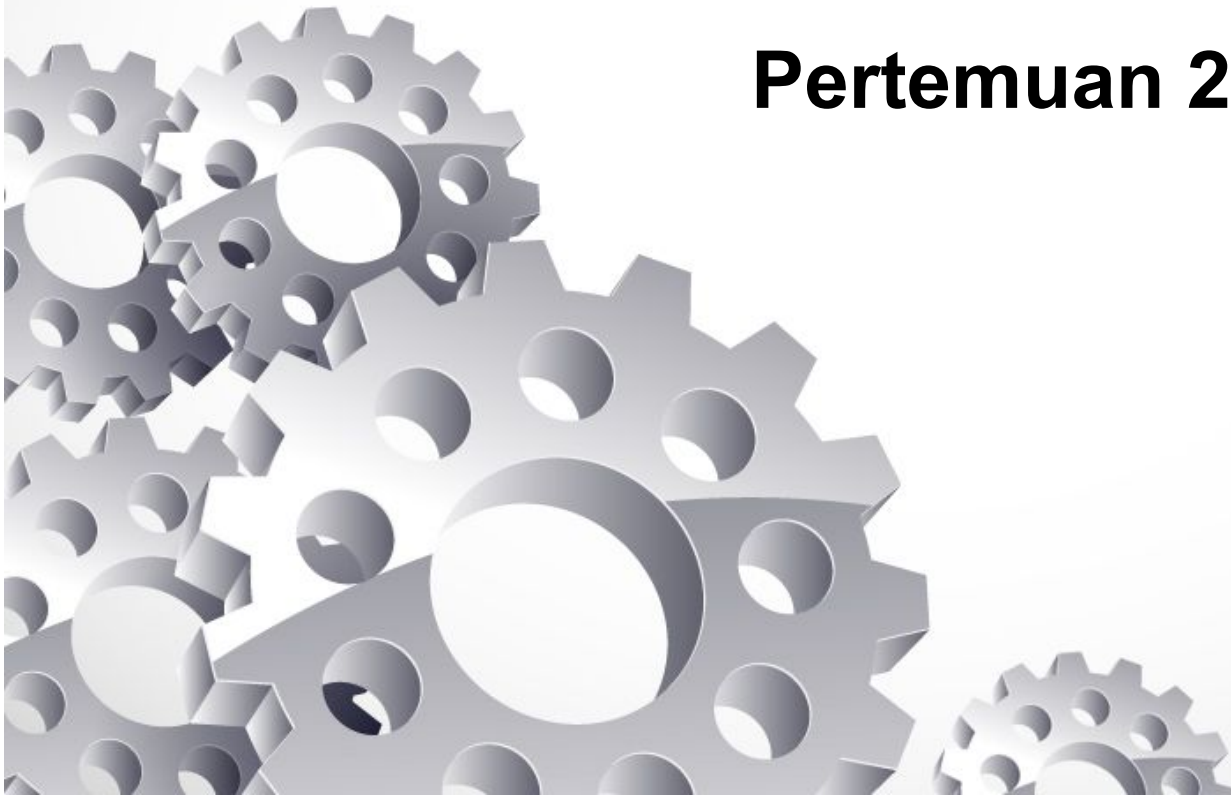


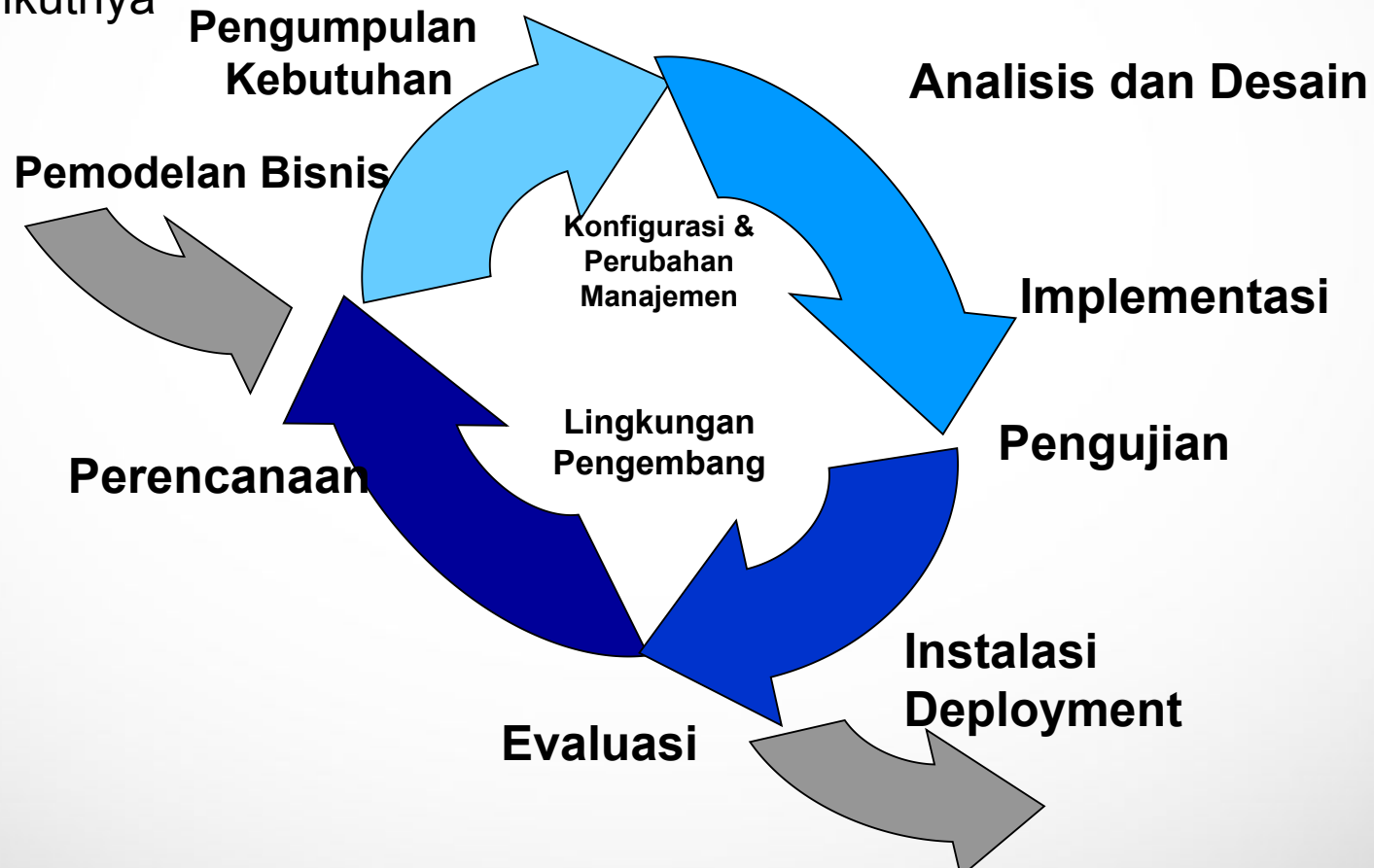
PENGANTAR RUP & UML

Pertemuan 2



PENGANTAR RUP

Rational Unified Process (RUP) atau dikenal juga dengan proses iteratif dan *incremental* merupakan sebuah pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan *incremental* (bertahap dengan proses menaik) dimana setiap iterasi memperbaiki iterasi berikutnya



PENGANTAR RUP



- Ciri utama metode ini adalah menggunakan use-case driven dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.
- RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language (UML)*

Phase RUP



Metode dari RUP ada empat fase sebagai berikut:

1. Inception

Pada tahap ini pengembang mendefinisikan batasan kegiatan :

- Melakukan analisis kebutuhan user.
- Melakukan perancangan awal perangkat lunak (perancangan arsitektural dan use case).
- Menentukan kebutuhan akan model bisnis diterapkan. (Biaya)
- Menentukan batasan - batasan dalam sebuah project.
- Memahami ruang lingkup dari proyek (termasuk biaya, waktu, kebutuhan resiko dsb)
- Hasil yg diharapkan : Pendefinisian Ruang lingkup, Perkiraan Biaya, dan perkiraan Jadwal.

