Menampilkan form penilaian dengan nim mahasiswa yang akan dinilai 4. menampilkan informasi mengenai tugas akhir mahasiswa 6. menampilkan borang penilaian 8. menerima nilai yang diberikan 10. Menyimpan hasil penilaian

Alur alternative No 10

Jika data nilai yang dimasukkan tidak valid, maka akan menampilkan pesan “tidak sukses disimpan”



LATIHAN

1. Bentuk kelompok proyek anda sebanyak 3 orang
2. Tentukan kasus yang akan diselesaikan oleh kelompok anda
3. Diskusikan kasus anda dengan dosen pengampu
4. Rancang requirement dan use case diagram proyek anda seperti yang ada pada bagian praktik



TUGAS

Diberikan oleh dosen pengampu



REFERENSI

Buku-buku pemodelan berorientasi objek

MODUL 10

CLASS DIAGRAM



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mendefinisikan kebutuhan class diagram dari system yang akan dibangun



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Rational Rose



DASAR TEORI

Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana diagram tersebut saling berkolaborasi untuk mencapai suatu tujuan. *Class diagram* umumnya tersusun dari elemen *class*, *interface*, *dependency*, *generalization* dan *association*.

Relasi ***dependency*** menunjukkan bagaimana ketergantungan terjadi antar *class* yang ada. Relasi ***generalization*** menunjukkan bagaimana suatu *class* menjadi *superclass* dari *class* lainnya dan *class* yang lain tersebut menjadi *subclass* dari *class* tersebut.

Relasi ***association*** menggambarkan navigasi antar ***class***, berapa banyak obyek lain bisa berhubungan dengan satu obyek (***multiplicity*** antar ***class***), dan apakah suatu *class* menjadi bagian dari *class* lainnya (***aggregation***).

Relasi ***agregasi*** adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu obyek merupakan bagian dari obyek yang lain.

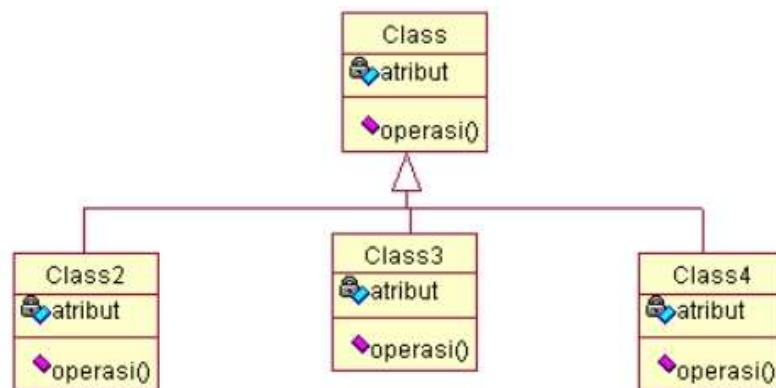
Class diagram juga digunakan untuk menggambarkan desain statis dari sistem yang sedang dibangun.

Bentuk dari *Class diagram* yang mempunyai hubungan asosiasi ditunjukkan pada gambar 10.1



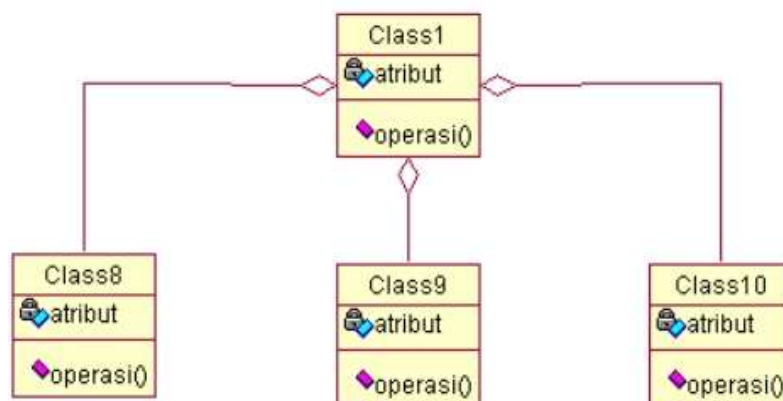
Gambar 10.1 Relasi Asosiasi antar Class

Bentuk dari *Class diagram* yang mempunyai hubungan *generalisasi* ditunjukkan pada gambar 10.2.



