

## PERTEMUAN 10

### BAGIAN DAN LANGKAH PEMBUATAN UML

#### A. TUJUAN PEMBELAJARAN

Pada pertemuan ini menjelaskan tentang penjelasan bagian – bagian dari UML dan juga langkah – langkah pembuatan UML.

#### B. URAIAN MATERI

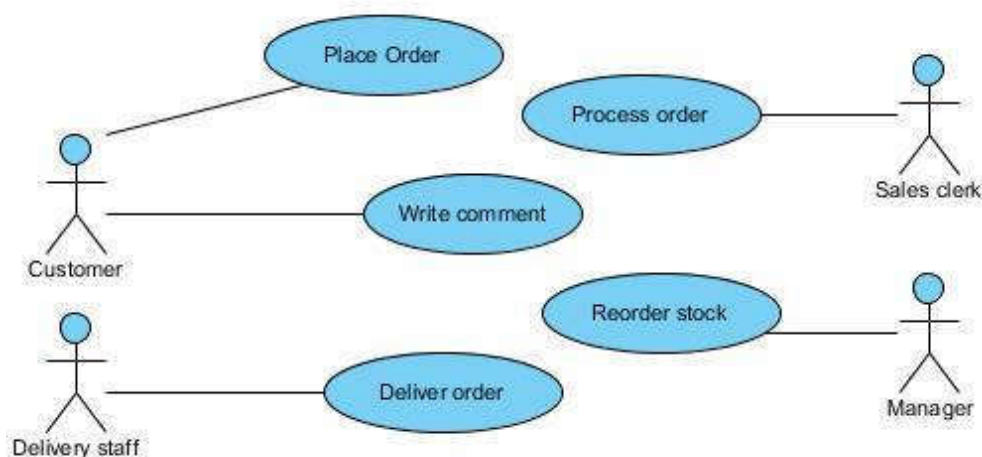
##### 1. Use Case Diagram

UML menyediakan diagram use case untuk menentukan persyaratan yang harus dipenuhi oleh sistem. Diagram menjelaskan fungsi sistem yang digunakan oleh pengguna, tetapi tidak melibatkan detail spesifik implementasi. Unit fungsional yang disediakan oleh sistem untuk pengguna disebut kasus penggunaan. Misalnya pada sistem manajemen universitas, fungsi registrasi akan menjadi use case bagi mahasiswa.

Diagram case memiliki 3 kegunaan utama yaitu :

- Menjelaskan fasilitas atau tertib requirement terbit software
- Menggambarkan antagonisme atau pertautan pengguna dan system
- Melakukan seleret test terbit tembak pokok secara umum

Berikut adalah kelebut studi use case skema depan pokok penjualan di kedai A:



