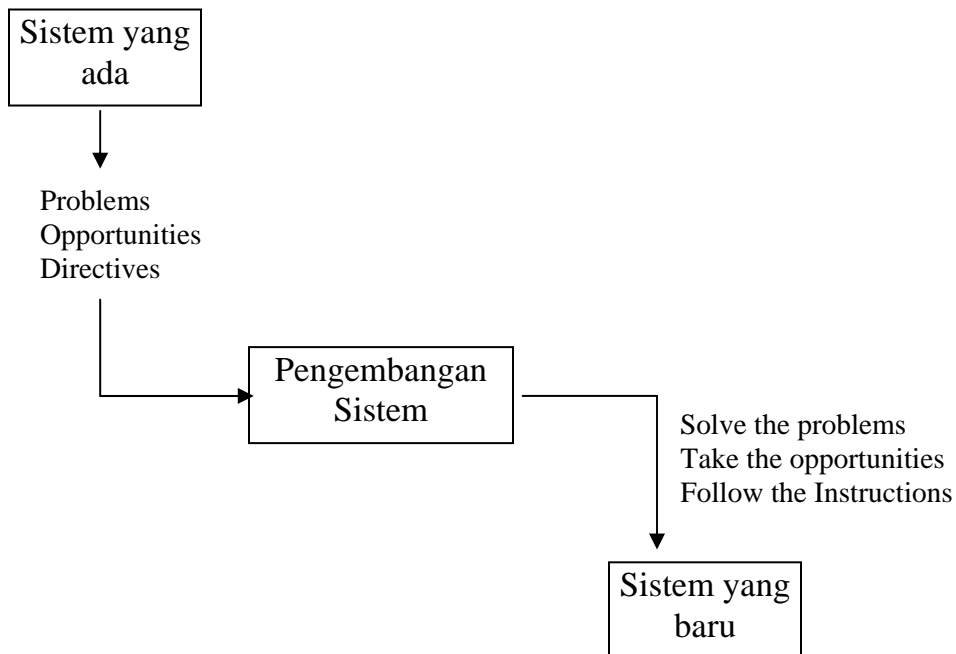


# TINJAUAN UMUM PENGEMBANGAN SISTEM

## Perlunya Pengembangan Sistem

- a. *Problems*
  - I. Ketidakberesan
  - II. Pertumbuhan Organisasi
- b. *Opportunities*
- c. *Directives*