Gambar 10.2 Relasi Generalisasi Class Induk ke Sub Class Sub Class

Sedangkan *Class diagram* yang mempunyai hubungan agregasi ditunjukkan pada gambar 10.3.



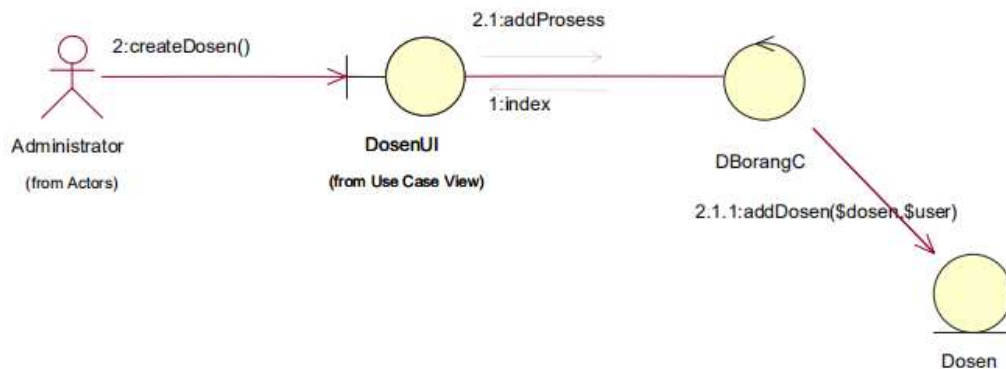
Gambar 10.3 Relasi Agregasi antar Class



PRAKTIK

Dari kasus yang ada di modul 9. Dapat digambarkan communication diagram dan class diagram untuk use case entry data dosen dan penilaian sebagai berikut.

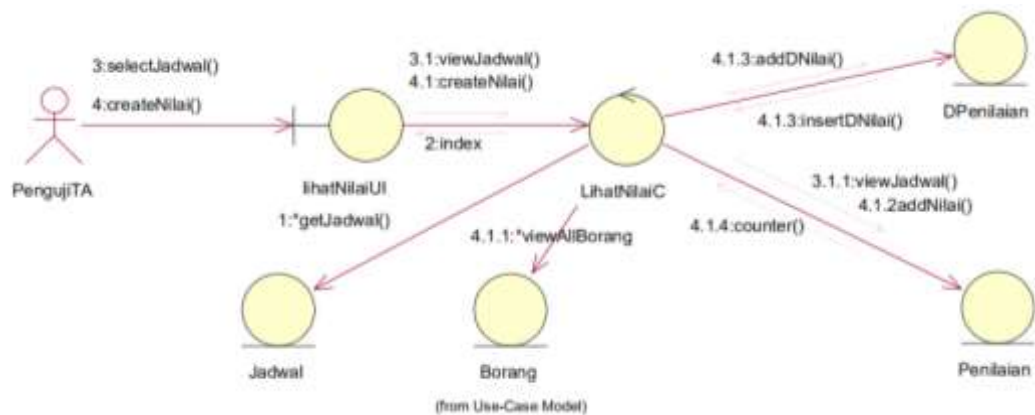
Analisis communication diagram untuk entry data dosen