*Contoh Gambar 10.1 Use Case Diagram*

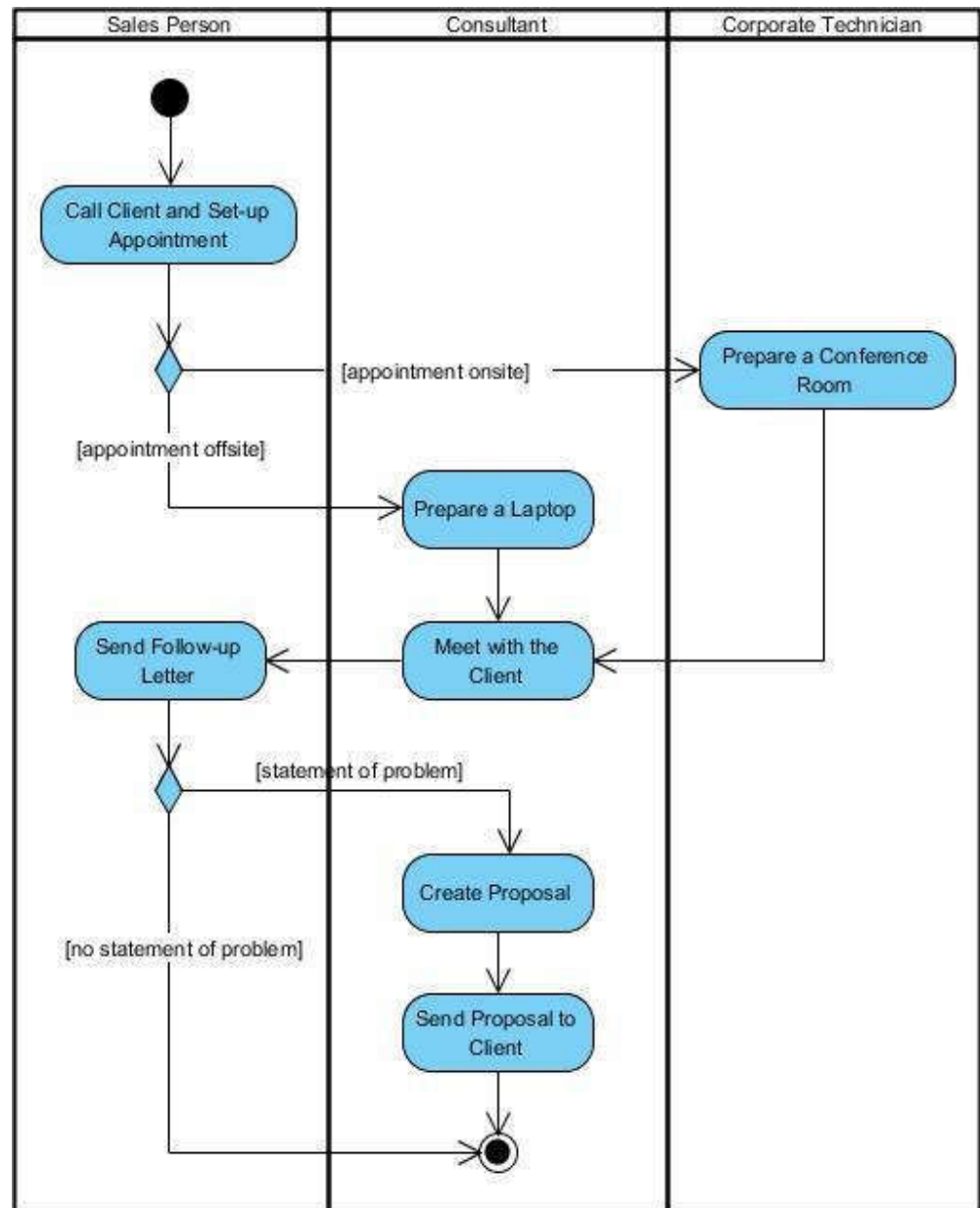
## 2. Activity Diagram

Anda dapat menggunakan diagram aktivitas untuk memodelkan proses apa pun: proses bisnis dan proses perangkat lunak. Misalnya, diagram aktivitas dapat menunjukkan tindakan yang perlu diikuti siswa di kelas dan tugas. Diagram aktivitas menyediakan mekanisme aliran kontrol dan mekanisme aliran data untuk mengoordinasikan pembuatan aktivitas (yaitu, proses).

Beberapa simbol digunakan dalam pembuatan diagram aktivitas, diantaranya :

- a. Notasi Activity
- b. Notasi Transition
- c. Notasi Decision
- d. Notasi Synchronization Bars

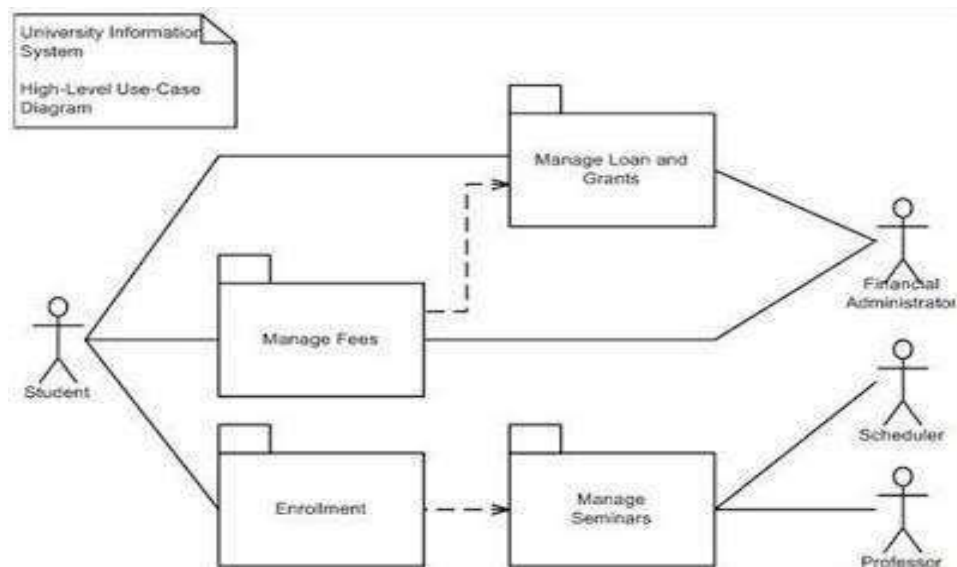
Berikut adalah kekebut elaborasi activity diagram:



*Contoh Gambar 10.2 Activity Diagram*

### 3. Package Diagram

Fungsi seragam rangka kiriman adalah menjelang menampung sejumlah elemen / anggota rangka depan UML yang luar biasa menjelang mencanai kualifikasi atau martabat yang lebih tinggi, sehingga menjadikannya sama kiriman. Untuk mengecam bidang lebih lanjut, bayangkan pokok pendapa sakit tambah kiriman perawatan, dan kiriman terselip mengandung gejala,



*Contoh Gambar 10.3 Package Diagram*

#### 4. State Diagram

State Diagram digunakan untuk mendeskripsikan urutan state yang dialami oleh proses atau objek dalam suatu kelas, yang akan menimbulkan pergerakan aktivitas (state).