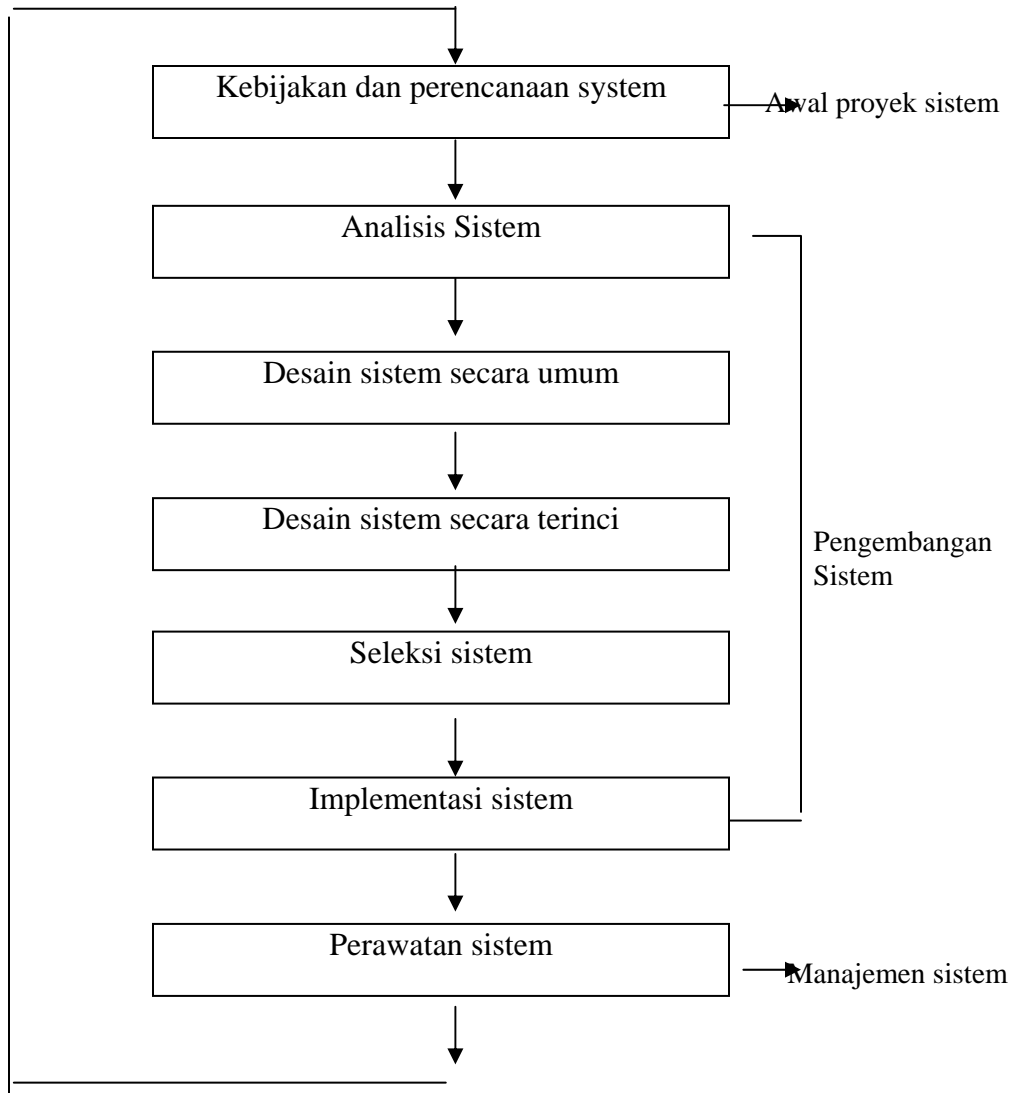
**P I E C E S**

## Prinsip Pengembangan Sistem

- a. Manajemen
- b. Investasi modal besar
- c. Orang yang terdidik
- d. Tidak harus urut
- e. Jangan takut membatalkan proyek
- f. Dokumentasi

## System Development Life Cycle (SDLC)

: Bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah di dalam tahapan tersebut dalam proses pengembangannya.


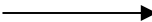

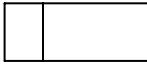


## Data Flow Diagram (DFD)

Diagram yang menggambarkan suatu system yang telah ada atau system baru yang akan dikembangkan , secara logika tanpa mempertimbangkan lingkungan fisik di mana data tersebut mengalir (misalnya lewat telepon, surat dsb) atau lingkungan fisik di mana data tersebut akan disimpan (misalnya file kartu, hard disk, tape, disket dsb).

→ alat pada *structured analysis and design*

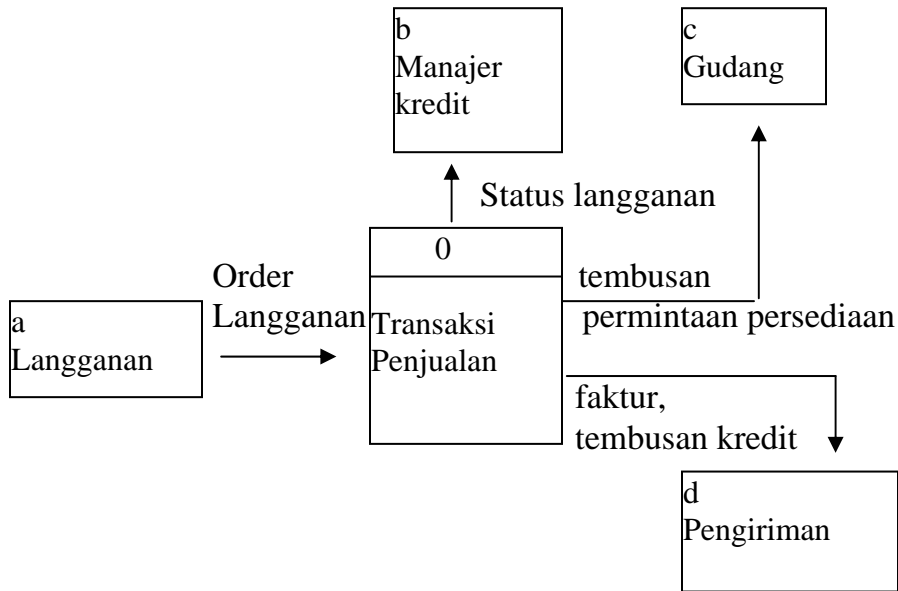
### Komponen dan Simbol :