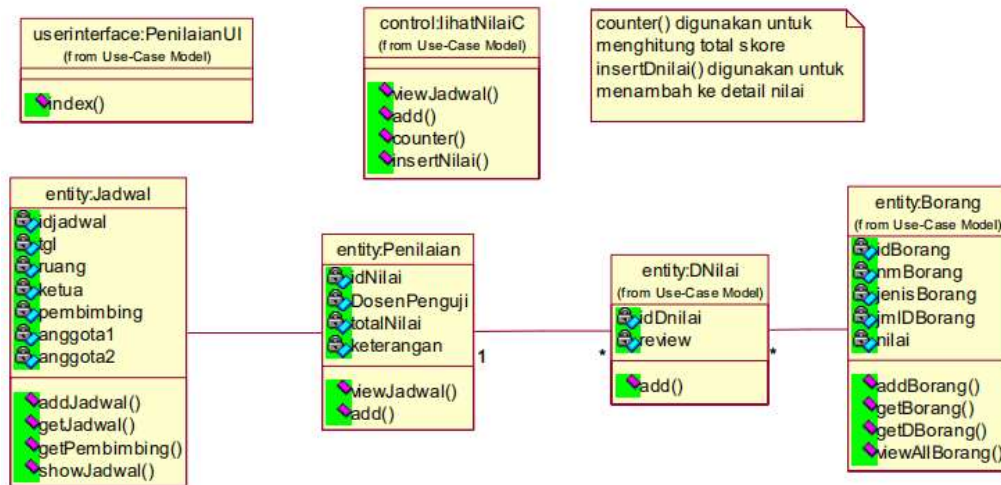
Analisis class diagram untuk entry data dosen



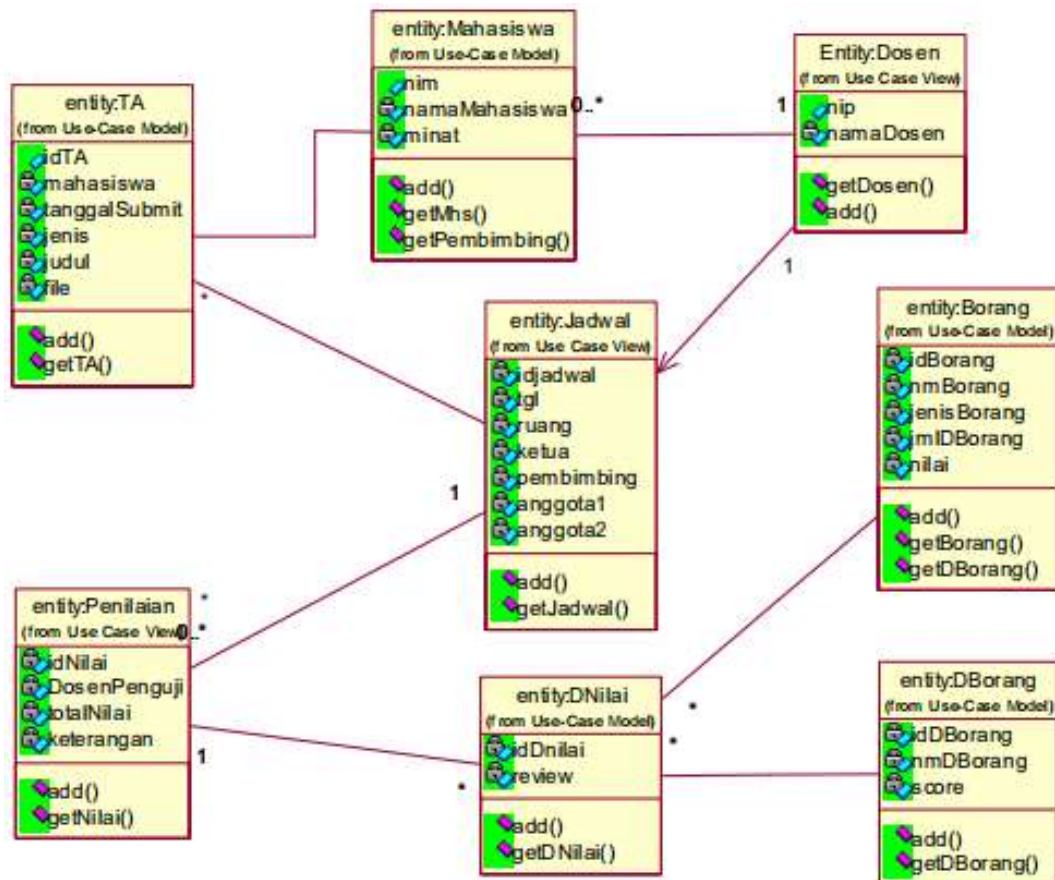
Analisis communication diagram untuk penilaian



