

## **PERTEMUAN 9**

### **PENGENALAN UML**

#### **A. TUJUAN PEMBELAJARAN**

Pada pertemuan ini menjelaskan tentang pengenalan serta penjelasan UML, dan di sertakan penjelasan bagian – bagian dari UML.

#### **B. URAIAN MATERI**

##### **1. Pengenalan UML**

UML ( Unified Modeling Language ) adalah Bahasa pada memodelan atau pemodelan visual umum digunakan untuk menentukan, memvisualisasikan, membuat, dan merekam artefak dalam software. Ini menangkap keputusan dan pemahaman tentang sistem yang akan dibangun. Ini digunakan untuk memahami, merancang, mengeksplorasi, mengkonfigurasi, memelihara, dan mengontrol informasi tentang sistem. Ini dirancang untuk semua metode pengembangan, tahapan siklus hidup, domain aplikasi, dan media. Bahasa Pemodelan ini bertujuan untuk mengumpulkan pengalaman dalam teknik pemodelan masa lampau dan menggabungkan praktik terbaik perangkat lunak saat ini ke dalam metode standar.

UML mencakup simbol, konsep dan pedoman semantik. Ini memiliki bagian dinamis, statis lingkungan dan organisasi. Ini dirancang untuk didukung oleh alat pemodelan visual interaktif dengan generator kode dan penulis laporan. Spesifikasi UML tidak menentukan proses standar, tetapi dimaksudkan untuk digunakan dengan proses pengembangan berulang. Ini dirancang untuk mendukung sebagian besar proses perkembangan berorientasi objek saat ini.

UML memberikan informasi tentang struktur statis dan perilaku dinamis sistem. Sistem dimodelkan sebagai kumpulan objek individu yang berinteraksi untuk menyelesaikan pekerjaan dan pada akhirnya menguntungkan pengguna eksternal. Struktur statis mendefinisikan jenis objek yang penting bagi sistem dan implementasinya, serta hubungan antar objek. Perilaku dinamis mendefinisikan sejarah objek yang berubah seiring waktu dan komunikasi antar

objek untuk mencapai tujuan. Pemodelan sistem dari beberapa perspektif independen tetapi terkait dapat memahami berbagai tujuan sistem.

UML juga berisi struktur organisasi untuk mengatur model ke dalam paket perangkat lunak yang memungkinkan tim perangkat lunak untuk membagi sistem besar menjadi beberapa bagian yang dapat digunakan untuk memahami dan mengontrol ketergantungan antara paket perangkat lunak dan mengelola lingkungan pengembangan yang kompleks. Versi unit model. Ini berisi struktur yang digunakan untuk mewakili keputusan implementasi dan mengatur elemen runtime menjadi komponen.

UML bukanlah bahasa pemrograman. Alat dapat menyediakan generator kode dari UML ke berbagai bahasa pemrograman, dan juga dapat membangun model rekayasa terbalik dari program yang ada. UML bukanlah bahasa yang sangat formal untuk membuktikan teorema. Ada banyak bahasa seperti itu, tetapi untuk sebagian besar tujuan, bahasa tersebut tidak mudah dipahami atau digunakan. UML adalah bahasa pemodelan universal. Untuk bidang profesional seperti tata letak GUI, desain sirkuit VLSI, atau kecerdasan buatan berbasis aturan, alat yang lebih profesional menggunakan bahasa tertentu mungkin sesuai.

UML adalah bahasa pemodelan diskrit. Ini tidak dimaksudkan untuk memodelkan sistem berkelanjutan di bidang teknik dan fisika. UML bertujuan untuk menjadi bahasa pemodelan tujuan umum untuk sistem diskrit (misalnya, sistem yang dibuat oleh perangkat lunak, firmware, atau logika digital).

## **2. Pengertian UML**

UML adalah bahasa yang digunakan untuk mendefinisikan, memvisualisasikan, membangun, dan merekam artefak (bagian dari informasi yang digunakan atau dihasilkan dalam proses pemodelan perangkat lunak, artefak ini dapat berupa model, deskripsi, atau perangkat lunak), seperti pemodelan bisnis dan alat non-sistem, dll. . . Selain itu, UML merupakan bahasa pemodelan yang menggunakan konsep berorientasi objek. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp.

Notasi yang diberikan oleh UML dapat membantu Anda memodelkan sistem dari berbagai sudut. UML tidak hanya digunakan untuk pemodelan perangkat lunak, tetapi UML digunakan di hampir semua area yang membutuhkan pemodelan

### **3. Bagian - bagian dari UML**

Bagian utama dari UML adalah view, diagram, elemen model dan mekanisme umum.

#### **a. View**

View di pergunakan untuk melihat sistem model dari berbagai aspek. Tampilannya bukanlah grafik, tetapi abstraksi yang mengandung diagram.

Beberapa jenis view di UML meliputi: use case view, logical view, component

Use case view

Menjelaskan fungsi yang akan digunakan oleh peserta eksternal. Peserta yang berinteraksi dengan sistem dapat berupa pengguna atau sistem lain, dan pandangan tersebut dijelaskan dalam bentuk diagram use case atau diagram aktivitas..

View ini terutama digunakan oleh pelanggan, desainer, pengembang, dan penguji sistem.

Logical view

Mengdeskripsikan bagaimana fungsi Sistem, pola statis (class, object, dan relationship), dan kerja sama dinamis kelahirannya saat selaur korban mengangakat suruhan ke korban lain tambah kelebihan tertentu.

View tersebut dideskripsikan sebagai struktur statis berupa diagram class, dan model dinamis berupa diagram state, sequence, collaboration, dan activity diagram. View ini untuk digunakan oleh desainer dan pengembang.

Componen View

Implementasi dan dependensi modul. Komponen sebagai jenis modul kode lainnya mewakili struktur dan ketergantungannya, serta alokasi sumber daya komponen dan informasi manajemen lainnya.

Vew ini dideskripsikan sebagai tampilan komponen untuk digunakan

pengembang (developer).