Phase RUP

2. *Elaboration* (perluasan/ perencanaan)

- Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasi fitur perangkat lunak hingga perilsan prototype versi Beta dari perangkat lunak.
- Pengimplementasian use case, sebagai perwujudan dari arsitektur system software.
- Menghilangkan kemungkinan-kemungkinan terbesar yang memungkinkan timbulnya sebuah resiko dalam proses perkembangan project itu sendiri, dikarenakan jika sampai terjadi perubahan terhadap project dalam fase berikutnya, akan menyulitkan pengembang project tersebut untuk kembali meninjau ulang.
- Merencanakan fase berikutnya yaitu construction



Phase RUP



3. **Construction** (konstruksi)

- Aktivitas yang terjadi dalam fase ini antara lain design, implementasi, dan serta pengujian software tersebut
- Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem.
- Tujuan utama dalam fase ini adalah perkembangan sebuah project dalam bentuk coding.

4. **Transition** (transisi)

- Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh user.
- Sosialisasi perangkat lunak
- Aktifitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

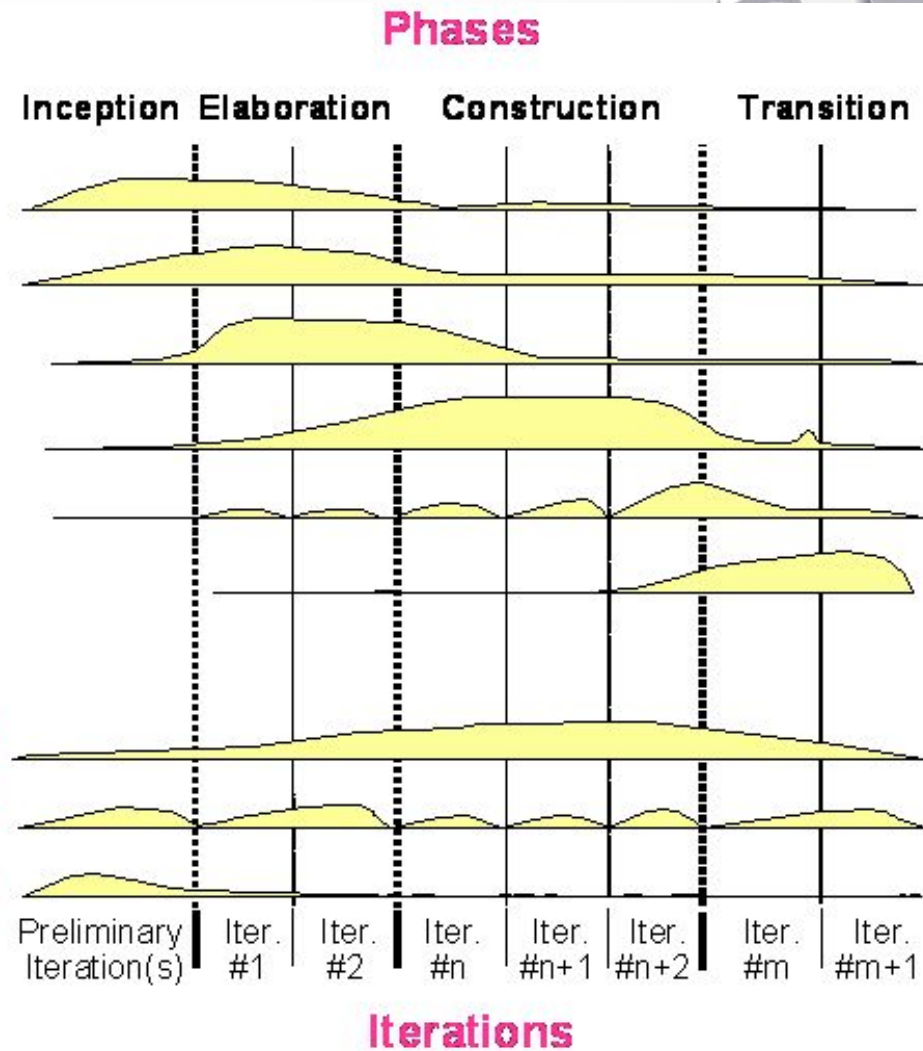
Phase RUP

Process Disciplines

Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment

Supporting Disciplines

Configuration Mgmt
Project Management
Environment



Bagian Business Modeling

Tujuan utama dalam business modeling di sini adalah untuk memungkinkan adanya komunikasi dan pengertian yang lebih baik dari business engineering dan software engineering.

Fase-fase yang terlibat dalam business modeling :

- a. **Inception** : pertama kalinya business modeling dideklarasikan dan didefinisikan.
- b. **Elaboration** : peninjauan kembali terhadap requirement bisnis untuk meminimalisasikan terjadinya perubahan pada tahap selanjutnya yaitu construction.
- c. **Construction** penerapan dari business modeling yang telah terdefinisi dalam bentuk coding.
- d. **Transition** : dimungkinkan apabila terjadi kesepakatan antara developer dengan end users dalam perawatan software yang telah dibuat.

Bagian Requirement

Tujuan dari tahap ini adalah mendeskripsikan apa yang harus dilakukan oleh system dan memungkinkan terjadinya kesepakatan antara customer dan developer tentang hal itu.

Fase-fase yang terlibat antara lain :

- 1.Inception** : requirement dari software pertama kali dibahas. Lebih terfokus pada requirement pengembangan software yang akan dipakai.
- 2.Elaboration** : mengurangi / meninjau kembali requirement dari software, dan dimungkinkan terjadi pergantian requirement dalam software yang akan dikembangkan.
- 3.Construction** : perwujudan requirement yang ada dalam bentuk coding dari software yang dikembangkan beserta pengujian apakah software sudah memenuhi requirement awal.
- 4.Transition** : requirement dalam fase ini berupa requirement dari end users untuk menambah aplikasi software, perawatan software.

Bagian Analysis dan Design

Tujuan dalam tahap ini adalah untuk menunjukkan bagaimana project akan diwujudkan dalam fase implementasi kelak. Hasil dari tahap ini adalah model design.