Analisis Class diagram untuk penilaian



Class Diagram gabungan untuk Stereotype Entity





LATIHAN

Gambarkan communication diagram dan class diagram dari kasus proyek kelompok anda masing-masing.



TUGAS

Diberikan oleh masing-masing pengampu



REFERENSI

Buku-buku pemodelan berorientasi objek

MODUL 11

SEQUENCE DIAGRAM



CAPAIAN PEMBELAJARAN

Mahasiswa dapat membuat Sequence Diagram untuk studi kasus tertentu



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Rational rose



DASAR TEORI

Sequence Diagram menjelaskan secara detail tentang urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case, interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

Pembuatan *sequence diagram* merupakan aktivitas yang paling kritis dari proses desain, karena artefak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program. Oleh karena itu, berharga untuk meluangkan waktu lebih lama di pembuatan *sequence diagram* ini untuk menghasilkan *sequence diagram* yang terdesain dengan baik.

Sequence diagram biasanya tersusun dari elemen obyek, interaction dan message. Interaction menghubungkan 2 obyek dengan pesannya. Diagram ini menjelaskan aspek dinamis dari sistem yang sedang dibangun.

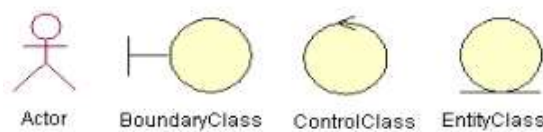
Di dalam *Sequence Diagram*, terdapat pelaku (*actor*), *boundary class*, *control class*, dan *entity class*.

- *Boundary Class* adalah kelas yang memodelkan interaksi antara satu atau lebih actor dengan sistem. *Boundary* memodelkan bagian dari sistem yang

bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.

- *Control Class* digunakan untuk memodelkan “perilaku mengatur”, khusus untuk satu atau beberapa *use-case* saja.
- *Entity Class* memodelkan informasi yang harus disimpan oleh sistem. *Entity Class* memperlihatkan struktur data dari suatu sistem.

Adapun notasi dari *actor*, *boundary class*, *control class* dan *entity class* adalah seperti tampak pada gambar 11.1.



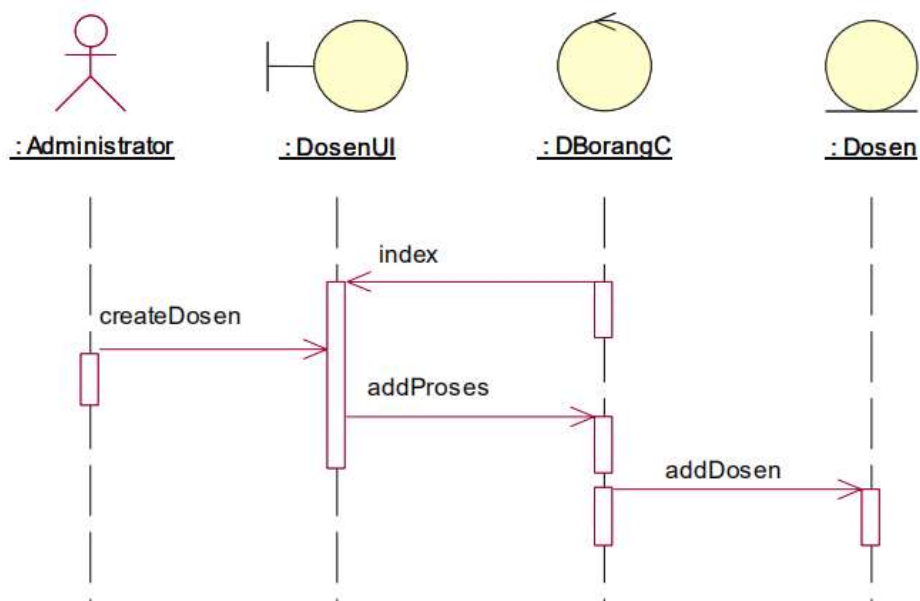
Gambar 11.1 Notasi/symbol dalam Sequence Diagram