#### Concurrency view

Pembagian sistem menjadi proses dan prosesor. View ini diwakili oleh skema dinamis (state, sequence, collaboration dan activity skema) dan skema implementasi (component dan deployment skema), dan digunakan oleh pengembang, integrator, dan penguji.

#### Deployment view

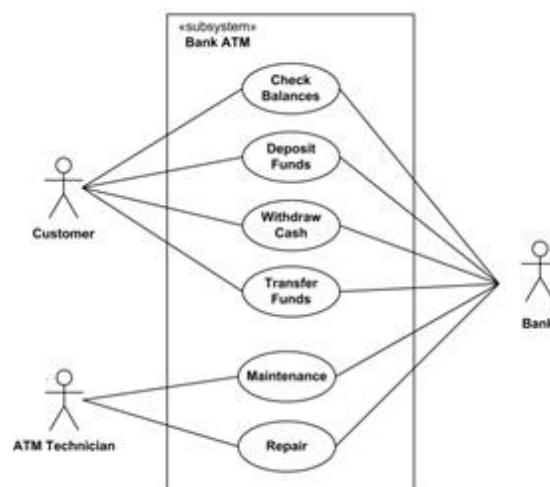
Mendekripsikan fisik sistem sebagai komputer dan perangkat (nodes) dan sebagai apa hubungannya dengan sistem lain. View ini dijelaskan dalam deployment diagram dan digunakan oleh pengembang, integrator, dan penguji.

### b. Diagram

Diagram grafik menunjukkan simbol elemen model, yang disusun untuk menggambarkan bagian atau aspek tertentu dari sistem. Grafik adalah bagian dari tampilan tertentu, dan biasanya ditetapkan ke view tertentu saat menggambar. Jenis diagram antara lain :

#### Use Case Diagram

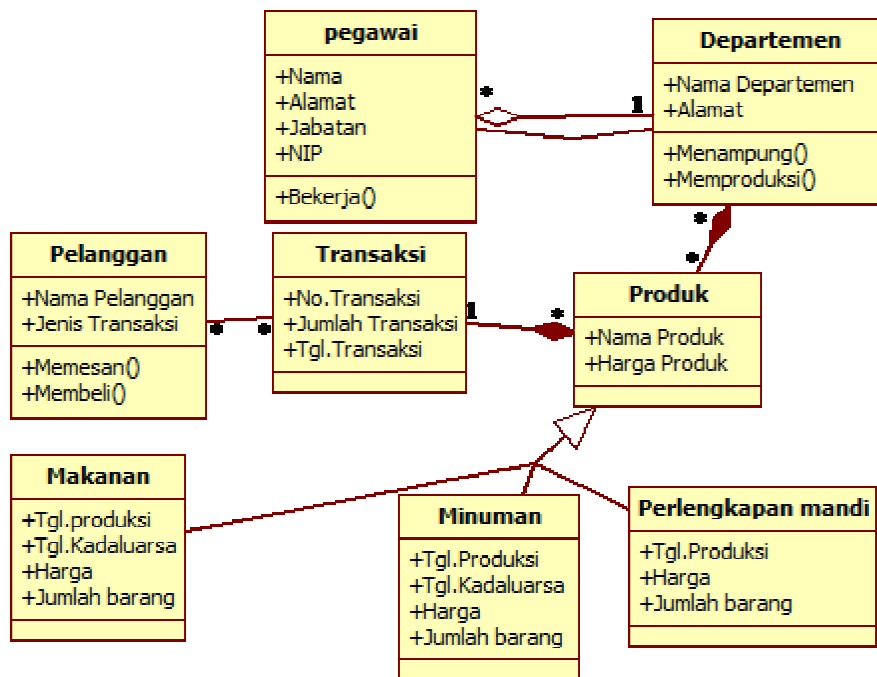
Menggambarkan sejumlah actor external dan hubungannya pakai use case yang diberikan oleh sistem. Use case adalah deskripsi fungsional yang disediakan oleh sistem bagian dalam bentuk teks sebagai dokumen simbol use case, tetapi juga bisa dilengkapi dalam activity diagram.



Gambar 9.3.2.1 Contoh Use Case Diagram

### Class Diagram

Jelaskan struktur kelas statis internal sistem. Kelas mewakili item yang ditangani oleh sistem. Kelas dapat dihubungkan satu sama lain dalam berbagai cara: asosiasi (terhubung satu sama lain), ketergantungan (kelas bergantung pada/menggunakan keluarga lain), khusus (satu keluarga adalah keluarga lain) atau paket (digabungkan menjadi satu unit)) Sebuah sistem biasanya memiliki beberapa diagram kelas.

