



# Rekayasa Perangkat Lunak

## Pertemuan 4

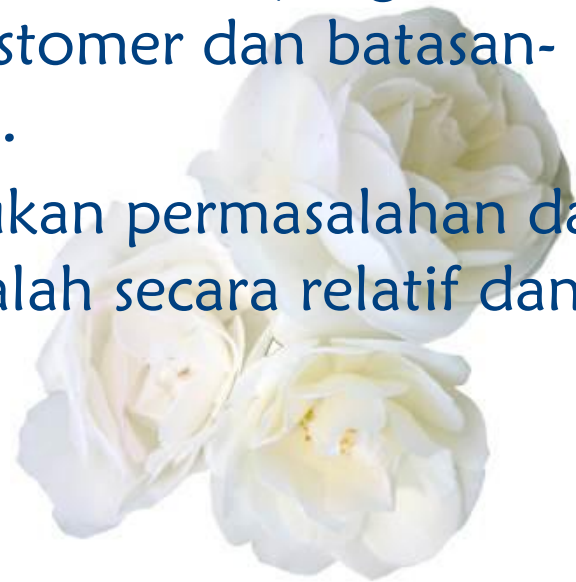
### Konsep dan Prinsip Analisis

### Dan

### Pemodelan Analisis

# Definisi

- Tahap Analisis : yaitu tahap dimana kita berusaha mengenali segenap permasalahan yang muncul pada pengguna, mengenali komponen-komponen sistem, obyek-obyek, hubungan antar obyek dan sebagainya
- Kebutuhan : Deskripsi fungsi kuantitas sistem yang berorientasi pada customer dan batasan-batasan yang membatasinya.
- Analisis Kebutuhan : Proses menemukan permasalahan dan memecahkan masalah secara relatif dan relevan.



- Pada tahap ini yang terlibat adalah tim spesifikasi/analisis dan customer (meliputi end-user, manajer dan staf lain yang terlibat).
- Tujuan Khusus : Untuk mengetahui kebutuhan Customer dengan sistem perangkat lunak yang diinginkan.



# Tujuan umum analisis sistem

- Mendefinisikan masalah secara tepat
- Menyusun alternatif penyelesaian
- Memilih dan mempertimbangkan satu dari alternatif tersebut
- Menyusun spesifikasi logis untuk penyelesaian
- Menyusun persyaratan fisik untuk penyelesaian
- Menyusun anggaran untuk fase desain sistem pengkodean dan implementasi sistem



# Bagaimana caranya ????

- Interview langsung
- Membuat Quisioner
- Menganalisa produk yang lama



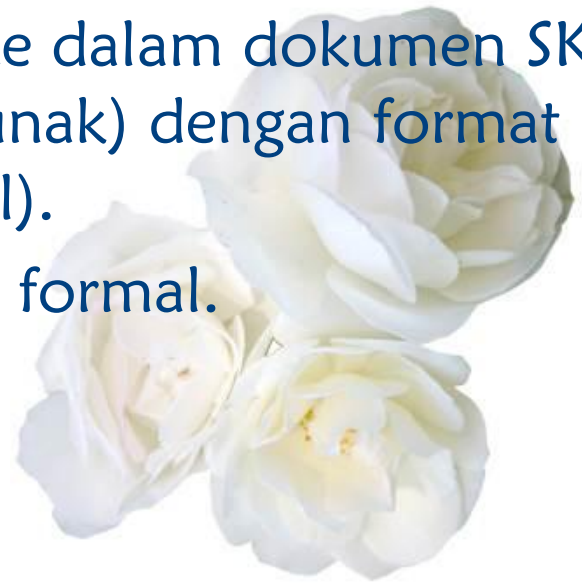
# Metric (Ukuran) Analisis Kebutuhan

- Mengukur apakah suatu kebutuhan didefinisikan dengan baik. Hal ini dapat dilihat antara lain dari persentase spesifikasi kebutuhan yang ambigu dan derajat kelengkapan kebutuhan yang didefinisikan
- Mengukur apakah inspeksi terhadap pendefinisian kebutuhan dilakukan secara efektif



# Lingkup Tahap Analisis

- Mengidentifikasi customer (termasuk pengguna)
- Mendefinisikan dan menspesifikasikan kebutuhan
- Membangun model analisis
- Mendefinisikan spesifikasi rinci untuk dijadikan panduan dalam melakukan perancangan
- Mendokumentasikan hasil analisis ke dalam dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) dengan format standar (misal : IEEE, NASA, ITB, dll).
- Melakukan pengkajian ulang secara formal.



