

# REKAYASA PERANGKAT LUNAK

2



1

# ANALISIS KEBUTUHAN PERANGKAT LUNAK

(Software Requirement Analysis)



# KEBUTUHAN

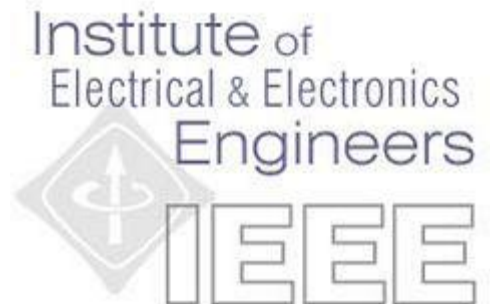
## *(Requirement)*

### ■ KEBUTUHAN *(Requirement)*

- ▷ Sesuatu yang diminta , dibutuhkan

### ■ Menurut IEEE

- Kondisi atau kemampuan yg diperlukan pemakai untuk menyelesaikan persoalan untuk mencapai sebuah tujuan
- Kondisi atau kemampuan yang harus dimiliki atau dipunyai oleh sistem atau komponen sistem untuk memenuhi kontrak, standar, spesifikasi, atau dokumen formal lainnya.



# KEBUTUHAN PERANGKAT LUNAK

“

*adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pengguna.*

# JENIS KEBUTUHAN

Secara kategoris, ada tiga buah jenis kebutuhan perangkat lunak [IEE93] :

1. Kebutuhan Fungsional (*Functional Requirement*)
2. Kebutuhan Antarmuka (*Interface Requirement*)
3. Kebutuhan Unjuk Kerja (*Performance Requirement*)

# KEBUTUHAN FUNGSIONAL

## *(Functional Requirement)*

- Disebut juga kebutuhan operasional, yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak.
- Contoh...
  - ▷ Perangkat lunak harus dapat menyimpan semua rincian data pesanan pelanggan.
  - ▷ Perangkat lunak harus dapat membuat laporan penjualan sesuai dengan periode waktu tertentu.
  - ▷ Perangkat lunak harus mampu menyajikan informasi jalur pengiriman barang terpendek.

# KEBUTUHAN ANTARMUKA

## *(Interface Requirement)*

- Kebutuhan antarmuka yang menghubungkan perangkat lunak dengan elemen perangkat keras, perangkat lunak, atau basis data.
- Contoh...
  - ▷ Perangkat untuk memasukkan data dapat berupa keyboard, mouse atau scanner.
  - ▷ Akses ke basisdata menggunakan ODBC (Open Database Connectivity).

# KEBUTUHAN UNJUK KERJA

## *(Performance Requirement)*

- Kebutuhan yang menetapkan karakteristik unjuk kerja yang harus dimiliki oleh perangkat lunak, misalnya: kecepatan, ketepatan, frekuensi.
- Contoh...
  - Perangkat lunak harus bisa mengolah data sampai 1 juta record untuk tiap transaksi.
  - Perangkat lunak harus dapat digunakan oleh multiuser sesuai dengan otoritas yang diberikan pada user.
  - Waktu tanggap penyajian informasi maksimal selama satu menit.



# ANALISIS KEBUTUHAN

“ *Analisis Kebutuhan Perangkat Lunak merupakan aktifitas awal dari siklus hidup pengembangan Perangkat Lunak*


Untuk proyek besar analisis kebutuhan dilaksanakan setelah aktifitas *sistem information engineering* dan *software projek planning*

# ANALISA KEBUTUHAN

- Proses mempelajari kebutuhan pemakai untuk mendapatkan definisi kebutuhan sistem atau perangkat lunak [IEE93].
- Proses untuk menetapkan fungsi dan unjuk kerja perangkat lunak, menyatakan antarmuka perangkat lunak dengan elemen-elemen sistem lain, dan menentukan kendala yang harus dihadapi perangkat lunak [PRE01].