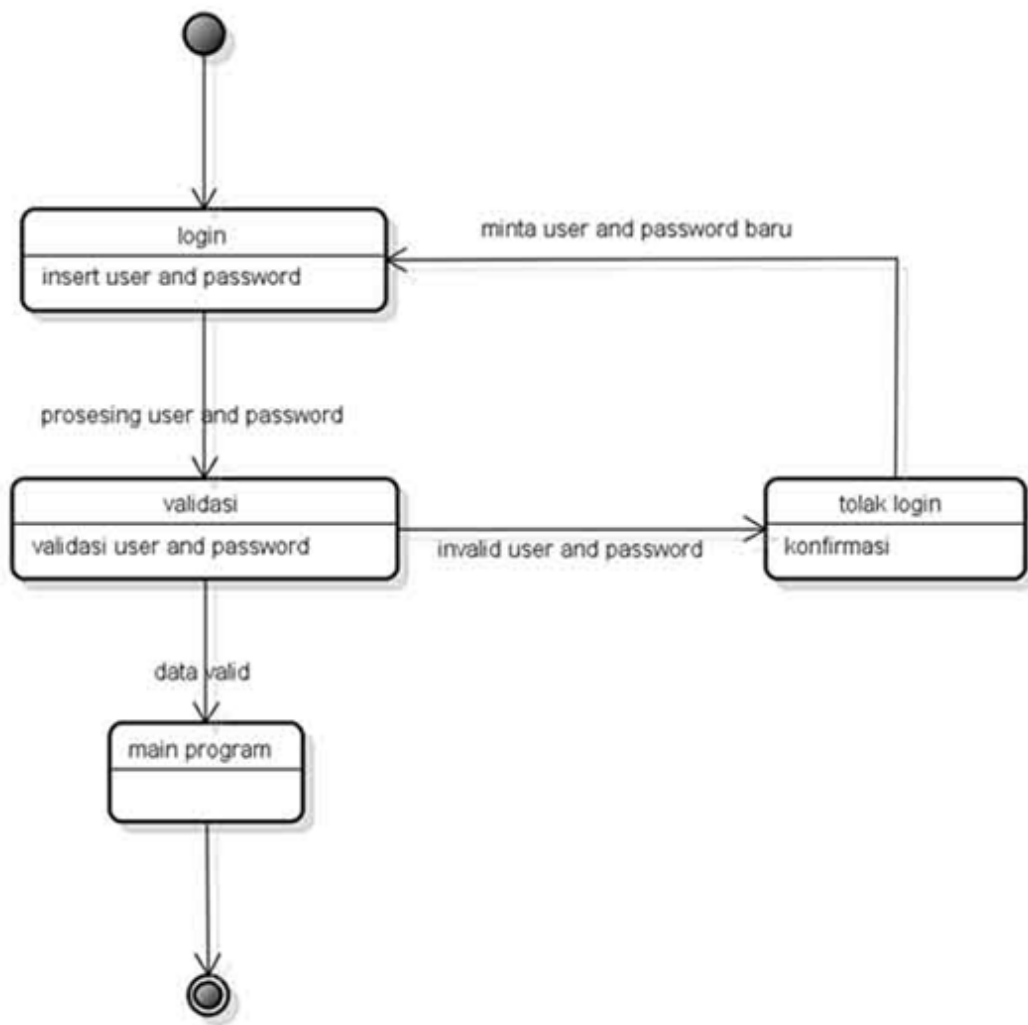


Gambar 9.3.2.2 Contoh Class Diagram

### State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh objek class dan ihwal yang menyebabkan suasana berubah. Acara bisa berupa objek lain yang mengirimkan pesan.

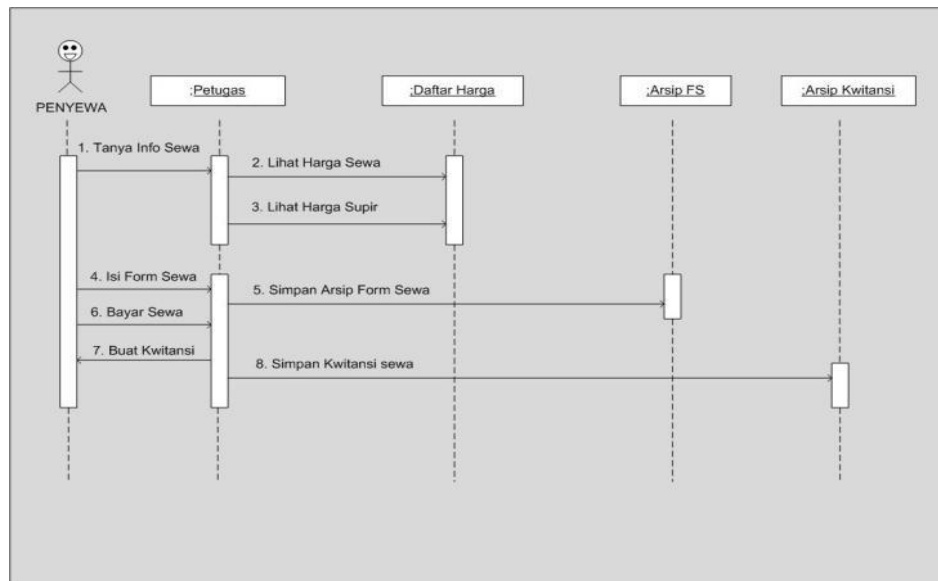
State class tidak di gambarkan semua class, semata-mata yang memegang beberapa state yang terdefinisi pakai kesetiaan dan mengenai class merangkak oleh state yang berbeda.



Gambar 9.3.2.3 Contoh State Diagram

#### 1) Sequence Diagram

Menggambarkan kolaborasi dinamis sela banyak objek. Tujuannya adalah kepada mempresentasikan alur instruksi yang dikirim antar tujuan dan koneksi antar sasaran, yang kelahirannya pada titik tertentu bagian dalam eksekusi sistem.

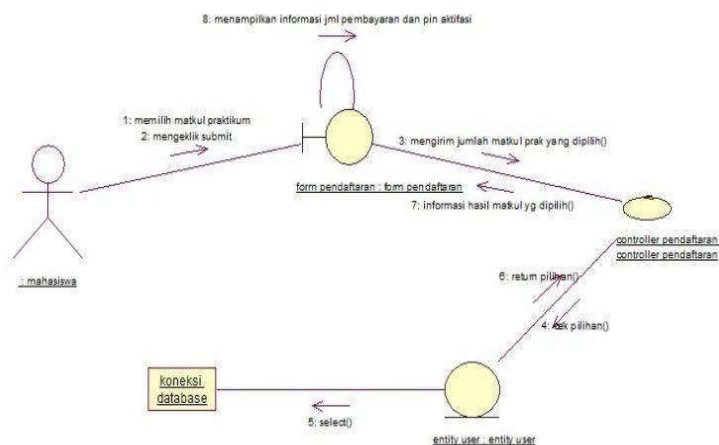


*Gambar 9.3.2.4 Contoh Sequence Diagram*

## 2) Collaboration Diagram

Menggambarkan kolaborasi dinamis, serupa rancangan urutan. Saat mempresentasikan pergantian pesan, collaboration rangka menceritakan tujuan dan hubungannya (mengacu ambang konteks).