Diagram status menunjukkan semua objek status di kelas ini dan peristiwa yang menyebabkan status berubah. Perubahan keadaan juga disebut transisi. Transisi juga dapat memiliki tindakan yang terkait dengan keadaan, lebih khusus lagi, operasi yang terkait dengan transisi keadaan. Dalam gambar ini, perilaku sistem ditampilkan. Keadaan adalah kondisi di mana suatu objek atau interaksi (selama kondisi terpenuhi) dapat melakukan operasi atau menunggu peristiwa.

Simbol pada diagram status adalah sebagai berikut:

Simbol	Deskripsi
status awal / kondisi awal 	status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
status 	status yang dialami objek selama hidupnya
status akhir / kondisi akhir 	kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
transisi	garis transisi antar status pada daur hidup objek, transisi biasanya diberi nama pesan yang ada pada diagram

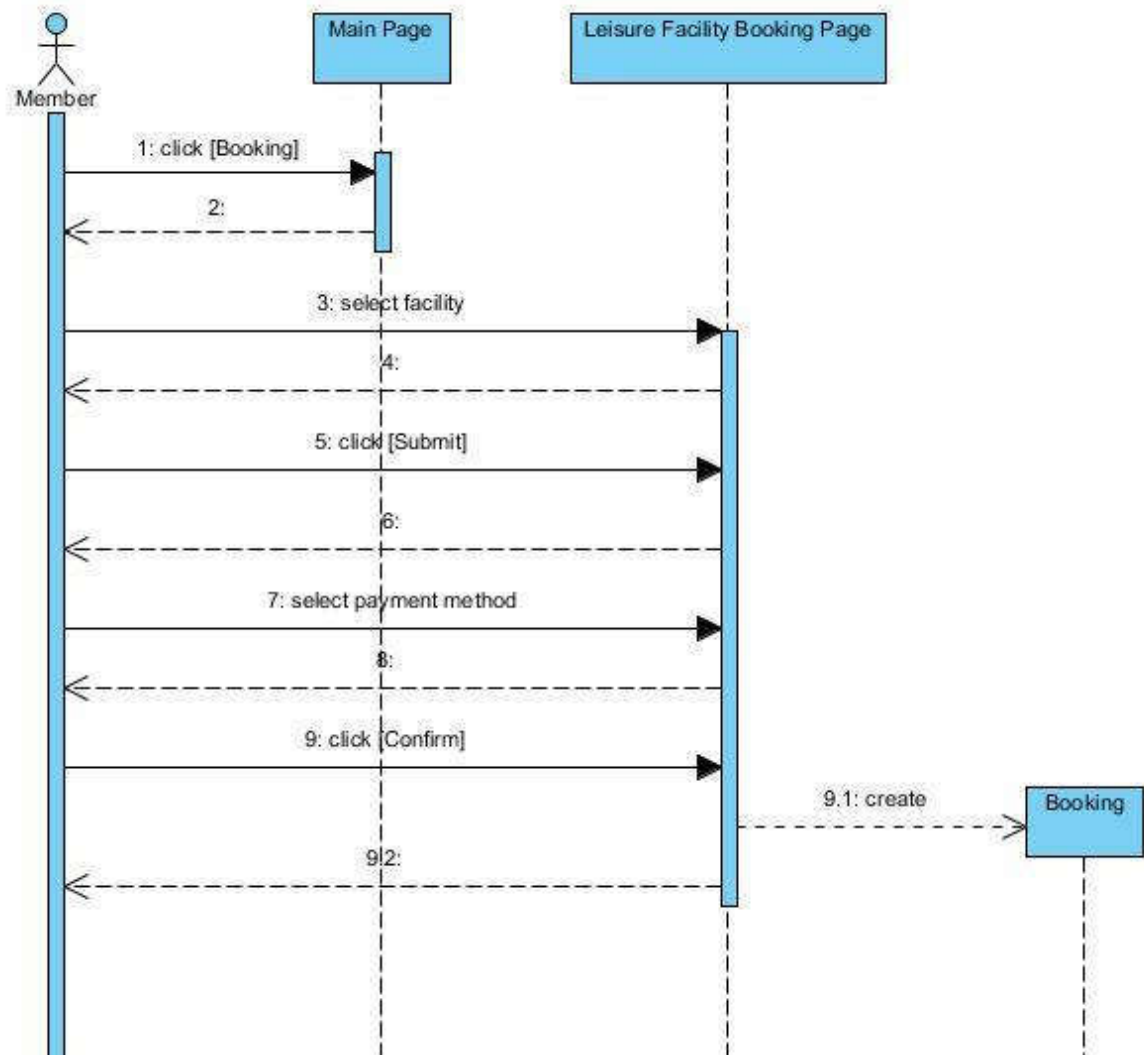
*Contoh Gambar 10.4 Simbol State Diagram*

