

Pertemuan 3

Pemodelan Object Oriented

Pembahasan

- 1. Model Object Oriented**
- 2. Pedoman Pemodelan Object Oriented**

1. Model Object Oriented

Objek dan Kelas

Objek

- Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah.
- Semua objek mempunyai identitas yang berbeda dengan lainnya. Istilah identitas berarti bahwa objek dibedakan oleh sifat yang melekat dan bukan dengan uraian sifat yang dimilikinya. Contohnya, kembar identik, walaupun tampak sama, tetapi merupakan dua orang yang berbeda.
- Kadang-kadang objek berarti suatu barang, maka digunakan istilah object instance, dan object class untuk menunjukkan satu grup dari barang yang sama.

Kelas

- Suatu object class menggambarkan kumpulan dari objek yang mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum.
Contoh: Orang, perusahaan, binatang, proses adalah objek. Setiap orang mempunyai umur, IQ, dan mungkin pekerjaan. Setiap proses mempunyai pemilik, prioritas, list dari sumber daya yang dibutuhkan.
- Objek dan object class sering sama sebagai benda dalam deskripsi masalah.
- Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup.
- Kelas Objek merupakan wadah bagi Objek. Dapat digunakan untuk menciptakan Objek.
- Objek mewakili fakta/keterangan dari sebuah kelas.

Istilah-istilah Objek

- Atribut : Data item yang menegaskan Objek
- Operasi : Fungsi di dalam kelas yang dikombinasikan ke bentuk tingkah laku kelas
- Metode : Pelaksanaan prosedur (badan dari kode yang mengeksekusi respon terhadap permintaan objek lain di dalam sistem).

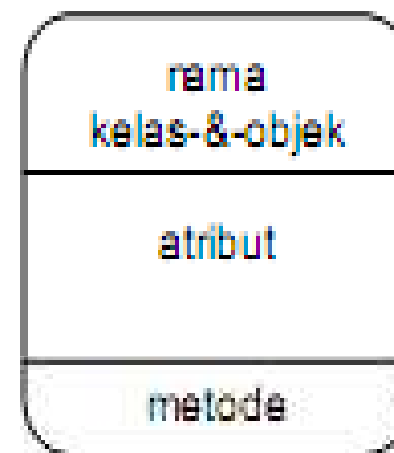
Diagram Objek

1. Kelas dan Objek

- Sebuah objek merupakan sebuah entitas yang mencakup data dan metode.
- Kelas merupakan satu atau lebih objek dengan persamaan atribut dan metode, sedangkan kelas-&-objek adalah kelas dengan satu atau lebih objek di dalamnya.
- Nama kelas adalah kata benda tunggal, atau kata sifat dan kata benda.
- Nama dari kelas-&-objek harus dapat menjelaskan objek tunggal dari suatu kelas.



