



PRINSIP DAN KONSEP DESAIN

Oleh: Rahmi Hidayati, S.Kom., M.Cs.

Tujuan Pembelajaran

- Mahasiswa mampu memahami desain data, arsitektur, interface dan procedural
- Mahasiswa mampu memahami prinsip dan konsep desain perangkat lunak

PENDAHULUAN

- Desain adalah langkah pertama dalam fase pengembangan bagi setiap produk atau sistem yang direkayasa.
- Desain dapat didefinisikan berbagai “proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya”.
- Tujuan *desainer* adalah untuk menghasilkan suatu model atau representasi dari entitas yang kemudian akan dibangun.

PROSES DESAIN

- *Software Design* : Suatu proses yang melewati serangkaian kebutuhan yang membentuk sebuah perangkat lunak.
- *Software design* dibagi dalam dua tahap yaitu :
 1. *Preliminary Design* : Pada tahap ini difokuskan dengan transformasi dari kebutuhan ke dalam data dan arsitektur *software*.
 2. *Detail Design* : Difokuskan pada penghalusan representasi arsitektur yang berisi struktur data detail dan algoritma untuk *software*.

DESAIN DAN KUALITAS PERANGKAT LUNAK

1. Memperlihatkan organisasi hirarki yang mengontrol elemen-elemen *software*.
2. Berkenaan dengan modul. *Software* secara logika terbagi dalam elemen-elemen yang membentuk fungsi dan sub fungsi.
3. Berisi representasi yang berbeda dan terpisah dari data dan prosedur.

DESAIN DAN KUALITAS PERANGKAT LUNAK

4. Membentuk modul (contoh : *procedure*) yang memperlihatkan karakteristik fungsi yang tidak saling bergantung.
5. Diturunkan dengan menggunakan metode perulangan yang didukung oleh informasi yang ada selama analisa kebutuhan *software*.

KONSEP-KONSEP DESAIN

- Apakah kriteria yang dapat dipakai untuk mempartisi *software* menjadi sejumlah komponen?
- Bagaimana fungsi atau struktur data dipisahkan dari suatu representasi konseptual *software*?
- Apakah ada kriteria yang seragam yang menetapkan kualitas teknik dari suatu *software* desain?

DESAIN PERANGKAT LUNAK

- Desain perangkat lunak berada pada inti teknik dari proses rekayasa perangkat lunak dan diaplikasikan tanpa memperhatikan model proses perangkat lunak yang digunakan.
- Begitu persyaratan perangkat lunak telah mulai dianalisis dan ditentukan, maka desain perangkat lunak menjadi yang pertama dari tiga aktivitas teknik yaitu : **desain, pembuatan kode** dan **pengujian** – yang diperlukan untuk membangun dan menguji perangkat lunak.

