

A close-up photograph of a hand holding a black fountain pen with gold accents, writing on a document. The document contains a table with columns of numbers and text, though it is out of focus. The background is blurred, showing what appears to be a desk or office environment.

Perancangan Perangkat Lunak

Pendahuluan (1)

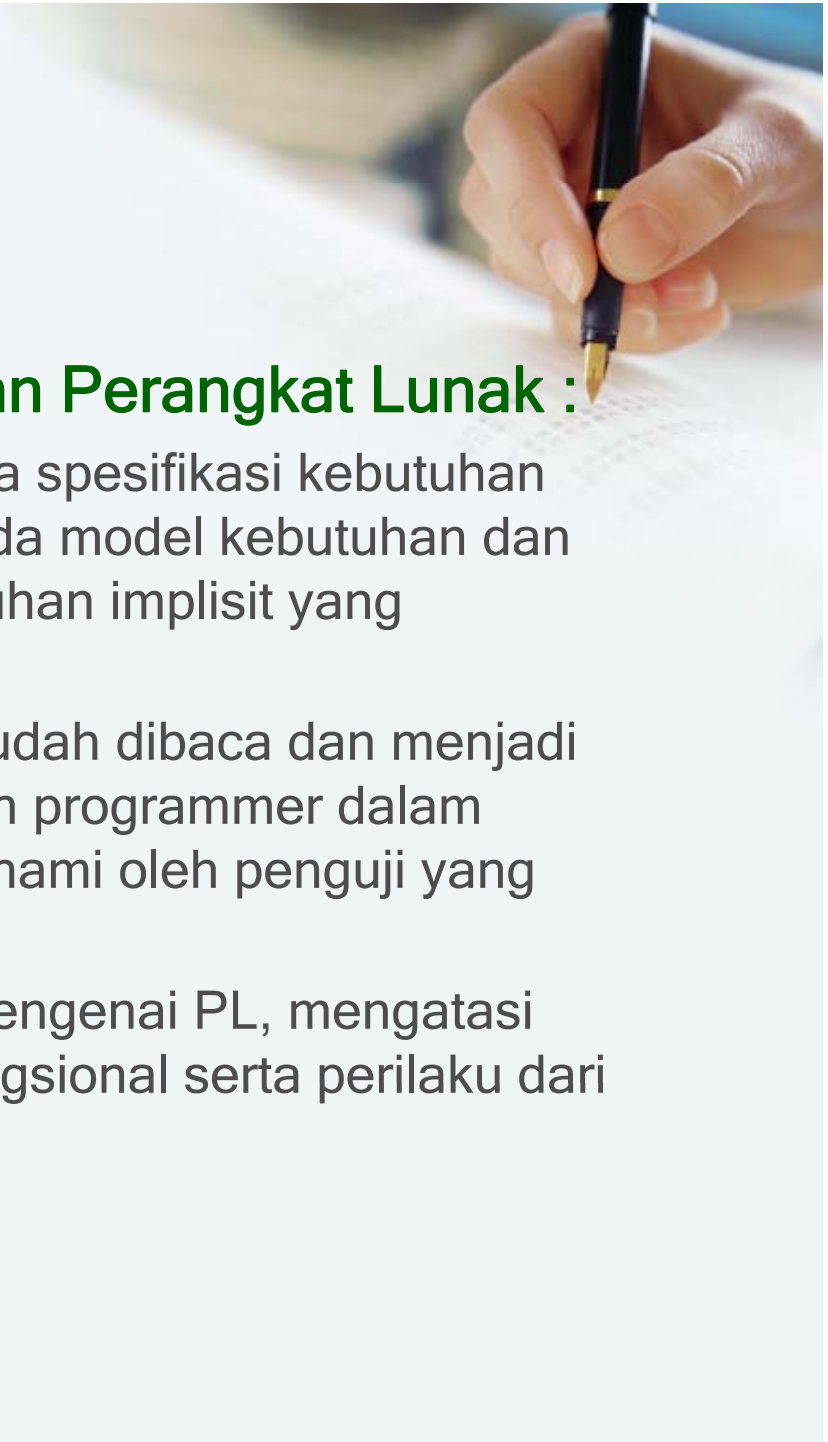
Perancangan Perangkat Lunak :

