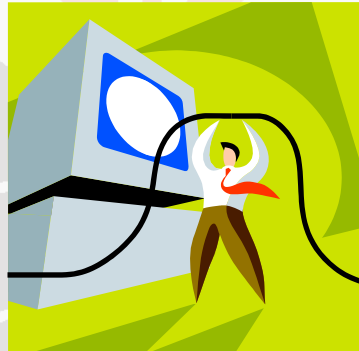


# Judul

## Deskripsi dan Spesifikasi *Kebutuhan Sistem Berbasis Komputer*



Oleh:  
**Tim Dit. TIK UPI**



## Tujuan

- Memperkenalkan konsep kebutuhan user dan Sistem
- Menggambarkan kebutuhan fungsional dan nonfungsional
- Menjelaskan dua teknik menggambarkan kebutuhan sistem
- Menjelaskan bagaimana kebutuhan software diorganisasikan dalam dokumen kebutuhan

## Materi

- Kebutuhan fungsional dan non-fungsional
- Kebutuhan user
- Kebutuhan sistem
- Dokumen kebutuhan software

# Rekayasa Kebutuhan

- Proses menetapkan layanan yang dibutuhkan konsumen terhadap sistem dan batasan operasi dan pengembangan
- Kebutuhan sendiri adalah diskripsi layanan sistem dan batasan yang dibangkitkan selama proses rekayasa kebutuhan

# Apa yang Dimaksud dengan Kebutuhan ?

- Susunan pernyataan abstrak level tinggi dari layanan atau batasan sistem ke dalam spesifikasi fungsional matematis
- Tidak terelakkan bahwa kebutuhan mempunyai dua fungsi

- ✓ merupakan dasar untuk penawaran kontrak sehingga harus terbuka untuk interpretasi
- ✓ merupakan dasar untuk kontrak itu sendiri sehingga harus didefinisikan dengan detail
- ✓ Kedua pernyataan diatas disebut kebutuhan

disebut  
kebutuhan

# Abstraksi Kebutuhan Perangkat Lunak (Davis)

“Jika sebuah perusahaan akan mengadakan kontrak untuk proyek pengembangan software besar, harus didefinisikan kebutuhan yang cukup pada saat dimana solusi belum terdefinisi.

Kebutuhan harus ditulis sehingga beberapa kontraktor dapat menawarkan kontrak, penawaran, kemungkinan, secara berbeda dengan kebutuhan organisasi client.

Bila kontrak sudah diserahkan, kontraktor harus menulis definisi sistem untuk client secara lebih detail sehingga client mengerti dan dapat mem-validasi software yang akan dikerjakan.

Kedua dokumen ini disebut *dokumen kebutuhan untuk sistem*”

# Tipe-Tipe Kebutuhan

## a. Kebutuhan User

Pernyataan dalam bahasa natural plus diagram layanan yang tersedia dan batasan operasional. Ditulis oleh konsumen

## b. Kebutuhan Sistem

Dokumen terstruktur berisi deskripsi detail dari layanan sistem. Ditulis sebagai kontrak antara klien dan kontraktor.

## c. Spesifikasi Software

Deskripsi software detail yang sebagai dasar untuk desain atau implementasi. Ditulis oleh developer