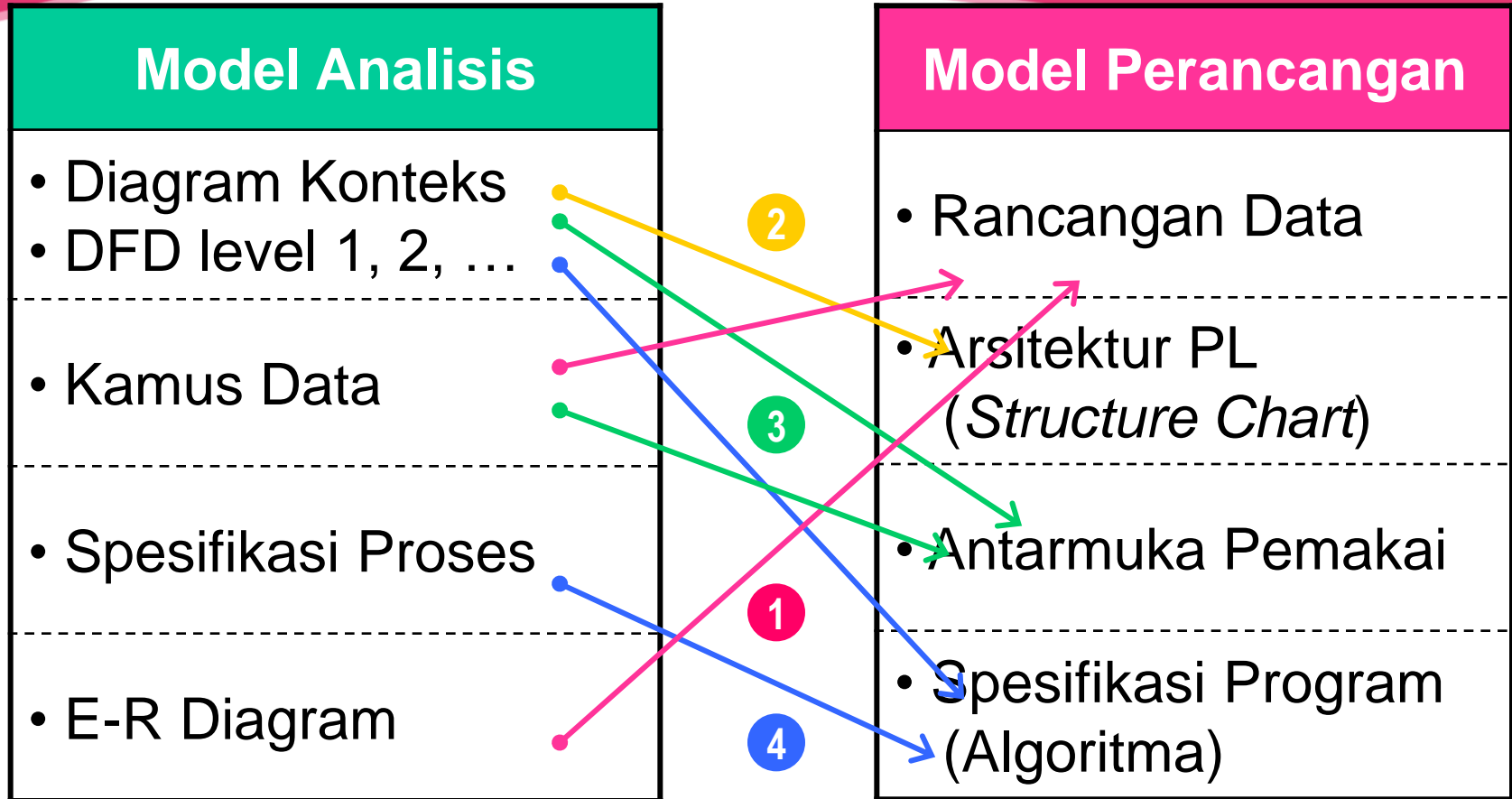
DESAIN PERANGKAT LUNAK

- Persyaratan perangkat lunak, yang dimanifestasi oleh data, fungsional dan model-model perilaku, mengisi langkah desain.
- Dengan menggunakan satu dari sejumlah metode desain, langkah desain menghasilkan :
 1. Desain data
 2. Desain arsitektur
 3. Desain *interface*
 4. Desain prosedural

DESAIN PERANGKAT LUNAK

- Desain Data
 - Struktur tabel basis data atau *file* data
 - Struktur data *internal*
- Desain Arsitektur
 - *Structure chart*
 - Struktur menu program
- Desain *Interface* (Antarmuka pemakai)
- Desain Prosedural (Spesifikasi program (algoritma))

DESAIN PERANGKAT LUNAK



PERANCANGAN BASIS DATA

- Transformasi **Diagram E-R** (*Conceptual Data Model, CDM*) menjadi model relasi (**skema relasi, tabel relasi**).
- Penentuan **atribut relasi** sesuai dengan **kamus data** yang telah dibuat.
- Normalisasi.
- Pendefinisian **struktur tabel**.
- Pembuatan **relasi antar tabel** (*Physical Data Model (PDM)*)