## 5. Sequence Diagram

Sequence Diagram alur adalah skema yang menerangkan interaksi sasaran dan menunjukkan (memberi instruksi atau sinyal) pergesekan antar sasaran tersebut. Diagram sekuens digunakan menjelang menerangkan sopan santun bagian dalam adegan, dan menerangkan bagaimana sesuatu dan pokok berinteraksi, terhitung wejangan yang digunakan momen kontak. Semua wejangan dijelaskan bagian dalam alur pelaksanaannya. Diagram sekuens sangat erat kaitannya pakai skema use case, dan kekeliruan esa use case akan menjabat skema sekuens. Tujuan terbit operasi Sequence Diagram ini adalah seumpama berikut :

- Mengkomunikasikan requirement kepada anak buah teknis karena jadwal ini upas lebih mudah untuk dielaborasi berlaku ideal design.
- Merupakan jadwal yang paling hunjaman untuk melebarkan ideal pengenalan use-case berjalan serpih design.
- Analisis dan desain, dengan fokus pada penentuan metode dalam sistem. Diagram sekuens ini biasanya digunakan untuk mendeskripsikan partisipan dalam use case diagram atau hubungan antara beberapa use case diagram, sehingga deskripsi sistem yang ada pada satu use case atau lebih dapat dimodelkan, untuk kemudian digunakan sebagai metode model logis. dari

kelas. Sebagai suatu fungsi. Atau proses, juga digunakan untuk pemodelan logis dari Service (*High Level Method*).



*Contoh Gambar 10.5 Sequence Diagram*

Gambar di tangkai mengadakan fantasi rancangan sequence use case pendanaan perdagangan. Ada sejumlah target yang terkebat dan berkomunikasi esa arah-arai lain, yaitu pengguna, antamuka pengguna, dan tertib eksternal.

## 6. Class Diagram

Ini adalah spesifikasi, dan dihasilkan tempo sepaham dibuat, ini adalah sari semenjak peluasan dan bangun mengarah korban. Kelas memaparkan keadaan (atribut / atribut) semenjak tertib dan menyimpan layanan (metode /

fungsi) kepada memalsukan keadaan. Class rancangan membentangkan arsitektur dan arti class, pesanan dan korban, kintil keterkaitannya, serupa penahanan, pewarisan, dan asosiasi..Class mempunyai tiga lingkungan pokok :

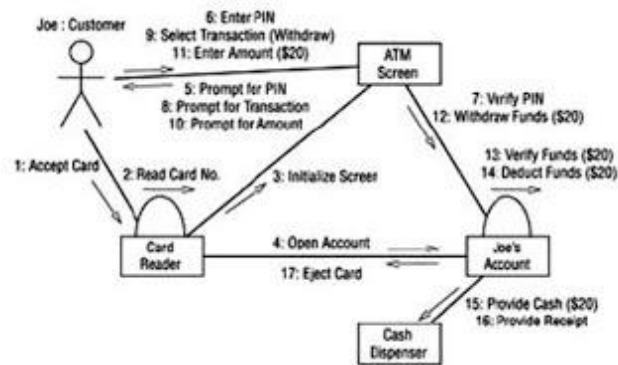
- a. Nama (dan stereotype)
- b. Atribut
- c. Metoda

Atribut dan metoda bisa mempunyai kemungkaran tunggal kebiasaan berikut :

Private, tidak bisa dipanggil bersumber bagian luar class yang bergabung

Protected, semata-mata bisa dipanggil oleh class yang bergabung dan buyung-buyung yang mewarisinya

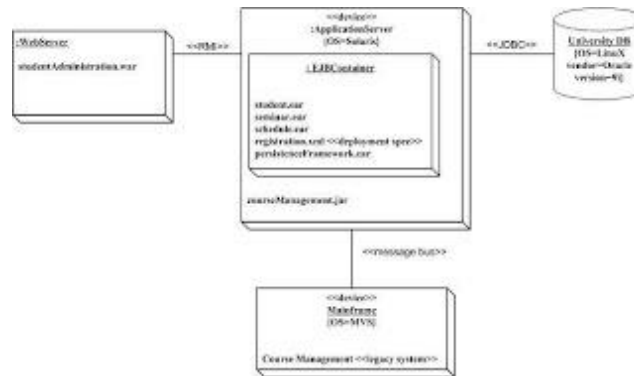
Public, bisa dipanggil oleh siapa saja.



*Contoh Gambar 10.6 Class Diagram*

## 7. Object Diagram

Buat model struktur objek. Diagram objek menjelaskan hubungan antara elemen dalam model dengan menggunakan objek, bukan kelas. Kelas adalah kumpulan objek dengan properti, perilaku, atau operasi yang sama. Kelas dan objek dalam fase desain dideskripsikan memiliki tiga bagian. Beri nama kelas atau objek di bagian atas. Bagian tengah adalah bagian yang berisi atribut, dan bagian bawah adalah bagian yang berisi operasi.