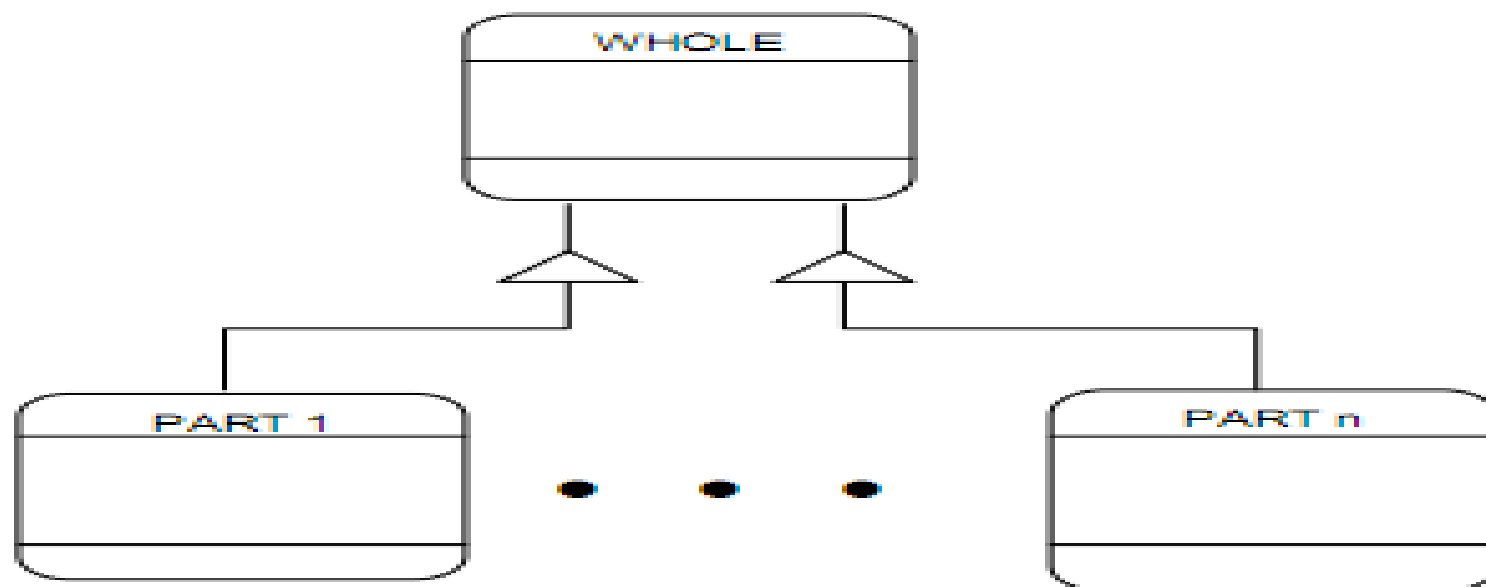
kelas



kelas dengan objek

2. Struktur Objek dan Hirarki Kelas

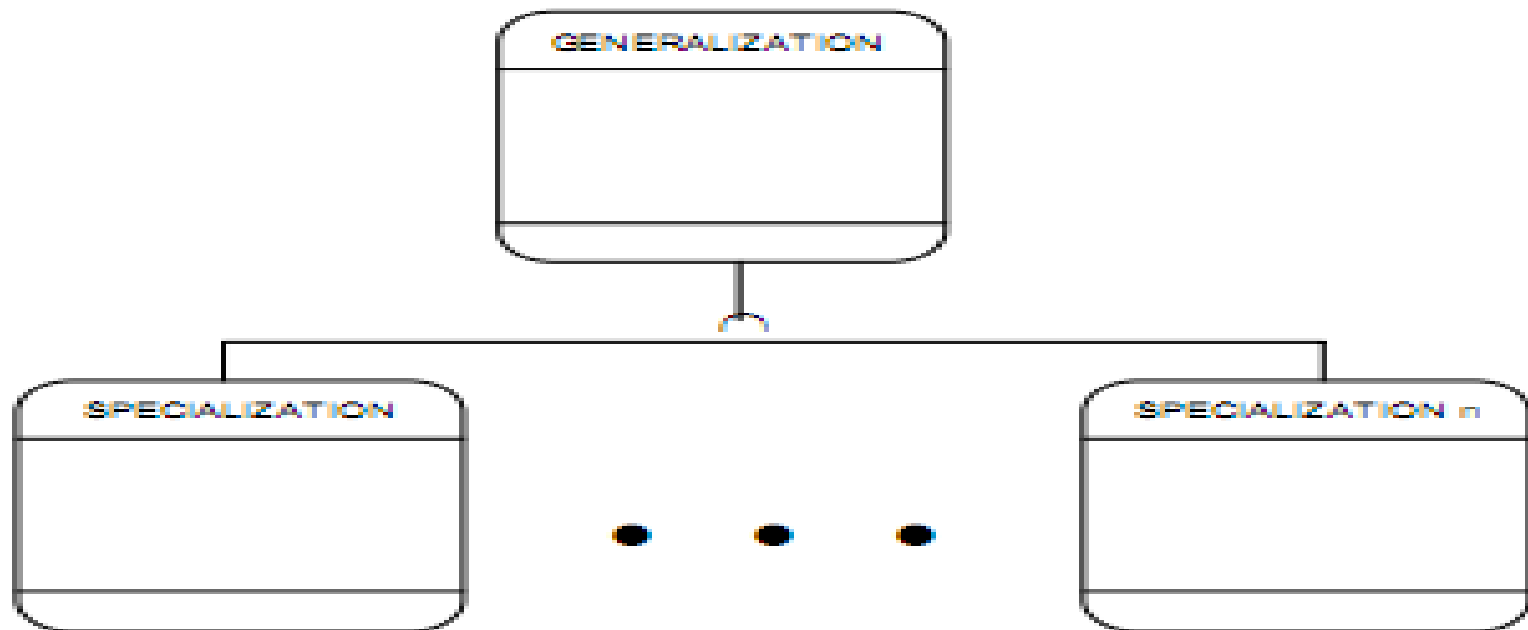
- a. Whole-Part Structure memperlihatkan hirarki dari suatu kelas sebagai komponen dari kelas lain yang disebut juga sub objek. Contohnya, kelas Mobil adalah Whole dan komponennya Mesin, Rangka, dll merupakan Part1, Part 2, ..., Partn.



b. Gen-Spec Structure memperlihatkan kelas sebagai spesialisasi dari kelas di atasnya.