PRAKTIK

Dari kasus yang ada di pertemuan ke-9, untuk use case entry data dosen dan penilaian dapat digambarkan sequence diagramnya sebagai berikut.

Sequence diagram untuk use case entry data dosen



```

sequenceDiagram
    participant User as :Pengguna
    participant UI as :PenilaianUI
    participant C as :PenilaianC
    participant Jadwal as :Jadwal
    participant Borang as :Borang
    participant Nilai as :Nilai
    participant DNilai as :DNilai

    User->>UI: login
    activate UI
    UI->>C: index
    deactivate UI
    activate C
    C->>Jadwal: viewJadwal
    deactivate C
    activate Jadwal
    Jadwal->>UI: selectJadwal
    deactivate Jadwal
    activate UI
    UI->>C: createNilai
    deactivate UI
    activate C
    C->>Borang: add
    deactivate C
    activate Borang
    Borang->>UI: selectBorang
    deactivate Borang
    activate UI
    UI->>C: showDBorang
    deactivate UI
    activate C
    C->>Borang: *viewAllBorang
    deactivate C
    activate Borang
    Borang->>C: *add
    deactivate Borang
    activate C
    C->>Nilai: *add
    deactivate C
    activate Nilai
    Nilai->>DNilai: insertDNilai
    deactivate Nilai
    activate DNilai
    DNilai->>C: counter
    deactivate DNilai
    deactivate C
    deactivate UI
    deactivate User
  
```







MODUL 12

ACTIVITY DIAGRAM



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mendefinisikan dan menggambarkan activity diagram



Rational rose

KEBUTUHAN ALAT/BAHAN/SOFTWARE



DASAR TEORI

Activity diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Menguntungkan untuk membuat *activity diagram* pada awal pemodelan proses untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behaviour* atau menggambarkan interaksi antara beberapa *use case*.

Elemen-elemen activity diagram :

1. Status *start* (mulai) dan *end* (akhir)
2. Aktivitas yang merepresentasikan sebuah langkah dalam *workflow*.
3. *Transition* menunjukkan terjadinya perubahan status aktivitas (*Transitions show what state follows another*).
4. Keputusan yang menunjukkan alternatif dalam *workflow*.
5. *Synchronization bars* yang menunjukkan *subflow parallel*. *Synchronization bars* dapat digunakan untuk menunjukkan *concurrent threads* pada *workflow* proses bisnis.

6. *Swimlanes* yang merepresentasikan *role* bisnis yang bertanggung jawab pada aktivitas yang berjalan.

Membuat Swimlanes

1. Klik kanan pada *use case* yang akan dibuat *activity diagram*, kemudian pilih *Select in Browser*. *Use case* yang dipilih akan tersorot pada *browser*.
2. Klik kanan *use case* yang tersorot di *browser*, kemudian klik *New, Activity Diagram*.
3. Beri nama *activity diagram*.
4. Buka *activity diagram* dengan double klik
5. Pilih *icon swimlane* dari *toolbar* dan klik ke dalam *activity diagram*.
6. Buka *Specification* dari *swimlane* dengan cara double klik *header swimlane* (*NewSwimlane*) pada diagram.
7. Beri nama *swimlane* dengan nama sesuai dengan *role* bisnis yang menjalankan aktivitas-aktivitas.
8. Klik OK.

