

Objek

- ▶ Objek adalah kombinasi antara struktur data dan perilaku dalam satu entitas dan mempunyai nilai tertentu yang membedakan entitas tersebut.

Berorientasi objek

- ▶ Berorientasi objek berarti pengorganisasian perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya.
- ▶ Konsep fundamental dalam analisis sistem berorientasi objek adalah objek itu sendiri

1. Encapsulation (Enkapsulasi)

- ▶ Enkapsulasi merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- ▶ Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya kecuali prosedur yang berada dalam objek itu sendiri.
- ▶ Contoh :
 - ▶ Informasi/properties objek rekening : No rekening, Nama , alamat dll
 - ▶ Perilaku/method objek rekening : buka, tutup, penarikan, penyimpanan, ubah nama, ubah alamat dll
 - ▶ Kita bungkus/encapsulate informasi dan perilaku tersebut pada objek rekening
 - ▶ Sehingga perubahan-perubahan pada sistem perbankan yang berkaitan dengan rekening diimplementasikan sederhana pada objek rekening

2. Inheritance (Pewarisan)

- ▶ Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data atau atribut dan metode dari induknya langsung.
- ▶ Atribut dan metode dari sebuah objek induk akan diturunkan kepada anak objek dan demikian seterusnya.
- ▶ Metode ini menggambarkan generalisasi dari sebuah kelas objek.

3. Polymorphism (Polimorfisme)

- ▶ Polimorfisme merupakan konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.
- ▶ Hal ini menunjukkan bahwa operasi yang sama mungkin saja mempunyai perbedaan dalam kelas yang berbeda atau bergantung kepada kelas yang menciptakan objek tersebut.

Pemodelan Berorientasi Objek

- ▶ Model berorientasi objek lebih mendekati keadaan nyata, dan dilengkapi dengan penyajian grafis dari system yang sangat bermanfaat untuk komunikasi dengan user dan pembuatan dokumentasi struktur dari system.

-
- ▶ Dari sisi desain, pemodelan berorientasi objek menggunakan tiga macam model antara lain:

1. Model Objek

2. Model Dinamik

3. Model Fungsional

I. Model Objek

- ▶ Model objek menggambarkan struktur statis dari suatu objek dalam system realisasinya yang berisi diagram objek yaitu suatu *graph* dimana *node*-nya adalah kelas yang mempunyai relasi antar kelas.

2. Model Dinamik

- ▶ Model dinamik menggambarkan aspek dari system yang berubah setiap saat yang dipergunakan untuk menyatakan aspek control dari system yang berisi *state diagram* yaitu suatu *graph* dimana *node*-nya adalah *state* dan *arc* adalah transisi antara *state* yang disebabkan oleh *event*.

3. Model Fungsional

- ▶ Model fungsional menggambarkan transformasi nilai data di dalam system yang *flow diagram* yaitu suatu *graph* dimana *node*-nya menyatakan proses dan *arc*-nya adalah aliran data.

UML (Unified Modeling Language)

- ▶ **UML (Unified Modeling Language)** sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).
- ▶ UML merupakan salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented.

- ▶ Untuk dapat memahami UML, membutuhkan bentuk konsep dari sebuah bahasa model dan mempelajari 3 (tiga) elemen utama dari UML, seperti building block, aturan-aturan yang menyatakan bagaimana building block diletakkan secara bersamaan, dan beberapa mekanisme umum (common).

3 hal yang terdapat dalam building block, diantaranya yaitu:

1. Benda/Things

Adalah abstraksi yang pertama dalam sebuah model

2. Hubungan/Relationship

Sebagai alat komunikasi dari benda-benda

3. Bagan/Diagrams

Sebagai kumpulan/group dari benda-benda/things

1. Benda/Things

Bentuk dari beberapa benda/things adalah sebagai berikut :

► ***Classes***

Yang diuraikan sebagai sekelompok dari object yang mempunyai attribute, operasi, hubungan yang semantik.

► ***Interfaces***

Merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen

► ***Use cases***

Adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor

► ***Nodes***

Merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem

2. Hubungan/Relationship

Ada 4 macam hubungan didalam penggunaan UML, yaitu :

Dependency, adalah hubungan semantik antara dua benda/things yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula.

