

# ANALISIS KEBUTUHAN PERANGKAT LUNAK BAG – 1

# PENDAHULUAN

---

- Analisis kebutuhan perangkat lunak (*software requirements analysis*) merupakan *aktivitas awal* dari *siklus hidup pengembangan perangkat lunak*.
- Untuk proyek-proyek perangkat lunak yang besar, analisis kebutuhan dilaksanakan setelah aktivitas *sistem information engineering* dan *software project planning*.

# TAHAPAN ANALISIS

---

Tahap Analisis adalah :

“Tahapan *pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun*”

# TAHAPAN ANALISIS

---

**Tahap analisis** meliputi :

1. Spesifikasi kebutuhan perangkat lunak,
2. Fungsi perangkat lunak yang dibutuhkan,
3. Performansi (unjuk kerja) sistem perangkat lunak,
4. Penjadwalan proyek,
5. Identifikasi sumber daya (manusia ,
6. Perangkat keras dan perangkat lunak yang dibutuhkan) dan
7. Taksiran biaya pengembangan perangkat lunak.

# KEGUNAAN ANALISIS

---

Kegunaan analisis adalah :

*“Untuk memodelkan permasalahan dunia nyata agar dapat dimengerti.”*

Tujuan aktivitas ini adalah :

*“Untuk mengetahui ruang lingkup produk (product space) dan pemakai yang akan menggunakannya.”*

Cat : Analisis yang baik akan mengungkapkan hal-hal yang penting dari permasalahan, dan mengabaikan yang tidak penting.

# PRINSIP ANALISIS

---

1. Menggambarkan *domain informasi* masalah
2. Mendefinisikan *fungsi perangkat lunak*
3. Menghasilkan *model yang menggambarkan informasi, fungsi dan kelakuan* yang dibagi secara rinci pada sebuah *model lapisan (hirarki)*
4. *Informasi pokok* pada tahap analisis memudahkan tahap implementasi yang lebih rinci.

# TUJUAN TAHAP ANALISIS

---

1. Menjabarkan *kebutuhan pemakai*
2. Meletakkan *dasar-dasar untuk tahap perancangan perangkat lunak*
3. Mendefinisikan *semua kebutuhan pemakai sesuai dengan lingkup kontrak yang disepakati kedua belah pihak* (pengembang dan pengguna).

# KEBUTUHAN (REQUIREMENT)

---

Menurut arti kamus, Kebutuhan (Requirement) adalah :

*“Sesuatu yang diminta”* atau *“Sesuatu yang dibutuhkan”*.

# KEBUTUHAN (REQUIREMENT)

---

Sedangkan menurut IEEE (*The Institute of Electrical and Electronics Engineers*) **kebutuhan** adalah :

1. *Kondisi* atau *kemampuan* yang *diperlukan pemakai* untuk *menyelesaikan suatu persoalan*, atau untuk *mencapai sebuah objek*.
2. *Kondisi* atau *kemampuan* yang *harus dipenuhi oleh sistem*, dalam arti *memenuhi kontrak, standar, spesifikasi atau dokumen formal lain yang diinginkan*.

# TAHAPAN KEBUTUHAN (REQUIREMENT)

---

1. *Dikenalinya* adanya sebuah *permasalahan* yang membutuhkan sebuah *penyelesaian*.
2. *Identifikasi* sebuah *permasalahan* mungkin dapat dilakukan dengan *berorientasi pada aplikasi, berorientasi pada bisnis, atau berorientasi pada kenaikan produktivitas (product improvement oriented)*.
3. *Munculnya ide* untuk *membuat sebuah perangkat lunak baru* (sebagai sebuah kemajuan).

# JENIS KEBUTUHAN (REQUIREMENT)

---

1. Behavioral
2. Non-behavioral

# 1. JENIS KEBUTUHAN - BEHAVIORAL

---

- Apa yang dilakukan oleh sistem (input dan output dari dan ke sistem).
- Hubungan informasi antara input dan output sehingga menghasilkan sebuah fungsi transformasi.

## 2. JENIS KEBUTUHAN – NON BEHAVIORAL

---

Mendefinisikan atribut sistem yang terkait untuk membentuk pekerjaan tersebut, termasuk deskripsi lengkap tentang :

- Efisiensi,
- Keamanan (*security*),
- *Rehability maintainability* (bagaimana perawatan untuk sistem), dan
- *Portability* (bisa dipindahkan dari satu perangkat keras ke perangkat keras lainnya).