No.	Komponen	Simbol
1.	<i>External entity</i> (kesatuan luar)	
2.	<i>Data flow</i> (arus data)	
3.	<i>Process</i> (proses)	
4.	<i>Data Store</i> (simpanan data)	

### Tingkatan (Levelisasi) dalam DFD :

1. *Context Diagram* (diagram konteks)
2. *Overview Diagram* (diagram level 0)
3. *Primitive Diagram* (diagram primitif)

Contoh : *context diagram* untuk sistem penjualan




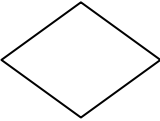

**Perbedaan DFD dengan Flowchart :**

No.	DFD	Flowchart
1.	Paralel	Terurut
2.	Arus data	Arus prosedur atau arus algoritma
3.	Tidak menunjukkan proses perulangan ( <i>loop</i> ) dan keputusan ( <i>decission</i> )	menunjukkan proses perulangan ( <i>loop</i> ) dan keputusan ( <i>decission</i> )

## Entity Relationship Diagram (ERD)

: Diagram yang menghubungkan antar entitas di dalam penyusunan/perancangan basis data.

### Komponen dan Simbol

No.	Komponen	Simbol
1.	<i>Entity</i> (entitas)	
2.	<i>Relationship</i> (keterhubungan)	
3.	<i>Atribut</i>	

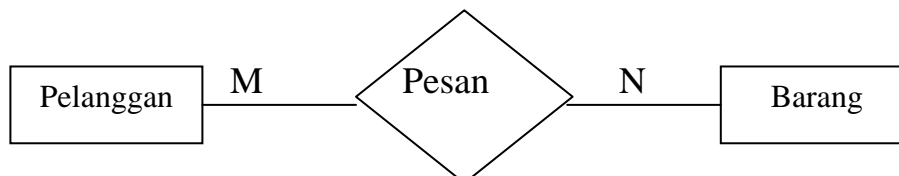
### Ratio Cardinality :

1. One to one : 1 : 1
2. One to Many : 1 : M
3. Many to One : M : 1
4. Many to many : M : N

### Relationship Degree :

1. Unary relationship
2. Binary relationship
3. Ternary relationship

**Contoh** : ERD untuk database penjualan



## ***Data Dictionary (Kamus Data)***

: Katalog fakta tentang data dan kebutuhan – kebutuhan informasi dari suatu sistem informasi. → dapat mendefinisikan data yang mengalir dengan lengkap.

### **Isi Kamus Data :**

1. Nama arus data
2. Alias
3. Bentuk data
4. Arus data
5. Penjelasan
6. Periode
7. Volume
8. Struktur data

**Contoh** : kamus data untuk arus data “tembusan permintaan persediaan”  
(dibuat dari suatu DFD)

### **Kamus Data**

Nama arus data	: Tembusan permintaan persediaan
Alias	: Faktur : Tembusan jurnal : Tembusan kredit
Bentuk data	: Dokumen cetakan komputer
Arus Data	: Proses 1.4P - Gudang Proses 1.4P - Bagian pengiriman Bagian pengiriman - Proses 3
Penjelasan	: Tembusan dari faktur penjualan untuk meminta barang dari gudang
Periode	: Setiap kali terjadi penjualan (harian)
Volume	: Volume rata-rata tiap hari adalah 100 & Volume puncak adalah 150
Struktur data	: Tembusan permintaan persediaan terdiri dari item data : KODE LANGGANAN NAMA LANGGANAN TANGGAL PENJUALAN NOMOR FAKTUR Satu sampai maksimum 10 kali : KODE BARANG NAMA BARANG UNIT JUAL HARGA SATUAN TOTAL HARGA TOTAL PENJUALAN POTONGAN PENJUALAN PAJAK PENJUALAN TOTAL DIBAYAR JENIS PENJUALAN

## Structured English dan Pseudocode

*Structured English* (SE) berbasis pada bahasa Inggris. → *Structured Indonesia* (SI) : berbasis pada bahasa Indonesia.

*Pseudocode* berbasis pada bahasa pemrograman yang sesungguhnya, seperti COBOL, FORTRAN atau Pascal, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada *programmer*.