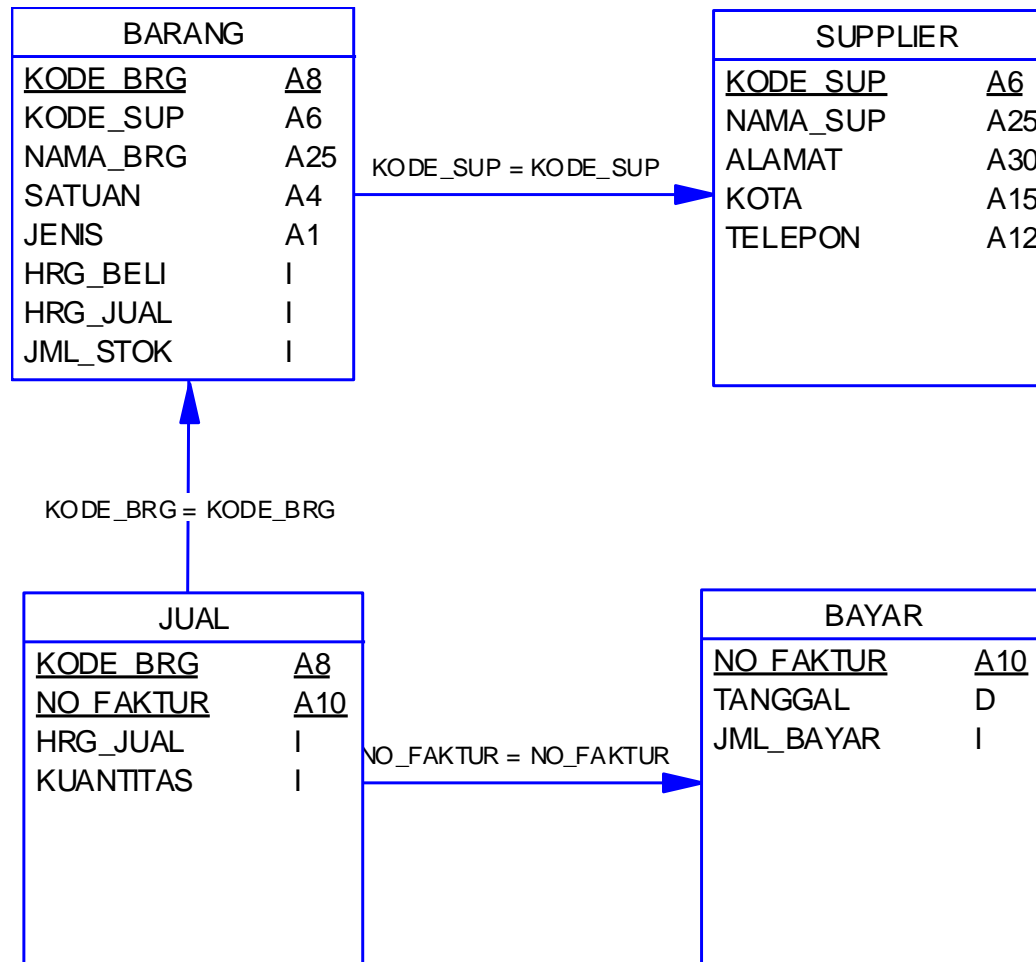
CONTOH STRUKTUR TABEL BASIS DATA

Tabel Penjualan

- Fungsi : Menyimpan data transaksi penjualan
- Jenis : Tabel Transaksi
- Primary Key : No_Faktur+Kode_Brg
- Foreign Key : Kode_Brg
- Struktur Tabel :

No.	Nama Field	Jenis	Lebar	Keterangan
1	No_Faktur	String	10	Nomor Faktur
2	Kode_Brg	String	8	Kode Barang
3	Hrg_Jual	Long Integer	8	Harga jual barang saat transaksi
4	Kuantitas	Integer	5	Banyaknya (kuantitas) barang

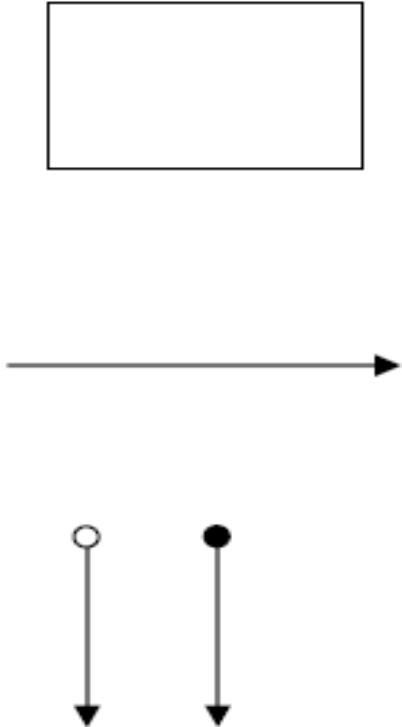
CONTOH RELASI ANTAR TABEL





ARSITEKTUR PERANGKAT LUNAK

- Gambaran bagaimana elemen atau komponen fungsional perangkat lunak disusun, diorganisasi dan di strukturkan sehingga:
 - **Hubungan** antar elemen atau komponen dapat dijelaskan.
 - **Interface** yang menghubungkan elemen atau komponen dapat didefinisikan.
 - **Wujud** dan **penempatan** elemen atau komponen dalam tempat penyimpanan sekunder **secara fisik** dapat ditetapkan.

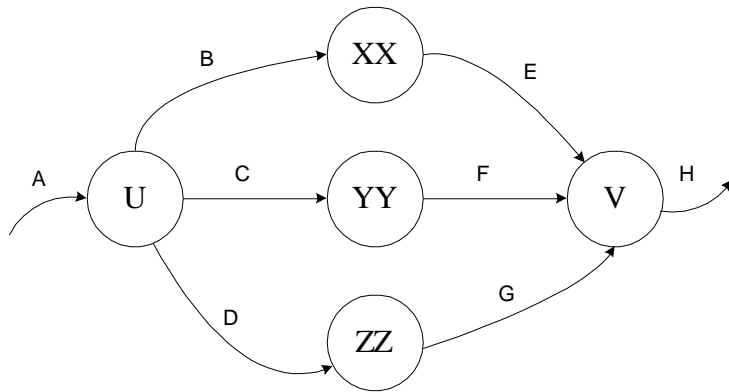
Symbol *Structure Chart*

SIMBOL	KETERANGAN
	<p><u>Module</u> Simbol ini menunjukkan suatu modul</p> <p><u>Connection</u> Digunakan untuk menghubungkan modul, atau simbol untuk menyatakan pemanggilan modul.</p> <p><u>Couple</u> Menunjukkan data atau elemen kontrol yang dikirimkan atau diterima dari satu modul. Panah dengan lingkaran kosong menunjukkan data, sedangkan panah dengan lingkaran yang diblok menunjukkan elemen kontrol.</p>