Membuat status Aktifitas (Aktifitas)

1. Klik *icon* status mulai di *toolbar* dan kemudian klik di *swimlane*.
2. Klik *icon* aktifitas di *toolbar* dan kemudian klik di *swimlane*.
3. Ganti nama *NewActivity* sesuai dengan aktifitas yang dilakukan.
4. Untuk menunjukkan aktifitas pada nomor tiga berhubungan dengan status mulai, klik *icon state transition* di *toolbar*..
5. Klik dan *drag transition* dari status mulai menuju ke aktifitas nomor tiga.
Catatan: untuk membuat aktifitas dan *transition* lainnya dapat dilakukan dengan mengulang langkah 2 sampai 5.

Membuat Decision point

1. Klik *icon decision point* di *toolbar* dan kemudian sambungkan *transition* menuju dan dari *decision point* ke aktifitas-aktifitas yang berhubungan.
2. Buka *decision specification* dengan cara double klik *decision point*.
3. Masukkan nama *decision point* sesuai dengan fungsinya.
4. Untuk setiap *transition* yang keluar dari *decision point*, double klik untuk membuka *specification*-nya.

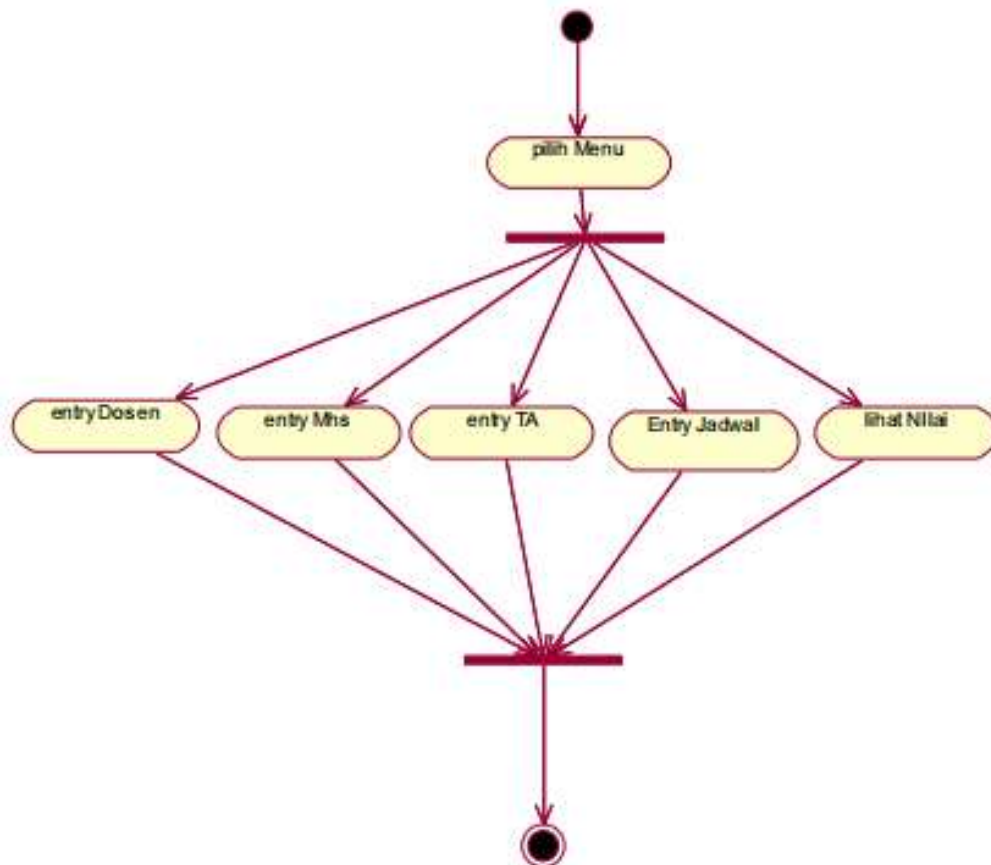
6. Pada *tab Detail*, masukkan label *guard condition* dengan fungsi yang sesuai di kotak *Guard Condition*. Arti *Guard Condition* adalah transition yang keluar dari *decision point* di-triger oleh *guard condition* pada *decision point*-nya.
7. Klik OK