# Definisi dan Spesifikasi Kebutuhan

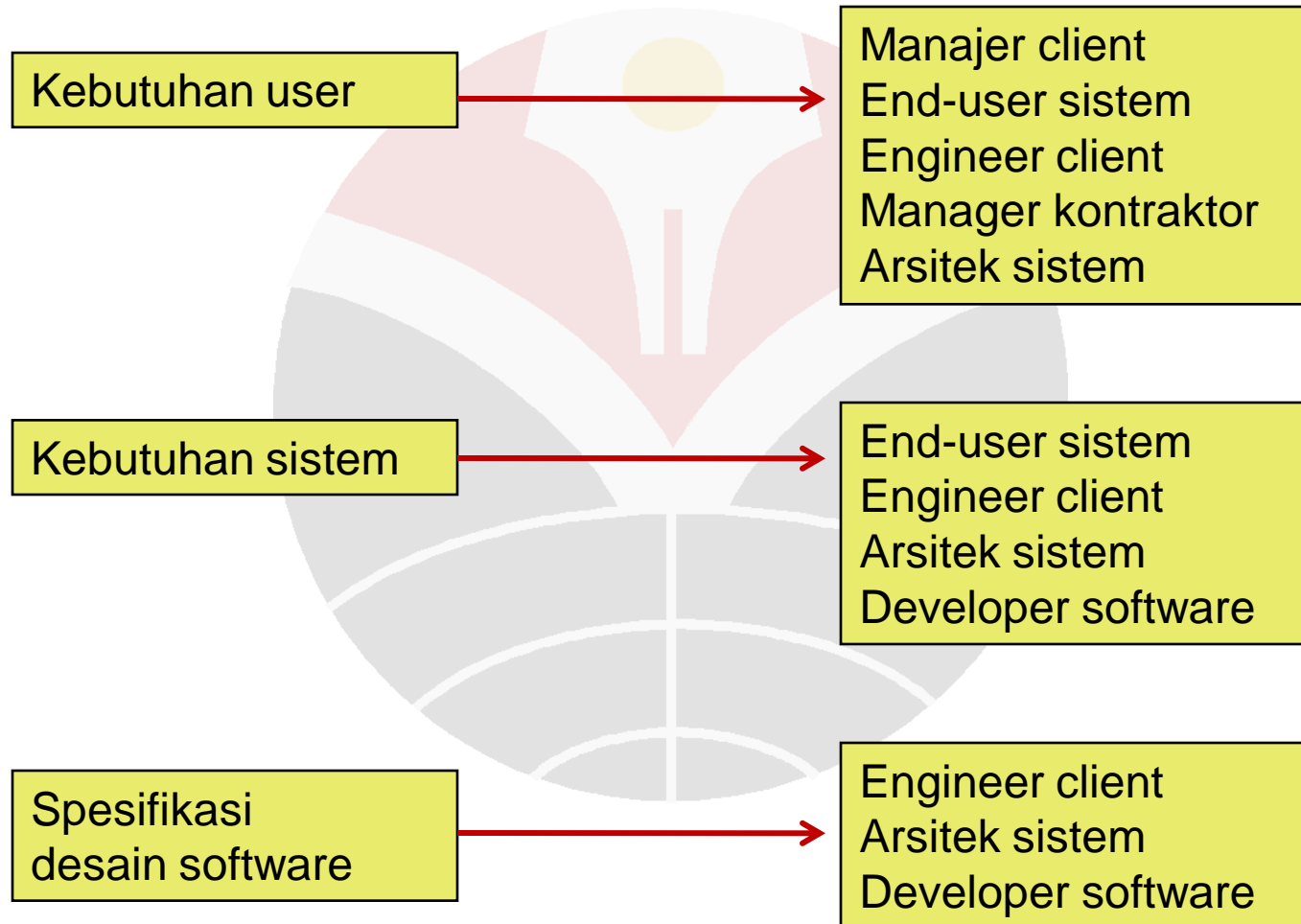
## Definisi Kebutuhan

1. Software harus menyediakan ketentuan menampilkan dan mengakses file yang dibuat oleh tool lain

## Spesifikasi Kebutuhan

- 1.1 User diberikan fasilitas untuk mendefinisikan tipe file eksternal
- 1.2 Setiap tipe file eksternal mempunyai alat untuk dihubungkan yang dapat diaplikasikan ke file
- 1.3 Setiap tipe file eksternal direpresentasikan sebagai icon tertentu pada tampilan user
- 1.4 Fasilitas disediakan untuk icon yang merepresentasikan tipe file eksternal yang didefinisikan oleh user
- 1.5 Jika user memilih icon untuk merepresentasikan file eksternal, efek pemilihan mengaplikasikan alat yang menghubungkan antara tipe file eksternal ke file yang direpresentasikan oleh icon terpilih