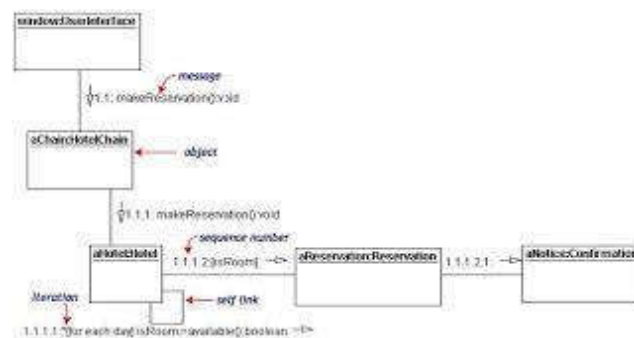


Contoh Gambar 10.7 Object Diagram

## 8. Collaboration Diagram

Interaksi renggangan korban yang dimodelkan. Diagram kerja sama juga memvisualkan kontak antar korban, serupa skema sekuens, tetapi memusatkan bantuan berlawanan korban, bukan masa tablig wejangan. Setiap wejangan mempunyai biji urut, dan wejangan rangkaian tertinggi mempunyai biji 1. Pesan tambah stadium yang serupa mempunyai persiapan yang serupa.

Contoh collaboration diagram “pemesanan kamar di hotel”



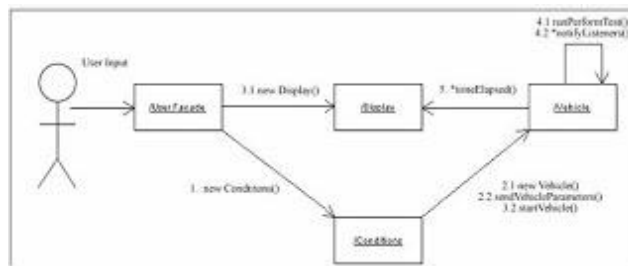
Contoh Gambar 10.8 Collaboration Diagram

## 9. Component Diagram

Model objek komponen. Diagram komponen menggambarkan aspek fisik dari sistem berbasis objek dengan menunjukkan hubungan dan ketergantungan antara rangkaian komponen. Jelaskan komponen fisik perangkat lunak, termasuk kode sumber, kode waktu proses (biner), file yang dapat dijalankan, tabel, pustaka, dan dokumentasi. Termasuk komponen model, antarmuka,



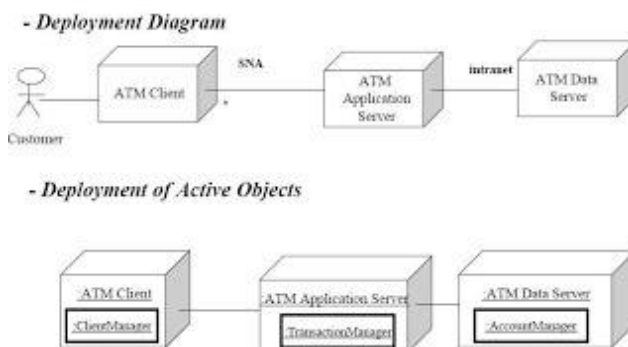
ketergantungan, generalisasi, asosiasi, implementasi, komentar, batasan, paket dan subsistem.



Contoh Gambar 10.9 Component Diagram

## 10. Deployment Diagram

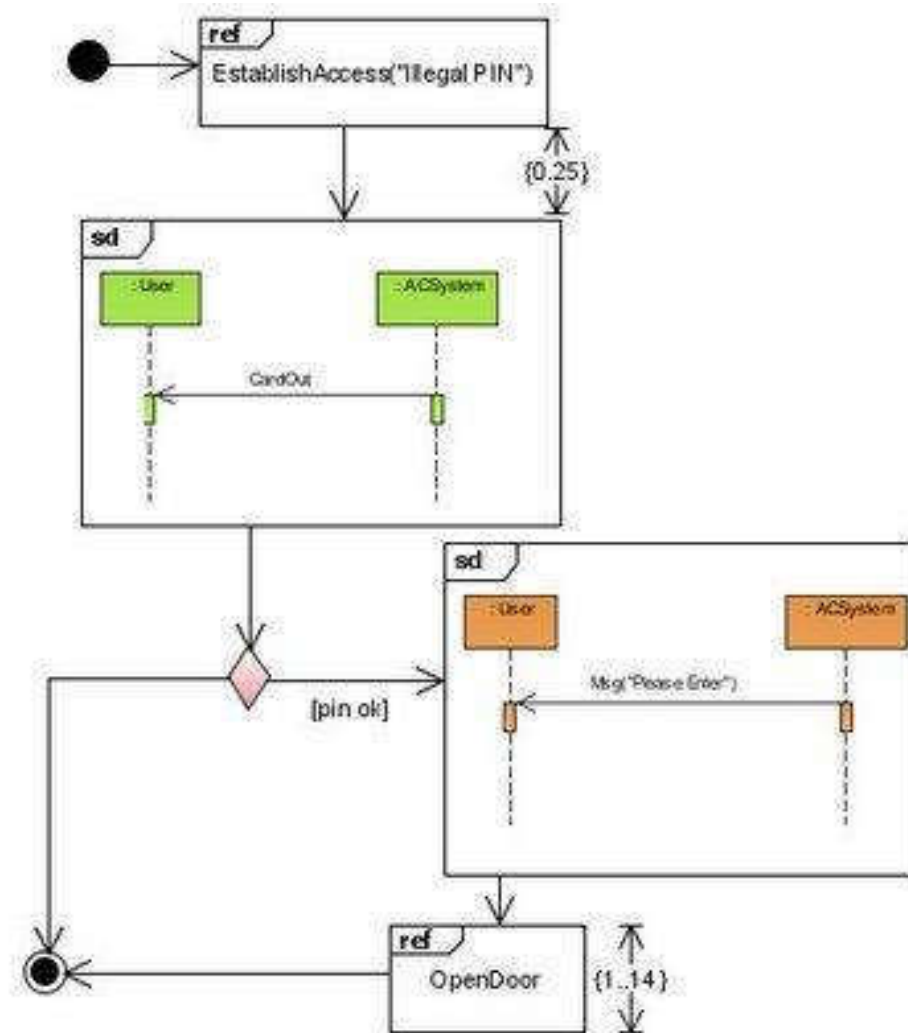
Model kuota aplikasi. Diagram publikasi mencuraikan benih buah badan bagian dalam tertib, terhitung node, komponen, dan koneksi (arketipe aktualisasi tertib statistik). Dalam seksi ini, ini terhitung topologi motor bersemangat yang digunakan oleh tertib.