Fase-fase yang terlibat :

1. **Inception** : pembahasan tentang business modeling dan requirement.
2. **Elaboration** : Fase inilah yang menjadi pusat perkembangan, peninjauan kembali dilakukan pada fase ini.
3. **Construction**: Project dikembangkan dalam bentuk coding

Bagian Implementasi

Tujuan dari implementasi di sini adalah mendefinisikan pengkodean secara terorganisasi, mengimplimentasikan classes dan objects dalam bentuk komponen-komponen, menguji perkembangan komponen-komponen dalam bentuk kesatuan, dan mengintegrasikan hasil-hasil dari tiap-tiap kelompok yang mengerjakan project.

Fase-fase yang terlibat :

1. **Inception** : Pada tahap ini implementasi berlaku dengan terjadinya percakapan antara end users dan developer mengenai software yang akan dikembangkan.
2. **Elaboration** : selain implementasi terhadap pembuatan use case, tahap ini juga memuat implementasi dari perkembangan perencanaan arsitektural dan sebagainya.
3. **Construction** : Implementasi dari rancangan software dilaksanakan.
4. **Transition** : Implementasi yang terjadi pada tahap ini adalah penyerahan software terhadap end users dan implementasi pada penerapan aplikasi software yang telah dikembangkan .

Bagian Testing

Tujuan dari testing adalah memverifikasi interaksi antar object, memverifikasi integrasi yang sesuai antar komponen, memverifikasi apakah semua requirement telah terpenuhi dengan benar, mengidentifikasi dan memastikan defect telah ditangani secara benar.

Fase-fase yang terlibat :

1. **Inception** : Dalam fase ini testing dilakukan apabila pemodelan bisnis dan requirement telah teridentifikasi. Testing dilakukan dengan tujuan menghasilkan kesepakatan antara end users dengan developer.
2. **Elaboration** : testing di sini merupakan testing setelah use case diimplementasikan, masih seputar tercapainya kesepakatan antara end users dengan developer.
3. **Construction** : testing kebanyakan dilakukan di akhir fase construction, karena setelah penyelesaian program-lah, testing baru dilaksanakan.
4. **Transition** : testing dilakukan sebelum penyerahan software kepada end users dengan keadaan yang sebenarnya.

Bagian Pengembangan

Tujuan dari pengembangan di sini adalah suksesnya menghasilkan suatu project dan mengantarkan project tersebut pada end users.

Fase-fase yang terlibat :

1. **Elaboration** : mulailah pengembangan tentang realitas dari software itu akan seperti apa dalam fase ini.
2. **Construction** : dalam fase ini pengembangan software secara nyata terjadi dengan adanya coding.
3. **Transition** : fase yang paling berpengaruh karena adanya penyerahan software dari developer kepada end users.



Bagian Manajemen Konfigurasi dan Perubahan

Area-area spesifik dalam bagian ini adalah manajemen konfigurasi, merubah management requirement, status dan management pengukuran.

Fase-fase yang terlibat :

1. **Inception** : terjadi diskusi mengenai konfigurasi dari system software yang diinginkan.
2. **Elaboration** : masih membahas seputar konfigurasi software, ditambah dengan perubahan-perubahan yang terjadi, terkait dengan diminimalisasikannya perubahan dalam fase selanjutnya.
3. **Construction** : dalam fase inilah akan terlihat jelas penerapan dari konfigurasi yang telah ditentukan, dan mungkin tidaknya konfigurasi yang diinginkan terpenuhi.
4. **Transition** : konfigurasi yang ada adalah mengenai konfigurasi system dalam keadaan yang sesungguhnya.

Pengantar UML



Unified Modeling Language (UML)

Aturan-aturan pemodelan yang digunakan untuk mendeskripsikan sistem perangkat lunak dalam bentuk kumpulan obyek.

Note :

UML bukan sebuah metode utk mengembangkan sistem tetapi notasi-notasi yang digunakan secara umum sebagai standar untuk pemodelan obyek.

PENGANTAR UML



UML disebut sebagai bahasa pemodelan bukan metode. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

UML merupakan bahasa standar untuk penulisan *Blueprint Software* yang digunakan untuk visualisasi (*Visualize*), Spesifikasi (*Specify*), pembentukan (*Construct*) dan Pendokumentasian (*Documentation*) alat - alat dari sistem perangkat lunak.

Tujuan Penggunaan UML



Tujuan Penggunaan UML adalah sebagai berikut :

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.

DIAGRAM UML (2.3)

1	Structure Diagram	kumpulan diagram yang digunakan utk menggambarkan struktur statis dari sistem yang dimodelkan	<ul style="list-style-type: none">- Class Diagram- Object Diagram- Component Diagram- Composite Structure Diagram- Package Diagram- Deployment Diagram
2	Behavior Diagrams	kumpulan diagram yang digunakan utk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem	<ul style="list-style-type: none">- Usecase Diagram- Activity Diagram- State Machine Diagram
3	Interaction Diagram	kumpulan diagram yg digunakan utk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem	<ul style="list-style-type: none">- Sequence Diagram- Communication Diagram- Timing Diagram- Interaction Overview Diagram

Diagram –Diagram dalam UML 2.0 (1)

Diagram	Description
Use Case	<ul style="list-style-type: none">➤ Menggambarkan interaksi antara sistem dan pengguna/sistem eksternal. Secara grafis, menggambarkan siapa yang akan menggunakan sistem dan dengan cara bagaimana pengguna berinteraksi dengan sistem.➤ Narasi use case digunakan untuk menggambarkan secara tekstual langkah-langkah dalam interaksi sistem dengan penggunanya.
Activity (Aktivitas)	Menggambarkan alur aktivitas sekuensial dari tiap use case/proses bisnis/logika sistem.
Class (Kelas)	Menggambarkan struktur obyek dari sistem, yang memuat obyek-obyek yang terdapat dalam sistem beserta hubungan/relasi antar obyek.

Diagram -Diagram dalam UML 2.0 (2)

Diagram	Description
Object (Obyek)	Serupa dengan class diagram, namun digunakan untuk memodelkan kejadian (instance) obyek dengan nilai-nilainya. Dipakai oleh pengembang sistem untuk memotret keadaan obyek sistem pada suatu saat tertentu.
State Machine	Memodelkan bagaimana kejadian dapat mengubah keadaan (state) dari suatu obyek sepanjang "hidupnya" obyek tersebut. Menggambarkan berbagai macam keadaan yang dapat dialami suatu obyek beserta transisi antar keadaan tsb.
Composite Structure	Mendekomposisi struktur internal kelas, komponen, atau use case.