# Penggunaan Definisi dan Spesifikasi Kebutuhan





# Jenis Kebutuhan

## **Kebutuhan fungsional**

Pernyataan layanan sistem yang harus disediakan, bagaimana sistem bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu

## **Kebutuhan non-fungsional**

Batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dll

## **Kebutuhan domain**

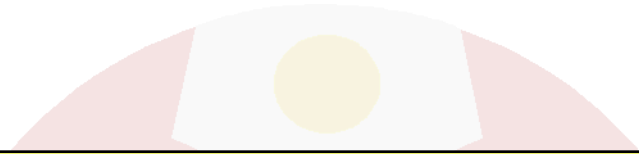
Kebutuhan yang datang dari domain aplikasi dari sistem dan yang menyatakan karakteristik dari domain tersebut

# Penggunaan

## Definisi dan Spesifikasi Kebutuhan

- Menggambarkan fungsionalitas atau layanan sistem
- Tergantung pada tipe software, harapan user dan tipe sistem dimana software digunakan
- Kebutuhan fungsional user merupakan pernyataan level tinggi dari apa yang seharusnya dilakukan sistem tetapi kebutuhan fungsional sistem menggambarkan layanan sistem secara detail

# Contoh Definisi Kebutuhan Fungsional



- User dapat mencari semua data mengenai aset yang telah habis
- Sistem menyediakan tampilan yang tepat untuk user untuk membaca dokumen-dokumen yang berkaitan dengan pengajuan pemanfaatan aset
- Setiap pesanan terhadap pemanfaatan aset harus dapat dialokasikan sebagai identifier yang unik (ORDER\_ID) dimana user dapat melakukan pencarian (tracking) terhadap dokumen-dokumen yang dimaksud

# Konsistensi dalam Kelengkapan Kebutuhan

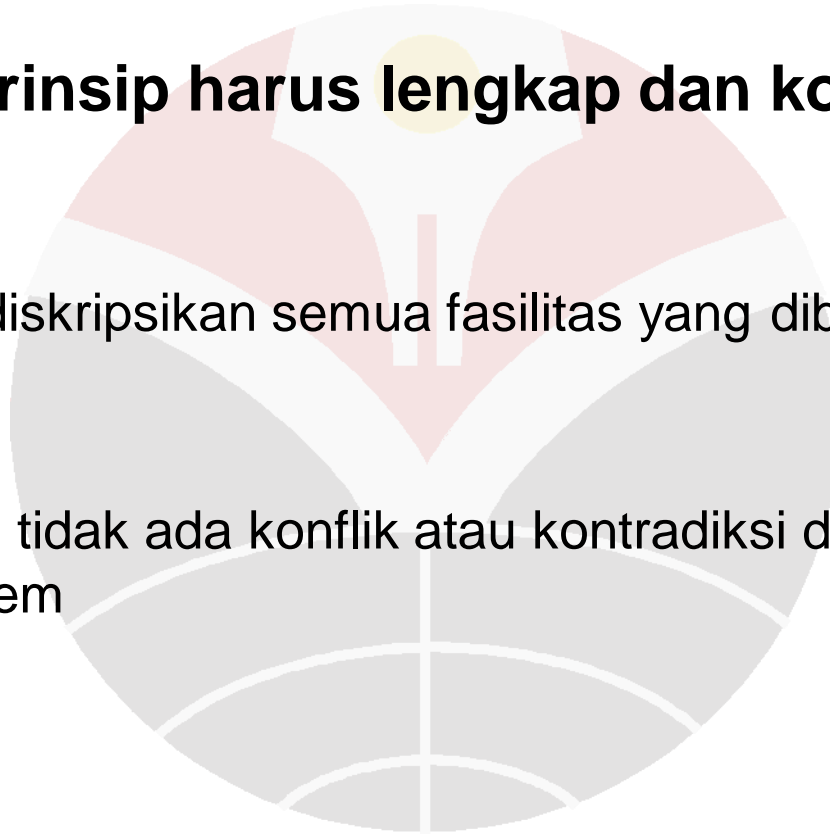
**Kebutuhan prinsip harus lengkap dan konsisten**