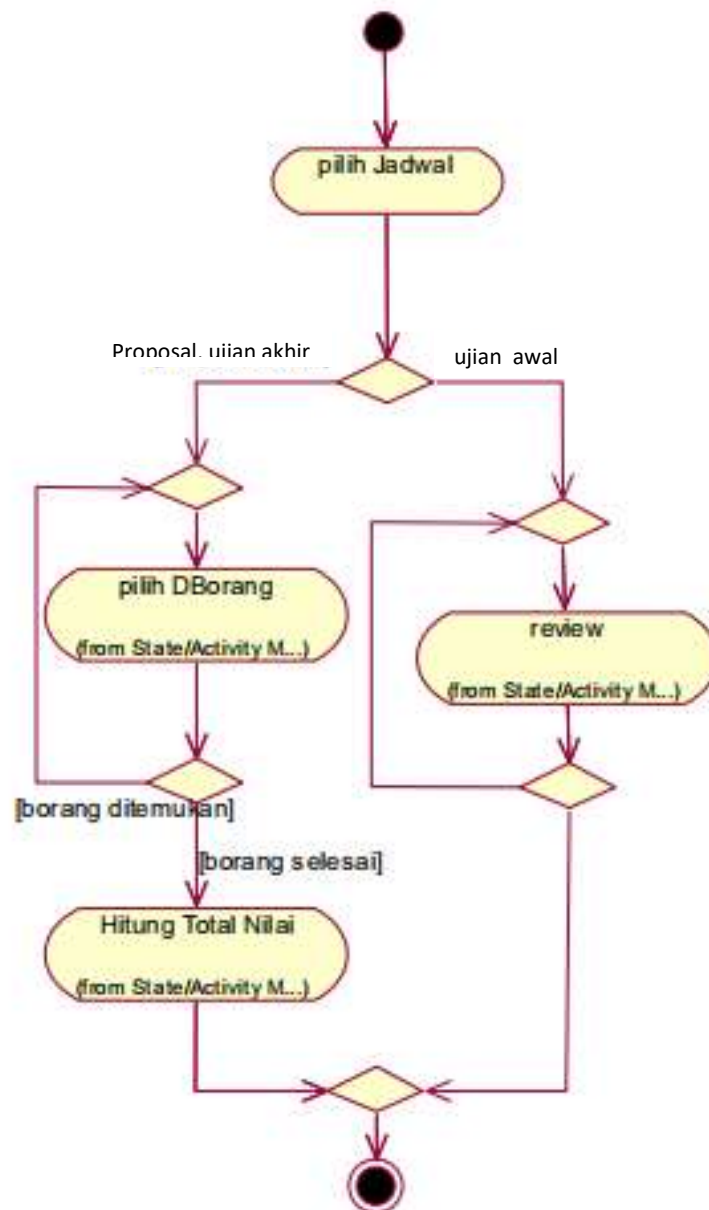
PRAKTIK

Untuk kasus yang ada pada modul 9, berikut beberapa activity diagram yang dibutuhkan.

Activity Diagram menu Administrator



Activity diagram penilaian



LATIHAN

Definisikan dan gambarkan seluruh aktivitas diagram yang dibutuhkan dari proyek kelompok anda



TUGAS

Diberikan oleh dosen pengampu



REFERENSI

Buku-buku pemodelan berorientasi objek

MODUL 13

INTERFACE PROTOTYPES



CAPAIAN PEMBELAJARAN

Mahasiswa dapat merancang kebutuhan antarmuka sistem



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Tools pengolah kata lengkap



DASAR TEORI

Pengertian Interaksi

Interaksi adalah komunikasi 2 arah antara manusia (user) dan sistem komputer. Interaksi menjadi maksimal apabila kedua belah pihak mampu memberikan stimulan dan respon (aksi & reaksi) yang saling mendukung, jika salah satu tidak bisa, maka interaksi akan mengalami hambatan atau bahkan menuju pembiasan tujuan.