- Kelas yang mempunyai sifat umum disebut Generalization, Superclass atau Topclass.
- Kelas yang mempunyai sifat khusus disebut Specialization.

Contohnya, kelas Mobil adalah Generalization, sedangkan Sedan, Truk, Minibus, dll merupakan Specialization1, Specialization2, dst



UML

UML (Unified Modeling Language) adalah metode pemodelan (tools/model) secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek dan memberikan standar penulisan sebuah sistem untuk pengembangan sebuah software yang dapat menyampaikan beberapa informasi untuk proses implementasi pengembangan software.

a. Sejarah UML

Pada Oktober 1994, Dr. James Rumbaugh bergabung dengan Perusahaan Rational software, dimana Grady Booch sudah bekerja disana sebelumnya. Grady Booch mengembangkan Object Oriented Design (OOD) dan Dr. James Rumbaugh mengembangkan Object Modeling Technique (OMT). Duet Mereka pada Oktober 1995 menghasilkan Unified Method versi 0.8.

- Musim gugur 1995 Dr. Ivar Jacobson ikut pula bergabung dengan duet Rumbaugh-Booch, dengan memperkenalkan tool use case. Trio tersebut pada bulan Juni 1996 menghasilkan Unified Modeling Language (UML) versi 0.9. Sebelumnya Dr. Ivar Jacobson mengembangkan Object Oriented Software Engineering (OOSE)
- Trio ini mengembangkan Rational Unified Process (RUP)
- Banyak perusahaan software merasakan bagaimana pentingnya UML dalam tujuan strategis mereka, sehingga beberapa perusahaan membentuk sebuah konsorsium yang terdiri dari perusahaan-perusahaan seperti Microsoft, Oracle, IBM, Hewlett-Packard, Intellicorp, I-Logix, DEC, Digital Equipment Corp. texas instrument

Konsep Dasar UML

Untuk dapat memahami UML diperlukan pemahaman tentang konsep bahasa pemodelan dan tiga elemen utama UML. Tiga elemen utama UML antara lain:

a. *Benda / Things / Objek*

Objek merupakan bagian paling statik dari sebuah model, yang menjelaskan elemen–elemen lainnya dari sebuah konsep. Bentuk dari beberapa objek :

1. Classes, sekelompok dari object yang mempunyai attribute, operasi, dan hubungan yang semantik
2. Interfaces, antar-muka yang menghubungkan dan melayani antarkelas dan atau elemen dan mendefinisikan sebuah kelompok dari spesifikasi pengoperasian.

3. Collaboration, interaksi dari sebuah kumpulan kelas–kelas atau elemen–elemen yang bekerja secara bersama–sama.
4. Use cases, pembentuk tingkah laku objek dalam sebuah model serta di realisasikan oleh sebuah collaboration.
5. Nodes, bentuk fisik dari elemen–elemen yang ada pada saat dijalankannya sebuah system

b. Hubungan / Relationship

Ada 4 macam hubungan dalam penggunaan UML, yaitu :

1. Dependency, hubungan semantik antara dua objek yang mana sebuah objek berubah mengakibatkan objek satunya akan berubah pula.