### Struktur dari pemrograman terstruktur

1. Struktur urut
2. Struktur keputusan
3. Struktur iterasi

#### Contoh struktur urut

- untuk SI :

Baca data Jam-kerja  
Hitung Gaji adalah Jam-kerja dikalikan Tarip  
Tampilkan Gaji

- untuk pseudocode :

Baca data Jam-kerja dari keyboard	Read Jam-kerja
Hitung Gaji = Jam-kerja * Tarip	Let Gaji = Jam-kerja * Tarip
Tampilkan Gaji di monitor	Print Gaji

#### Contoh struktur keputusan :

##### Struktur If-Then

- untuk SI : If Nilai lebih besar dari 60 Then tambah 1 ke Lulus

- untuk pseudocode :

If Nilai > 60 Then  
    Lulus = Lulus + 1  
Endif

#### Contoh struktur Iterasi :

##### Struktur For

- untuk SI :



Baca Jumlah-barang  
Untuk masing-masing barang  
    Baca data Nilai-penjualan (Unit dan Harga)  
    Hitung Nilai-barang yaitu Unit dikalikan Harga  
    Tampilkan dalam satu baris hasilnya  
    Hitung Jumlah  
Tampilkan nilai Jumlah

- untuk pseudocode :

Jumlah = 0	10 JUMLAH = 0
Baca Jumlah-barang dari monitor	20 INPUT JUMLAHBARANG
Do I = 1 to Jumlah-barang	30 FOR I = 1 TO
Read data Unit dan Harga dari	JUMLAHBARANG
monitor	40 INPUT UNIT,HARGA
Nilai-barang = Unit * Harga	50 NILAIBARANG=UNIT*HA
Tampilkan baris laporan di	RGA
printer	60 LPRINT UNIT, HARGA,
Jumlah = jumlah + Nilai-barang	NILAIBARANG
Enddo	70 JUMLAH = JUMLAH +
Tampilkan Jumlah di printer	NILAIBARANG
	80 NEXT I
	90 LPRINT "JUMLAH
	PENJUALAN ="; JUMLAH

### ***Decision Tabel (Tabel Keputusan)***

Tabel yang digunakan sebagai alat bantu untuk menyelesaikan logika di dalam program.

**Struktur dari tabel keputusan adalah sbb :**

	Rules					
	1	2	3	4	...	N
Condition Stub					Condition Entry	
Action Stub					Action Entry	

**Latihan : Buatlah tabel keputusan untuk kasus berikut ini :**

Jika unit yang dipesan untuk suatu barang tertentu sama atau melebihi jumlah minimum untuk mendapatkan potongan dan pemesannya adalah dealer, maka akan mendapatkan potongan, sebaliknya bila unit yang dipesan kurang dari unit minimum untuk mendapatkan potongan, walaupun yang membeli dealer, maka tidak akan mendapat potongan. Bila unit barang persediaan di gudang tidak mencukupi, maka yang dikirim adalah unit yang ada dan kemudian dibuatkan catatan kekurangannya (*Backorder*). Kalau unit persediaan mencukupi maka semua pesanan dikirim.

**Jawaban :**

	Rules							
	1	2	3	4	5	6	7	8
a. Apakah unit dipesan $\geq$ unit min. potongan?	Y	Y	Y	Y	T	T	T	T
b. Apakah pemesannya adalah dealer?	Y	Y	T	T	Y	Y	T	T
c. Apakah unit persediaan mencukupi?	Y	T	Y	T	T	T	Y	T
a. Mendapatkan potongan	X	X	-	-	-	-	-	-
b. Tidak mendapatkan potongan	-	-	X	X	X	X	X	X
c. Kirim semua yang dipesan	X	-	X	-	X	-	X	-
d. Kirim yang ada saja	-	X	-	X	-	X	-	X
e. Buat catatan kekurangannya	-	X	-	X	-	X	-	X

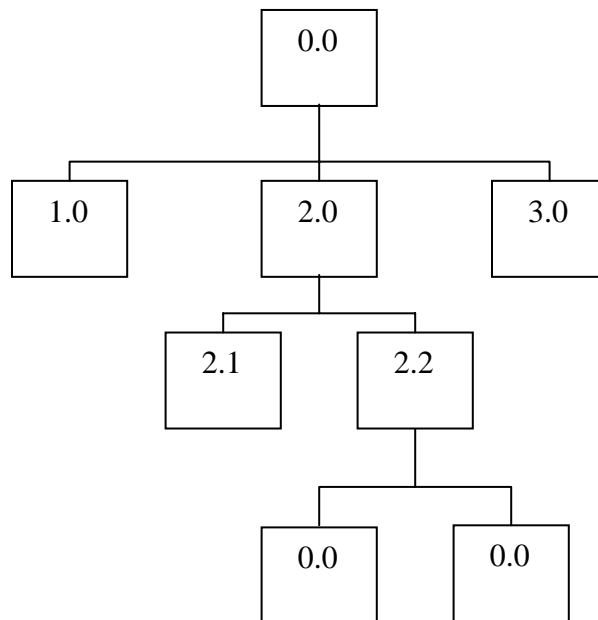
## Hierarchy plus Input-Process-Output (HIPO)