## **Lengkap**

Harus mendiskripsikan semua fasilitas yang dibutuhkan

## **Konsisten**

Seharusnya tidak ada konflik atau kontradiksi dalam diskripsi fasilitas sistem



# Kebutuhan Non-Fungsional

- Mendefinisikan properti sistem dan batasan sistem, seperti kehandalah, waktu respon, kebutuhan penyimpan. Batasan misalnya kapabilitas perangkat I/O, representasi sistem dll
- Kebutuhan proses juga menetapkan penggunaan sistem CASE khusus, bahasa pemrograman atau metode pengembangan
- Kebutuhan non-fungsional lebih kritis daripada kebutuhan fungsional. Jika tidak dapat bertemu, sistem menjadi tidak berguna



# Klasifikasi Kebutuhan Non Fungsional

## **Kebutuhan Produk**

Kebutuhan yang menetapkan bahwa produk yang dikirim harus berjalan dengan cara tertentu, contoh kecepatan eksekusi, kehandalan dll

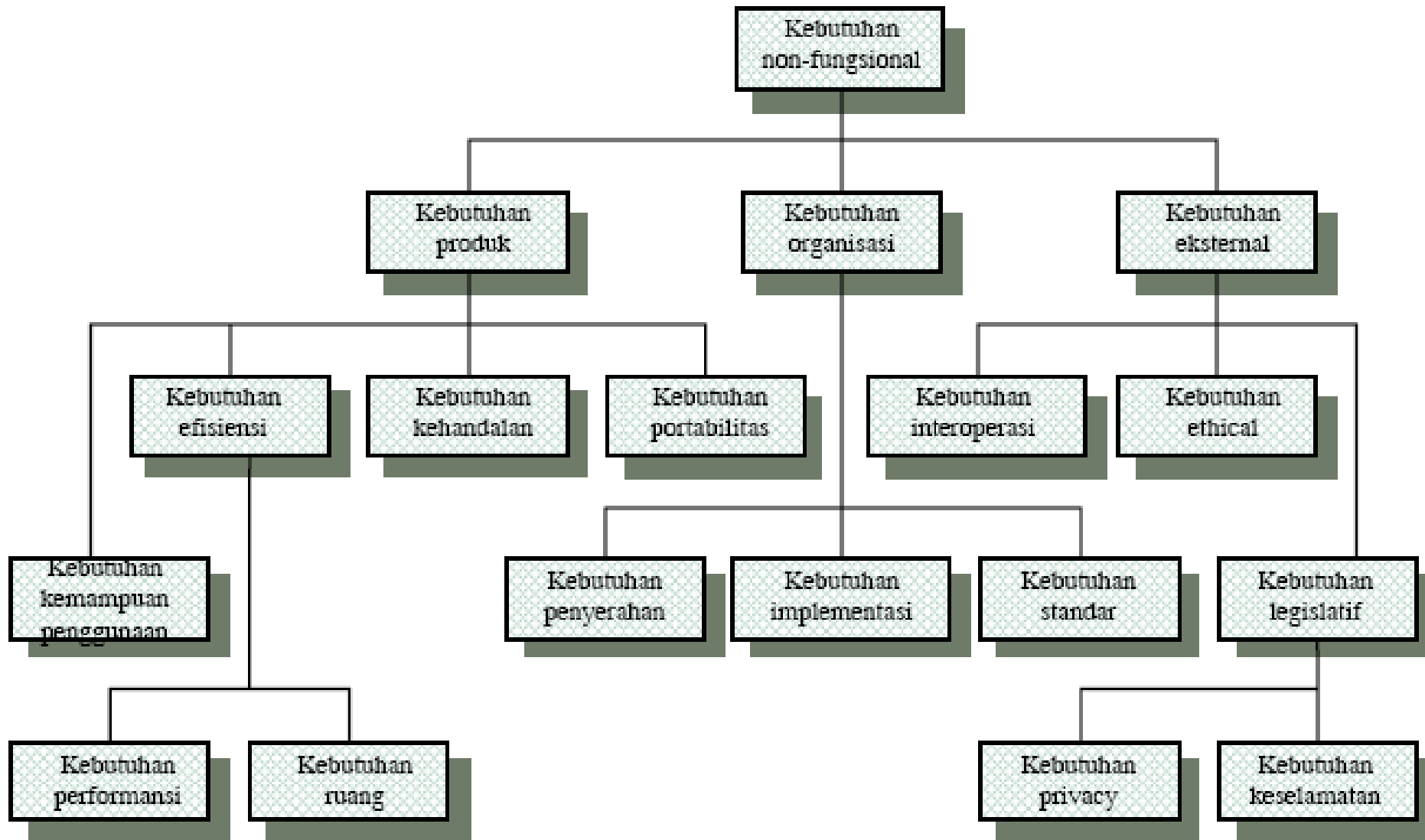
## **Kebutuhan Organisasi**

Kebutuhan sebagai akibat dari kebijakan organisasi dan prosedur misalnya standar proses yang digunakan, kebutuhan implementasi dll

## **Kebutuhan Eksternal**

Kebutuhan yang muncul dari faktor eksternal sistem dan proses pengembangan misalnya kebutuhan antar operasi, kebutuhan legislatif dll

# Tipe-tipe Kebutuhan Non-Fungsional



# Contoh Kebutuhan Non-Fungsional

## **Kebutuhan Produk**

Sistem aplikasi yang dibangun harus dapat bekerja secara simultan 6 hari kerja, selama 10 jam non stop, tanpa gagal dan error