Prinsip Utama Mendesain Antarmuka (Interface)

Berikut ini beberapa hal yang menjadi prinsip utama mendesain antarmuka yang baik dengan memperhatikan karakteristik manusia & komputer:

1. User compatibility

Antarmuka merupakan topeng dari sebuah sistem atau sebuah pintu gerbang masuk ke sistem dengan diwujudkan ke dalam sebuah aplikasi software. Oleh karena itu sebuah software seolah-olah mengenal user-nya, mengenal karakteristik user-nya, dari sifat sampai kebiasaan manusia secara umum. Hal tersebut harus terpikirkan oleh desainer dan tidak dianjurkan merancang antarmuka dengan didasarkan pada dirinya sendiri

2. Product compatibility

Sebuah aplikasi yang bertopengkan antarmuka harus sesuai dengan sistem aslinya. Adanya aplikasi software diharapkan dapat menjaga produk yang dihasilkan dan dihasilkan produk yang jauh lebih baik. Contoh : aplikasi sistem melalui antarmuka diharapkan menghasilkan report/laporan serta informasi yang detail dan akurat dibandingkan dengan sistem manual.

3. Task compatibility

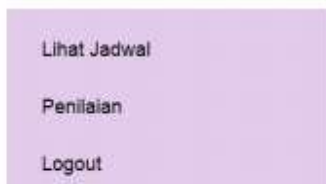
Sebuah aplikasi yang bertopengkan antarmuka harus mampu membantu para user dalam menyelesaikan tugasnya. Semua pekerjaan serta tugas-tugas user harus diadopsi di dalam aplikasi tersebut melalui antarmuka



PRAKTIK

Dari kasus yang ada pada modul 9 dapat dirancang berbagai rancangan antarmuka sebagai berikut.

1. Menu utama Dosen



Menu Lihat Jadwal digunakan untuk melihat jadwal ujian TA mahasiswa yang diuji, dan menu Penilaian digunakan untuk melakukan penilaian TA dari mahasiswa yang diuji dengan melakukan pengisian borang TA.

2. Lihat jadwal

PENILAIAN UGAS AKHIR					
Mahasiswa	Judul	Tanggal	Ruang	Pembimbing	Penguji
Septiawan	Analisis sentiment	08-11-2019	U.3.1	Amir	Ketua : Tutik Penguji1: Yuni Penguji2: Rio
Iwan	Sistem pakar	10-11-2019	U.3.2	Dedi	Ketua : Tutik Penguji1: Dion Penguji2: Joko

Lihat Jadwal digunakan untuk melihat jadwal semua mahasiswa yang akan diuji oleh dosen bersangkutan pada bulan tersebut.

3. Pengisian borang

Untuk melakukan pengisian borang dosen harus mengisi detail borang penilaian dengan memilih salah satu point-point penilaian sesuai kriteria dan review jika ada catatan tambahan.



LATIHAN

Rancang seluruh antarmuka yang dibutuhkan oleh kasus kelompok anda.



TUGAS

Diberikan oleh dosen pengampu



REFERENSI

Buku-buku terkait dengan perancangan system dan antarmuka system.

MODUL 14
DOKUMENTASI DAN PRESENTASI
PROYEK PERANCANGAN SISTEM INFORMASI



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mendokumentasikan dan mengkomunikasikan proyek perancangan system informasi yang telah dilakukan



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Perangkat olah kata dan presentasi



DASAR TEORI

Teori presentasi dan komunikasi



PRAKTIK

Presentasikan perancangan system kelompok anda