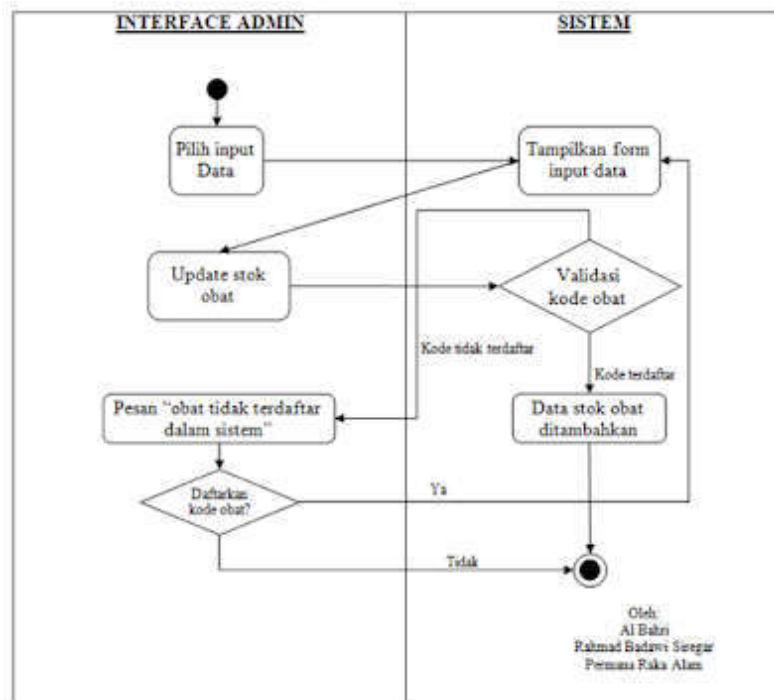
Jika fokus berada pada waktu atau urutan, gunakansequence diagram, tetapi jika fokus pada konteks,Gunakan collaboration diagram.



*Gambar 9.3.2.5 Contoh Collaboration Diagram*

### 3) Activity Diagram

Mendeskripsikan rangkaian proses dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang terbentuk dalam operasi, sehingga dapat juga digunakan untuk aktivitas lain, seperti use case atau interaksi

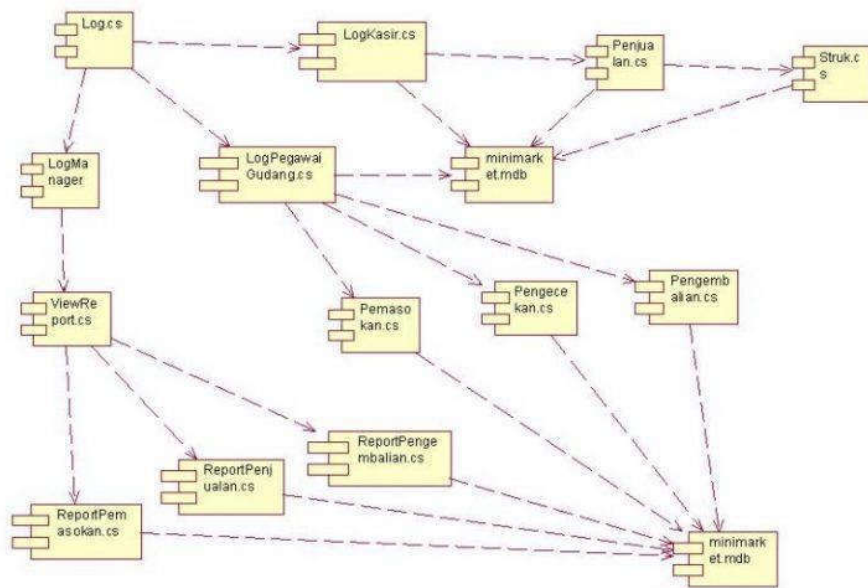


Gambar 9.3.2.1 Contoh Activity Diagram

### 4) Componen Diagram

Menggambarkan struktur kode fisik komponen. Komponen bisa bercorak source code, elemen biner, atau executable. Komponen mengandung logic class atau class yang diimplementasikan kepada menegakkan pemetaan berasal view logical ke component view.





*Gambar 9.3.2.7 Contoh Component Diagram*

#### 4. Gambaran dari UML

##### a. UML sebagai Bahasa Pemodelan

UML adalah bahasa pemodelan, mempunyai kosakata, dan menjadikan cara kepada mengeluarkan kekhawatiran mengenai konsep sistem dan fisika. Contoh sistem software intensif tekanan suara menunjukkan pandangan yang berbeda bersumber arsitektur sistem, mirip tambah siklus kehidupan pengembangan perangkat lunak kompilasi / pengembangan. Menggunakan UML akan memberi mengerti Anda cara membuat dan mempersembahkan formulir model yang baik, tetapi UML tidak bisa memasukkan mengerti kaca apa yang akan dibangun dan kapan harus bermanfaat kaca. Ini adalah institusi bagian dalam muslihat pengembangan perangkat lunak.

##### b. UML untuk Mengambarkan system

UML tidak hanya berupa rangkaian simbol grafik, tetapi setiap simbol bagian dalam pesan UML mengadakan distribusi semantik yang baik. Dengan hukum ini, satu dam dapat menggambar arketipe UML, sementara dam lain atau alat serupa lainnya bisa menafsirkan arketipe dengan jelas. Ini akan mengurangi kekhilafan yang disebabkan oleh friksi irama bagian dalam

persinggungan antara arketipe konseptual dan model lainnya.

UML menggambarkan model yang dapat dipahami dan disajikan dalam model teks suatu bahasa pemrograman. Anda dapat menebak contoh dalam model sistem berbasis web, tetapi Anda tidak dapat langsung memproses contoh tersebut dengan mempelajari kode sistem. Dengan menggunakan model UML, Anda dapat menggunakan bahasa pemrograman untuk membuat model dan merepresentasikan sistem Web.

c. UML untuk Menspesifikasikan system

Ini berarti membuat model yang mirip, eksplisit Dan lengkap. faktanya, uml mewakili seluruh pembahasan uraian penting, desain, dan perintah penjelmaan yang harus di buat tatkala peluasan dan operasi software intensif.

d. UML untuk membangun system

UML tidak suatu intonasi pemograman visual, tapi untuk pemodelan UML dapat di sambungkan secara langsung menjelang bahasa pemograman visual.