# Pentingnya Analisis Kebutuhan

- Pendefinisian kebutuhan yang baik dapat menjadi faktor sukses pelaksanaan pengembangan perangkat lunak. Sebaliknya akan menyebabkan banyak kegagalan.
  - Menurut hasil survey DeMarco, 56% kegagalan proyek perangkat lunak adalah karena ketidaklengkapan pendefinisian kebutuhan.
- 

# Pentingnya Analisis Kebutuhan

- Produk perangkat lunak yang **tidak sempurna** akan dihasilkan karena *kesalahan* pada saat menentukan spesifikasi kebutuhan.
- Jika kesalahan tersebut diketahui di akhir siklus hidup pengembangan, usaha untuk memperbaikinya akan sangat mahal (sekitar 82% dari total biaya perbaikan).

# TUJUAN ANALISIS KEBUTUHAN

1. Memahami masalah secara menyeluruh (komprehensif) yang ada pada perangkat lunak yang akan dikembangkan seperti ruang lingkup produk perangkat lunak (product space) dan pemakai yang akan menggunakannya.
2. Mendefinisikan apa yang harus dikerjakan oleh perangkat lunak untuk memenuhi keinginan pelanggan.

# Tahap Analisis Kebutuhan

- Tahap kebutuhan perangkat lunak dimulai dengan [DAV93]:
  - Adanya **masalah** yang membutuhkan penyelesaian:
    - orientasi aplikasi, misalnya *inventory*
    - orientasi bisnis, misalnya produk baru, peramalan pendapatan
    - orientasi peningkatan produk, misalnya pemeliharaan
  - Munculnya **ide** untuk membuat sebuah perangkat lunak baru.

# TAHAPAN ANALISIS KEBUTUHAN

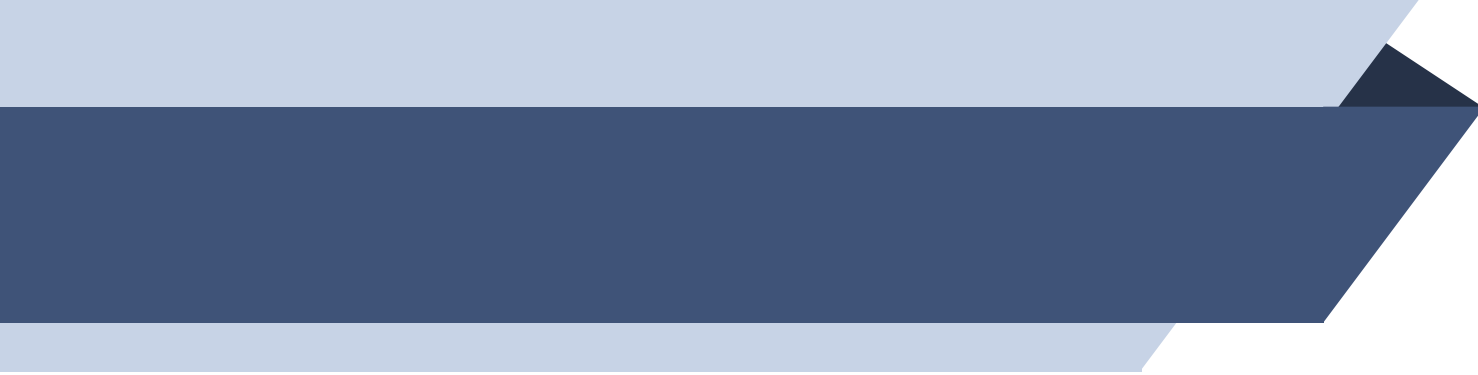
- Secara teknis pelaksanaan pekerjaan analisis kebutuhan perangkat lunak pada dasarnya terdiri dari urutan aktivitas:
  1. Mempelajari dan memahami persoalan
  2. Mengidentifikasi kebutuhan pemakai
  3. Mendefinisikan kebutuhan perangkat lunak
  4. Membuat dokumen spesifikasi kebutuhan perangkat lunak
  5. Mengkaji ulang (review) kebutuhan

# Mempelajari dan memahami persoalan


Pada tahap ini, seorang analis mempelajari masalah yang ada pada perangkat lunak yang dikembangkan, sehingga dapat ditentukan :

- siapa pemakai yang menggunakan perangkat lunak.
- dimana perangkat lunak akan digunakan .
- pekerjaan apa saja dari pemakai yang akan dibantu oleh perangkat lunak.
- apa saja cakupan dari pekerjaan tersebut, dan bagaimana mekanisme pelaksanaannya.
- apa yang menjadi kendala dilihat dari sisi teknologi yang digunakan atau dari sisi hukum dan standar.