## **Kebutuhan Organisasi**

Proses pengembangan sistem dan penyerahan dokumen seharusnya sesuai mekanisme dan aturan atau SOP yang tercantum dalam kebijakan yang telah didefinisikan dalam dokumen .....

## **Kebutuhan Eksternal**

Sistem seharusnya tidak tertutup untuk segala informasi personal mengenai konsumen dan dapat melakukan pencarian berdasarkan nama dan alamat sedemikian rupa sehingga tidak hanya berdasarkan nomor referensi konsumen ke operator sistem



# Kebutuhan dan Tujuan

Kebutuhan Non-fungsional kemungkinan sangat sulit untuk diterapkan dan kebutuhan yang tidak tepat sulit diverifikasi

*Hal yang harus diperhatikan :*

## Tujuan

Tujuan umum dari user misalnya kemudahan penggunaan

## Kebutuhan non-fungsional yang dapat diverifikasi

Pernyataan menggunakan beberapa ukuran yang dapat dites secara obyektif

# Contoh Kebutuhan Non-Fungsional

(berdasarkan tujuan dan kemudahan verifikasi)

## Tujuan Sistem

Sistem seharusnya mudah digunakan oleh pengguna dan diorganisasikan sehingga error user dapat diminimalkan

## Kebutuhan non-fungsional yang dapat diverifikasi

Pengguna seharusnya dapat menggunakan semua fungsi sistem setelah training selesai. Setelah training ini, jumlah rata-rata error yang dibuat oleh user yang berpengalaman tidak lebih dari 2 kali setiap hari