- Tahap akhir aktivitas Pemodelan dalam Rekayasa PL (sebelumnya adalah spesifikasi kebutuhan dan analisis pemodelan)
- Landasan awal untuk aktivitas Konstruksi (penulisan program dan pengujian)
- “Cetak Biru” dari perangkat lunak

Pendahuluan (2)

Karakteristik umum Perancangan Perangkat Lunak :

- Dapat mengimplementasikan semua spesifikasi kebutuhan yang secara eksplisit ditemukan pada model kebutuhan dan dapat mengakomodir semua kebutuhan implisit yang diinginkan oleh stakeholder
- Menghasilkan produk kerja yang mudah dibaca dan menjadi panduan yang mudah dipahami oleh programmer dalam menuliskan kode program dan dipahami oleh penguji yang melakukan pengujian PL
- Menyediakan gambaran lengkap mengenai PL, mengatasi permasalahan pada ranah data, fungsional serta perilaku dari sudut implementasi



Pendahuluan (3)

Ciri Perancangan Perangkat Lunak yang Baik :

1. Mencerminkan arsitektur yang :
 1. Menggunakan pola arsitektur sistem yang telah dikenal sebelumnya
 2. Tersusun dari komponen yang memiliki karakteristik perancangan PL yang baik
 3. Dapat diimplementasikan dalam bentuk evolusioner dan memfasilitasi proses implementasi dan pengujian PL
2. Bersifat modular, dapat dibagi menjadi elemen-elemen serta subsistem di dalamnya
3. Memuat secara jelas representasi data, arsitektur, antarmuka dan komponen-komponennya



Pendahuluan (4)

Ciri Perancangan Perangkat Lunak yang Baik : (lanjt)

4. Memuat struktur data yang sesuai untuk kelas-kelas sedemikian rupa sehingga saat diimplementasikan penggambaran pola data yang dapat dikenali
5. Memuat rancangan komponen-komponen yang memperlihatkan karakteristik fungsional yang bersifat mandiri
6. Memuat antarmuka yang mengurangi kompleksitas hubungan antara komponen-komponen dalam PL dengan lingkungan eksternalnya
7. Diperoleh dari metode perulangan yang dikenali oleh informasi yang diperoleh selama analisis kebutuhan PL dilakukan
8. Direpresentasikan dalam notasi yang secara efektif mengkomunikasikan maknanya



Desain PL sebagai sebuah Masalah yang Sulit

- Tidak ada rumusan yang pasti
- Tidak ada aturan kapan berhenti
- Solusi tidak hanya benar atau salah
- Setiap masalah yang sulit adalah gejala dari masalah lain