Alat dokumentasi program → alat desain dan teknik dokumentasi dalam SDLC. HIPO berbasis pada fungsi.

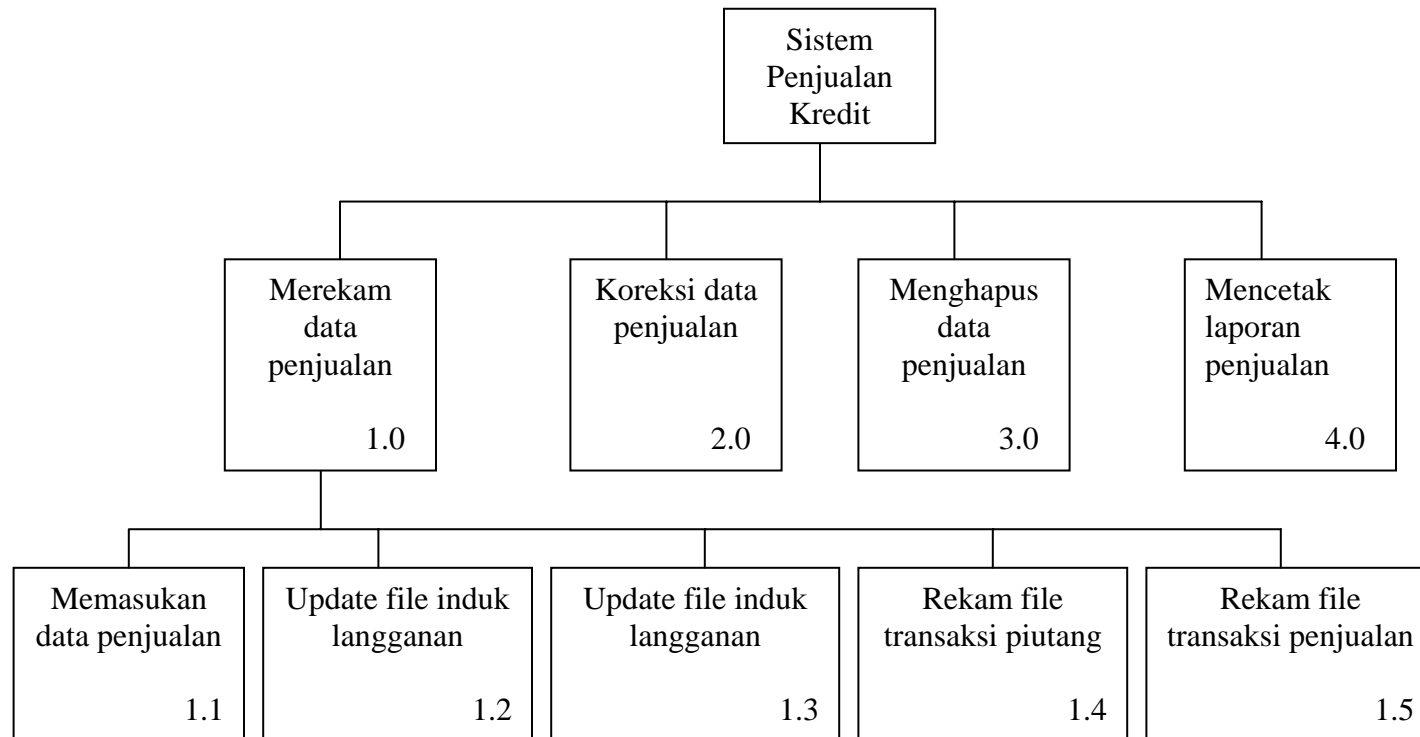
**Diagram dalam paket HIPO :**

### Visual Table of Content (VTOC)

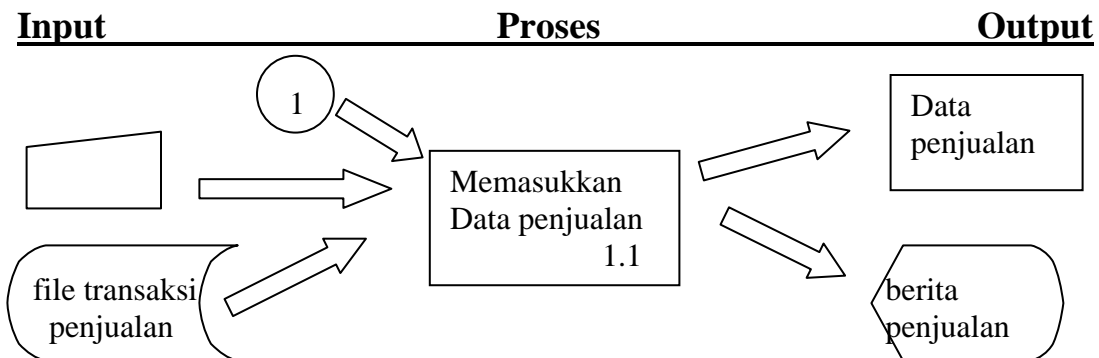
Diagram yang menggambarkan hubungan dari fungsi-fungsi di sistem secara berjenjang.



**Contoh VTOC suatu sistem penjualan kredit :**



## Overview Diagrams



## RANCANGAN SISTEM GENERAL

Berhubungan dengan kategori sistem :

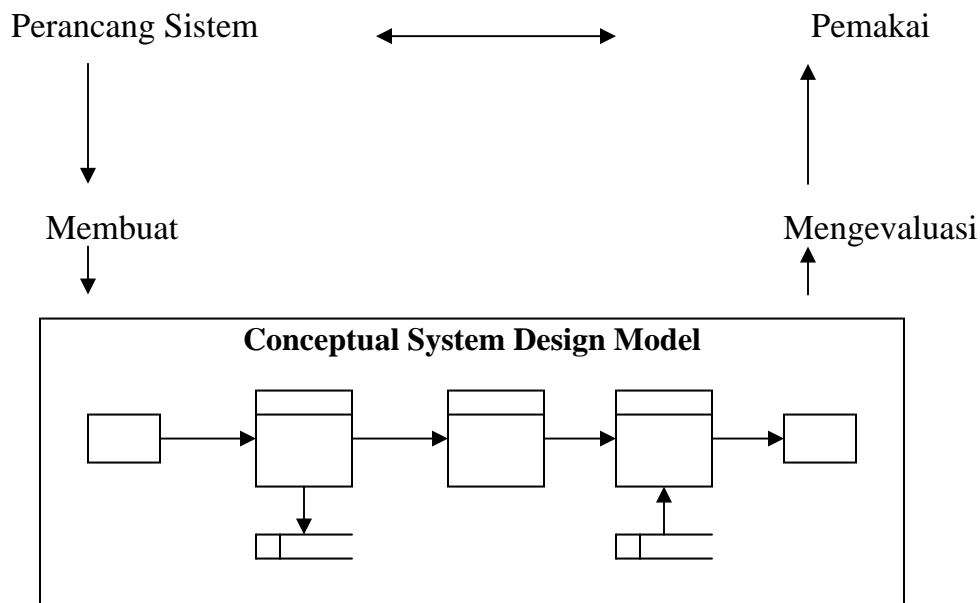
- a. Sistem dengan dasar global
- b. Memperhatikan seluruh organisasi
- c. Besar dan Kompleks dengan volume tinggi dari transaksi
- d. Diproses dengan batch dan real time
- e. Pondasi system informasi
  
- f. Sistem dengan dasar group  
Dibangun untuk unit,cabang,departemen, atau bagian dalam sebuah organisasi  
  
Contoh : - Group perencana keuangan sebuah perusahaan  
          - Bagian penagihan sebuah rumah sakit
  
- g. Sistem dengan dasar lokal
- h. Dirancang untuk beberapa orang
- i. Dapat berdiri sendiri untuk seorang pemakai atau dihubungkan dengan beberapa bagian dari sebuah system dengan dasar group atau global
- j. Contoh : EIS

## Rapid Application Development (RAD)

Untuk Menciptakan rancangan-desain sistem yang dapat bekerja untuk menyesuaikan kebutuhan pemakai dengan tepat

### Empat (4) elemen RAD :

- a. JAD
  - a. Efektif terutama bila sistem dengan dasar global sedang dalam pengembangan
  - b. Dapat digunakan pula untuk group dan lokal
  - c. Kata kunci Gabungan.