Bermaksud untuk berguna sama anutan yang bisa dimapping ke tekanan pemograman sebagai java, VB, C++ ataupun table ambang database relational atau pemilihan tetap depan database berorientasi object.

e. UML untuk pendokumentasikan system

Dengan kata lain, UML menunjukkan dokumen dan semua detail arsitektur sistem, uml menyediakan bahasa menampilkan permintaan dan tes. uml menyediakan bahasa untuk kegiatan pemodelan dalam rencana proyek dan mnejemen rilis.

## 5. Area Penggunaan UML

UML di Gunakan paling efektif pada domain seperti :

- a. System Informasi Perusahaan
- b. System Pernagkan , Perekonomian
- c. System bidang telekomuniikasi
- d. System bidang transportasi

- e. System bidang penerbangan
- f. System bidang perdagangan
- g. System bidang pelayanan elektronik
- h. System bidang pengetahuan
- i. System bidang pelayanan berbasis web

Tetapi uml tidak terbatas bagi pemodelan software pada faktanya uml banyak untuk pemodelan system non software seperti :

- a. Bagian dari kerja pada system perundangan
- b. Bagian struktural juga kekuatan dari sebuah system kepedulian kesehatan pasien
- c. Bagian dari desain hardware

## **6. Komponen – komponen yang terlibat dalam Use Case Diagram**

### **a. Actor**

Dengan kata lain, UML menunjukkan dokumen dan semua detail arsitektur sistem, uml juga menyediakan bahasa untuk menampilkan permintaan dan tes. UML menyediakan bahasa untuk kegiatan pemodelan dalam perencanaan proyek dan manajemen rilis..

Ada banyak kemungkinan yang menyebabkan partisipasi ini terkait dengan sistem, diantaranya :

- 1) Bagi yang tertarik dengan sistem, ada arus informasi dan arus informasi yang masuk ke dalam sistem
- 2) Orang atau kelompok yang akan mengelola sistem.
- 3) Sumber daya eksternal yang digunakan oleh sistem.
- 4) Sistem lain yang berinteraksi dengan sistem yang akan dibuat.

### **b. Use Case**

Use Case merupakan bentuk dari fungsi sistem agar pelanggan atau pengguna sistem dapat memahami dan memahami fungsi sebuah system yang akan di buat.

Use Case juga menggambarkan dari sistem perspektif users (pengguna), jadi use case lebih fokus pada fungsi sistem yang ada, daripada proses berbasis peristiwa atau urutan peristiwa.

### c. Relasi dalam Use Case

Ada beberapa relasi terdapat pada use case diagram :

- 1) Asosiasi, menghubungkan link antar elemen.
- 2) Generalisasi, juga dikenal sebagai pewarisan, suatu elemen dapat menjadi spesialisasi elemen lainnya.
- 3) Ketergantungan, suatu elemen bergantung pada elemen lain dalam beberapa cara.
- 4) Agregasi, suatu bentuk asosiasi di mana satu elemen mengandung elemen lainnya

### d. Use Case Diagram

Use case diagram adalah bentuk grafis dari beberapa atau semua actor ,Use case, dan interaksi di antaranya Yang di perkenalkan dari suatu system .

## 7. Class Diagram

### a. Definisi Objek dan Class

Objek adalah deskripsi dari suatu entitas, apakah itu dunia nyata atau konsep dengan batasan dan definisi yang tepat. Objek dapat mewakili hal-hal yang berwujud, seperti komputer, mobil, atau konsep seperti pemrosesan bahan kimia, transaksi bank, dan pesanan pembelian. Setiap objek dalam sistem memiliki tiga ciri yaitu state (status), behaviour (atribut) dan identity (identitas).

Cara identifikasi :

- 1) mengelompokan berdasarkan kata/frase benda pada skenario.
- 2) Berdasarkan daftar kategori objek, antara lain:
  - a) Benda-benda fisik, misalnya: telepon
  - b) Spesifikasi/desain/deskripsi, misalnya: deskripsi pesawat terbang
  - c) Lokasi, misalnya: gudang Transaksi, misalnya: penjualan
  - d) Barang-barang yang terlibat dalam transaksi, misalnya: penjualan barang
  - e) Peran, misalnya: pelanggan
  - f) Kontainer, seperti pesawat terbang

- g) Perlengkapan, misalnya: PABX
- h) Kata benda abstrak, seperti: kecanduan
- i) Acara, misalnya: pendaratan
- j) Aturan atau kebijakan, misalnya: aturan diskon
- k) Direktori atau bahan referensi, misalnya: daftar pelanggan

Class adalah deskripsi sekelompok objek berdasarkan hubungan antara atribut (atribut), atribut (operasi), objek, dan semantik umum. Class adalah templat yang digunakan untuk membentuk objek. Setiap objek adalah contoh dari banyak class, dan sebuah objek tidak bisa menjadi turunan dari lebih dari satu class.

Dalam UML, sebuah class diwakili oleh persegi yang terbagi. Di bagian atas adalah nama class. Bagian tengah adalah struktur class (atribut), dan bagian bawah adalah sifat class (operasi) .

#### b. Status( State ), Behaviour dan Identify

Keadaan suatu benda adalah syarat keberadaannya. Keadaan objek akan berubah seiring waktu dan ditentukan oleh beberapa property (atribut) dengan nilai atribut dan hubungan antara objek dan objek lainnya.

Sifat (Behaviour) menentukan bagaimana objek menanggapi permintaan dari objek lain dan mewakili setiap objek yang dapat diselesaikan. Properti ini diimplementasikan oleh banyak operasi objek.