Metodologi Pengembangan Sistem

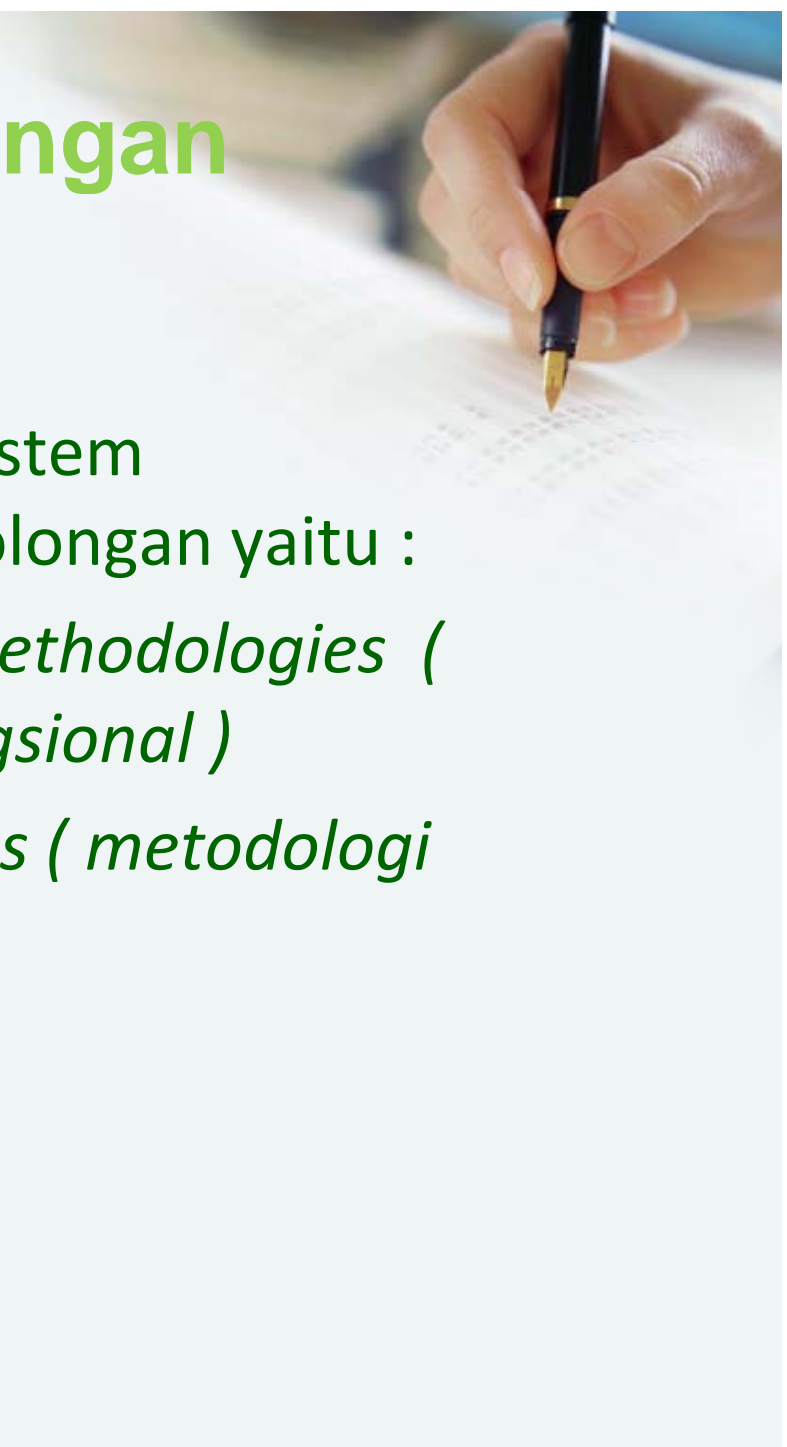
A close-up photograph of a person's hand holding a black pen, writing on a white document. The document has some faint, illegible text on it. The background is blurred, showing what appears to be a desk or workspace.


- Metodologi adalah kesatuan metode-metode , prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan , seni, atau disiplin lainnya.
- Metode adalah suatu cara / teknik yang sistematis untuk mengerjakan sesuatu.

Metodologi Pengembangan Sistem

Metodologi pengembangan sistem diklasifikasikan menjadi 3 golongan yaitu :

- *Functional decomposition methodologies (metodologi Pemecahan fungsional)*
- *Data oriented methodologies (metodologi orientasi data)*
- *Prescriptive methodologies*



- 
- A close-up photograph of a person's hand holding a black pen with a gold nib, writing on a white document. The document has some faint, illegible text on it. The background is a light, out-of-focus surface.
- Functional decomposition
 - Data Flow Design (SA/SD)
 - Design based on Data Structures (JSD/JSP)
 - OO is gOOd, isn't it