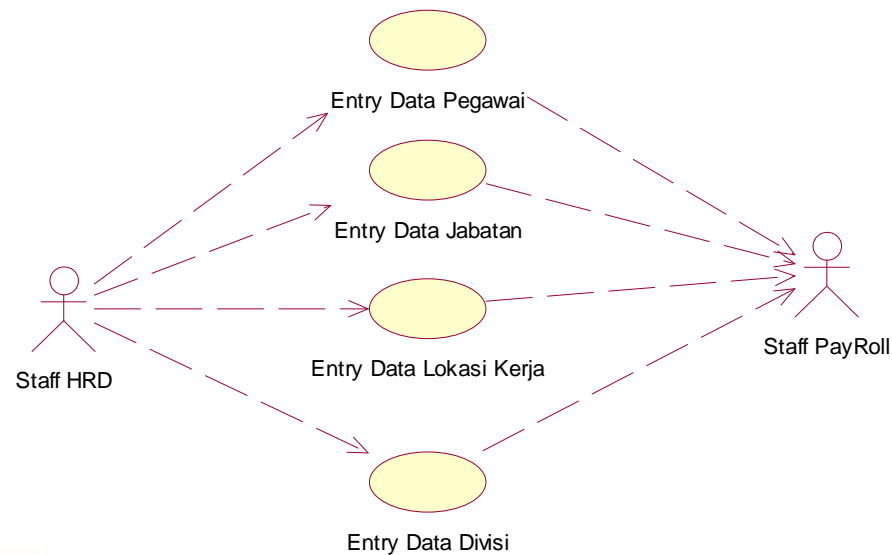
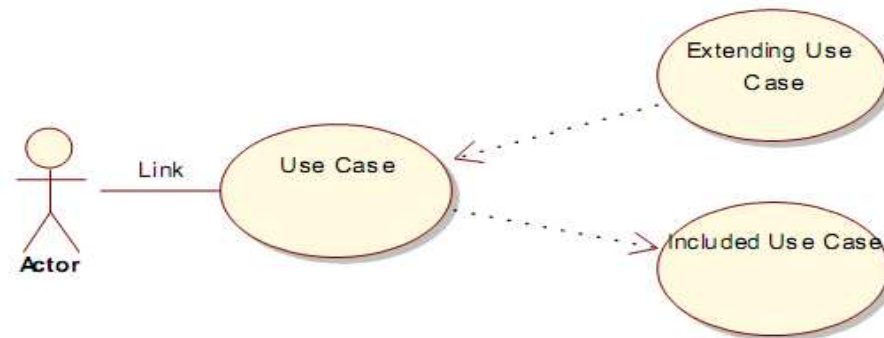
2. Association, hubungan antar benda secara struktural yang terhubung diantara objek dalam kesatuan objek.
3. Generalizations, hubungan khusus dalam objek anak yang menggantikan objek induk dan memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada objek induk
4. Realizations, hubungan semantik antarpengelompokkan yang menjamin adanya ikatan diantaranya yang diwujudkan diantara interface dan kelas atau elements, serta antara use cases dan collaborations.

c. Bagan atau Diagrams

Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu:

- 1) Diagram Use Case, menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram Use Case berguna dalam tiga hal :
 - a. Menjelaskan fasilitas yang ada (requirements)
 - b. Komunika dengan klien
 - c. Membuat test dari kasus – kasus secara umum

Contoh Use Case Diagram

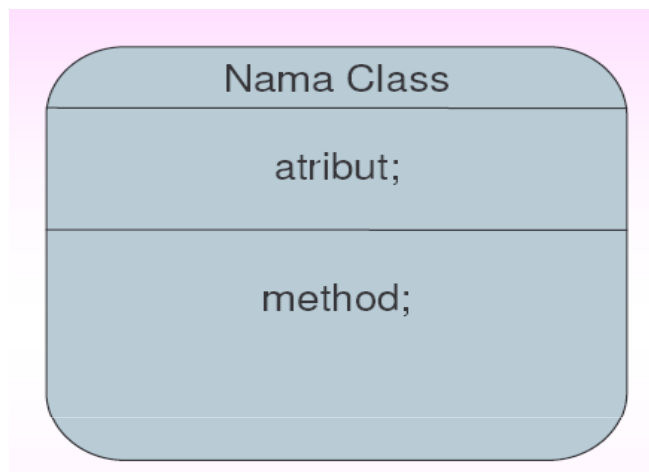


2) Diagram Class, memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka.

Diagram Class mempunyai 3 macam relationships (hubungan), sebagai berikut :

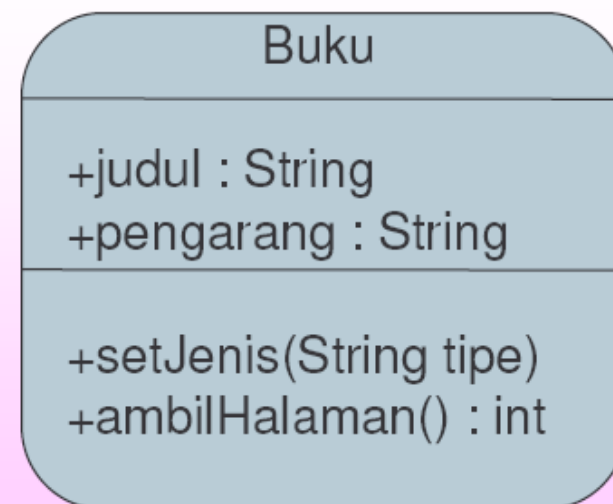
- a. **Association**, suatu hubungan antara bagian dari dua kelas yang terjadi jika salah satu bagian dari kelas mengetahui kelas yang lain dalam melakukan suatu kegiatan..
- b. **Aggregation**, hubungan association dimana salah satu kelasnya merupakan bagian dari suatu kumpulan dan memiliki titik pusat yang mencakup keseluruhan bagian.
- c. **Generalization**, hubungan turunan dengan mengasumsikan satu kelas merupakan suatu kelas super dari kelas yang lain.