Umumnya sebuah dependency digambarkan sebuah panah dengan garis terputus-putus.

Association, hubungan antar benda struktural yang terhubung diantara obyek.

Umumnya assosiation digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.

Generalizations, adalah menggambarkan hubungan khusus dalam obyek anak/child yang menggantikan obyek parent / induk. Digambarkan dengan garis panah.

Realizations, merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara interface dan kelas atau elements, serta antara use cases dan collaborations.

Bagian-Bagian UML

1. **View**
2. **Diagram**

1. View

Digunakan untuk melihat sistem yang dimodelkan dengan aspek yang berbeda.

- Jenis-jenis dari view adalah :

1. Use case View
2. Logical view
3. Component view
4. Concurrency view
5. Deployment view

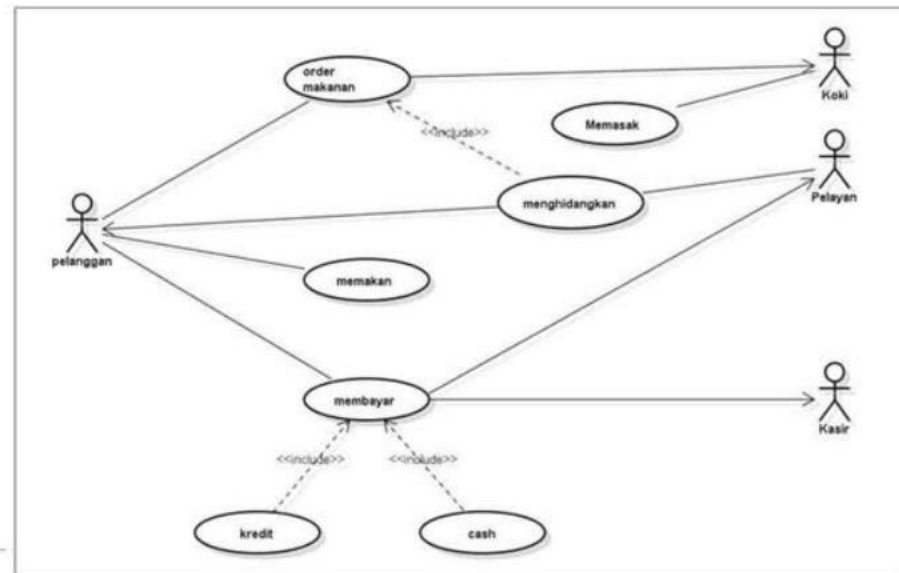
2. Diagram

Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model.

- Diagram pada UML :
 - Use Case Diagram
 - Activity Diagram
 - Sequence Diagram
 - Class Diagram
 - Communication Diagram
 - Component Diagram
 - Deployment Diagram
 - Composite Structure Diagram
 - Interaction Overview Diagram
 - Object Diagram
 - Package Diagram
 - Timing Diagram
-