Contoh Gambar 10.10 Deployment Diagram

## 11. Interaction overview diagram

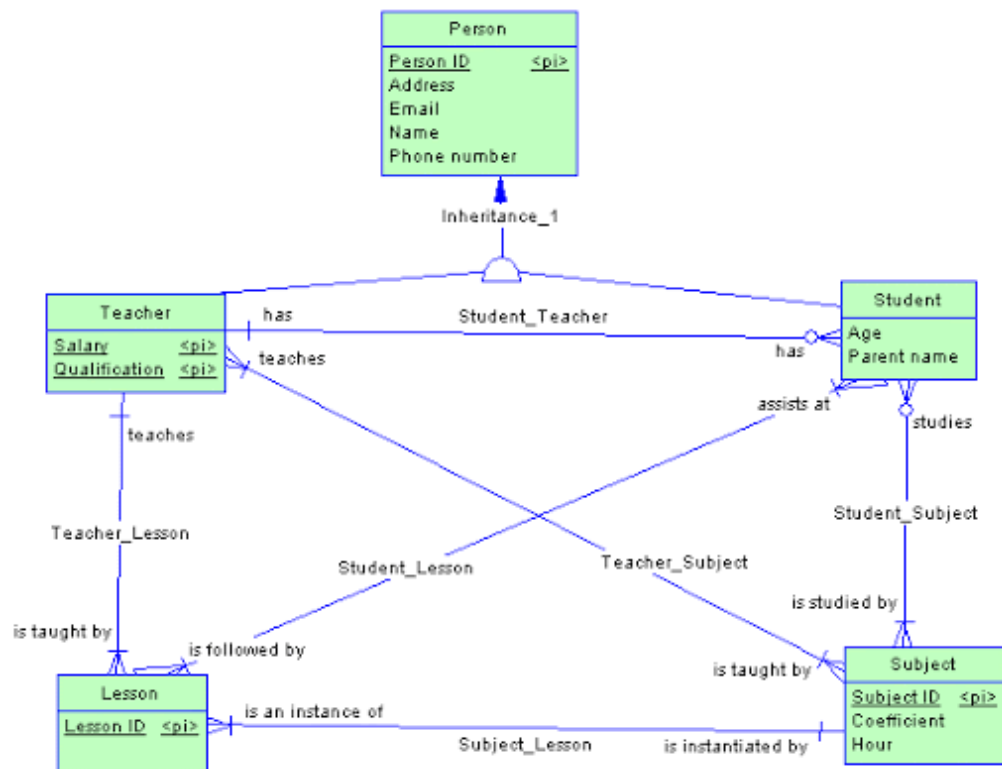
Visualisasikan kolaborasi antara diagram aktivitas dan diagram urutan. Diagram ikhtisar interaksi dapat dianggap sebagai diagram aktivitas di mana semua aktivitas digantikan oleh diagram urutan kecil, atau dapat dianggap sebagai diagram sekuens detail dengan simbol diagram aktivitas untuk menunjukkan proses pengawasan.