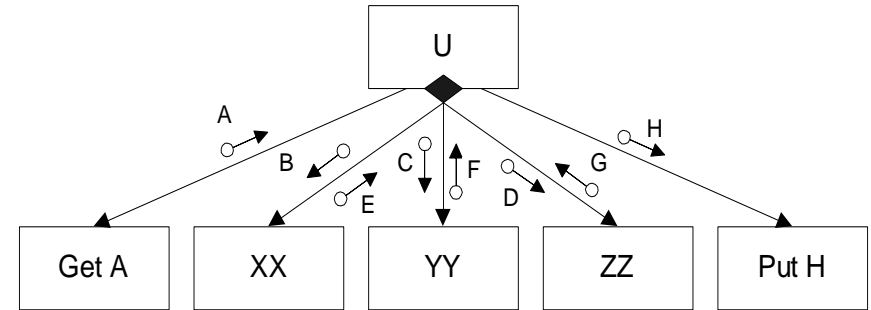
Simbol Tambahan *Structure Chart*

SIMBOL	KETERANGAN
 	<p><u>Loop</u> Menunjukkan suatu pengulangan di dalam modul</p> <p><u>Decision</u> Menunjukkan suatu penyeleksian kondisi di dalam modul, atau menunjukkan transform center</p>

TRANSFORMASI DFD - STRUCTURE CHART



???