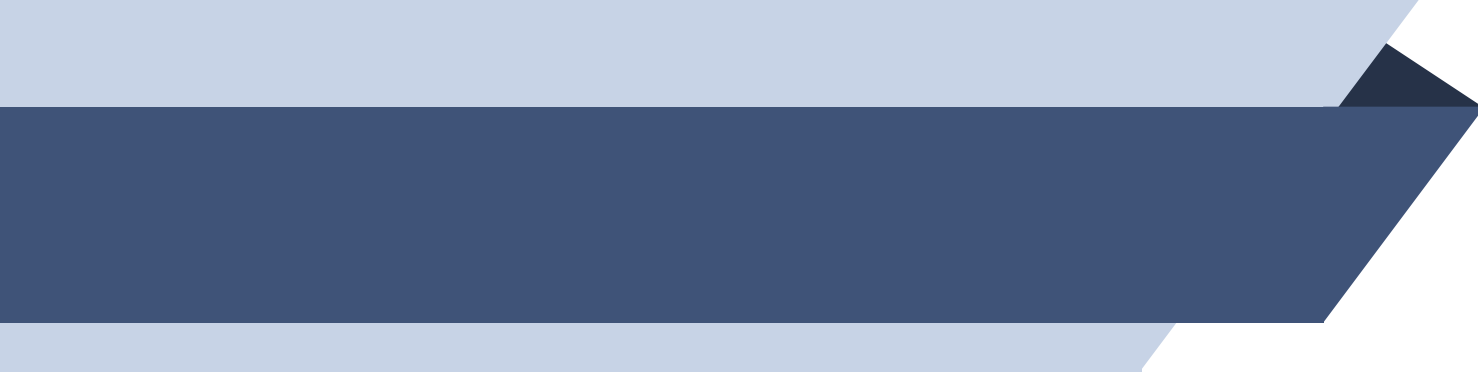
Cara yang digunakan oleh pengembang khususnya analis dalam memahami masalah perangkat lunak biasanya dilakukan

- wawancara dengan pemakai
  - observasi atau pengamatan lapangan
  - kuesioner
  - mempelajari referensi atau dokumen-dokumen yang digunakan, seperti dokumen hasil analisa dan perancangan perangkat lunak.
- 


## Mengidentifikasi kebutuhan pemakai

Pada tahap identifikasi kebutuhan pemakai (user requirement) pada prakteknya menjadi satu pelaksanaannya dengan pemahaman masalah. Hanya saja substansi yang ditanyakan ada sedikit perbedaan, yaitu

- fungsi apa yang diinginkan pada perangkat lunak.
- data atau informasi apa saja yang akan diproses.
- kelakuan sistem apa yang diharapkan.
- antarmuka apa yang tersedia (software interfaces, hardware interfaces, user interfaces, dan communication interfaces)