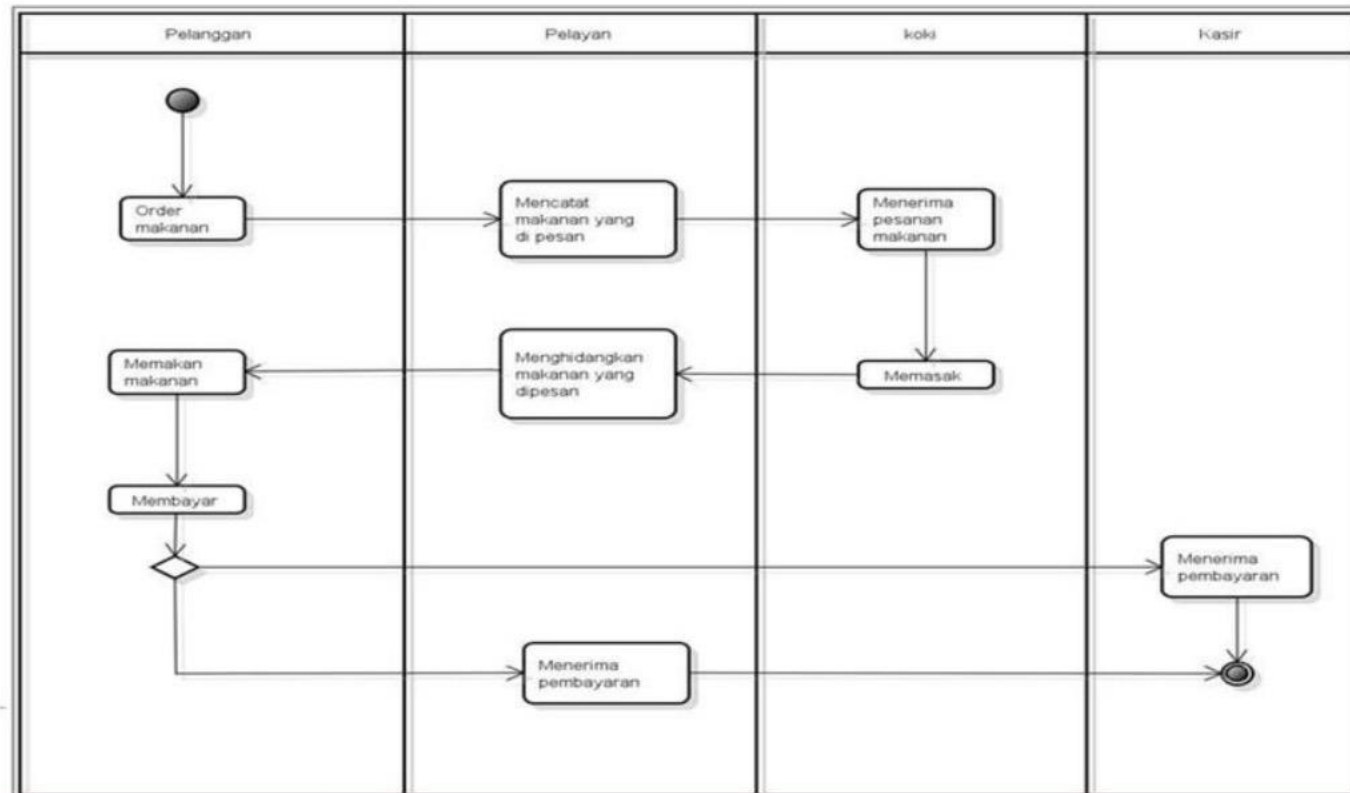
USE CASE DIAGRAM

- ▶ **Use case diagram** yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor.
- ▶ Use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.



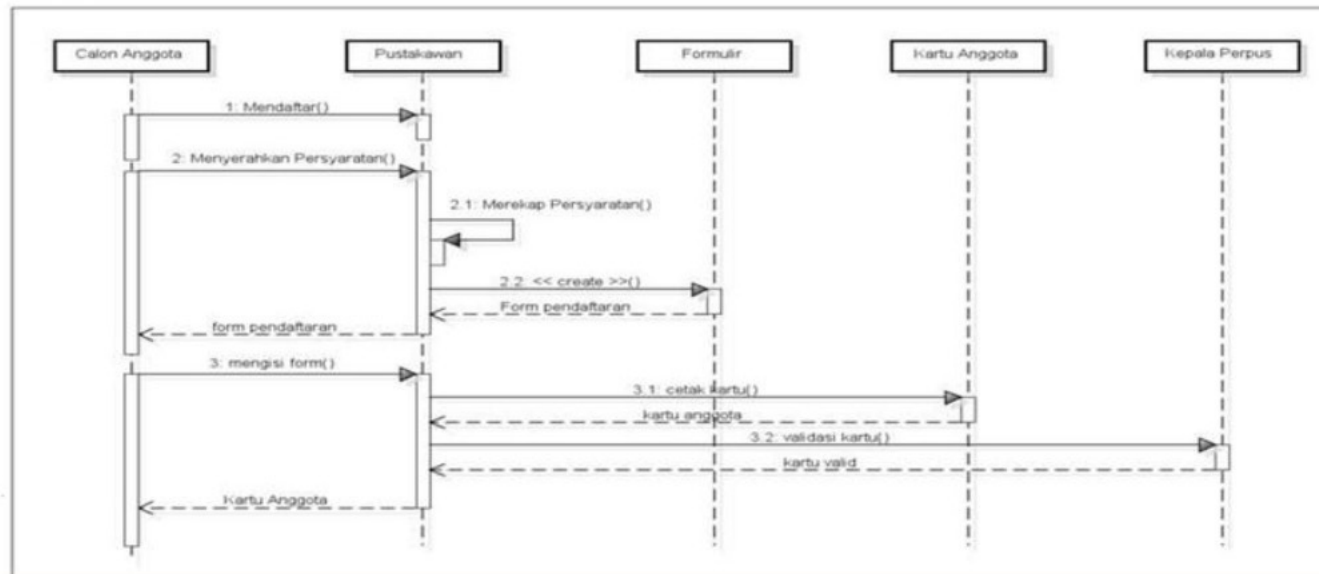
ACTIVITY DIAGRAM

- ▶ Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.