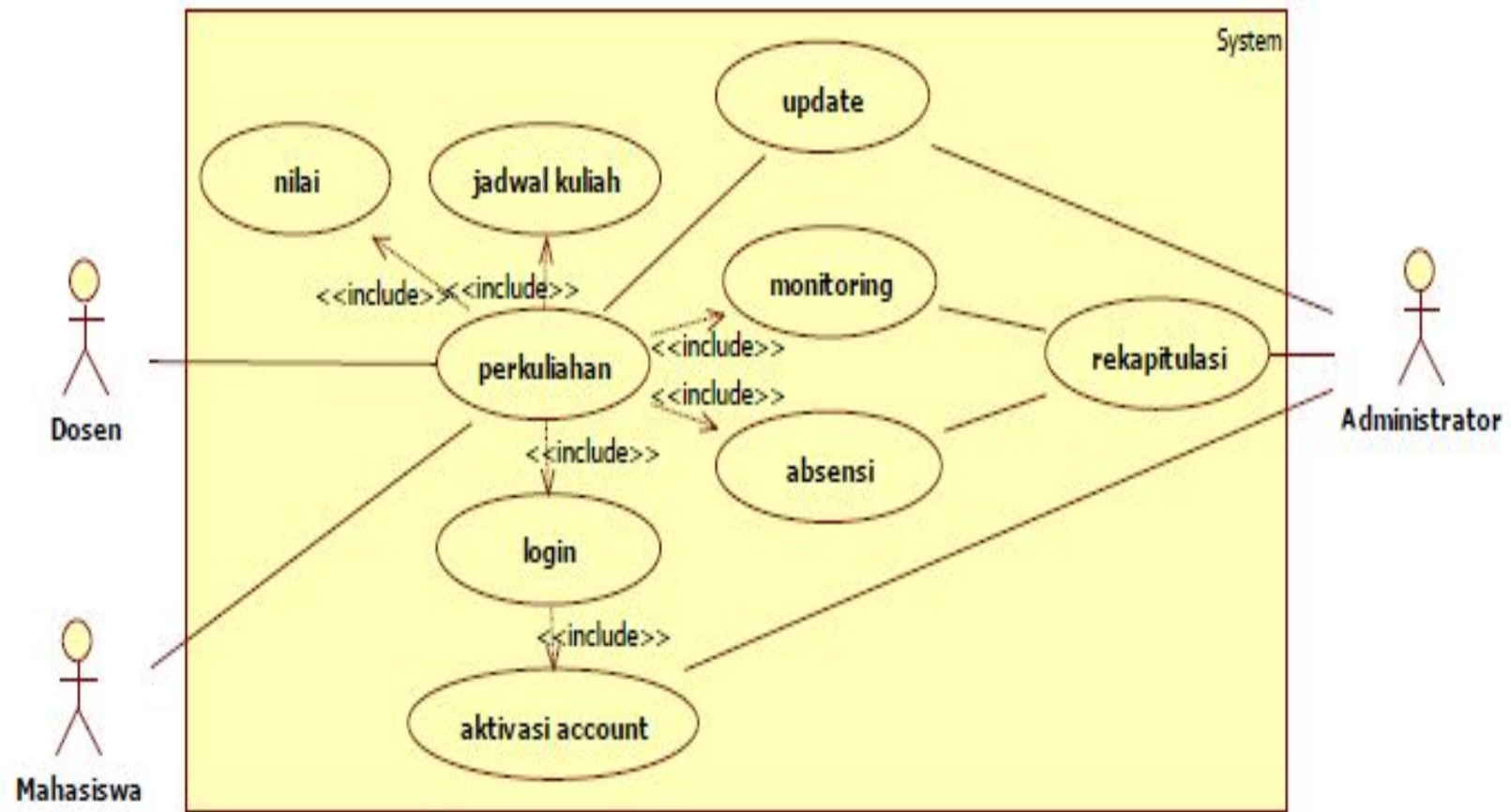
Diagram -Diagram dalam UML 2.0 (3)

Diagram	Description
Sequence (Sekuen)	Menggambarkan secara grafis bgmn obyek-obyek berinteraksi satu sama lain melalui pesan selama eksekusi suatu use case/operasi. Mengilustrasikan bagaimana pesan dikirim dan diterima oleh obyek dan bgmn urutannya.
Communication (Komunikasi)	(Dalam UML 1.X disebut Collaboration diagram). Menggambarkan interaksi obyek-obyek melalui pesan. Lebih berfokus pada struktur internal obyek dalam format jaringan, ketimbang urutannya seperti pada diagram sekuen.
Interaction Overview (Overview Interaksi)	Mengkombinasikan fitur-fitur dari diagram sekuen dan diagram aktivitas untuk menunjukkan bgmn obyek-obyek berinteraksi dalam tiap aktivitas dalam suatu use case.

Diagram -Diagram dalam UML 2.0 (4)

Diagram	Description
Timing	Bentuk diagram interaksi yang lain yang berfokus pada kendala waktu dalam perubahan keadaan dari suatu obyek tunggal atau kelompok. Bermanfaat untuk merancang embedded software bagi suatu piranti.
Component	Menggambarkan organisasi kode program yang dibagi dalam beberapa komponen dan bagaimana komponen-komponen tsb berinteraksi.
Deployment	Menggambarkan konfigurasi komponen software dalam arsitektur fisik dari "node-node" sistem hardware.
Package	Menggambarkan bagaiman kelas-kelas atau unit-unit dalam UML yang lain diorganisasi ke dalam paket (berhubungan dengan package dalam Java, C++ atau .NET) serta ketergantungan antar paket

Usecase



Skenario use case bertujuan untuk mendeskripsikan atau menjelaskan diagram use case.

Nama Use Case : Absensi.

Aktor : Mahasiswa.

Deskripsi : Sebagai dokumentasi untuk mahasiswa yang sudah mengikuti atau memasuki perkuliahan.