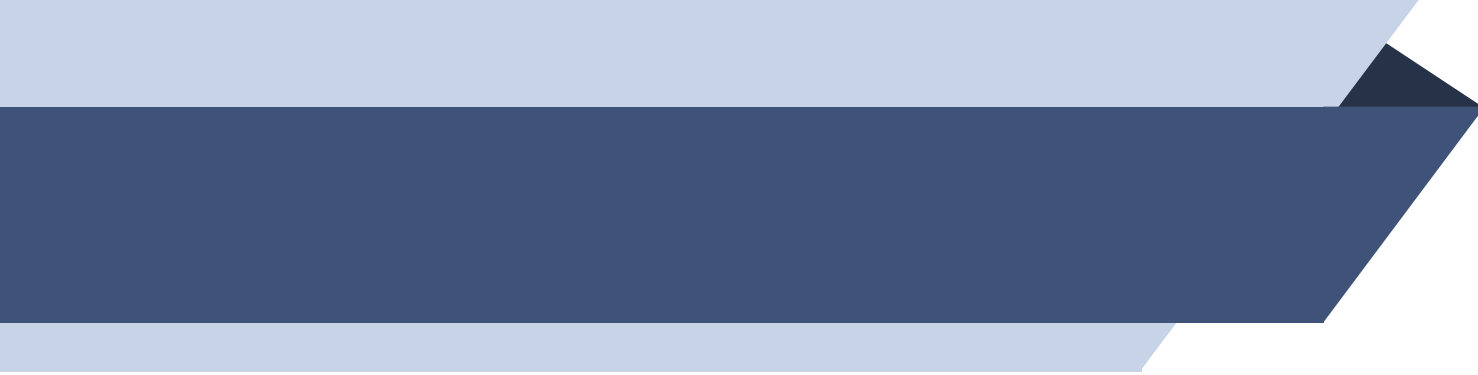



Untuk menangkap kebutuhan dari pemakai dengan baik, terutama kesamaan persepsi. seorang analis membutuhkan

- komunikasi dan brainstorming yang intensif dengan pelanggan.
  - pembuatan prototype perangkat lunak atau screenshoot.
  - Data atau dokumen yang lengkap.
- 

## Mendefinisikan kebutuhan perangkat lunak

- Saat melakukan pengidentifikasian kebutuhan pemakai, informasi yang diperoleh masih belum terstruktur. Biasanya pemakai akan mengungkapkan apa yang diinginkan dengan bahasa sehari-hari yang biasa mereka gunakan.
- Sebagai contoh, ungkapan kebutuhan pemakai dibagian akutansi.
  - saya ingin data yang dimasukkan oleh bagian penjualan bisa langsung dijurnal.
  - Informasi neraca keuangan bisa saya lihat kapan saja.

- 
- Kemudian pada tahap ini, kebutuhan pemakai yang belum terstruktur tersebut akan akan dianalisis, diklasifikasikan, dan diterjemahkan menjadi kebutuhan fungsional, antarmuka dan unjuk kerja perangkat lunak.
- 

## Contoh :

- Kebutuhan “data yang dimasukkan oleh bagian penjualan bisa langsung di jurnal” setelah dianalisis, diklasifikasikan dan diterjemahkan, mungkin akan menghasilkan pendefinisian kebutuhan sebagai berikut.

## “ ■ Kebutuhan Fungsional

- Entri dan rekam data transaksi penjualan.
- Retrieve data transaksi penjualan untuk periode tertentu (periode sesuai dengan inputan periode yang diinputkan pada keyboard).
- Rekam data akumulasi transaksi penjualan periode tertentu ke jurnal umum berikut account pasangannya (kas).