- a. Model rancangan di benak perancang
- b. Model rancangan di benak pemakai
- c. Model rancangan dideskripsikan dengan sebuah model seperti : DFD, ERD.

## 2. Tim SWAT (Specialist With Advanced Tools)

Terdiri dari  $\frac{3}{4}$  professional sistem dilengkapi perangkat CASE

### 3. Perangkat CASE (*Computer Aided Systems and Software Engineering*)

Digunakan tim SWAT untuk meningkatkan produktivitas pengembangan system dan kualitas kerja untuk :

- a. Menambah kedisiplinan pada pengembangan system
- b. Mengurangi kesalahan dan kehilangan pada rancangan
- c. Mengurangi pengerjaan ulang system

### 4. Prototyping

- d. Dengan JAD : pemakai diperlihatkan apa yang mereka dapat, memberi kesempatan mereka untuk bereaksi.
- e. CASE : fasilitas untuk prototyping
- f. TIM SWAT : dapat menciptakan rancangan pada layar, model yang bervariasi dan berdialog serta berinteraksi dengan pemakai
- g. Peraturan 80:20

**Dua Pendekatan luas untuk rancangan system :**

#### **I. Berorientasi Struktur**

Didasarkan pada metodologi perangkat model dan teknik  
Dibagi dua (2) :

##### **Berorientasi Proses**

1. Dengan memeriksa input, output dan proses sehingga dapat ditentukan kebutuhan pemakai
2. Melihat semua laporan, tampilan di layar, perhitungan, keputusan yang diperlukan untuk sesuatu proses
3. DFD
4. Efektif untuk aplikasi berdasarkan transaksi → I,P,O sangat stabil

##### **Berorientasi Data**

1. Tidak tahu dengan pasti apa yang akan dihasilkan oleh system
2. Output berubah setiap beberapa bulan
3. Bekerja sangat dekat dengan pemakai dari system
4. Saat proses dari sebuah system, seperti juga Input dan Output relatif tidak terdefinisi.

5. Fokus : untuk menentukan kebutuhan data → untuk menentukan keputusan yang didasarkan pada data tersebut.

Kedua Orientasi tersebut adalah untuk mengidentifikasi semua atribut yang diperlukan oleh sistem yang sedang dikembangkan.

## Membuat kamus data untuk sebuah sistem

Menjelaskan setiap atribut dan sub atribut untuk sebuah system dan mendefinisikan karakteristik seperti : ukuran, jenis, deskripsi, akses khusus.

Contoh : **Sistem pembelian** :

- a. Kebutuhan yang dapat didefinisikan
  - I. Melacak order pembelian dari tanggal beli
  - II. Melacak alamat pembeli, order pembelian (nama barang, jumlah)
  
- b. Kebutuhan yang tidak dapat didefinisikan :

Komponen pendukung keputusan yang baru untuk membantu pembeli dalam memilih pemasok/supplier.

→ kualitas, kinerja masa lalu, ketersediaan produk

## II. Berorientasi Obyek

**Tujuan** : Merancang dan membangun system melalui perakitan obyek perangkat lunak yang digunakan , daripada menulis *Programming Language* dari awal.

**Elemen Utama** :

### 1. Obyek.

- a. Semua yang dihadapi dalam lingkungan
- b. Mempunyai tingkah laku tertentu
- c. OOP
- d. Desain system dipusatkan pada identifikasi obyek lebih dari menspesifikasikan atribut dan kode program untuk memanipulasinya.

### 2. Classes

- a. Seperangkat obyek yang terbagi atas struktur biasa dan tingkah laku.
- b. Sebuah jenis ; sebuah obyek tunggal.
- c. Contoh : Class pelanggan dapat dimasukan 2 contoh : obyek pelanggan Agen dan Obyek Pelanggan Eceran



## **Empat langkah untuk melakukan Desain berorientasi obyek :**

- h. Mengidentifikasi kelas obyek
- i. Mengidentifikasi hubungan antara kelas obyek
- j. Mengidentifikasi atribut utama, bukan spesifikasi detail
- k. Menentukan hubungan penerimaan dan membangun hirarki kelas