Metodologi Pemecahan Fungsional

- *Functional decomposition methodologies (metodologi Pemecahan fungsional)*

Menekankan pada pemecahan dari sistem ke dalam subsistem-subsistem yang lebih kecil, sehingga lebih mudah dipahami, dirancang dan diterapkan. Yang termasuk metodologi ini :

- *HIPO (Hierarchy Input Process Output)*
- *Stepwise refinement (SR) atau Iterative Stepwise Refinement (ISR)*
- *Information hiding*



Metodologi Orientasi Data

- *Data oriented methodologies (metodologi orientasi data)*
Menekankan pada karakteristik dari data yang akan diproses. Dapat dikelompokkan menjadi 2 kelas yaitu :
 - Data Flow Oriented Methodologies
Yang termasuk dalam metodologi ini adalah *DFD*, *SADT (Structured Analysis and Design Technique)*; *Composite Design* ; *Structured System Analysis & Design (SSAD)*
 - Data Structure Oriented Methodologies
Yang termasuk metodologi ini adalah *JSD*; *W/O*



Prescriptive Methodologies

- *Prescriptive methodologies*

Metodologi yang dikembangkan oleh pabrik PL. Tersedia dalam paket-paket program. Yang termasuk metodologi ini adalah :

- *ISDOS (Information System Design and Optimization System)*
- *PLEXSYS*
- *PRIDE*
- *SDM/70*
- *SPEKTRUM*
- *SRES (Software Requirement Engineering System)*
- *SREM (Software Requirement Engineering Methodology)*

Alat Pengembangan Sistem

A close-up photograph of a person's hand holding a black pen with a gold nib, writing on a white document. The document has some faint, illegible text on it. The background is blurred, showing what appears to be a desk or workspace.

- Alat-alat Pengembangan sistem yang berbentuk grafik diantaranya :
 - DFD, ERD, WO, Jackson
 - UML (berorientasi objek)