# TAHAP ANALISIS KEBUTUHAN PL (1)

---

1. *Menentukan kebutuhan (requirement)* Lebih banyak berhubungan dengan pemakai. Hasil belum terstruktur.
  - Data atau informasi apa yang akan diproses
  - Fungsi apa yang diinginkan
  - Kelakuan sistem apa yang diharapkan
  - Antarmuka apa yang tersedia (*user interfaces, hardware interfaces, software interface, dan communications interfaces*)

# TAHAP ANALISIS KEBUTUHAN PL (2)

---

## 2. *Sintesis*

Mengubah kebutuhan yang belum terstruktur menjadi model atau gambar dengan memanfaatkan teknik dan metode analisis tertentu.

## 3. *Membuat dokumen Software Requirements Spesification (SRS).*

Sudah merupakan analisis yang lebih rinci, sebagai tahap awal perancangan.

# METODA ANALISIS KEBUTUHAN PL

---

1. *Berorientasi Aliran Data (Data Flow Oriented atau Functional Oriented)*
2. *Berorientasi Struktur Data*
3. *Berorientasi objek*

# 1. BERORIENTASI ALIRAN DATA (1)

---

- *Sudut pandang analisis* pada pendekatan ini difokuskan pada aspek *functional* dan *behavioral* (perilaku laku) sistem.
- *Pengembang* harus mengetahui :
  1. *Fungsi-fungsi* atau *proses-proses* apa saja yang ada dalam sistem,
  2. *Data* apa yang menjadi *masukannya*,
  3. *Dimana* data tersebut disimpan,
  4. *Transformasi* apa yang akan dilakukan terhadap data tersebut, dan
  5. Apa yang menjadi *hasil transformasinya*.

# 1. BERORIENTASI ALIRAN DATA (2)

---

Selain itu pengembang harus mengetahui :

1. Keadaan (*state*),
2. Perubahan (*transition*),
3. Kondisi (*condition*), dan
4. Aksi (*action*) dari sistem.

# 1. BERORIENTASI ALIRAN DATA (3)

---

Salah satu metode yang paling populer untuk pendekatan ini adalah *Analisis Terstruktur (Structured Analysis)* yang dikembangkan oleh *Tom DeMarco, Chris Gane, Trish Sarson*, dan *Edward Yourdon*.

# 1. BERORIENTASI ALIRAN DATA (4)

---

Pada metode ini, hasil analisis dan perancangan dimodelkan dengan menggunakan beberapa perangkat permodelan seperti :

- *Data Flow Diagram (DFD)* dan *Kamus Data (data dictionary)* untuk menggambarkan fungsi-fungsi dari sistem.
- *Entity-Relationship Diagram (ERD)* untuk menggambarkan data yang disimpan (*data storage*).
- *State Transition Diagram (STD)* untuk menggambarkan perilaku sistem.
- *Structure Chart* untuk menggambarkan struktur program

## 2. BERORIENTASI STRUKTUR DATA (1)

---

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

## 2. BERORIENTASI STRUKTUR DATA (2)

---

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 rancangannya.
- **Jackson Sistem Development (JSD)** : Dikembangkan oleh M.A. Jackson [1975] dengan menggunakan perangkat permodelan yang disebut *Structure Diagram* dan *Sistem Spesification Diagram*.

# 3. BERORIENTASI OBJEK (1)

---

- Berbeda dengan pendekatan-pendekatan sebelumnya, 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*.

# 3. BERORIENTASI OBJEK (2)

---

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 Modelling Technique (OMT)* dari James Rumbaugh [1987].
- *Object Oriented Software Engineering (OOSE)*

# LANGKAH LANGKAH ANALISIS SISTEM BERJALAN (1)

---

## 1. *Analisis Masalah*

“**Mengumpulkan** dan **memilah-milah** masalah-masalah yang merupakan **inti** dari **ide** pembangunan perangkat lunak.”

## 2. *Analisis Prosedur Manual*

“**Menuliskan skenario** tentang **prosedur-prosedur yang berlaku**. Manual belum tentu prosedur yang tidak menggunakan **komputer** sebagai alat bantu.”

# LANGKAH LANGKAH ANALISIS SISTEM BERJALAN (2)

---

## Contoh Analisis Prosedur Manual : **Prosedur Penjualan Barang**

1. Pembeli memilih barang yang ada di counter kemudian menyerahkannya ke kasir.
2. Kasir mencatat data penjualan di mesin kasir. Mesin kasir akan menampilkan informasi jumlah pembayaran kepada kasir.
3. Kasir memberitahukan jumlah pembayaran kepada pembeli untuk kemudian pembeli membayar sesuai dengan jumlah pembayaran.
4. Kasir mencatat data pembayaran di mesin kasir. Mesin kasir akan mencetak nota penjualan barang yang berisi informasi penjualan barang. **[Dan seterusnya...]**

# LANGKAH LANGKAH ANALISIS SISTEM BERJALAN (3)

---

## 3. *Analisis Aliran Dokumen Manual*

“*Pencatatan dan pemodelan alur dokumen-dokumen manual yang digunakan pada suatu prosedur manual. Alat bantu yang bisa digunakan adalah flowmap*”