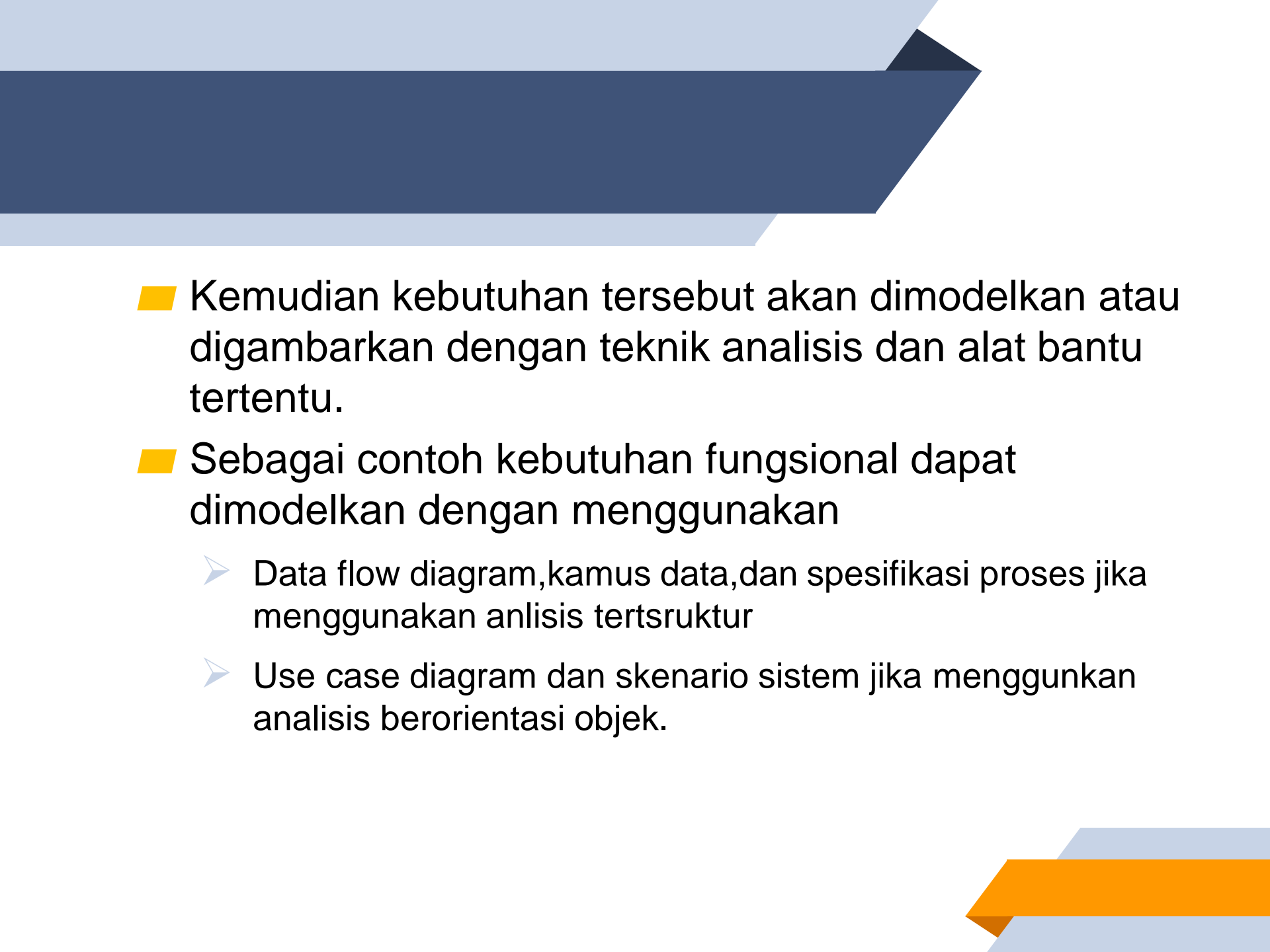
## “ ■ **Kebutuhan Antarmuka**

- ▷ Antarmuka pemakai untuk memasukkan dan merekam data penjualan.
- ▷ Antarmuka pemakai untuk menyajikan dan menjurnal informasi transaksi penjualan pada periode tertentu.
- ▷ Antarmuka untuk jaringan lokal yang menghubungkan perangkat lunak aplikasi dibagian penjualan dengan perangkat lunak aplikasi dibagian akutansi.



## “ ■ Kebutuhan Unjuk Kerja

- proses jurnal hanya bisa dilakukan sekali setelah data transaksi penjualan direkam.
- Adanya otoritas pemakaian perangkat lunak dan akses data sesuai dengan bagian pekerjaan masing-masing.

- 
- Kemudian kebutuhan tersebut akan dimodelkan atau digambarkan dengan teknik analisis dan alat bantu tertentu.
  - Sebagai contoh kebutuhan fungsional dapat dimodelkan dengan menggunakan
    - Data flow diagram, kamus data, dan spesifikasi proses jika menggunakan analisis terstruktur
    - Use case diagram dan skenario sistem jika menggunakan analisis berorientasi objek.