Identitas (Identify) artinya setiap object yang unik Pada UML

Rational Objectory Process merekomendasikan pencarian kelas dalam sistem yang sedang dibangun dengan mencari kelas (batas, kontrol, dan entitas). Ketiga stereotipe ini menggambarkan sudut pandang model-view-controller, sehingga analis dapat membagi sistem dengan memisahkan sudut pandang dari domain dan kontrol yang dibutuhkan oleh sistem..

Karena proses analisis dan desain adalah proses berulang, daftar kelas akan berubah seiring waktu. Kelas awal mungkin bukan kelas yang akan diimplementasikan. Oleh karena itu, kelas kandidat biasanya digunakan untuk menggambarkan himpunan kelas awal yang ditemukan dalam sistem.

Merancang sebuah class diagram, Rational Unified Process, yang

merupakan hasil realisasi use case untuk mengembangkan Rational Objectory Process, yang menggambarkan realisasi setiap use case pada model use case. Jelaskan bagaimana menggunakan beberapa diagram untuk realisasi use case, termasuk diagram kelas dan diagram interaksi milik realisasi use case.

Untuk mengilustrasikan realisasi use case di sini, kita akan menggunakan diagram kelas yang termasuk dalam realisasi use case. Setiap use case yang ada dibagi lagi, sehingga Anda dapat melihat entitas mana yang terlibat dalam realisasi use case. Entitas akan menjadi kandidat kelas dalam grafik.

## **8. Interaction Diagram**

### **a. Use Case Realization**

Fungsi use case diwakili oleh alur . Skenario digunakan untuk mendeskripsikan bagaimana use case diimplementasikan sebagai Interaksi antar objek.

Realisasi use case menggambarkan realisasi setiap use case dalam model use case. Untuk menggambarkan bagaimana menggunakan beberapa diagram untuk realisasi kasus penggunaan, termasuk diagram kelas yang dimiliki oleh realisasi use case dan interaksi.

Diagram interaksi model digunakan untuk menggambarkan bagaimana objek bekerja bersama dalam beberapa cara. Interaction Diagram menggambarkan perilaku use case.

### **b. Sequence Diagram**

Diagram sequence menguraikan koneksi jarak berlebihan tujuan bagian dalam rentetan kronologis. Tujuannya adalah menjelang menuangkan untai wasiat yang dikirim antar tujuan dan koneksi antar tujuan yang kelahirannya hadirat flek terbatas bagian dalam eksekusi sistem.

Di UML, tujuan bagian dalam rancangan sequence diwakili oleh bujur sangkar yang mengandung label tujuan yang digarisbawahi. Pada tujuan terpendam 3 hukum nomenklatur yaitu: label tujuan, label tujuan dan class tempuh label class.

### c. Collaboration Diagram

Diagram collaboration adalah resam lain kepada menguraikan skrip sistem. Diagram ini mengilustrasikan afiliasi tujuan-tujuan yang teratur di seputar tujuan dan koneksi kisi-kisi esa tujuan tambah tujuan lainnya.

Collaboration diagram berisi :

- 1) Object yang menerangkan pakai segiempat.
- 2) Hubungan diantara object yang digambarkan pakai rel penghubung.
- 3) Pesan yang selalu di gambarkan pakai wacana dan pendar berbunga object yang menggotong titipan ke peserta titipan.

### d. Perbedaan Sequence Diagram dan Collaboration Diagram

Sequence skema menahan lembaga menjelang menyelidiki pokok berlandasan waktu (apa yang kelahirannya perdana kali, lepas apa yang kelahirannya). user akan lebih mudah menyampaikan dan memafhumi skema ragam ini. Oleh karena itu, ini sangat praktis bagian dalam babak pembahasan awal.

Sedangkan Collaboration Diagram cenderung memberikan pemandangan panorama selama kolaborasi, proses kolaborasi terdiri dari objek sekitar dan hubungan antar objek. Ketika hubungan desain direalisasikan, diagram tampaknya digunakan dalam tahap pengembangan dan desain.



## 9. State Transition Diagram dan Activity Diagram

### a. State Transiton Diagram

Use case dan skrip meninggalkan hukum kepada memaparkan sopan santun sistem, yaitu koneksi kisi-kisi target bagian dalam sistem. Terkadang terbiasa kepada memata-matai sopan santun dekat target. State transition penampang memperlihatkan nilai tunggal target, kejadian atau suruhan yang memicu perputaran berpokok tunggal nilai ke nilai lain, dan sikap yang dihasilkan berpokok transmudasi state.

State transition penampang Ini tidak akan dibuat kepada setiap class di sistem. Hanya buat instraction penampang kepada marga tambah sopan santun dinamis. Anda bisa menilik intraction penampang kepada

menetapkan dynamic target bagian dalam sistem, yaitu target yang memeluk dan menggotong berlebihan suruhan. State transition penampang juga sangat konstruktif kepada meneropong sopan santun whole class dan tandu control class.

Simbol	Nama	Keterangan
	State	State melambangkan keadaan sebuah objek, baik atribut itu sendiri maupun hubungan objek tersebut dengan objek lain dari sistem
	Transition	Transition melambangkan suatu kejadian yang menyebabkan perubahan state dalam sistem. Setiap State Transition merupakan penghubung 2 buah state.

Gambar 9.9.1 Simbol – simbol State Transition Diagram

#### b. States

State Ini adalah keadaan memuaskan kondisi tertentu selama umur objek, melakukan operasi tertentu atau menunggu suatu peristiwa. Keadaan suatu objek dapat dicirikan oleh nilai dari satu atau lebih atribut dari class.

State transition diagram Termasuk semua pesan yang dapat dikirim dan diterima dari objek. Skema ini merepresentasikan jalur melalui state transition diagram . Interval waktu antara dua pesan yang dikirim oleh suatu objek mewakili suatu keadaan. Oleh karena itu, tentukan diagram sequence untuk mengetahui keadaan objek (lihat jarak antar garis yang mewakili pesan yang diterima objek).