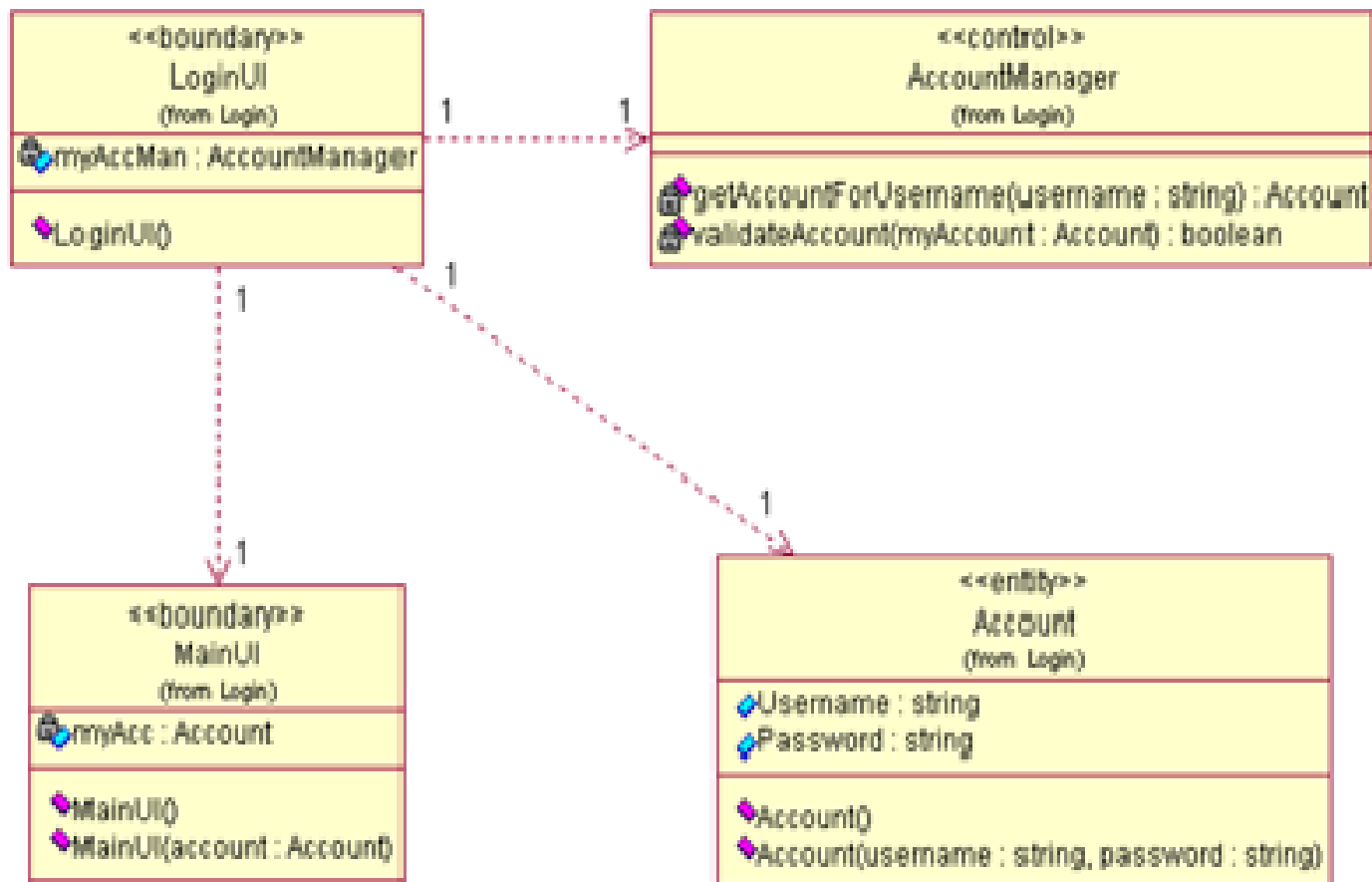
Contoh Class Diagram



Class Buku terdiri dari :