## 4. *Analisis Aturan Bisnis*

“*Identifikasi dan pencatatan terhadap aturan-aturan baik tertulis ataupun lisan yang berlaku di lingkungan sistem dan memberikan pengaruh terhadap pembangunan sistem.*”

# LANGKAH LANGKAH ANALISIS SISTEM BERJALAN (4)

---

## Contoh Analisis Bisnis Rule : Aturan Bisnis Penjualan Barang

1. Diskon 10% akan diberikan apabila merupakan anggota/member.
2. Diskon khusus akan diberikan pada barang-barang yang berlabel sale.
3. Pembayaran bisa dilakukan secara tunai atau menggunakan kartu Debit dan kartu kredit.

# TOOLS ANALISIS SISTEM BERJALAN

---

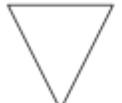
- *Flow map* dapat dijadikan **Tools** untuk melakukan *analisis sistem yang sedang berjalan* secara *manual*
- Merupakan diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi
- Merupakan *diagram alir* yang menunjukkan *arus dari dokumen, aliran data fisis, entitas-entitas* sistem informasi dan kegiatan operasi yang berhubungan dengan *sistem informasi*.

# FLOW MAP : RULES OF THUMB

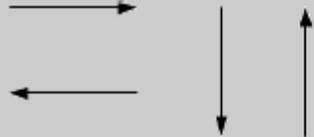
---

1. Memodelkan aliran dokumen pada sistem yang sedang berjalan.
2. Bentuk dokumen bisa manual atau berupa file komputer.
3. Satu alur aliran dokumen terdiri dari input -> proses -> output.
4. Apabila ada kondisi yang dikenakan alur pada poin 3 tetap diperhatikan.
5. Tidak boleh ada dokumen yang hilang dalam runtunan prosesnya.

# FLOW MAP : SYMBOL

SIMBOL	NAMA SIMBOL	FUNGSI
	Dokumen Manual	Menunjukkan dokumen sebagai masukan dan keluaran dalam proses manual
	Proses Manual	Menunjukkan proses yang dilakukan tanpa bantuan komputer
	Kondisi	Menunjukkan ada suatu kondisi yang harus diperiksa untuk melihat hasil keluaran
	Arsip	Menggambarkan kumpulan dokumen-dokumen sejenis yang disimpan

# FLOW MAP : SYMBOL

SIMBOL	NAMA SIMBOL	FUNGSI
	Aliran Dokumen	Menunjukkan aliran dokumen
	Input Data Manual	Menunjukkan data untuk membentuk dokumen komputerisasi
	Proses terkomputerisasi	Menggambarkan proses yang dilakukan dengan bantuan komputer
	File/Database	Menggambarkan penyimpanan jika menggunakan proses terkomputerisasi

# MEMBUAT FLOW MAP

---

1. Bagi diagram ke dalam *kolom-kolom*
2. Setiap *kolom* diberi nama *entitas* yang terlibat (orang, bag./ unit organisasi, perusahaan lain, atau pimpinan)
3. Diagram harus dibaca dari *atas ke bawah* dan dari *kiri ke kanan*
4. Setiap kolom terdapat *siklus pengolahan data : I-P-O (input-proses-output)* bila kolom tersebut adalah entitas yg melakukan kegiatan
5. Ketika *menyebrangi garis* yang memisahkan antara satu kolom dg kolom lain, gunakan *simbol konektor*.
6. Cara mengakses *file komputer* adalah melalui *simbol proses komputer*
7. Prosedur kerja yang *kejadiannya tidak bersamaan* dapat digambarkan melalui *flowmap yang terpisah*.

# CONTOH PEMBUATAN FLOW MAP (1)

---

## KASUS :

Sebuah lembaga pendidikan mempunyai prosedur penilaian sbb:

1. Pengajar meyerahkan nilai kursus ke bag. Akademik untuk dicatat / diinput ke komputer setelah seluruh nilai terkumpul.
2. Kumpulan nilai tadi disimpan ke file nilai dengan mengakses file siswa.
3. Berdasarkan file nilai, petugas menghitung nilai akhir dan membuat transkrip nilai untuk diserahkan ke siswa.