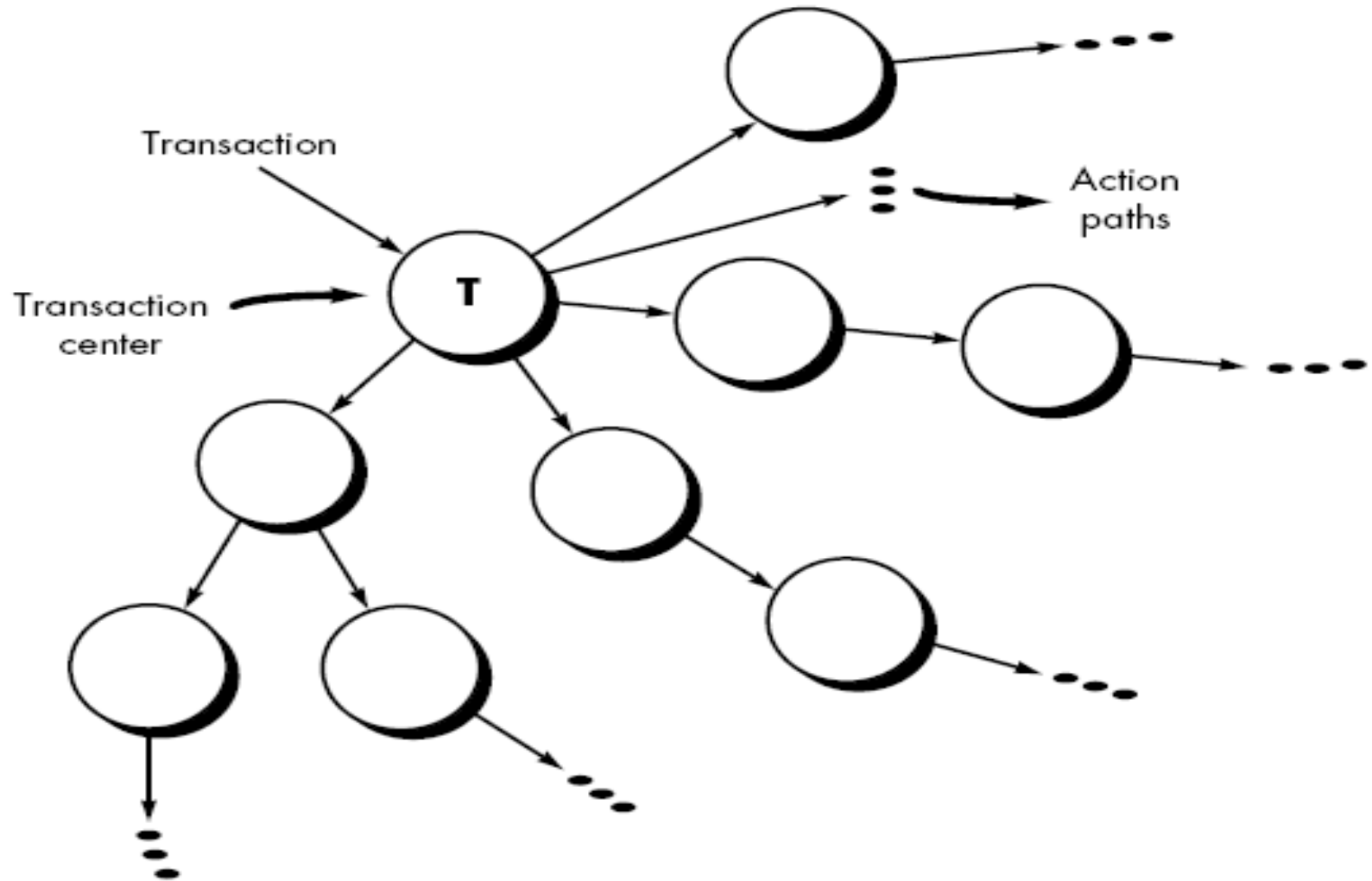


- Menggunakan teknik:
 - Analisis transformasi
 - Analisis transaksi

ALIRAN TRANSFORMASI

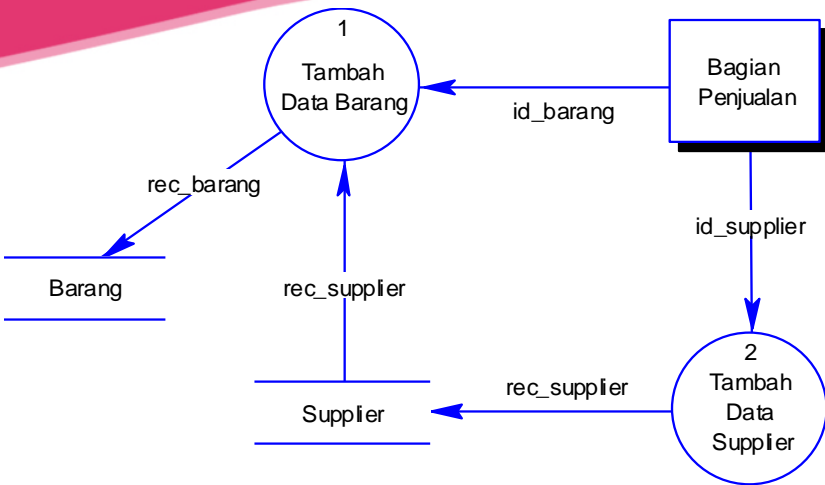
Mentransformasikan data *eksternal* ke bentuk *internal* diidentifikasi sebagai aliran masuk, terjadi transisi, data masuk dilewatkan melalui pusat transformasi dan bergerak keluar melalui jalur keluar.