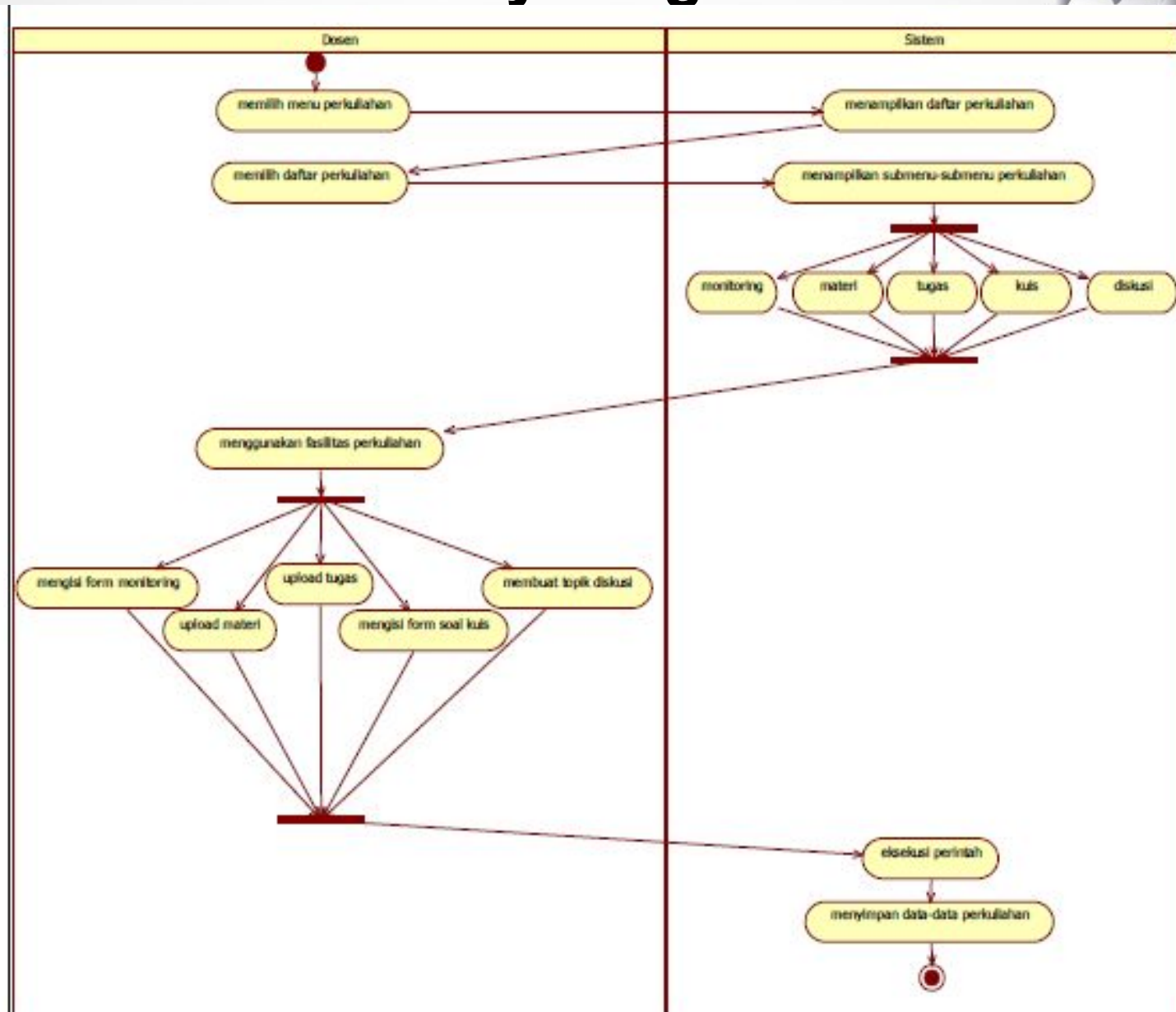
Aliran Utama

Kondisi Awal : user (mahasiswa) sudah login.

No	Aksi Aktor	No	Reaksi Sistem
1	Memilih menu absensi.	2	Menampilkan form absensi.
3	Input data absensi.	4	Menyimpan data absensi ke dalam database.

Kondisi Akhir : data absensi tersimpan ke dalam database.