# CONTOH PEMBUATAN FLOW MAP (2)

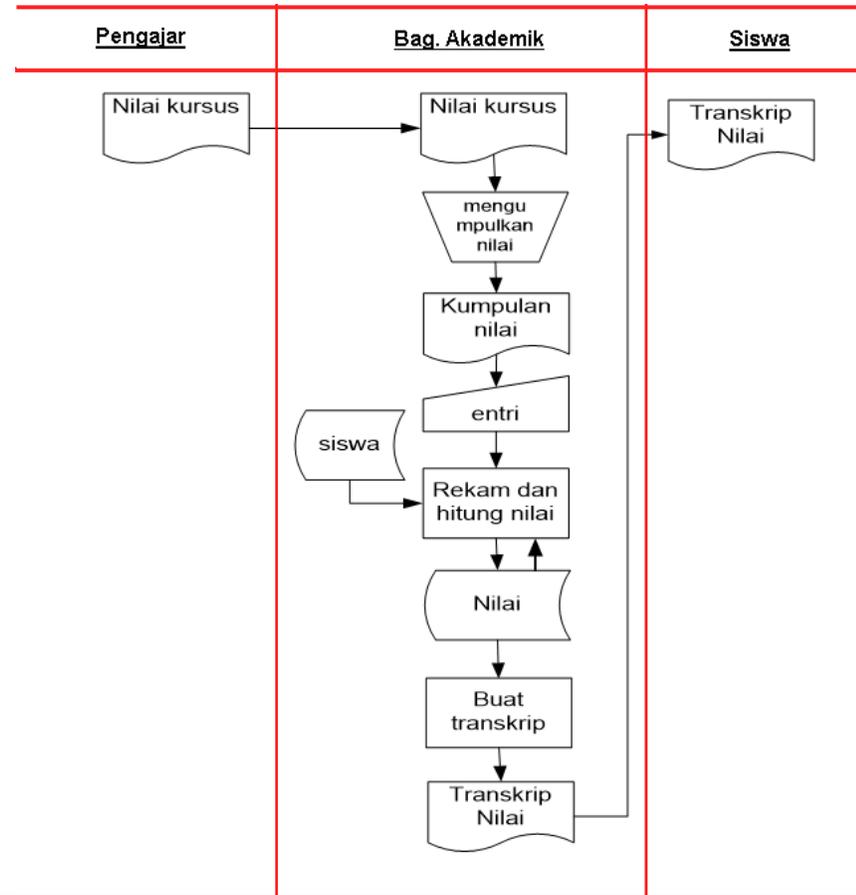
---

## PENYELESAIAN :

- Daftar entitas :
  - Pengajar
  - Bag. Akademik
  - Siswa
- Dokumen :
  - Nilai kursus
  - Kumpulan nilai
  - Transkrip nilai
- File :
  - siswa
  - nilai

# CONTOH PEMBUATAN FLOW MAP (3)

FLOW MAP :



# TABEL ENTITAS DAN KEGIATAN

---

- Digunakan untuk *memperjelas* informasi yang ada dalam uraian prosedur serta dapat membantu mempermudah penyelesaian *diagram prosedur / flowmap* secara benar.
- Cara membuat :  
Tuliskan masing-masing entitas di bagian kiri tabel (kolom ke-1) dan kegiatan setiap entitas di bagian kanan tabel (kolom ke-2).

# TABEL ENTITAS DAN KEGIATAN

---

## Tabel Entitas dan Kegiatan Alternatif 1

No.	Entitas	Kegiatan
1	Pengajar	1. Menyerahkan Nilai kursus
2	Bag.Akademik	2. Mengumpulkan nilai-nilai 3. Memasukkan data 4. Menghitung dan merekam nilai 5. Mencetak transkrip nilai
3	Siswa	6. Menerima Transkrip nilai

# TABEL ENTITAS DAN KEGIATAN

## Tabel Entitas dan Kegiatan Alternatif 2

Entitas	Pengajar	Bag. Akademik	Siswa
Kegiatan	1. Menyerahkan Nilai kursus	2. Mengumpulkan nilai-nilai 3. Memasukkan data 4. Menghitung dan merekam nilai 5. Mencetak transkrip nilai	6. Menerima Transkrip nilai

# TUGAS / PR MANDIRI

---

Sistem Penjualan barang tunai secara langsung di minimarket :

1. Pembeli membawa barang-barang yang dibeli ke Bagian Kasir
2. Petugas Kasir menghitung jumlah barang yg dibeli dan membuat struk Penjualan sebagai bukti transaksi penjualan barang untuk pembeli setelah menyerahkan pembayaran.
3. Di akhir jam kerja, petugas Kasir menghitung jumlah uang yang diterima dari seluruh transaksi penjualan barang dan membuat Laporan Penjualan.
4. Salinan struk dan Laporan Penjualan diserahkan ke Admin Penjualan.
5. Admin penjualan memeriksa apakah jumlah uang yang diterima sesuai dengan Laporan Penjualan dan struk penjualannya.
6. Jika sudah sesuai, maka Admin penjualan akan memberi paraf dan mengarsipkan Laporan dan struk tersebut. Jika tidak sesuai, Admin penjualan akan mengoreksinya sebelum memberi paraf dan mengarsipkannya.
7. Laporan Penjualan diarsipkan oleh Admin Penjualan.
8. Buatlah **Flow map** dan **Tabel Entitas/kegiatannya** ...