Aliran Transaksi

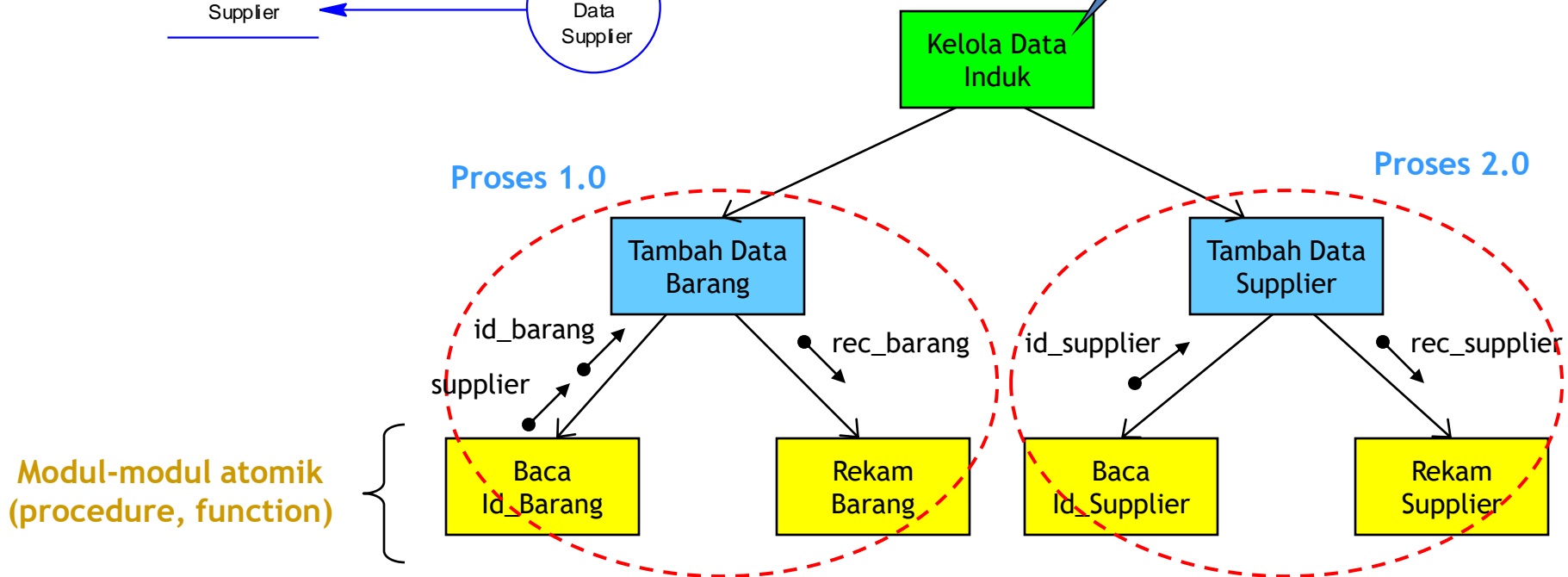


CONTOH ARSITEKTUR PERANGKAT LUNAK

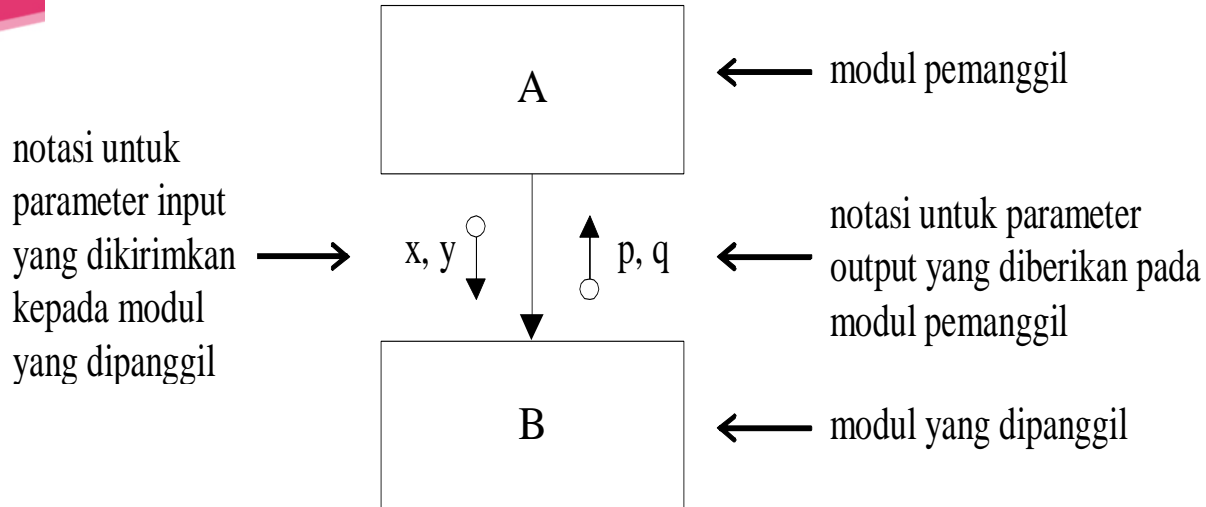
Model Analisis (DFD level atomik)



Arsitektur Perangkat Lunak (Structure Chart)



STRUCTURE CHART (1) : PASCAL



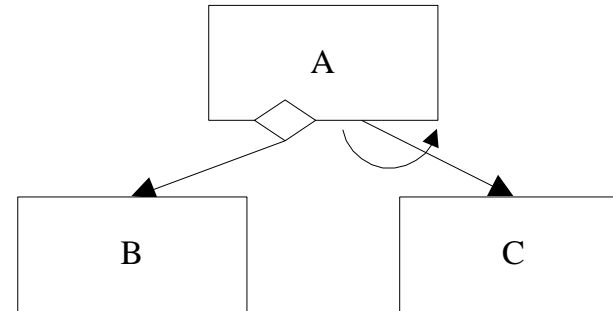
- Modul A memanggil modul B dengan data x dan y sebagai parameter-nya.
- Modul B mengirimkan data p dan q sebagai *return value* ke modul A.

```
Procedure A;  
  Var p, q : Real;  
  Procedure B(x, y : Real);  
  Begin  
    p := ... { manipulasi nilai p }  
    q := ... { manipulasi nilai q }  
  End;  
Begin  
  B(x, y); { call procedure B }  
End;
```

Potongan kode program dalam bahasa Pascal

STRUCTURE CHART (2) : PASCAL

- Modul A akan memanggil modul B **jika** kondisi dalam modul A dipenuhi.
- Modul A akan memanggil modul C secara **berulang**.