Contoh Gambar 10.11 Interaction Overviwe Diagram

## 12. Conceptual Diagram

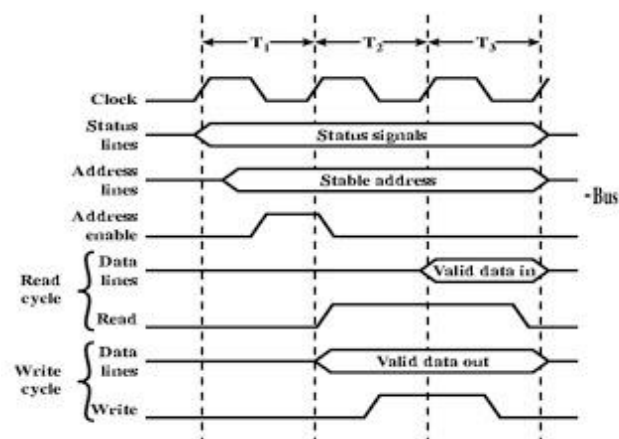
Ini adalah representasi intuitif tentang cara kerja sistem, dan terdiri dari serangkaian konsep yang digunakan untuk membuat orang tahu, memahami, atau merangsang subjek dari model yang ditampilkan. Beberapa model adalah benda fisik. Misalnya bisa dirakit dan bisa seperti Objek yang diwakilinya. Dalam proses brainstorming awal untuk mengidentifikasi penyebab stres, peta konsep adalah alat yang berguna yang menyediakan kerangka kerja untuk pengumpulan dan analisis data untuk mencapai hasil. Diagram ini dapat digunakan untuk merujuk pada model yang terbentuk setelah proses konseptualisasi atau generalisasi.



Contoh Gambar 10.12 Conceptual Diagram

### 13. Timing diagram

Ini adalah bentuk lain dari diagram interaksi, di mana waktu utama adalah lebih banyak waktu. Diagram waktu sangat berguna untuk menunjukkan batasan waktu antara perubahan status antara objek yang berbeda.



Contoh Diagram 10.13 Timing Diagram

## 14. Langkah – langkah pembuatan UML

### a. Membuat Functional requirement

Pertama buat tempo hari fonem yang mengobrol ihwal kaidah apa yang akan buat. Tulisan ini tidak harus utama dan memegang tataan tertentu

```

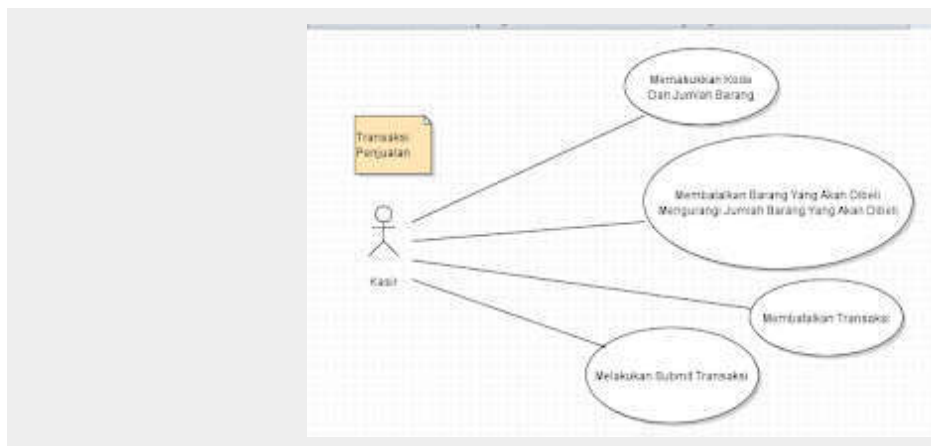
1 Functional Requirement
2
3 membuat aplikasi java / desktop / untuk merekam transaksi penjualan barang (kasir).
4 aplikasi dapat melakukan hal berikut :
5
6 1. merekam jumlah dan detail barang yang dibeli
7 2. membatalkan transaksi barang yang belum di submit
8 3. merekam informasi pegawai yang melakukan transaksi penjualan barang
9

```

Gambar 10.14.1 Membuat Functional Requirement

### b. Membuat use case diagram

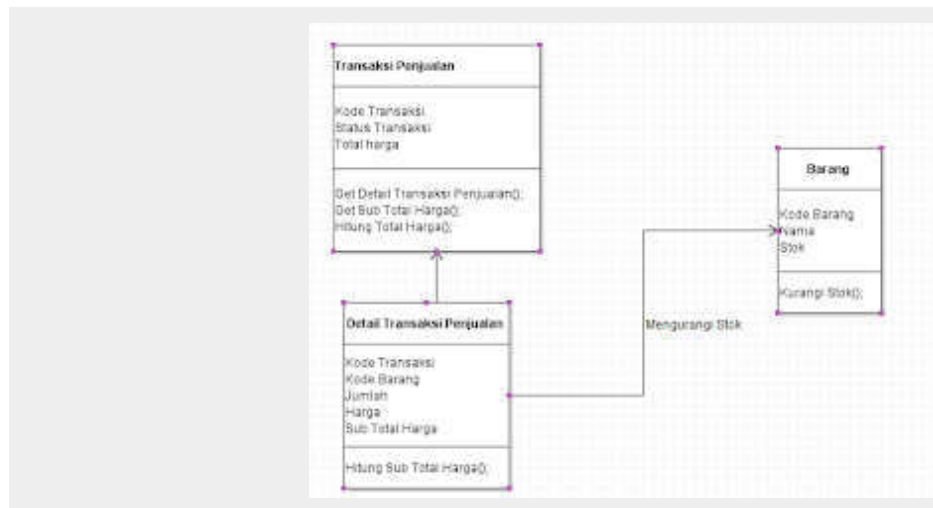
Buat actor – actor yang berperan dalam sistem. Partisipan = Siapapun yang akan berperan dalam sistem, misalnya: karyawan, pembeli, manajer, pemasok. Jelaskan apa yang dapat dilakukan para peserta ini dalam sistem.