#### c. State Transitions

State transition Termasuk semua pesan yang dapat dikirim dan diterima dari objek. Skema ini merepresentasikan jalur melalui diagram transisi status. Interval waktu antara dua pesan yang dikirim oleh suatu objek mewakili suatu keadaan. Oleh karena itu, tentukan diagram sekuens untuk mengetahui keadaan objek (lihat jarak antar garis yang mewakili pesan yang diterima objek).



Ada dua metode untuk transisi state - otomatis dan non-otomatis. Transisi status otomatis terjadi saat status awal aktivitas selesai-tidak ada peristiwa yang terkait dengan transisi status tanpa nama. Transisi status non-otomatis yang disebabkan oleh peristiwa terkenal (dari objek atau di luar sistem). Kedua jenis transisi status dianggap membuat waktu menjadi nol dan tidak terputus. Transisi keadaan diwakili oleh panah, yang menunjuk dari keadaan awal ke keadaan berikutnya.

d. Special States

Dua keadaan khusus telah ditambahkan ke diagram transisi keadaan. Yang pertama adalah keadaan awal. Saat membuat objek, setiap grafik hanya boleh memiliki satu status awal

e. State Transition Details

Transisi status dapat memiliki kondisi operasi dan / atau perlindungan yang terkait dengannya, dan juga dapat memicu peristiwa. Tindakan adalah tindakan yang terjadi saat transisi keadaan terjadi. Peristiwa adalah pesan yang dikirim ke objek lain di sistem. Kondisi perlindungan adalah ekspresi Boolean dari nilai atribut, dan transisi status hanya diperbolehkan jika kondisinya benar. Tindakan dan perlindungan keduanya merupakan perilaku objek, dan biasanya menjadi operasi. Operasi ini biasanya terisolasi -yaitu, operasi ini hanya digunakan oleh objek itu sendiri.

f. State Details

Action –action yang menepi seluruh state transtation ke sewarna state ganjur di tempatkan seperti padu entry action episode bagian dalam state. Demikian juga, action – action yang mengiringi serata ahli state transition bertiup semenjak arah-arrah state bertelur di tempatkan seperti padu serbuan bergiat episode bagian dalam state. Kelakuan yang kelahirannya episode bagian dalam state disebut padu activity.

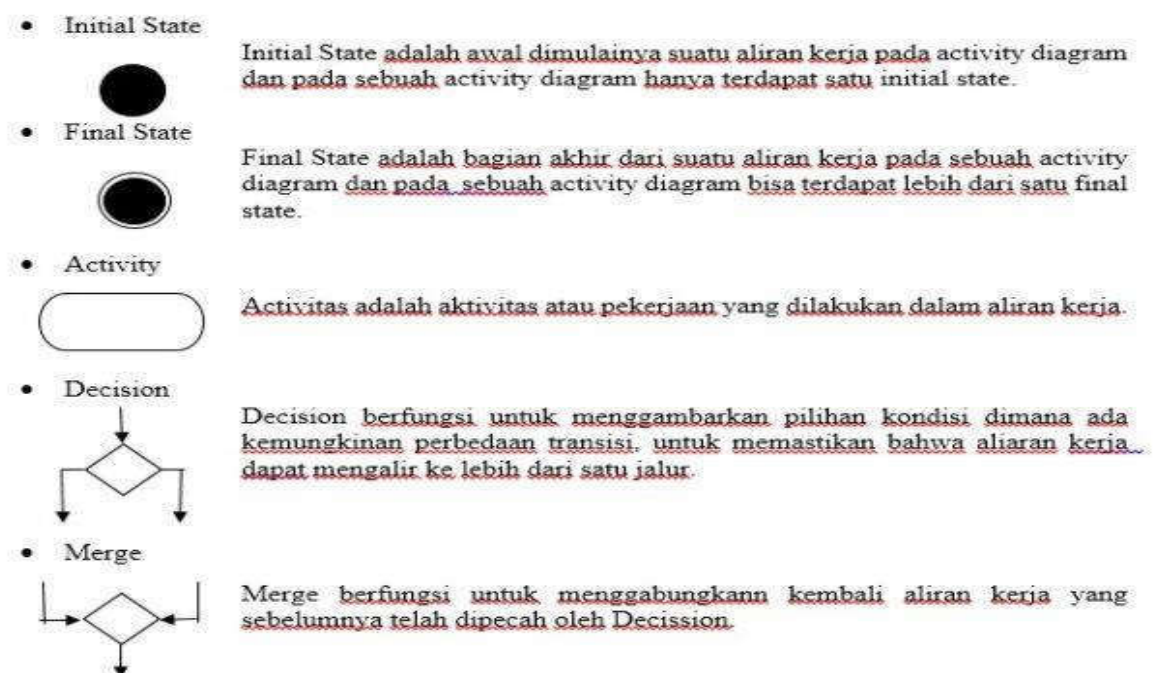
Perilaku tersebut dapat berupa tindakan sederhana, atau dapat berupa peristiwa yang dikirim ke objek lain. Menurut tindakan dan tindakan perlindungan, perilaku ini biasanya dipetakan ke operasi pada suatu objek.

## 10. Activity Diagram

Activity denah Buat arketipe jjejeran tugas usaha niaga dan rentetan sikap bagian dalam usaha. Diagram ini sangat analog tambah denah jjejeran karena memodelkan jjejeran tugas berbunga tunggal sikap ke sikap lain atau berbunga sikap ke status. Ini membangun kepada menyelenggarakan denah sikap di mula pemodelan usaha kepada sehat mengerti kepaduan usaha. Diagram sikap juga bisa digunakan kepada mencatat tutur kata seleret atau afiliasi jarak sejumlah use case.

Elemen-elemen activity diagram :

- Status start (mulai) dan end (akhir)
- Aktifitas yang membayangkan secorak gelagat bagian dalam workflow.
- Transition menyinggir terjadi deformasi kadar kelakuan (Transitions show what state follows another).
- Keputusan yang menyinggir pilihan bagian dalam workflow.
- Synchronization bars yang menyinggir subflow parallel. Synchronization bars bisa digunakan menjelang menyinggir concurrent threads depan workflow jalan kulak.
- Swimlanes yang membayangkan role kulak yang bertanggung sambut depan kelakuan yang berjalan.