- **Atribut : judul, pengarang**
- **Method : ambilJenis(),
ambilHalaman();**





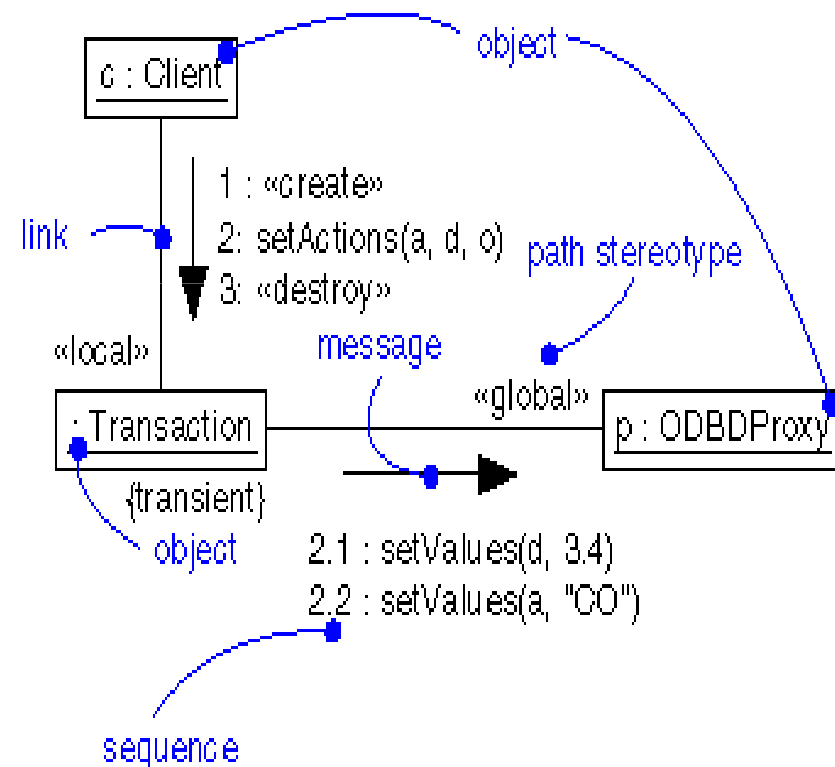
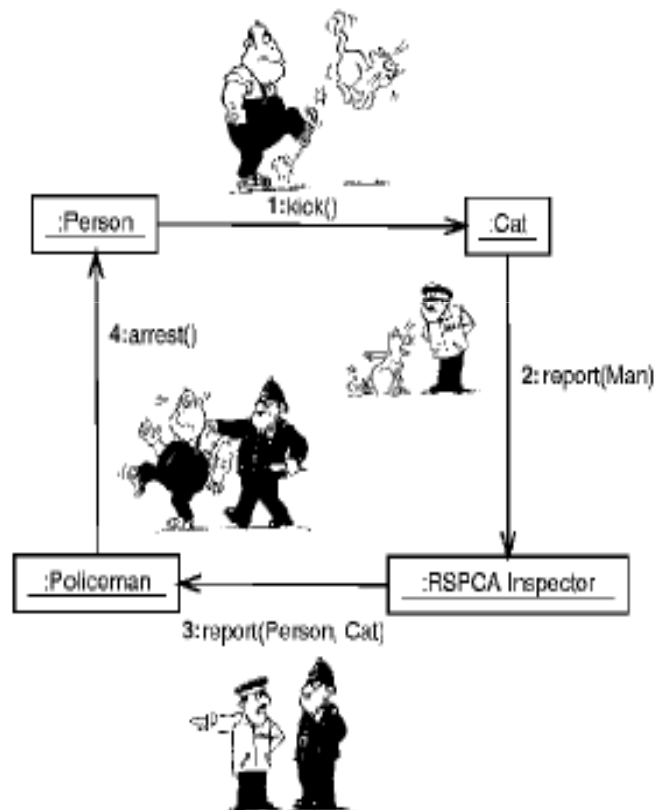
- 3) Diagram Package dan Object, merupakan kumpulan elemen–elemen logika UML yang bertujuan untuk mengelompokkan Diagram class yang lebih kompleks.

- 4) Diagram Sequence, merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan yang diatur berdasarkan waktu.

- 5) Diagram Collaboration juga merupakan diagram Interaction berfungsi membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu informasi itu dikirimkan dan bukan pada waktu penyampaian message.