# Prinsip-prinsip analisis

- Dominan informasi dari suatu masalah harus direpresentasikan dan dipahami.
- Fungsi – fungsi yang akan dilakukan oleh perangkat lunak harus di definisikan.
- Tingkah laku perangkat lunak (sebagai suatu urutan kejadian eksternal) harus diwakilkan.
- Model – model yang menggambarkan informasi, fungsi, dan tingkah laku harus dipecah – pecah dalam suatu cara yang membongkar suatu detail dalam bentuk lapisan (atau hirarki).
- Proses analisis harus bergerak dari informasi dasar ke detail implementasi.





# Spesifikasi kebutuhan PL

- Penulisan Definisi Kebutuhan:

1. Menggunakan bahasa natural (english):: meskipun secara universal dimengerti, namun tetap dimungkinkan muncul hal-hal berikut yang harus dihindari, antara lain :

- ✓ ketidakjelasan disebabkan dokumen yang dibuat sulit dibaca karena terlalu rinci
- ✓ pencampuradukan antara kebutuhan fungsional dan kebutuhan non fungsional
- ✓ beberapa kebutuhan diekspresikan dengan satu pernyataan

2. Menggunakan format standar (misal : IEEE)

- Spesifikasi Kebutuhan adalah deskripsi rinci dan terukur dari fungsi-fungsi dan batasan-batasan sistem yang telah didefinisikan



# Penulisan Spesifikasi Kebutuhan :

- Structured language specifications
  - ✓ Bentuk terbatas dari bahasa natural (english) yang digunakan untuk mengekspresikan kebutuhan
  - ✓ Menghilangkan beberapa problem yang diakibatkan oleh ambiguitas dan fleksibilitas
  - ✓ Sering didukung dengan penggunaan pendekatan berbasis form( Form-based specifications)
- Notasi Grafis
- Spesifikasi matematis



# Jenis kebutuhan

- Kebutuhan Data

- Kebutuhan Fungsional

Pendefinisian layanan yang harus disediakan, bagaimana reaksi sistem terhadap input dan apa yang harus dilakukan sistem pada situasi khusus.

- Kebutuhan Non Fungsional

Kendala pada pelayanan atau fungsi sistem seperti kendala waktu, kendala proses pengembangan, standard, dll.

- Kebutuhan Antarmuka



# Pemodelan Analisis



# Prinsip Pemodelan Analisis

- Memodelkan domain data
- Memodelkan Fungsi
- Memodelkan perilaku sistem
- Mempartisi model
- Mulai dengan fokus pada esensi dari permasalahan



- Memodelkan Domain Data, terdiri atas:
  - ✓ Mendefinisikan objek data
  - ✓ Mendeskripsikan atribut data
  - ✓ Mendefinisikan keterhubungan data
- Memodelkan Fungsi , terdiri atas:
  - ✓ Mengidentifikasi fungsi-fungsi yang mentransformasikan objek data
  - ✓ Mengidentifikasi bagaimana aliran data yang terdapat pada sistem
  - ✓ Mengidentifikasi entitas yang memproduksi data dan memanfaatkan informasi yang dihasilkan sistem



- Memodelkan Perilaku Sistem, meliputi :
  - ✓ Mengidentifikasi semua state yang dihasilkan sistem
  - ✓ Menspesifikasikan event yang menyebabkan perubahan dari satu state ke state yang lain
  
- Mempartisi model, meliputi :
  - ✓ Melakukan perincian objek data
  - ✓ Menyusun hirarki fungsional
  - ✓ Merepresentasikan perilaku sistem pada tiap level