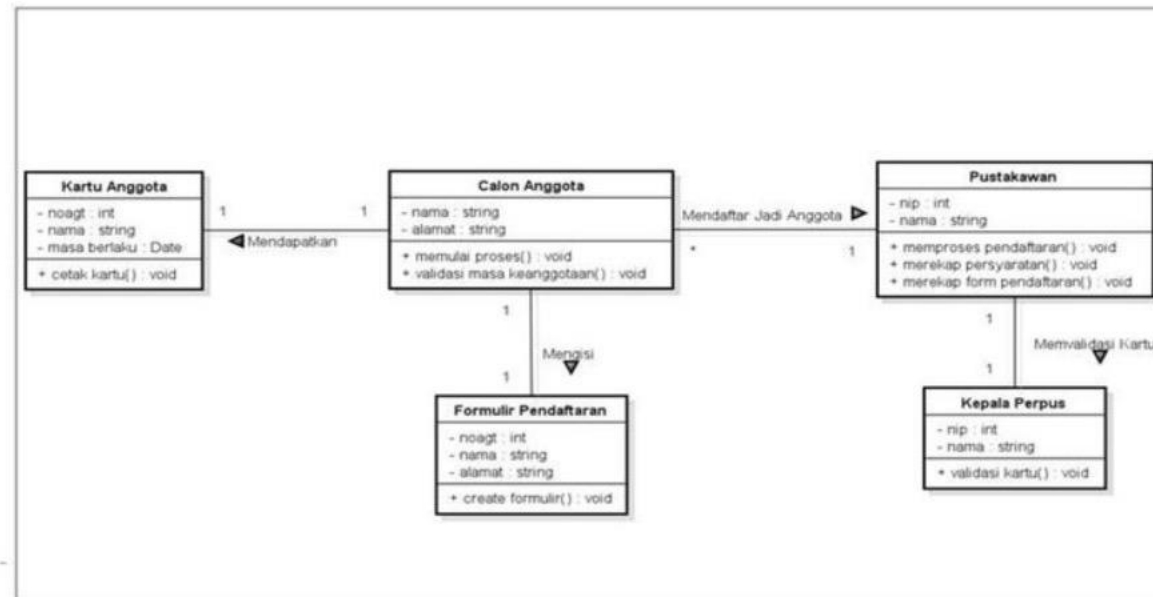
SEQUENCE DIAGRAM

- ▶ Sequence diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu. Sequence diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram.