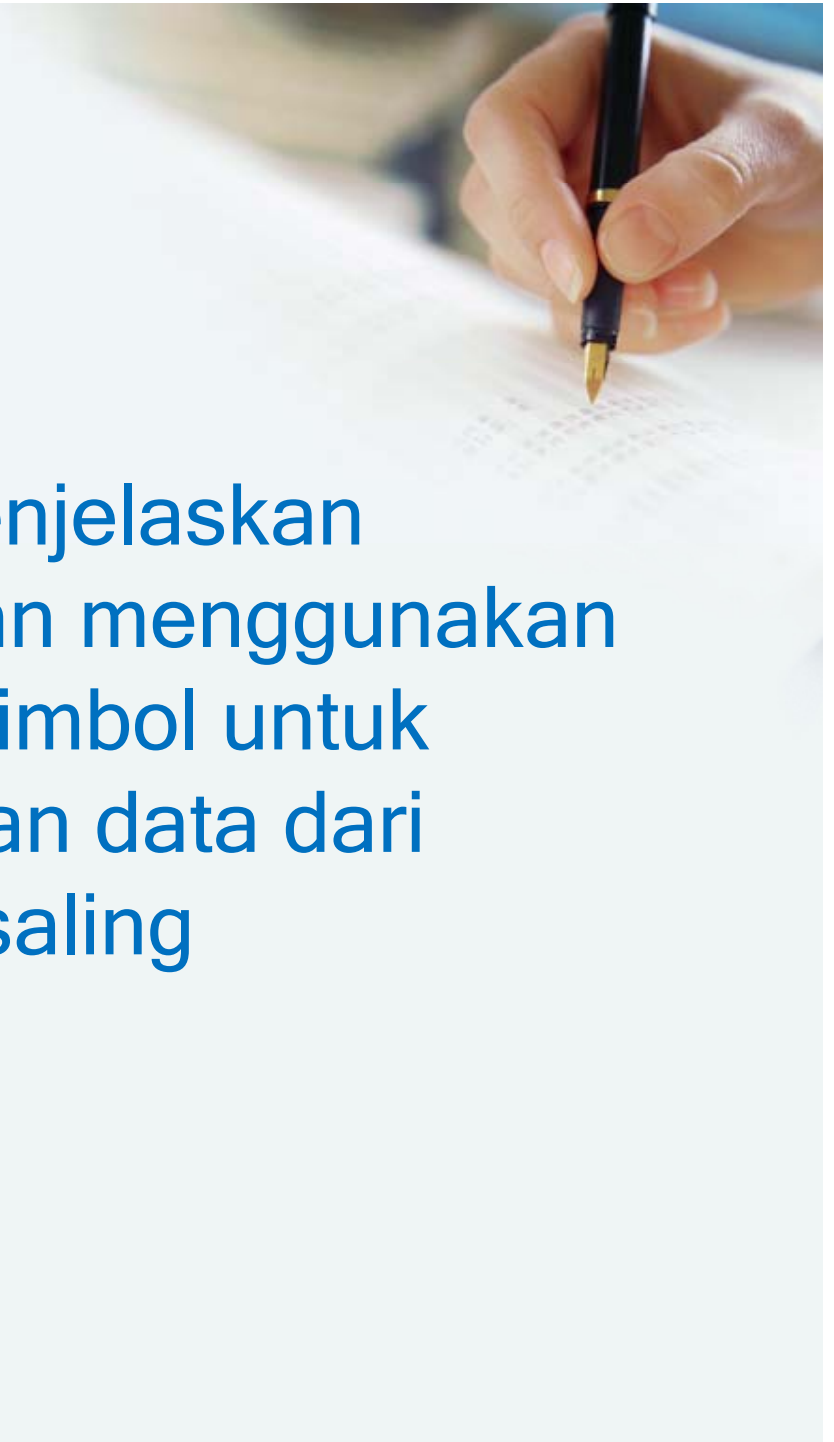


DFD

DATA FLOW DIAGRAM

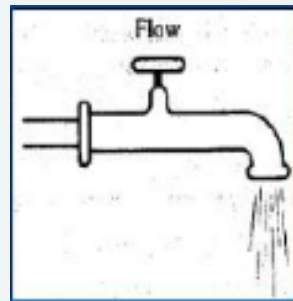
Pengertian DFD

- Suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk atau simbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan.

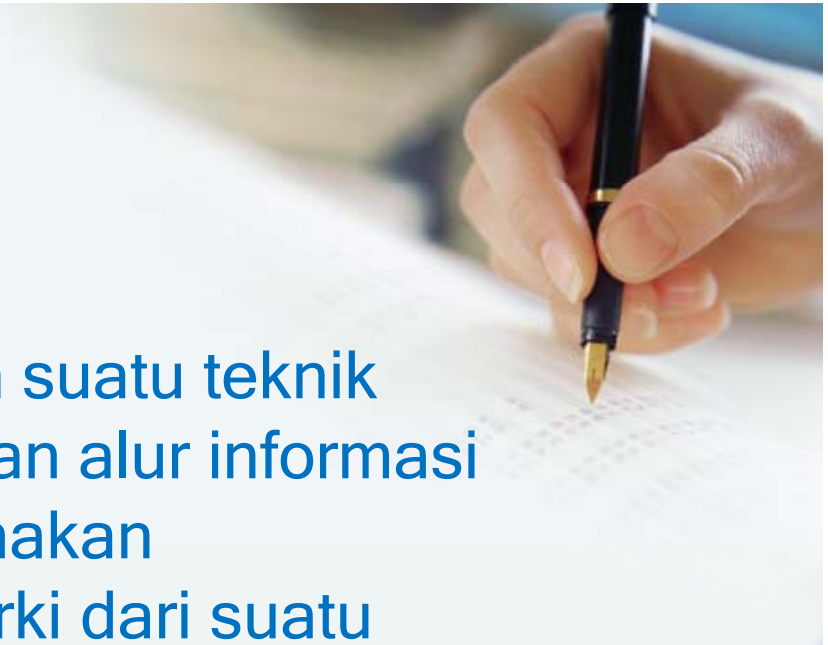


Pengertian DFD

- Data Flow Diagram adalah suatu teknik grafik yang menggambarkan alur informasi dan transformasi menggunakan dekomposisi fungsi terhirarki dari suatu proses dalam suatu sistem