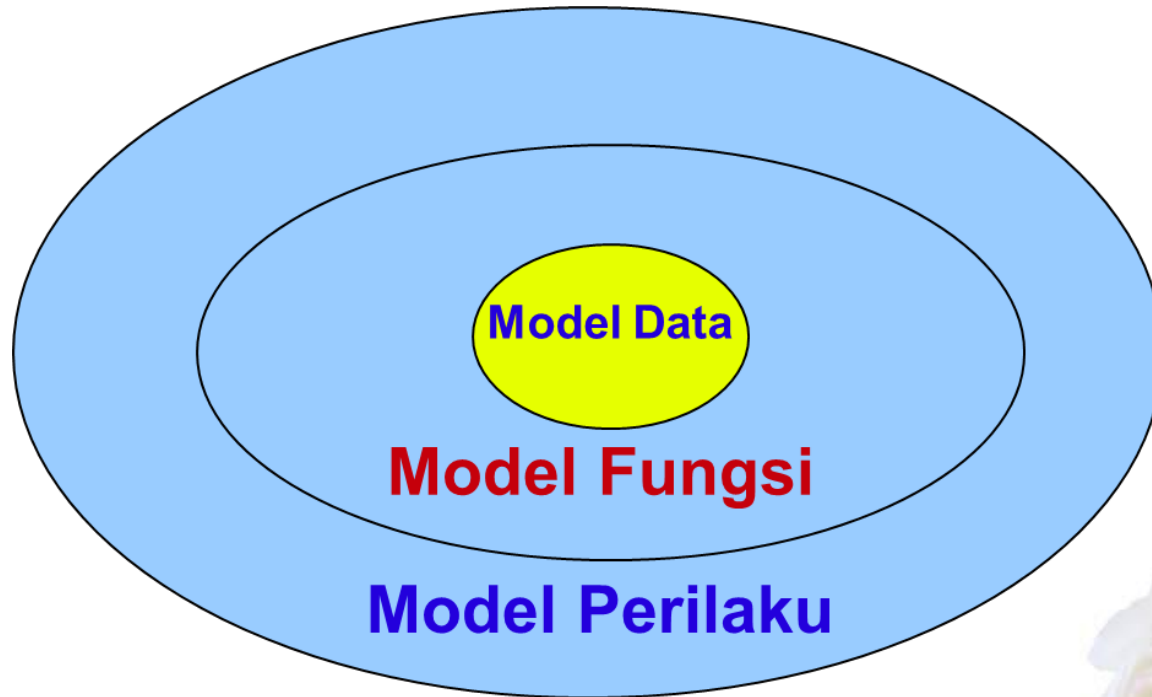
# Fungsi Pemodelan analisis

- Untuk memberikan gambaran logik perangkat lunak, data yang diperlukan dan model perilaku dari sistem perangkat lunak yang dikembangkan.





# Pemodelan Analisis



# Pemodelan analisis dimulai dengan :

- Mendefinisikan pernyataan ruang lingkup sistem
- Mendefinisikan objek-objek dan operasi
- Melakukan pemodelan data
- Melakukan pemodelan fungsi
- Melakukan pemodelan perilaku
- Mendefinisikan Spesifikasi Proses
- Mendeskripsikan Kamus Data



# Elemen Model Analisis

- Menggambarkan apa yang dibutuhkan untuk pelanggan
- Membangun dasar bagi pembuatan desain perangkat lunak
- Membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun.

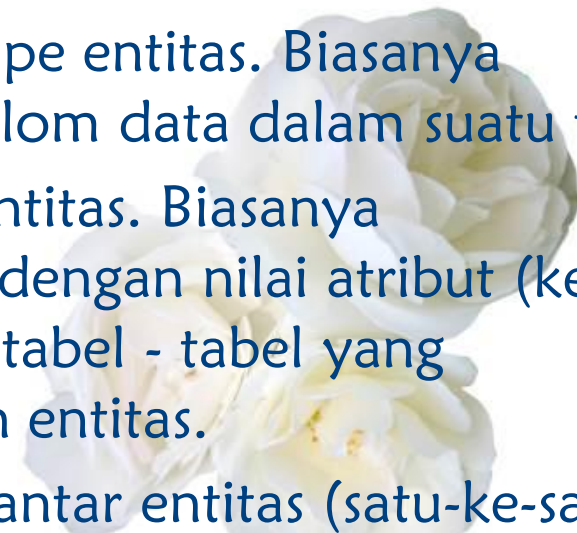


# Pemodelan Data

- Mendeskripsikan data yang terlibat dalam PL
- Tool yang digunakan:
  - Entity Relationship Diagram.
    - ✓ Diagram keterhubungan antar objek data
  - Data Object Description
    - ✓ Deskripsi atribut dari setiap objek data
  - Data Dictionary
    - ✓ Deskripsi semua objek data yang dibutuhkan maupun dihasilkan PL



# Komponen Model E-R

- **Entitas** : Orang, tempat, objek, event, konsep.
  - **Tipe Entitas** : Kategori / kelas untuk instan entitas sejenis.  
Biasanya dipresentasikan sebagai suatu Tabel.
  - **Instan Entitas** : Individu anggota suatu entitas, misalnya Andi, Rudy (orang), Jakarta, Surabaya (Kota), dsbnya.  
Biasanya direpresentasikan sebagai baris data dalam suatu tabel (Record).
  - **Atribut** : Ciri dan Karakteristik suatu tipe entitas. Biasanya direpresentasikan sebagai kolom data dalam suatu tabel
  - **Instan Hubungan** : Hubungan antar entitas. Biasanya direpresentasikan dengan nilai atribut (key) yang sama dalam tabel - tabel yang merepresentasikan entitas.
  - **Tipe Hubungan** : Kategori hubungan antar entitas (satu-ke-satu, satu-ke-banyak, banyak-ke-banyak).
- 