Activity Diagram



antara kelas
e diagram,

```

classDiagram
    class program {
        idprogram : String
        namaprogram : String
        update()
        tampil()
    }
    class interes {
        idinteres : String
        namaint : String
        update()
        tampil()
    }
    class administrator {
        idadmin : String
        username : String
        password : String
        login()
        akses()
        update()
    }
    class materi {
        idmateri : String
        namamateri : String
        filemateri : String
        update()
        upload()
        download()
    }
    class soalkuis {
        idsoal : Int
        soal : String
        j1 : String
        j2 : String
        j3 : String
        j4 : String
        j5 : String
        j6 : String
        update()
        tampil()
        menyamakan()
    }
    class kuis {
        idkuis : String
        judulkuis : String
        update()
        tampil()
        upload()
        download()
    }
    class nilai {
        idnilai : String
        nilai : Int
        update()
        tampil()
    }
    class nilaikuis {
        idnilaikuis : String
        nilaikuis : Int
        update()
        tampil()
    }
    class nilaitugas {
        idnilaiugas : String
        nilaiugas : Int
        fileugas : String
        update()
        tampil()
    }
    class tugas {
        idtugas : String
        judulugas : String
        fileugas : String
        update()
        tampil()
        upload()
        download()
    }
    class komentar {
        idkomentar : String
        komentar : String
        komentator : String
        update()
        tampil()
    }
    class monitoring {
        idmonitoring : String
        pertemuan : Int
        tanggalperkuliahan : Date
        lokasiperkuliahan : String
        update()
        tampil()
        rekap()
        download()
    }
    class absensi {
        idabsen : String
        update()
        tampil()
        rekap()
        download()
    }
    class jadwalkuliah {
        idjadwal : String
        hari : Date
        perkuliahan : Time
        kuliahselesai : Time
        ruangan : String
        update()
        tampil()
        download()
    }
    class kelas {
        idkelas : String
        namakls : String
        semester : String
        tahunadm : Date
        update()
        tampil()
    }
    class dosen {
        iddosen : String
        nip : String
        namadn : String
        gelaripn : String
        gelarblk : String
        email : String
        password : String
        login()
        akses()
        update()
        tampil()
    }
    class mahasiswa {
        nim : String
        namamhs : String
        kelamin : String
        email : String
        password : String
        login()
        akses()
        update()
        tampil()
    }
    class status {
        idstatus : String
        namastatus : String
        update()
    }
    class mastakuliah {
        idmatkul : String
        namamatkul : String
        sks : Int
        update()
        tampil()
    }
    class jenismatkul {
        idjenismatkul : String
        namajenismatkul : String
        update()
        tampil()
    }
    class pengumuman {
        idpengumuman : String
        judulpeng : String
        isipengumuman : String
        sumber : String
        tanggal : Date
        update()
        tampil()
    }
    class diskusi {
        idtopik : String
        judultopik : String
        moderator : String
        bahasan : String
        update()
        tampil()
        dikomentari()
    }

    program "1" -- "1" kelas
    program "1" -- "1" jadwalkuliah
    program "1" -- "1" monitoring
    program "1" -- "1" absensi
    program "1" -- "1" materi
    program "1" -- "1" soalkuis
    program "1" -- "1" kuis
    program "1" -- "1" nilai
    program "1" -- "1" nilaikuis
    program "1" -- "1" nilaitugas
    program "1" -- "1" tugas
    program "1" -- "1" komentar
    program "1" -- "1" monitoring
    program "1" -- "1" absensi
    program "1" -- "1" jadwalkuliah
    program "1" -- "1" kelas
    program "1" -- "1" dosen
    program "1" -- "1" mahasiswa
    program "1" -- "1" status
    program "1" -- "1" mastakuliah
    program "1" -- "1" jenismatkul
    program "1" -- "1" pengumuman
    program "1" -- "1" diskusi

    kelas "1" -- "1..*" jadwalkuliah
    kelas "1" -- "1..*" monitoring
    kelas "1" -- "1..*" absensi
    kelas "1" -- "1..*" materi
    kelas "1" -- "1..*" soalkuis
    kelas "1" -- "1..*" kuis
    kelas "1" -- "1..*" nilai
    kelas "1" -- "1..*" nilaikuis
    kelas "1" -- "1..*" nilaitugas
    kelas "1" -- "1..*" tugas
    kelas "1" -- "1..*" komentar
    kelas "1" -- "1..*" monitoring
    kelas "1" -- "1..*" absensi
    kelas "1" -- "1..*" jadwalkuliah
    kelas "1" -- "1..*" kelas
    kelas "1" -- "1..*" dosen
    kelas "1" -- "1..*" mahasiswa
    kelas "1" -- "1..*" status
    kelas "1" -- "1..*" mastakuliah
    kelas "1" -- "1..*" jenismatkul
    kelas "1" -- "1..*" pengumuman
    kelas "1" -- "1..*" diskusi

    jadwalkuliah "1..*" -- "1..*" monitoring
    jadwalkuliah "1..*" -- "1..*" absensi
    jadwalkuliah "1..*" -- "1..*" materi
    jadwalkuliah "1..*" -- "1..*" soalkuis
    jadwalkuliah "1..*" -- "1..*" kuis
    jadwalkuliah "1..*" -- "1..*" nilai
    jadwalkuliah "1..*" -- "1..*" nilaikuis
    jadwalkuliah "1..*" -- "1..*" nilaitugas
    jadwalkuliah "1..*" -- "1..*" tugas
    jadwalkuliah "1..*" -- "1..*" komentar
    jadwalkuliah "1..*" -- "1..*" monitoring
    jadwalkuliah "1..*" -- "1..*" absensi
    jadwalkuliah "1..*" -- "1..*" jadwalkuliah
    jadwalkuliah "1..*" -- "1..*" kelas
    jadwalkuliah "1..*" -- "1..*" dosen
    jadwalkuliah "1..*" -- "1..*" mahasiswa
    jadwalkuliah "1..*" -- "1..*" status
    jadwalkuliah "1..*" -- "1..*" mastakuliah
    jadwalkuliah "1..*" -- "1..*" jenismatkul
    jadwalkuliah "1..*" -- "1..*" pengumuman