## Membuat dokumen spesifikasi kebutuhan perangkat lunak

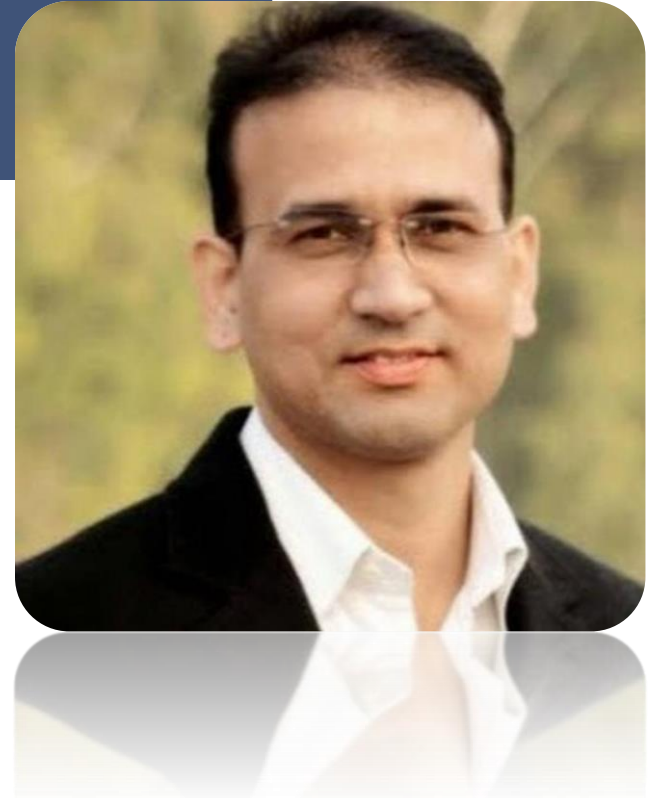
- Semua kebutuhan yang telah didefinisikan selanjutnya dibuat dokumentasinya yaitu Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau Software Requirement Specification (SRS).

## Mengkaji ulang (review) kebutuhan

- Proses untuk mengkaji ulang (validasi) kebutuhan apakah SKPL sudah konsisten, lengkap, dan sesuai dengan yang diinginkan oleh pemakai.

■ Sedangkan menurut Pressman [PRE01], analisis kebutuhan perangkat lunak dapat dibagi menjadi lima area pekerjaan, yaitu:

1. Pengenalan masalah
2. Evaluasi dan sistesis
3. Pemodelan
4. Spesifikasi
5. Tinjau ulang (review)



# METODE ANALISIS

# METODE ANALISIS

“ *Metode atau teknik untuk melakukan analisis kebutuhan perangkat lunak dapat dikelompokkan berdasarkan pendekatan yang diambil pada saat melakukan aktivitas tersebut*

# METODE ANALISIS

- Analisis Terstruktur (*Structured Analysis*)
- Berorientasi Aliran Data (*Data Flow Oriented atau Functional Oriented*)
- Berorientasi Struktur Data (*Data Structured Oriented*)
- Berorientasi Objek (*Object Oriented*)



# Berorientasi Aliran Data

*(Data Flow Oriented atau Functional Oriented )*

- Pada metode ini, hasil analisis dan perancangan dimodelkan dengan menggunakan beberapa perangkat pemodelan seperti:
  - Data Flow Diagram (DFD) dan Kamus Data (data dictionary) untuk menggambarkan fungsi-fungsi dari sistem (system functions).
  - Entity-Relationship Diagram (ERD) untuk menggambarkan data yang disimpan (data stored).
  - State Transition Diagram (STD) untuk menggambarkan perilaku sistem.
  - Structure Chart untuk menggambarkan struktur program.

# BERORIENTASI STRUKTUR DATA

*(Data Structured Oriented)*

“ Analisis dengan pendekatan ini difokuskan pada struktur data, dimana struktur tersebut dapat dinyatakan secara hirarki dengan menggunakan konstruksi sequence, selection dan repetition.

# Berorientasi Struktur Data

## *(Data Structured Oriented)*

■ Beberapa metode berorientasi struktur data ini diantaranya adalah:

▷ **Data Structured System Development (DSSD)**

Diperkenalkan pertama kali oleh J.D. Warnier [1974] dan kemudian oleh Ken Orr [1977], sehingga sering disebut juga metode Warnier-Orr. Metode ini menggunakan perangkat entity diagram, assembly line diagram dan Warnier-Orr diagram untuk memodelkan hasil analisis dan perancangannya.