Potongan kode program dalam bahasa Pascal

```
Procedure C;  
Begin  
    ...  
End;  
Procedure B;  
Begin  
    ...  
End;  
Procedure A;  
Begin  
    If True Then B; {call procedure B}  
    While True Do C; {call procedure C}  
End;
```

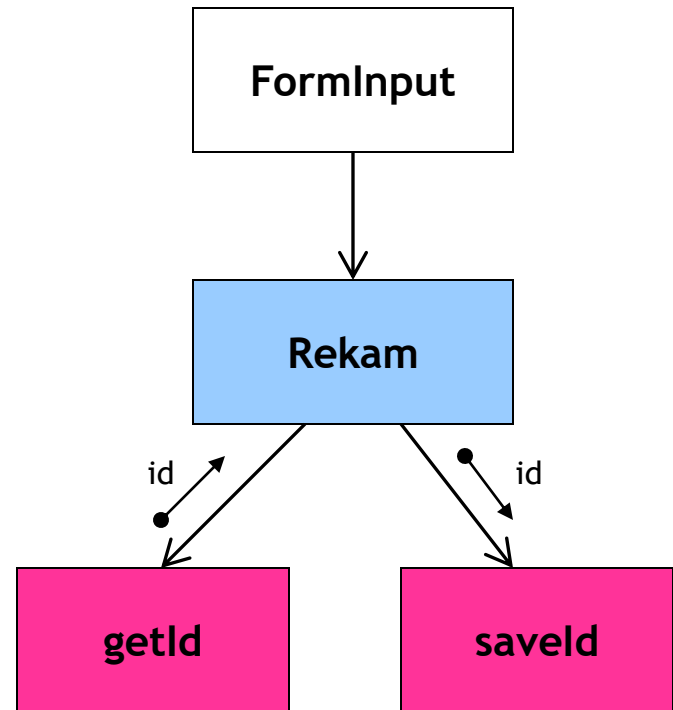
STRUCTURE CHART (3) : PHP

FormInput.html

```
<html>
...
<form method=post action=Rekam.php>
...
</html>
```

Rekam.php

```
<?
// Rekam.php
function getId() {
}
function saveId(id) {
}
id = getId();
saveId(id)
?>
```



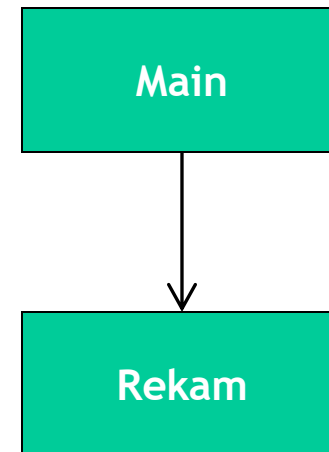
STRUCTURE CHART (4) : DELPHI

main.pas

```
unit main;  
...  
var  
    Form1: TForm1;  
implementation  
uses Rekam;  
  
procedure TForm1.Click(Sender: TObject);  
begin  
    frmRekam.Show;  
end;  
  
end.
```

rekam.pas

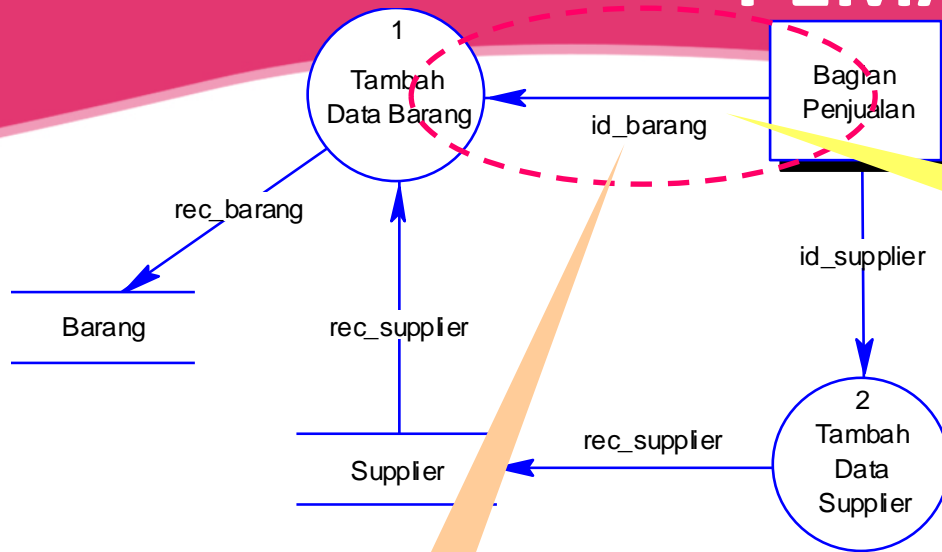
```
unit Rekam;  
...  
var  
    frmRekam: TForm1;  
implementation  
...  
end.
```