    jadwalkuliah "1..*" -- "1..*" diskusi

    monitoring "1..*" -- "1..*" absensi
    monitoring "1..*" -- "1..*" materi
    monitoring "1..*" -- "1..*" soalkuis
    monitoring "1..*" -- "1..*" kuis
    monitoring "1..*" -- "1..*" nilai
    monitoring "1..*" -- "1..*" nilaikuis
    monitoring "1..*" -- "1..*" nilaitugas
    monitoring "1..*" -- "1..*" tugas
    monitoring "1..*" -- "1..*" komentar
    monitoring "1..*" -- "1..*" monitoring
    monitoring "1..*" -- "1..*" absensi
    monitoring "1..*" -- "1..*" jadwalkuliah
    monitoring "1..*" -- "1..*" kelas
    monitoring "1..*" -- "1..*" dosen
    monitoring "1..*" -- "1..*" mahasiswa
    monitoring "1..*" -- "1..*" status
    monitoring "1..*" -- "1..*" mastakuliah
    monitoring "1..*" -- "1..*" jenismatkul
    monitoring "1..*" -- "1..*" pengumuman
    monitoring "1..*" -- "1..*" diskusi

    absensi "1..*" -- "1..*" materi
    absensi "1..*" -- "1..*" soalkuis
    absensi "1..*" -- "1..*" kuis
    absensi "1..*" -- "1..*" nilai
    absensi "1..*" -- "1..*" nilaikuis
    absensi "1..*" -- "1..*" nilaitugas
    absensi "1..*" -- "1..*" tugas
    absensi "1..*" -- "1..*" komentar
    absensi "1..*" -- "1..*" monitoring
    absensi "1..*" -- "1..*" absensi
    absensi "1..*" -- "1..*" jadwalkuliah
    absensi "1..*" -- "1..*" kelas
    absensi "1..*" -- "1..*" dosen
    absensi "1..*" -- "1..*" mahasiswa
    absensi "1..*" -- "1..*" status
    absensi "1..*" -- "1..*" mastakuliah
    absensi "1..*" -- "1..*" jenismatkul
    absensi "1..*" -- "1..*" pengumuman
    absensi "1..*" -- "1..*" diskusi

    materi "1..*" -- "1..*" soalkuis
    materi "1..*" -- "1..*" kuis
    materi "1..*" -- "1..*" nilai
    materi "1..*" -- "1..*" nilaikuis
    materi "1..*" -- "1..*" nilaitugas
    materi "1..*" -- "1..*" tugas
    materi "1..*" -- "1..*" komentar
    materi "1..*" -- "1..*" monitoring
    materi "1..*" -- "1..*" absensi
    materi "1..*" -- "1..*" jadwalkuliah
    materi "1..*" -- "1..*" kelas
    materi "1..*" -- "1..*" dosen
    materi "1..*" -- "1..*" mahasiswa
    materi "1..*" -- "1..*" status
    materi "1..*" -- "1..*" mastakuliah
    materi "1..*" -- "1..*" jenismatkul
    materi "1..*" -- "1..*" pengumuman
    materi "1..*" -- "1..*" diskusi

    soalkuis "1..*" -- "1..*" kuis
    soalkuis "1..*" -- "1..*" nilai
    soalkuis "1..*" -- "1..*" nilaikuis
    soalkuis "1..*" -- "1..*" nilaitugas
    soalkuis "1..*" -- "1..*" tugas
    soalkuis "1..*" -- "1..*" komentar
    soalkuis "1..*" -- "1..*" monitoring
    soalkuis "1..*" -- "1..*" absensi
    soalkuis "1..*" -- "1..*" jadwalkuliah
    soalkuis "1..*" -- "1..*" kelas
    soalkuis "1..*" -- "1..*" dosen
    soalkuis "1..*" -- "1..*" mahasiswa
    soalkuis "1..*" -- "1..*" status
    soalkuis "1..*" -- "1..*" mastakuliah
    soalkuis "1..*" -- "1..*" jenismatkul
    soalkuis "1..*" -- "1..*" pengumuman
    soalkuis "1..*" -- "1..*" diskusi

    kuis "1..*" -- "1..*" nilai
    kuis "1..*" -- "1..*" nilaikuis
    kuis "1..*" -- "1..*" nilaitugas
    kuis "1..*" -- "1..*" tugas
    kuis "1..*" -- "1..*" komentar
    kuis "1..*" -- "1..*" monitoring
    kuis "1..*" -- "1..*" absensi
    kuis "1..*" -- "1..*" jadwalkuliah
    kuis "1..*" -- "1..*" kelas
    kuis "1..*" -- "1..*" dosen
    kuis "1..*" -- "1..*" mahasiswa
    kuis "1..*" -- "1..*" status
    kuis "1..*" -- "1..*" mastakuliah
    kuis "1..*" -- "1..*" jenismatkul
    kuis "1..*" -- "1..*" pengumuman
    kuis "1..*" -- "1..*" diskusi

    nilai "1..*" -- "1..*" nilaikuis
    nilai "1..*" -- "1..*" nilaitugas
    nilai "1..*" -- "1..*" tugas
    nilai "1..*" -- "1..*" komentar
    nilai "1..*" -- "1..*" monitoring
    nilai "1..*" -- "1..*" absensi
    nilai "1..*" -- "1..*" jadwalkuliah
    nilai "1..*" -- "1..*" kelas
    nilai "1..*" -- "1..*" dosen
    nilai "1..*" -- "1..*" mahasiswa
    nilai "1..*" -- "1..*" status
    nilai "1..*" -- "1..*" mastakuliah
    nilai "1..*" -- "1..*" jenismatkul
    nilai "1..*" -- "1..*" pengumuman
    nilai "1..*" -- "1..*" diskusi

    nilaikuis "1..*" -- "1..*" nilaitugas
    nilaikuis "1..*" -- "1..*" tugas
    nilaikuis "1..*" -- "1..*" komentar
    nilaikuis "1..*" -- "1..*" monitoring
    nilaikuis "1..*" -- "1..*" absensi
    nilaikuis "1..*" -- "1..*" jadwalkuliah
    nilaikuis "1..*" -- "1..*" kelas
    nilaikuis "1..*" -- "1..*" dosen
    nilaikuis "1..*" -- "1..*" mahasiswa
    nilaikuis "1..*" -- "1..*" status
    nilaikuis "1..*" -- "1..*" mastakuliah
    nilaikuis "1..*" -- "1..*" jenismatkul
    nilaikuis "1..*" -- "1..*" pengumuman
    nilaikuis "1..*" -- "1..*" diskusi

    nilaitugas "1..*" -- "1..*" tugas
    nilaitugas "1..*" -- "1..*" komentar
    nilaitugas "1..*" -- "1..*" monitoring
    nilaitugas "1..*" -- "1..*" absensi
    nilaitugas "1..*" -- "1..*" jadwalkuliah
    nilaitugas "1..*" -- "1..*" kelas
    nilaitugas "1..*" -- "1..*" dosen
    nilaitugas "1..*" -- "1..*" mahasiswa
    nilaitugas "1..*" -- "1..*" status
    nilaitugas "1..*" -- "1..*" mastakuliah
    nilaitugas "1..*" -- "1..*" jenismatkul
    nilaitugas "1..*" -- "1..*" pengumuman
    nilaitugas "1..*" -- "1..*" diskusi

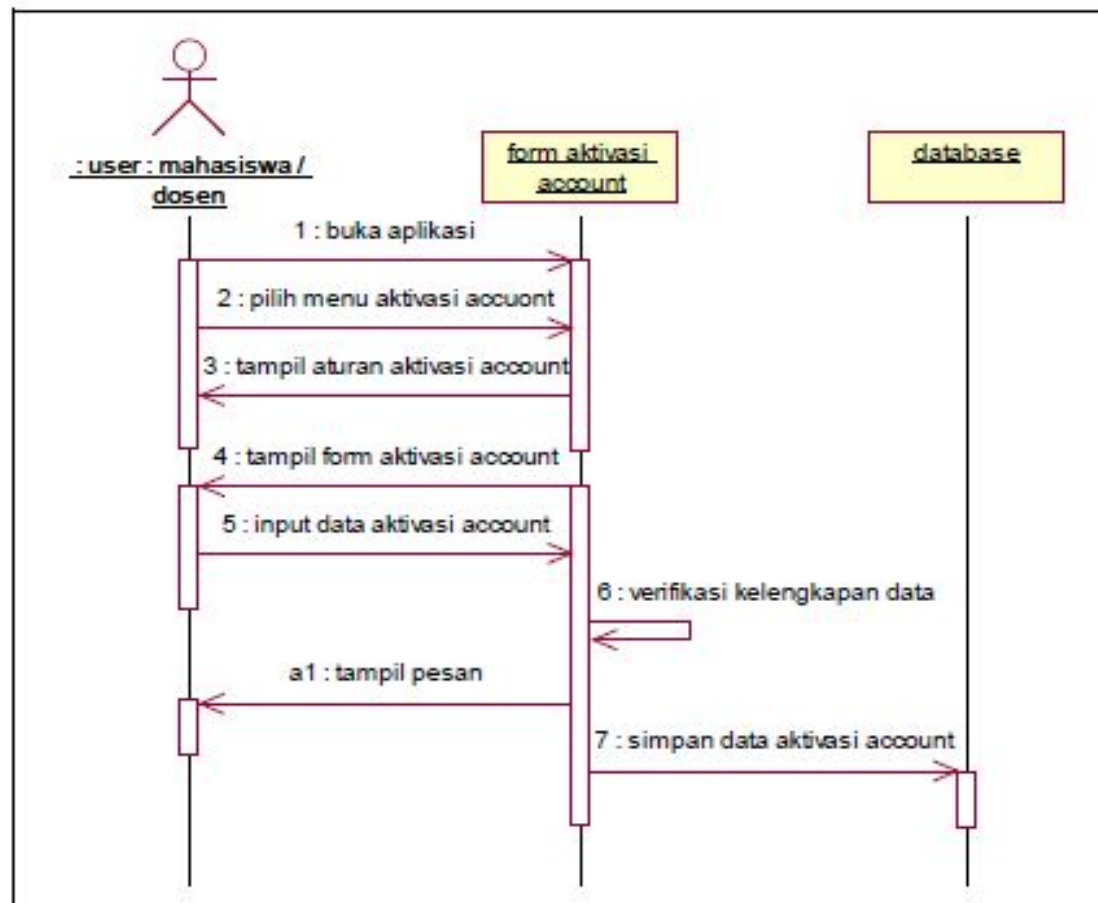
    tugas "1..*" -- "1..*" komentar
    tugas "1..*" -- "1..*" monitoring
    tugas "1..*" -- "1..*" absensi
    tugas "1..*" -- "1..*" jadwalkuliah
    tugas "1..*" -- "1..*" kelas
    tugas "1..*" -- "1..*" dosen
    tugas "1..*" -- "1..*" mahasiswa
    tugas "1..*" -- "1..*" status
    tugas "1..*" -- "1..*" mastakuliah
    tugas "1..*" -- "1..*" jenismatkul
    tugas "1..*" -- "1..*" pengumuman
    tugas "1..*" -- "1..*" diskusi

    komentar "1..*" -- "1..*" monitoring
    komentar "1..*" -- "1..*" absensi
    komentar "1..*" -- "1..*" jadwalkuliah
    komentar "1..*" -- "1..*" kelas
    komentar "1..*" -- "1..*" dosen
    komentar "1..*" -- "1..*" mahasiswa
    komentar "1..*" -- "1..*" status
    komentar "1..*" -- "1..*" mastakuliah
    komentar "
```