Contoh Gambar 9.10 Komponen Activity Diagram

## 11. Kafinement

### 1. Class Refinement

Dimana sequence diagram Pada tahap ini interaksi antar kelas akan ditampilkan pada diagram kelas modul berikutnya. Pada tahap tersebut masih mendeskripsikan diagram sequence berdasarkan interaksi antar objek. Tidak ada kelas antarmuka pengguna yang ditemukan yang akan menjadi formulir dalam aplikasi yang akan dibuat selama tahap implementasi.

### 2. Class User Interface

Pada tahap implementasi, Anda harus mendefinisikan kelas antar muka pengguna, walaupun pada dasarnya kelas ini hanya merupakan spesifikasi untuk kelas lain, namun tetap diperlukan untuk pemrograman. Di kelas antarmuka pengguna, metode adalah komponen yang digunakan saat membuat aplikasi. Jika Anda menggunakan Visual Basic, metode di kelas antarmuka pengguna adalah komponen Visual Basic itu sendiri.

## 12. Sejarah Singkat UML

UML secara aturan diluncurkan pada Oktober 1994, waktu Rumbaugh berburu tambah Booch, setara perusahaan gawai kepala dingin relasional. Proyek ini berfokus pada perpaduan adat Booch dan OMT. UML penerbitan 0.8 adalah adat tercampur yang dirilis pada Oktober 1995. Pada waktu yang sama, Jacobson berburu tambah Relasional, dan spektrum UML merambat melangkahi OOSE.

Dokumen UML keluaran 0.9 kesudahannya dirilis pada rembulan Juni 1996. Meskipun pada perian 1996, ia menyoroti dan memeluk kepercayaan jeratan balasan berusul lingkungan penggunaan gawai tenang. Selama ini, hidup spesifik bahwa sejumlah parlemen gawai tenang duga mengakui UML seumpama reka bentuk kulak mereka. Kemudian, sejumlah parlemen bersama-serupa menyesuaikan koneksi UML, dan parlemen-parlemen ini akan mendedikasikan asal dayanya kepada melaksanakan, mengembangkan, dan menggenapi UML.

Ada berlebihan mitra yang taksiran berkontribusi ambang UML 1.0,

terhitung Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Relational, Texas Instruments, dan Unisys. Melalui kerjasama ini, UML 1.0 diproduksi, yang mengadakan lagu kalimat pemodelan yang terdefinisi tambah ketakziman dan ekstrem yang tusukan menjelang berbagai loka masalah. Menyediakan UML 1.0 menjelang menstandarisasi Object Management Group (OMG). Dan berperan bahasa pemodelan dasar ambang Januari 1997.

Antara Januari dan Juli 1997, karena kira OMG, rombongan rampaian termuat memperluas kontribusinya tambah menyatukan Adersen Consulting, Ericsson, ObjectTimeLimeted, Platinum Technology, Ptech, Reich Technologies, Softeam, Sterling Software, dan Taskon. Versi perbaikan berpokok publikasi UML (versi 1.1) diberikan menjelang OMG menjelang penyeragaman muka Juli 1997. Pada kamar September 1997, OMG Analysis and Design Task Force (ADTF) dan OMG ArchitectureBoard mengikuti publikasi ini. Akhirnya muka Juli 1997, UML versi 1.1 bekerja standar.

Kelompok Kerja Revisi OMG (RTF) yang dipimpin oleh Cris Kobryn terus bertanggung jawab atas pemeliharaan UML. RTP merilis editorial UML 1.2 pada bulan Juni 1998. RTF juga merilis UML 1.3 dan panduan pengguna pada tahun 1998, dan menyediakan pembersihan teknis.

### **C. SOAL LATIHAN/TUGAS**

1. Jelaskan pengertian UML !
2. Sebutkan bagian – bagian dari UML !
3. Sebutkan komponen yang terlibat dalam use case diagram !
4. Jelaskan pengertian dari UML !
5. Tulislah sejarah UML secara singkat !

**D. REFERENSI**

Grady Booch, Object-Oriented Analysis and Design with Application,  
Benjamin/Cummings, 1991

Practical UML A Hands-On Introduction for Developers,

Sri Dharwiyanti, Pengantar unified modeling language (UML),2003  
IlmuKomputer.com

[[http://www.togethersoft.com/services/practical\\_guides/umlonlinecourse/index.html](http://www.togethersoft.com/services/practical_guides/umlonlinecourse/index.html)]

## GLOSARIUM

**Diagram** Suatu bayang-bayang kepada menunjukkan atau menjelaskan suatu informasi yang akan disajikan

**Designer** Adalah penghidupan yang mengarang ilustrasi, tipografi, fotografi, atau grafis motion

**Developer** adalah pekerjaan yang berada satu tingkat di atas programmer.

**Visualizing** Adalah suatu penggunaan bagian dalam penyusunan gambar, rancangan atau animasi kepada kemampuan suatu informasi

**Database** Adalah himpunan berbagai informasi dan data yang terpendam dan sistematis di bagian dalam komputer secara sistematis yang bisa diperiksa, dikerjakan atau dimanipulasi tambah memperuntukkan kegiatan komputer kepada merebut data berpokok landasan informasi tersebut.

**Software** adalah istilah spesifik untuk keterangan yang diformat dan disimpan secara digital, termasuk rancangan komputer, dokumentasinya, dan berbagai data yang upas dibaca, dan ditulis oleh komputer. Dengan perkataan lain, potongan susunan komputer yang tidak berwujud

Windows yang bisa mengimplementasikan sebanjar perintah.

**Hardware** adalah jenis *file* yang digunakan pada Sistem Operasi Windows yang dapat menjalankan serangkaian perintah.

**Softwere** adalah istilah karakteristik menjelang bukti yang diformat dan disimpan secara digital, terhitung rancangan komputer, dokumentasinya, dan berbagai fakta yang racun dibaca, dan ditulis oleh komputer.