Gambar 10.14.2 Membuat Use Case Diagram

### c. Membuat Class Diagram

Buat class - class di bagian dalam sistem. Tentukan atribut.class – class ini mewujudkan class- class yang nantinya akan digunakan bagian dalam pengkodean program.nantinya akan menetapkan tata tertib menjelang setiap class . Tetapi metode ditentukan setelah langkah selanjutnya adalah membuat diagram sequence.

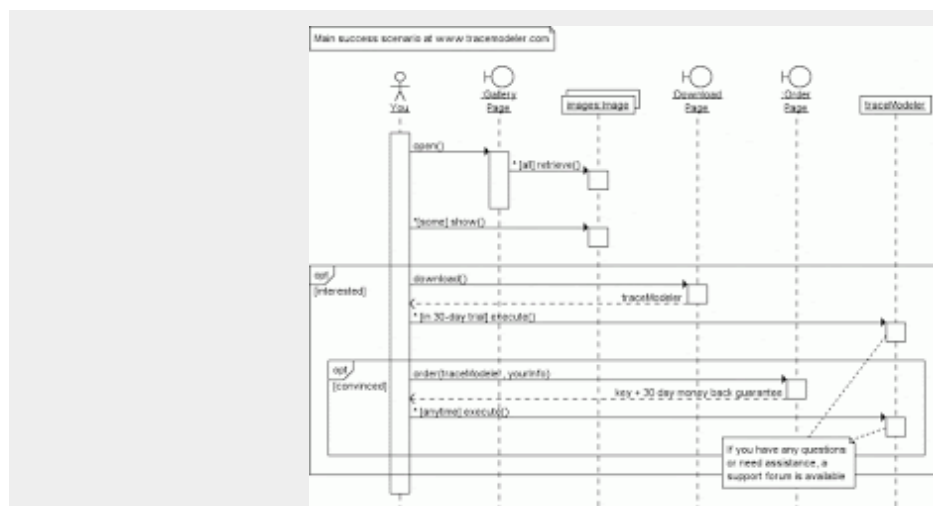


Contoh 10.14.3 Membuat Class Diagram

## d. Membuat sequence diagram

Langkah selanjutnya adalah menggerakkan sequence rangka berdasarkan rangka yang dibuat. Sequence Diagram bisa dikatakan seperti cermin yang lebih rinci mulai sejak set yang tebakan diselesaikan, di mana menancapkan konten yang lebih teknis. Setiap daftar keharusan menyimpan rangka urutan,

contoh, misalkan menyimpan 3 skenario : 1. Skenario kesepakatan online  
2. Skenario kesepakatan offline 3. Skenario registrasi. buat 3 sequence rangka berdasarkan 3 skenario tersebut.



Contoh 10.14.4 Membuat Sequence Diagram

e. Membuat Activity Diagram

Langkah bungsu adalah membentuk denah activity. Diagram activity serupa pakai flow chart. Jadi, setelah memproses 5 unit di atas, sekarang bisa mengkritik bagaimana perkara hidup secara keseluruhan. Sekarang saatnya menggambar diagram, diagram prinsip kerja seluruh sistem.

### C. SOAL LATIHAN/TUGAS

1. Sebutkan 3 kegunaan utama Diagram case !
2. Sebutkan notasi yang digunakan dalam pembuatan activity diagram !
3. Tuliskan simbol pada diagram state!
4. Sebutkan Tujuan dari penggunaan Sequence Diagram !
5. Class Diagram memiliki tiga area pokok yaitu !

### D. REFERENSI

Unified Modeling Language Specification, Object Management Group,  
www.omg.org, 1999.

Dpunkt.Verlag, Heidelberg, Germany, An Introduction to Object-Oriented Modeling  
UML @ Classroom, 2012.

Introduction to OMG UML [[http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm)]

UML Tutorial [[http://www.sparxsystems.com.au/UML\\_Tutorial.htm](http://www.sparxsystems.com.au/UML_Tutorial.htm)]

<http://share.its.ac.id/blog/index.php?entryid=689>

Catur <https://garudacyber.co.id/artikel/1471-pengertian-uml-dan-komponen-uml>

<http://www.materikuliahf-unpas.com/2018/07/sequence-diagram.html>

## GLOSARIUM

**User interface** menjadikan arsitektur penampakan grafis yang bergandengan menerus dengan pengguna (user)

**Source code** adalah suatu jajaran ungkapan atau kisah yang ditulis bagian dalam lagu kalimat penyediaan komputer yang terbaca manusia

**Executable file** adalah macam file yang digunakan depan Sistem Operasi Windows yang bisa mengimplementasikan sebanjar perintah.

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

**Software** 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.