# Ukuran Kebutuhan

Properti	Ukuran
Kecepatan	Transaksi proses/detik Waktu respon user/event Waktu refresh screen
Ukuran	K byte Jumlah chip RAM
Kemudahan penggunaan	Waktu training Jumlah frame bantuan
Kehandalan	Rata-rata waktu kegagalan Kemungkinan ketidak tersediaan Rata-rata kejadian kegagalan Ketersediaan
Kekuatan	Waktu restart setelah kegagalan Persentase event yang menyebabkan kegagalan Kemungkinan korupsi data pada saat kegagalan
Portabilitas	Persentase pernyataan ketergantungan target Jumlah sistem target

# Kebutuhan Domain

- Diperoleh dari domain aplikasi dan menggambarkan karakteristik sistem dan fitur yang merefleksikan domain
- Berupa kebutuhan fungsional baru, batasan pada kebutuhan yang sudah ada atau mendefinisikan komputasi tertentu
- Jika kebutuhan domain tidak terpenuhi, sistem mungkin tidak dapat bekerja



## Contoh Kebutuhan Domain

- Setiap keterlambatan pengembalian buku adalah pengembalian yang dilakukan setelah 7 hari masa pinjam, dan konstanta denda untuk setiap keterlambatan adalah Rp. 5.000 / hari keterlambatan
- Karena faktor pembatasan copyright, maka setiap user yang akan mengakses lengkap data jurnal, harus melengkapi keanggotaan terlebih dahulu, sehingga yang ditampilkan pada sistem hanyalah abstraksi
- dst

# Permasalahan Dalam Kebutuhan Domain

## Kemampuan untuk dimengerti

- Kebutuhan dinyatakan dalam bahasa domain aplikasi
- Biasanya tidak dimengerti oleh software engineer yang mengembangkan sistem

## Ketidaklengkapan

- Domain spesialis mengerti area dengan baik sehingga mereka tidak berfikir untuk membuat kebutuhan domain secara eksplisit



# Kebutuhan User

- Menjelaskan kebutuhan fungsional dan nonfungsional sehingga user yang tidak mempunyai pengetahuan teknis detail dapat mengerti sistem
- Kebutuhan user didefinisikan menggunakan bahasa natural, tabel dan diagram
  - Ketidakjelasan
    - o Kecermatan sulit diwujudkan tanpa membuat dokumen yang sulit dibaca
  - Kebutuhan yang membingungkan
    - o Kebutuhan fungsional dan non-fungsional cenderung dicampur aduk
  - Penggabungan kebutuhan
    - o Beberapa kebutuhan yang berbeda dinyatakan bersama-sama

# Permasalahan dengan Bahasa Alami

## **Ketidakjelasan**

Kecermatan sulit diwujudkan tanpa membuat dokumen yang sulit dibaca

## **Kebutuhan yang membingungkan**

Kebutuhan fungsional dan non-fungsional cenderung dicampur aduk

## **Penggabungan kebutuhan**

Beberapa kebutuhan yang berbeda dinyatakan bersama-sama



# Definisi Kebutuhan Berbasis Bahasa Pemrograman (PDL)

- Kebutuhan dapat didefinisikan secara operasional menggunakan bahasa seperti bahasa pemrograman tetapi lebih fleksibel dalam ekspresi
- Cocok untuk 2 situasi yaitu
  - ✓ Dimana operasi ditentukan sebagai deretan aksi dan urutan sangat penting
  - ✓ Dimana antar muka hardware dan software harus dispesifikasi
- Kelemahannya
  - ✓ PDL tidak cukup ekspresif untuk mendefinisikan konsep domain
  - ✓ Spesifikasi akan dianggap sebagai desain daripada spesifikasi
  - ✓ Notasi hanya dapat dimengerti oleh orang yang mempunyai pengetahuan bahasa pemrograman