▷ **Jackson System Development (JSD)**

Dikembangkan oleh M.A. Jackson [1975] dengan menggunakan perangkat pemodelan yang disebut structure diagram dan system specification diagram.

# BERORIENTASI OBJEK

## *(Object Oriented)*

“ Pendekatan berorientasi objek memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Pada pendekatan ini, informasi dan proses yang dipunyai oleh suatu Objek “dienkapsulasi” (dibungkus) dalam satu kesatuan.

# Berorientasi Objek (*Object Oriented*)

■ Beberapa metode pengembangan sistem yang berorientasi objek ini diantaranya adalah:

- Object Oriented Analysis (OOA) dan Object Oriented Design (OOD) dari Peter Coad dan Edward Yourdon (1990).
- Object Modeling Technique (OMT) dari James Rumbaugh (1987).
- Object Oriented Software Engineering (OOSE).

# SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

*Software Requirements Specification  
(SRS)*

# SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

*Software Requirements Specification (SRS)*

“

*sebuah dokumen yang berisi pernyataan lengkap dari apa yang dapat dilakukan oleh perangkat lunak, tanpa menjelaskan bagaimana hal tersebut dikerjakan oleh perangkat lunak.*


# Tujuan Pembuatan SRS

- Pemakai potensial (pelanggan) dari sistem
- Pengembang sistem






## Tujuan.....


1. Mendefinisikan keinginan yang biasanya dinyatakan dalam bentuk penjelasan umum.
  2. Tujuan kedua:
    - ▶ Sarana komunikasi antara pelanggan, pemakai, analis, dan perancang perangkat lunak.
    - ▶ Dasar untuk merencanakan dan melaksanakan aktivitas pengujian sistem.
    - ▶ Acuan untuk melakukan perbaikan dan perubahan perangkat lunak.
- 

# Syarat Pembentukan SRS

- Mudah diidentifikasi
- Diuraikan dengan jelas, simple, sederhana, dan tidak ambiguous
- Bisa divalidasi dan bisa dites (test reliable, test accessible)
- Mampu untuk ditelusuri kembali (traceability)



## Hal-hal yg harus dihindari saat pembentukan :

- Over specification (penjelasan berlebih dan berulang-ulang sehingga menjadi tidak jelas)
  - Tindakan unconcistency (seperti menggunakan istilah yang tidak konsisten)
  - Ambiguity dalam kata atau kalimat seperti menyatakan keterukuran kebutuhan secara tidak jelas misalkan menggunakan kata-kata : minimal, maksimal, optimal, cepat, user friendly, efisien, fleksible dan lainnya.
  - Menuliskan “mimpi-mimpi”, yaitu hal-hal yang tidak bisa dilakukan
- 

# Atribut Penulisan SRS yang Baik

■ Dokumen SRS yang baik (sempurna) akan ditulis secara:


1. Benar (correct)
2. Tepat (precise)
3. Unambiguouity
4. Lengkap (complete)
5. Bisa diverifikasi (verifiable)
6. Konsisten
7. Understandable
8. Bisa dimodifikasi (modifiedable)


# Atribut Penulisan SRS yang Baik

9. Dapat ditelusuri (traceable)
10. Harus dapat dibedakan bagian what (bagian spesifikasi) dan how (bagian yang menjelaskan bagaimana menyelesaikan what tadi).
11. Dapat mencakup dan melingkupi seluruh sistem
12. Dapat melingkupi semua lingkungan operasional, misalnya interaksi fisik dan operasional.
13. Bisa menggambarkan sistem seperti yang dilihat oleh pemakai.
14. Harus toleran (bisa menerima) terhadap ketidaklengkapan, ketidakpastian(ambiguous) dan ketidakkonsistenan.
15. Harus bisa dilokalisasi dengan sebuah coupling, yaitu hubungan ketergantungan antara dua model yang tidak terlalu erat.