Tujuan: Membuat collaboration diagram, Membuat link (relasi) antar obyek, Menambah message (pesan) pada link, Memetakan message pada operasi, Buat collaboration diagram dari sequence diagram dan sebaliknya

Contoh Collaboration Diagram

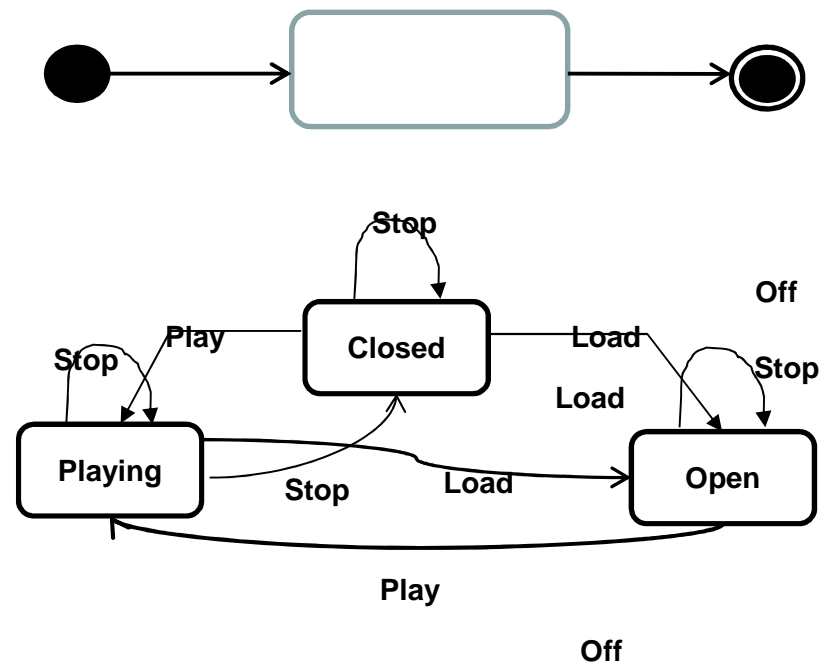


6) Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

Diagram StateChart merupakan indikator yang menunjukkan kemungkinan dari keadaan obyek dan proses yang menyebabkan perubahan pada keadaannya.

Contoh State Chart Diagram

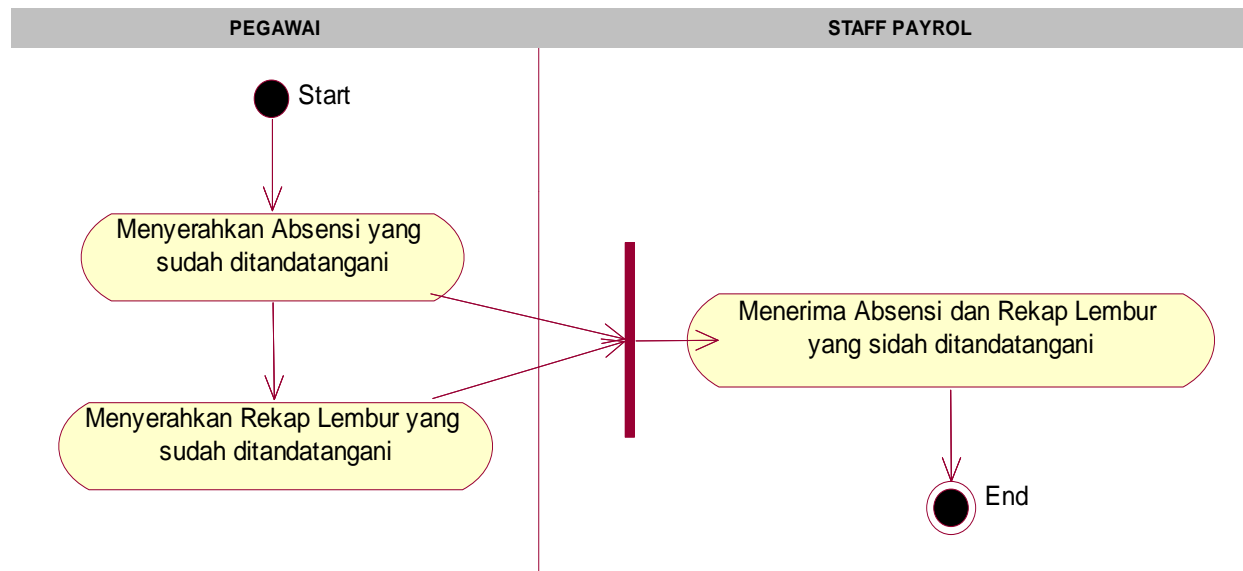
Simbol UML untuk state chart diagram



- 7) Diagram Activity, menunjukkan bagaimana aktifitas – aktifitas tersebut bergantung satu sama lain dan berfokus pada aktifitas – aktifitas yang terjadi yang terkait dalam suatu proses tunggal.

menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir

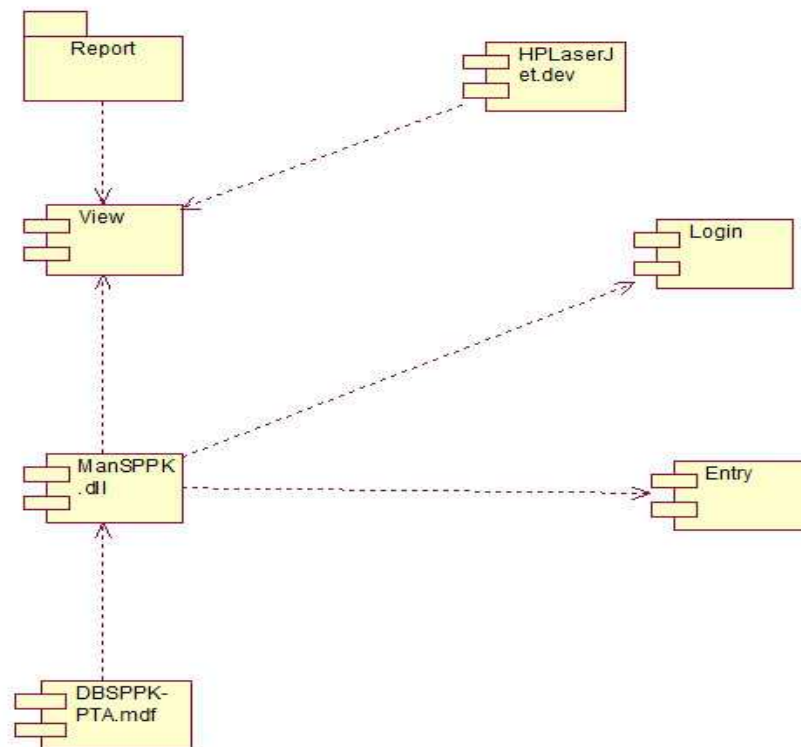
Contoh Activity Diagram



- 8) Diagram Component adalah sebuah kode–kode modul yang merupakan fisik sebenarnya dari diagram Class.

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya.

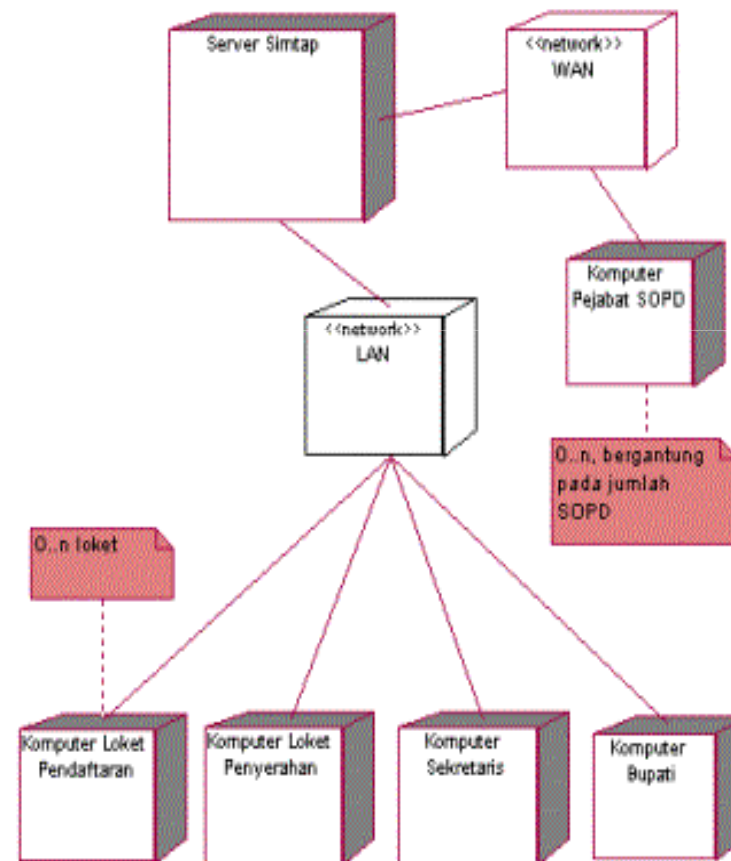
Contoh Component Diagram



- 9) Diagram Deployment menerangkan bahwa konfigurasi fisik software dan hardware.

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal

Contoh Deployment Diagram



2. Pedoman Pemodelan Object Oriented

1. Menentukan kelas dan objek
2. Menentukan struktur objek dan hirarki kelas
3. Menentukan subjek
4. Menentukan atribut
5. Menentukan metoda
6. Menentukan message