# Contoh Kebutuhan Berbasis PDL

(bagian dari spesifikasi ATM)

```
class ATM{
    // declarations here
    public static void main (String args[]) throws InvalidCard {
        try {
            thisCard.read () ; // may throw InvalidCard exception
            pin = KeyPad.readPin () ; attempts = 1 ;
            while ( !thisCard.pin.equals (pin) & attempts < 4 )
                {
                    pin = KeyPad.readPin () ; attempts = attempts + 1 ;
                }
            if (!thisCard.pin.equals (pin))
                throw new InvalidCard ("Bad RN");
            thisBalance = thisCard.getBalance () ;
            do { Screen.prompt (" Please select a service ") ;
                service = Screen.touchKey () ;
                switch (service) {
                    case Services.withdrawalWithReceipt:
                        receiptRequired = true ;
                }
            } while (true) ;
        }
    }
}
```



# Spesifikasi Kebutuhan Antarmuka

- Sebagian besar sistem harus beroperasi dengan sistem lain dan antar muka operasi harus ditentukan sebagai bagian kebutuhan
- Tiga tipe antar muka
  - ✓ Antar muka prosedural
  - ✓ Struktur data yang dapat ditukar
  - ✓ Representasi data
- Notasi formal adalah teknik efektif untuk spesifikasi antar muka



# Deskripsi Kebutuhan Antarmuka Berbasis PDL

```
interface PrintServer {  
  
    // defines an abstract printer server  
    // requires:      interface Printer, interface PrintDoc  
    // provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  
    void initialize ( Printer p ) ;  
    void print ( Printer p, PrintDoc d ) ;  
    void displayPrintQueue ( Printer p ) ;  
    void cancelPrintJob (Printer p, PrintDoc d) ;  
    void switchPrinter (Printer p1, Printer p2, PrintDoc d) ;  
} //PrintServer
```