PERANCANGAN ANTARMUKA PEMAKAI

- Secara fisik antarmuka pemakai yang dirancang adalah tampilan layar (*form*, halaman web).
- Jenisnya dapat berupa:
 - Menu pilihan
 - Form isian (*entry*)
 - Penyajian informasi (*report, query*)
 - Kotak dialog, jika diperlukan
 - Fasilitas bantuan (*Help*), jika diperlukan

IDENTIFIKASI RANCANGAN ANTARMUKA PEMAKAI



Ada interaksi antara pengguna dengan PL

Harus ada user interface untuk Tambah Data Barang!



Ada data yang diberikan oleh pengguna ke PL

$id_barang = kode_brg + nama_brg + satuan + jenis + hrg_beli + hrg_jual + jml_stok + kode_sup$



Tambah Data Barang		X
Kode Barang:	<input type="text"/>	
Nama Barang:	<input type="text"/>	
Satuan:	<input type="text"/>	
Jenis:	<input type="checkbox"/> 1:Milik <input type="checkbox"/> 2:Konsinyasi	
Harga Beli:	Rp. <input type="text"/>	
Harga Jual:	Rp. <input type="text"/>	
Jumlah Stok:	<input type="text"/> unit	
Kode Supplier:	<input type="text"/> ▼	
<input type="button" value="Rekam"/> <input type="button" value="Batal"/>		

PENULISAN SPESIFIKASI PROGRAM

- **Deskripsi prosedural** (algoritma) untuk semua **modul-modul program** yang menjadi elemen-elemen struktural dari arsitektur perangkat lunak:
 - Prosedur
 - Fungsi
- Merupakan penjelasan lebih rinci dan teknis dari spesifikasi proses.
- Ditulis dengan menggunakan notasi **pseudo-code**, atau notasi yang mirip dengan bahasa pemrograman yang digunakan.

Proses 1.1 Tambah Data Barang

Begin

While data barang masih ada Do

Baca identitas barang

Verifikasi

If not valid Then

tulis pesan

Else rekam ke tabel barang

End

Procedure btnRekamBarangClick

Kamus

{ Deklarasi variabel; TEdit, TDBLookupCombo, TTable terdefinisi }

eKode, eNama, eSatuan, eJenis, eHrgBeli, eHrgJual, eJmlStok: TEdit

DBLookupCombo1: TDBLookupCombo

TabelBarang, TabelSupplier: TTable

Algoritma

{ Buka tabel barang dan supplier }

TabelBarang.Open

TabelSupplier.Open

{ Baca identitas barang melalui komponen TEdit dan validasi }

{ Rekam ke tabel barang }

TabelBarang.Append

TabelBarang.FieldName('Kode_Brg').AsString := eKode.Text

TabelBarang.FieldName('Nama_Brg').AsString := eNama.Text

TabelBarang.FieldName('Satuan').AsString := eSatuan.Text

TabelBarang.FieldName('Jenis').AsInteger:=StrToInt(eJenis.Text)

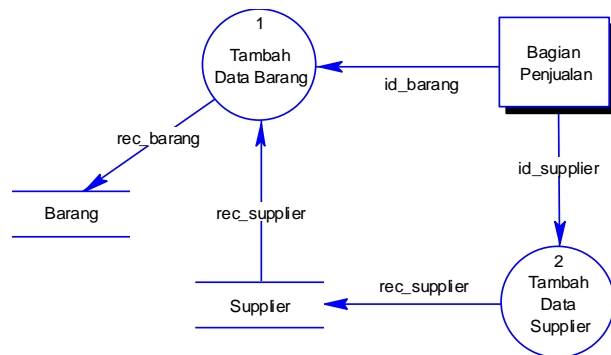
TabelBarang.FieldName('Hrg_Beli').AsInteger:=StrToInt(eHrgBeli.Text)

TabelBarang.FieldName('Hrg_Jual').AsInteger:=StrToInt(eHrgJual.Text)

TabelBarang.FieldName('Jml_Stok').AsInteger:=StrToInt(eJmlStok.Text)

TabelBarang.FieldName('Kode_Sup').AsString := DBLookupCombo1.Value;

TabelBarang.Post



Latihan

- Sebutkan tahapan pada software desain.
- Sebutkan bagian pada model analisis dan model perancangan.
- Jelaskan notasi yang ada pada struktur chart.
- Jelaskan perbedaan antara analisis transformasi dan analisis transaksi.
- Berikan contoh struktur chart pada salah satu bahasa pemrograman.



TERIMA KASIH