Yang terlibat dalam pembuatan SKPL:

1. Pemakai (user)
  2. Client
  3. System analyst (system engineer)
  4. Software engineer
  5. Programmer
  6. Test integration group
  7. Maintenance group
  8. Technical Support
  9. Staff dan Clerical Work
- 



## Keberhasilan pengembangan perangkat lunak bisa dilihat dari 10 aspek atau titik pandang:

1. Ketelitian dari pembuatnya
  2. Kualitas dari spesifikasi perangkat lunak yang dihasilkan (baik, jika ada sedikit kesalahan)
  3. Integritas
  4. Ketelitian
  5. Proses pembuatan yang mantap
  6. Mudah dikembangkan
  7. Jumlah versi tidak banyak
  8. Ketelitian dari model pengembangan yang digunakan untuk meramal atribut perangkat lunak
  9. Efektivitas rencana tes dan integrasi
  10. Tingkat persiapan untuk sistem perawatan (mempersiapkan pencarian bugs)
- 

# Dokumen SRS

format dokumen SRS baku menurut ANSI/IEEE std 830 1984 adalah:

1. PENDAHULUAN
  - 1.1 Tujuan
  - 1.2 Ruang Lingkup
  - 1.3 Definisi
  - 1.4 Referensi
  - 1.5 Sistematika
2. DESKRIPSI UMUM
  - 2.1 Perspektif
  - 2.2 Kegunaan
  - 2.3 Karakteristik Pengguna
  - 2.4 Batasan-batasan
  - 2.5 Asumsi dan Ketergantungan
3. SPESIFIKASI KEBUTUHAN
  - 3.1 Kebutuhan Fungsional
    - 3.1.1 Pendahuluan
    - 3.1.2 Input
    - 3.1.3 Proses
    - 3.1.4 Output
  - 3.2 Kebutuhan Antarmuka Eksternal
    - 3.2.1 Antarmuka Pengguna
    - 3.2.2 Antarmuka Perangkat Keras
    - 3.2.3 Antarmuka Perangkat Lunak
    - 3.2.4 Antarmuka Komunikasi
  - 3.3 Kebutuhan Performansi
  - 3.4 Kendala Disain
    - 3.4.1 Standard Compliance
    - 3.4.2 Perangkat Keras
  - 3.5 Atribut
    - 3.5.1 Keamanan Sistem
    - 3.5.2 Pemeliharaan
  - 3.6 Kebutuhan Lain
    - 3.6.1 Database
    - 3.6.2 Pengoperasian
    - 3.6.3 Penyesuaian Tempat



1. PENDAHULUAN
  - 1.1 Tujuan
  - 1.2 Ruang Lingkup
  - 1.3 Definisi
  - 1.4 Referensi
  - 1.5 Sistematika
2. DESKRIPSI UMUM
  - 2.1 Perspektif
  - 2.2 Kegunaan
  - 2.3 Karakteristik Pengguna
  - 2.4 Batasan-batasan
  - 2.5 Asumsi dan Ketergantungan
3. SPESIFIKASI KEBUTUHAN
  - 3.1 Kebutuhan Fungsional
    - 3.1.1 Pendahuluan
    - 3.1.2 Input
    - 3.1.3 Proses
    - 3.1.4 Output
  - 3.2 Kebutuhan Antarmuka Eksternal
    - 3.2.1 Antarmuka Pengguna
    - 3.2.2 Antarmuka Perangkat Keras
    - 3.2.3 Antarmuka Perangkat Lunak
    - 3.2.4 Antarmuka Komunikasi
  - 3.3 Kebutuhan Performansi
  - 3.4 Kendala Desain
    - 3.4.1 Standard Compliance
    - 3.4.2 Perangkat Keras
  - 3.5 Atribut
    - 3.5.1 Keamanan Sistem
    - 3.5.2 Pemeliharaan
  - 3.6 Kebutuhan Lain
    - 3.6.1 Database
    - 3.6.2 Pengoperasian
    - 3.6.3 Penyesuaian Tempat