# Notasi Dasar ERD

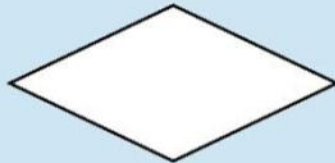
## Basic symbols



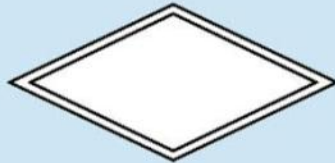
Strong entity



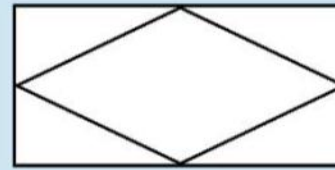
Weak entity



Relationship



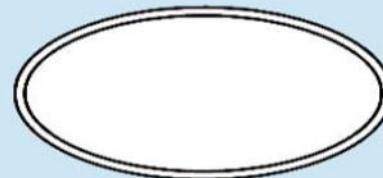
Identifying relationship



Associative entity



Attribute



Multivalued attribute



Derived attribute

# Kardinalitas Hubungan

- Satu-ke-Satu

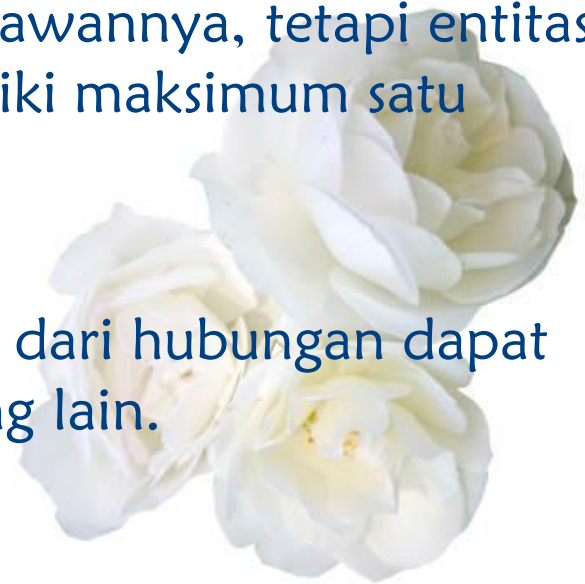
Setiap entitas dalam hubungan memiliki satu dan hanya satu entitas pasangan.

- Satu-ke-Banyak

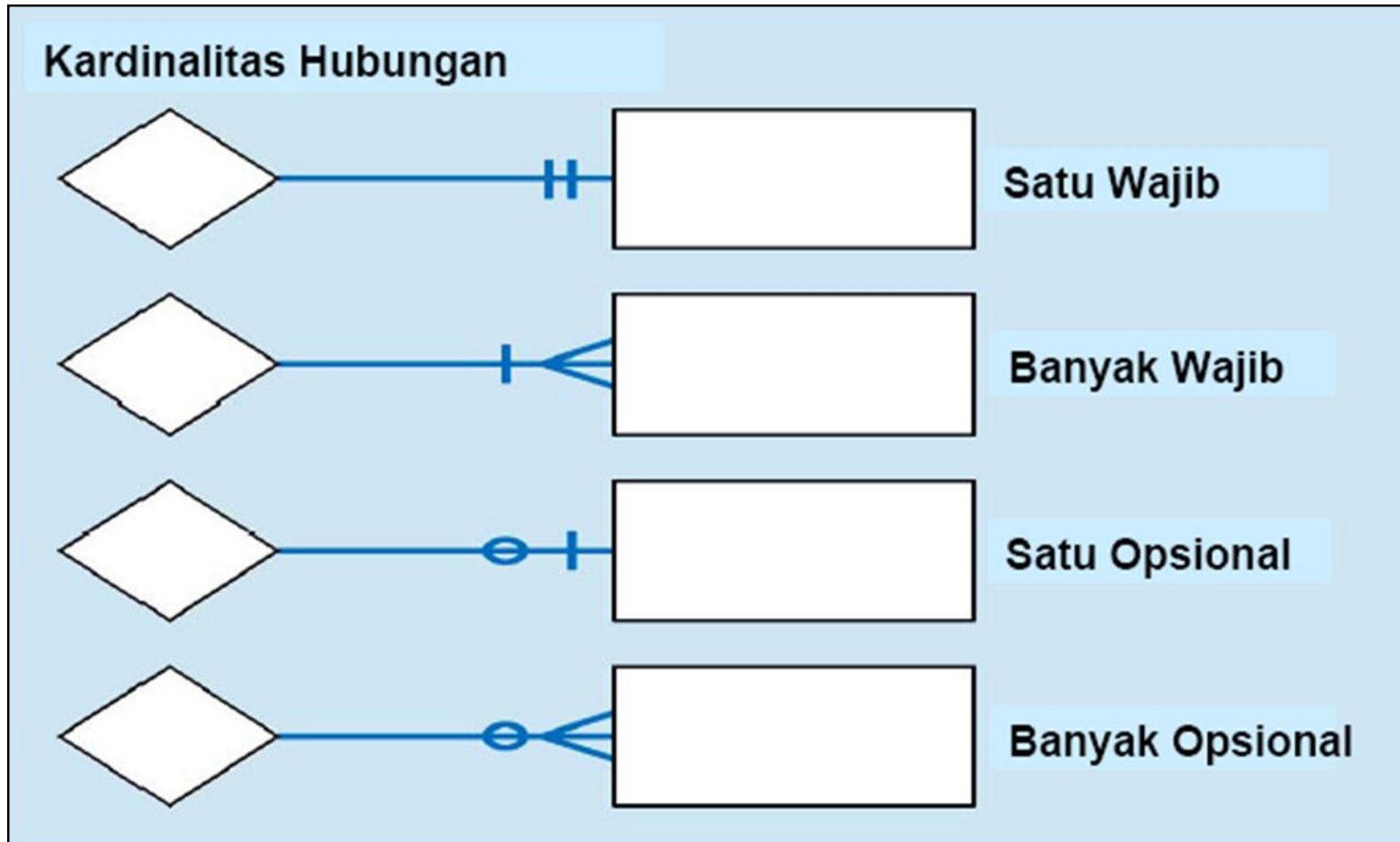
Satu entitas di satu pihak dalam suatu hubungan dapat memiliki beberapa pasangan di pihak lawannya, tetapi entitas di pihak lawannya hanya boleh memiliki maksimum satu pasangan.