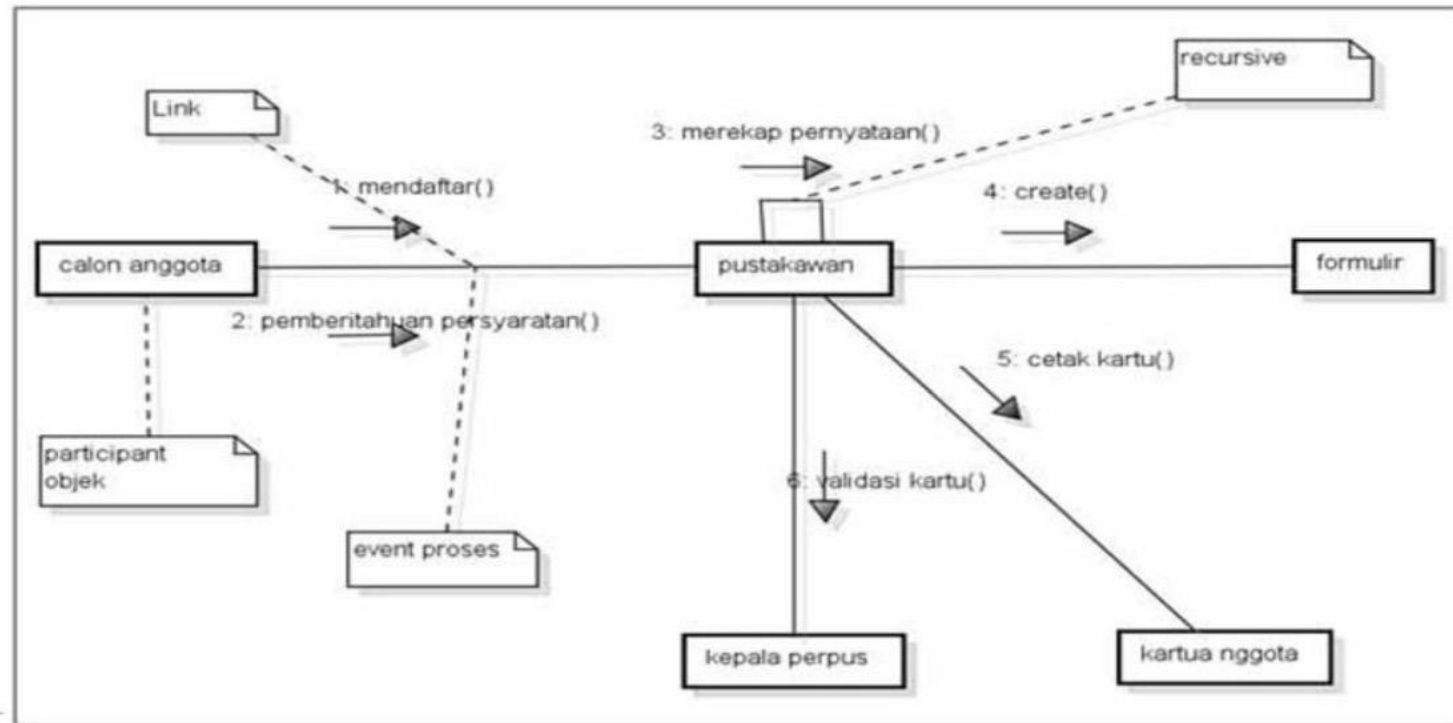
CLASS DIAGRAM

- Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.



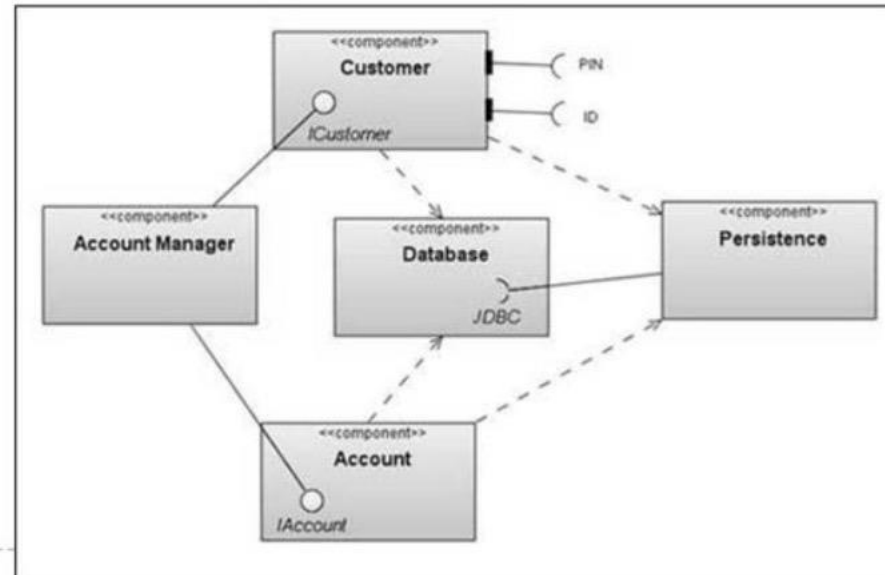
COMMUNICATION DIAGRAM

- ▶ Communication diagram yaitu salah satu jenis diagram pada UML yang dapat menggambarkan tahapan terjadinya suatu aktivitas dan diagram ini juga menggambarkan interaksi antara objek yang ada pada sistem.