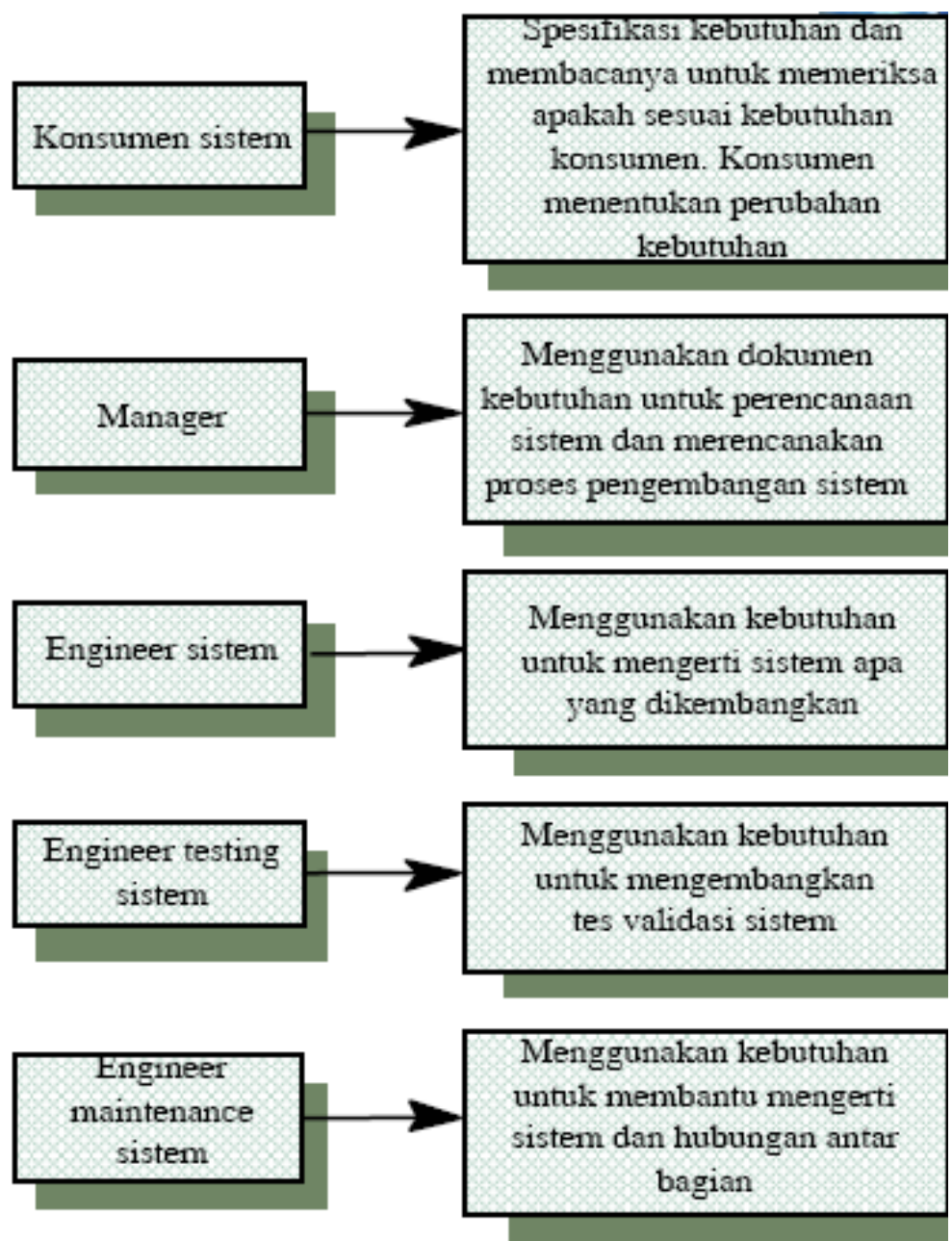


# Dokumen Kebutuhan

- Dokumen kebutuhan adalah pernyataan resmi dari apa yang dibutuhkan oleh developer sistem
- Terdiri dari definisi dan spesifikasi kebutuhan
- BUKAN dokumen desain. Se jauh mungkin, menentukan APA yang harus dikerjakan sistem daripada BAGAIMANA mengerjakannya



# User Dari Dokumen Kebutuhan



# Kebutuhan dari Dokumen Kebutuhan

- Menentukan perilaku sistem eksternal
- Menentukan batasan implementasi
- Mudah diubah
- Berlaku sebagai alat referensi untuk maintenance
- Menyimpan siklus hidup sistem, misalnya memprediksi perubahan
- Menentukan karakter respon untuk even yang tak diharapkan



# Standar IEEE untuk Kebutuhan

- Pendahuluan
- Deskripsi Umum
- Kebutuhan spesifik
- Lampiran
- Indeks
- Merupakan struktur generik yang harus disesuaikan sistem spesifik

## Rujukan Standar:

IEEE Std 830-1998 *IEEE Recommended Practice for Software Requirements Specifications*



# Struktur Umum Dokumen Kebutuhan

- Pendahuluan
- Daftar Istilah
- Definisi Kebutuhan User
- Arsitektur Sistem
- Spesifikasi kebutuhan sistem
- Model sistem
- Evolusi sistem
- Lampiran
- Indeks

## Rujukan Standar:

IEEE Std 830-1998 *IEEE Recommended Practice for Software Requirements Specifications*

# Summary

- Kebutuhan menentukan apa yang seharusnya dilakukan sistem dan menentukan batasan operasi dan implementasi
- Kebutuhan fungsional menentukan servis sistem yang harus disediakan
- Kebutuhan non-fungsional membatasi pengembangan sistem atau pengembangan proses
- Kebutuhan user adalah pernyataan level tinggi apa yang sistem harus kerjakan

# Summary

- Kebutuhan sistem dapat ditulis dalam bahasa alami, tabel dan diagram
- Kebutuhan sistem dimaksudkan untuk mengkomunikasikan fungsi yang harus disediakan sistem
- Kebutuhan sistem dapat ditulis dalam bahasa natural terstruktur, PDL atau bahasa formal
- Dokumen kebutuhan software adalah pernyataan persetujuan dari kebutuhan sistem





**Selesai  
Terima Kasih**