Gambar 4.35 Diagram Kelas Sistem Aplikasi Kuliah Online

Sequence Diagram

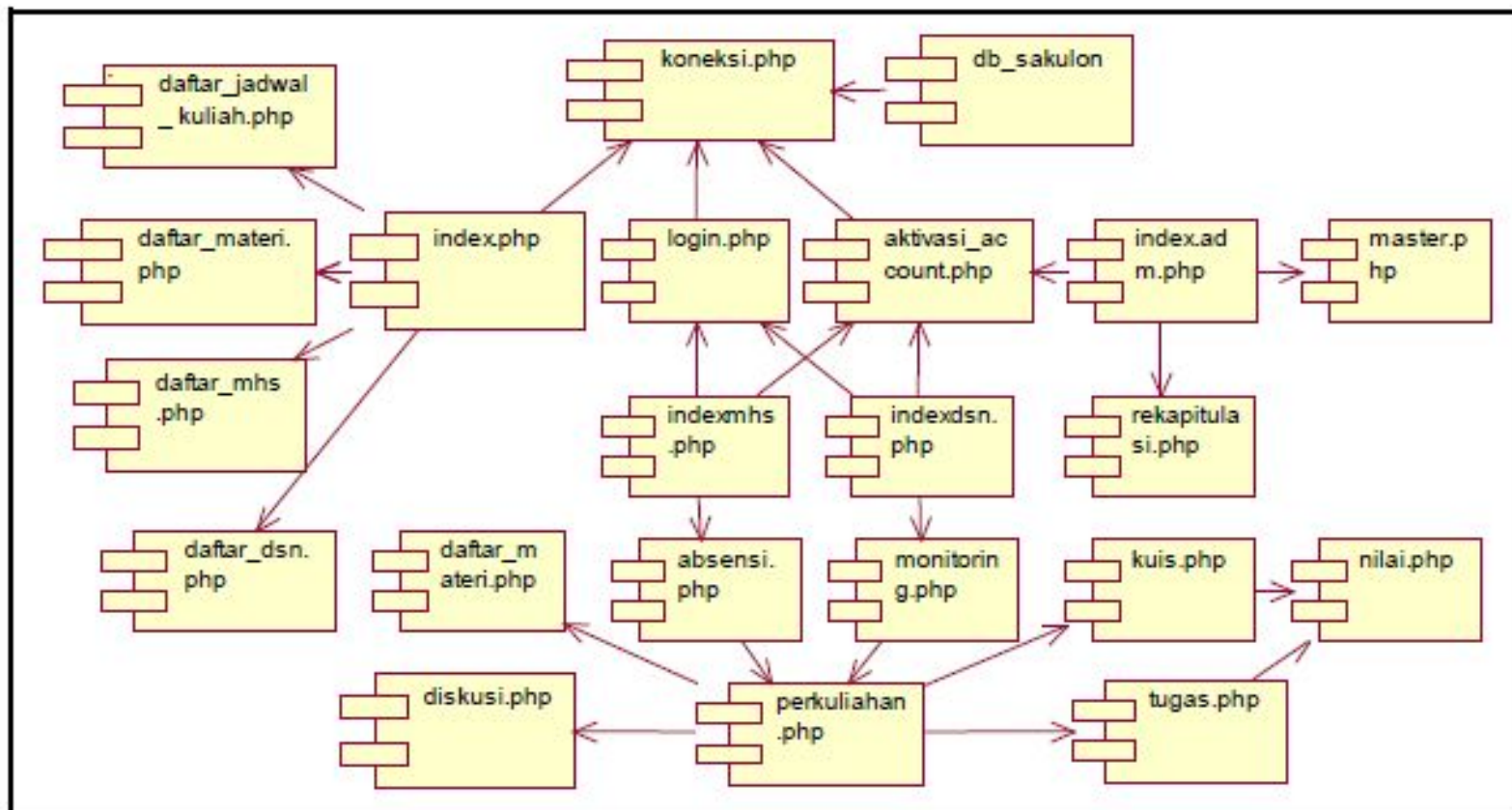
Diagram sekuensial atau sequence diagram digunakan untuk menunjukkan aliran fungsionalitas dalam use case. Sequence diagram digunakan untuk memberikan gambaran detail dari setiap use case diagram yang telah dibuat sebelumnya.



Gambar 4.23 Diagram Sekuensial Aktivasi Account untuk Mahasiswa dan Dosen

Component Diagram

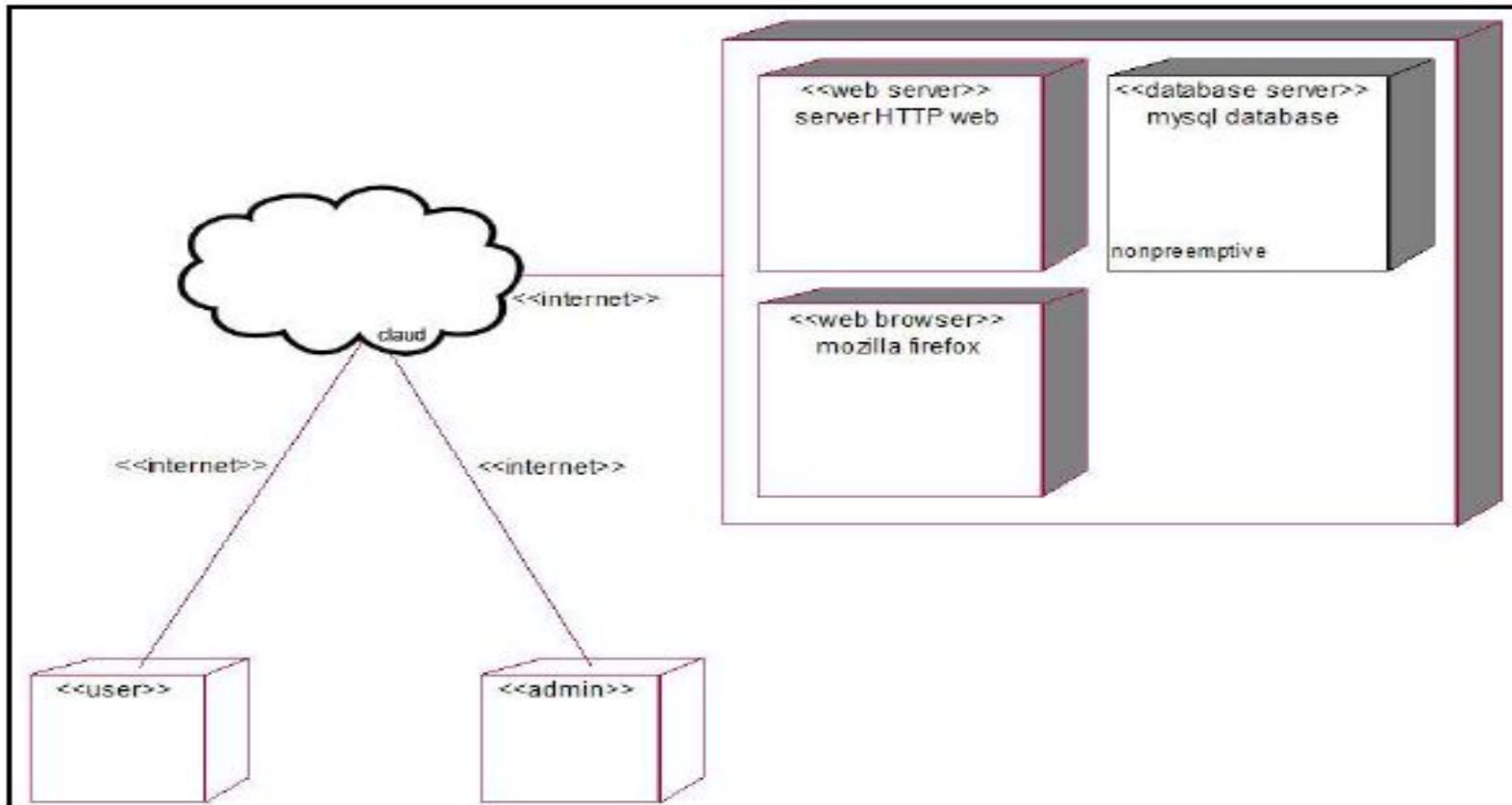
Diagram komponen atau component diagram menunjukkan model secara fisik komponen perangkat lunak pada sistem dan hubungannya.



Gambar 4.37 Diagram Komponen Sistem Aplikasi Kuliah Online

Deployment Diagram

Diagram deployment atau deployment diagram menampilkan rancangan fisik jaringan dimana berbagai komponen akan terdapat disana.



Gambar 4.38 Diagram *Deployment* Sistem Aplikasi Kuliah Online