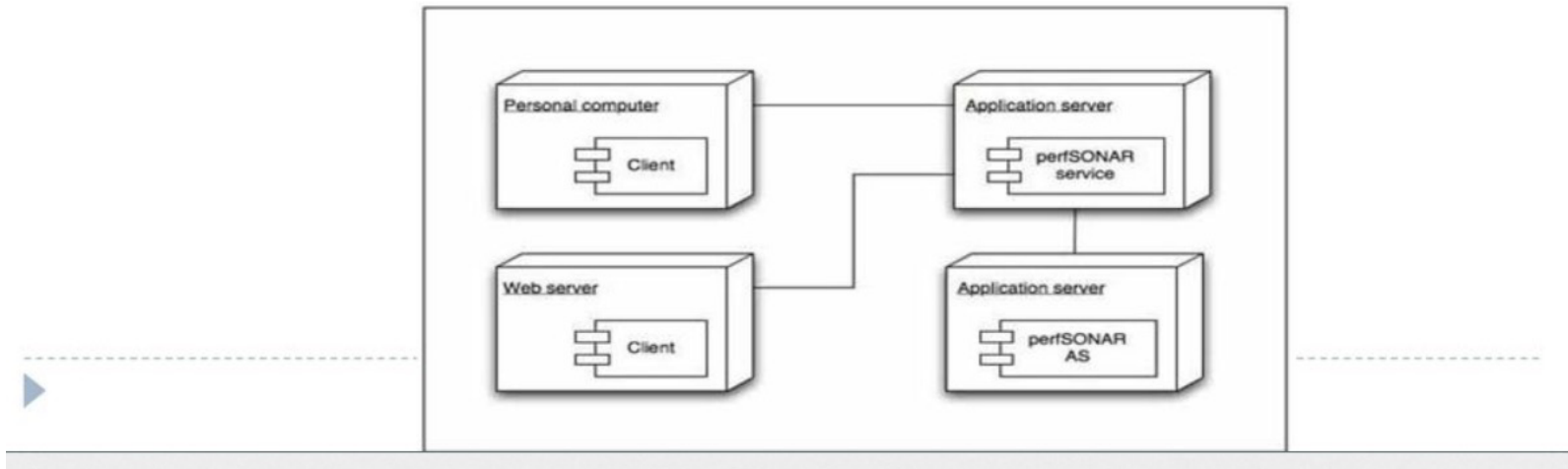
COMPONENT DIAGRAM

- ▶ Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan software pada suatu sistem. Component diagram merupakan penerapan software dari satu ataupun lebih class, dan biasanya berupa file data atau .exe, source code, table, dokumen dsb.



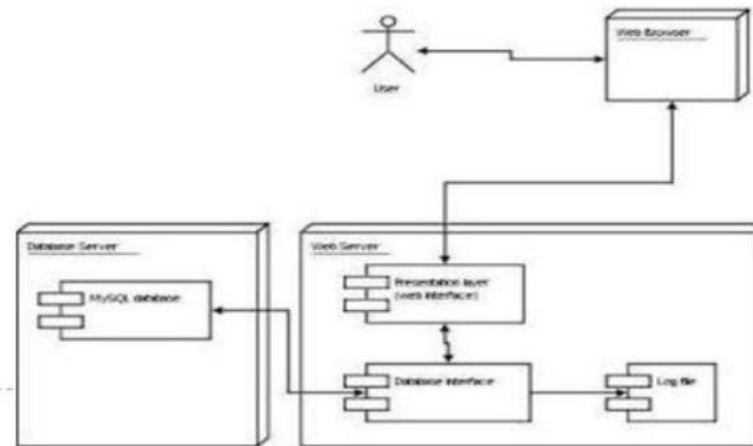
DEPLOYMENT DIAGRAM

- Deployment diagram yaitu salah satu diagram pada UML yang menunjukkan letak software pada hardware yang digunakan sistem.



Composite structure diagram

- ▶ Composite structure diagram yaitu salah satu jenis diagram pada UML yang menggambarkan struktur internal dari penklasifikasian (class, component atau use case) dan termasuk titik-titik interaksi penklasifikasian sebagian lainnya dari suatu sistem. Ini hampir mirip seperti class diagram akan tetapi composite structure diagram menggambarkan bagian-bagian dari individu kelas saja bukan semua kelas.



Statemachine diagram

- ▶ Statemachine diagram yaitu salah satu jenis diagram pada UML yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.