- Banyak-ke-Banyak

Entitas – entitas di masing – masing sisi dari hubungan dapat memiliki beberapa pasangan di sisi yang lain.

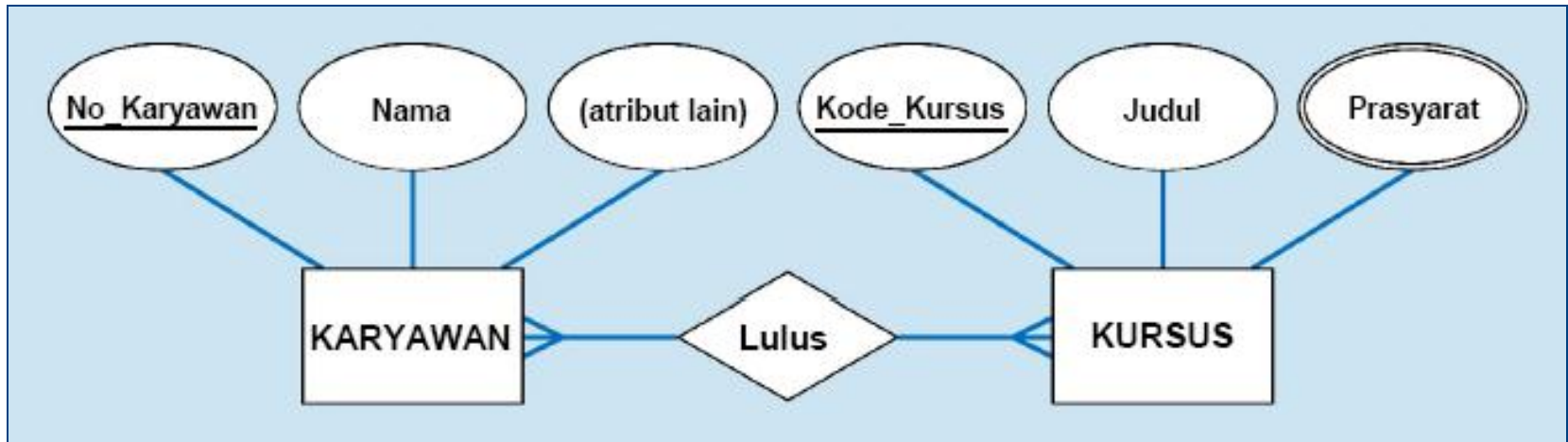


# Kardinalitas





# Contoh



## Catatan :

- Untuk kasus di mana data yang terlibat dalam sistem tidak memerlukan penyimpanan (basis data) atau jika objek-objek data tersebut tidak memiliki keterhubungan satu sama lain, maka ERD tidak perlu digambarkan.