## **Perbedaan antara 2 orientasi rancangan**

**I. Modularity** → derajat dari standarisasi dan kebebasan modul, serta variasi dalam penggunaan.

### **1. Orientasi struktur**

Modul adalah sebuah unit dari kode P.L yang melakukan sebuah fungsi yang didefinisikan dengan baik

### **2. Orientasi Obyek**

Modul : sebuah obyek yang menggabungkan atribut dan kode program untuk berlaku dalam suatu cara tertentu

## **II. Desain Top-Bottom dan Bottom-Up**

a. Top-Bottom → orientasi struktur

Memulai dengan sebuah deskripsi abstrak → kemudian menyaring pada langkah berikutnya.

b. Bottom-Up → orientasi obyek

## **ANALISIS SISTEM**

: Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan – kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Langkah – langkah Dasar :

- a. Identify*
- b. Understand*
- c. Analyze*
- d. Report*

### **I. Mengidentifikasi masalah :**

- I. Mengidentifikasi penyebab masalah
- II. Mengidentifikasi titik keputusan
- III. Mengidentifikasi personil-personil kunci

### **II. Memahami Kerja dari sistem yang ada**

- Menentukan jenis penelitian
- Merencanakan jadwal penelitian
  - Mengatur jadwal wawancara

- Mengatur jadwal observasi
- Mengatur jadwal pengambilan sample
- Membuat penugasan penelitian
- Membuat agenda wawancara
- Mengumpulkan hasil penelitian

### **III. Menganalisis hasil penelitian**

#### **A. Menganalisis kelemahan sistem**

- Mengapa dikerjakan ?
- Perlukah dikerjakan ?
- Apakah telah dikerjakan dengan baik ?

→ *Relevance, capacity, efficiency, timeliness, accessibility, flexibility, accuracy, reliability, security, economy, simplicity.*

#### **Menganalisis distribusi pekerjaan**

- Apakah tugas dan tanggungjawab telah didefinisikan dan diterapkan dengan jelas
- Apakah tugas dan tanggungjawab didistribusikan dengan efektif untuk masing-masing personil dan unit-unit organisasi.

#### **Menganalisis pengukuran pekerjaan**

- Apakah kebijaksanaan dan prosedur telah dipahami dan diikuti ?
- Apakah produktivitas karyawan memuaskan ?
- Apakah unit-unit organisasi telah bekerjasama dan terkoordinasi dengan baik menjaga arus data lancar ?
- Apakah masing-masing kegiatan telah mencapai sarannya ?
- Apakah terjadi operasi-operasi yang tumpang tindih ?
- Seberapa perlu hasil dari tiap-tiap operasi ?
- Apakah terdapat operasi yang menghambat arus data ?
- Apakah volume puncak dari data dapat ditangani dengan baik ?
- Apakah terdapat standar kinerja yang baik dan selalu dimutakhirkan ?

#### **Menganalisis Keandalan**

- Apakah jumlah kesalahan yang terjadi di masing-masing operasi diminimumkan ?
- Apakah operasi-operasi telah direncanakan dengan baik dan terkendali ?

## **Menganalisis Dokumen**

- Seberapa perlu dokumen-dokumen yang ada ?
- Apakah masing-masing dokumen telah dirancang untuk penggunaan yang efektif ?
- Apakah tembusan dari dokumen perlu ?

## **Menganalisis Laporan**

- Dapatkah laporan-laporan yang dipersiapkan dengan mudah dari file dan dokumen-dokumen yang ada ?
- Apakah terdapat duplikasi di file, catatan-catatan dan laporan-laporan ?

## **Menganalisis Teknologi**

Apakah fasilitas dari system informasi (dalam bentuk personil, peralatan dan fasilitas lainnya) cukup untuk menangani volume rata-rata data tanpa terjadi penundaan yang berarti ?

## **B. Menganalisis kebutuhan informasi pemakai/manajemen**

### **IV. Membuat Laporan Hasil Analisis**

Contoh Daftar Isi dari suatu laporan hasil analisis system untuk proyek pengendalian pemasaran dan penjualan :

- IV. Alasan melakukan analisis system
- V. Permasalahan-permasalahan
- VI. Identifikasi penyebab masalah
- VII. Identifikasi titik keputusan
- VIII. Penelitian yang dilakukan
- IX. Hasil analisis
- X. Kesimpulan analisis
- XI. Rekomendasi Umum

Lampiran-lampiran