- Apa sebagai masukan.
- Apa sebagai keluaran.
- Bagaimana terjadi proses perubahan didalamnya



Pengertian DFD

- DFD mempunyai level-level mulai dari yang terkecil, yaitu level 0 (context diagram).
- Context diagram merupakan gambaran paling umum dari sistem, yang hanya memiliki satu proses saja untuk mewakili seluruh sistem.



Pengertian DFD

- Semakin bertambahnya level dalam DFD akan semakin detail digambarkannya proses-proses yang ada pada sistem, tetapi yang boleh bertambah hanya proses dan data flow saja.



Pengertian DFD

- Data source jumlahnya harus tetap dengan yang ada pada context diagram.
- Khusus untuk data store, pada context diagram masih belum digambarkan, akan tampak pada level 1 dan konsisten jumlahnya sampai pada level berikutnya.



Notasi DFD

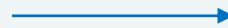
- entitas



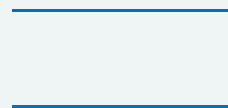
- proses



- data flows



- data stores



Membuat Data Flow Diagrams

Langkah-langkah:

1. Buat daftar aktifitas
2. Membangun DFD level Context
(Mengidentifikasi entitas eksternal dan proses)
3. Membangun DFD Level 0
(mengidentifikasi sub proses yang dikelola)
4. Membangun DFD Level 1- n
(mengidentifikasi arus data aktual dan simpanan data)
5. Memeriksa terhadap aturan DFD



Petunjuk Penamaan DFD

- External Entity → Kata benda
- Data Flow → Nama data
- Proses → kata kerja berupa frasa
 - Nama sistem
 - Nama subsistem
- Data Store → Kata benda



Membuat Data Flow Diagram

Contoh 1 “Lemonade Stand”



Membuat Data Flow Diagrams

Contoh

Operasi lemonade stand akan digunakan untuk menunjukkan pembuatan data flow diagrams.



Langkah-langkah:

1. Membuat daftar aktifitas
 - Old way: no Use-Case Diagram
 - New way: use Use-Case Diagram
2. Membangun DFD Level Context (identifikasi sumber dan pengguna)
3. Membangun DFD Level 0 (mengidentifikasi sub proses yang dikelola)
4. Construct Level 1- n DFD (mengidentifikasi alur data dan simpanan data aktual)



Membuat Data Flow Diagrams

Contoh

Pikirkan kegiatan yang berlangsung di stand lemonade/limun.



1. Membuat daftar aktifitas

Customer Order
Serve Product
Collect Payment
Produce Product
Store Product



Membuat Data Flow Diagrams

Contoh

Juga pikirkan kegiatan tambahan yang diperlukan untuk mendukung kegiatan utama.



1. Membuat daftar aktifitas

Customer Order
Serve Product
Collect Payment
Produce Product
Store Product
Order Raw Materials
Pay for Raw Materials
Pay for Labor



Membuat Data Flow Diagrams

Contoh

Kelompokkan kegiatan ini dalam beberapa mode logis, mungkin bidang fungsional.



1. Membuat daftar aktifitas

Customer Order
Serve Product
Collect Payment

Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor



Membuat Data Flow Diagrams

Contoh

Membuat diagram level context mengidentifikasi sumber dan pengguna.

Customer Order
Serve Product
Collect Payment

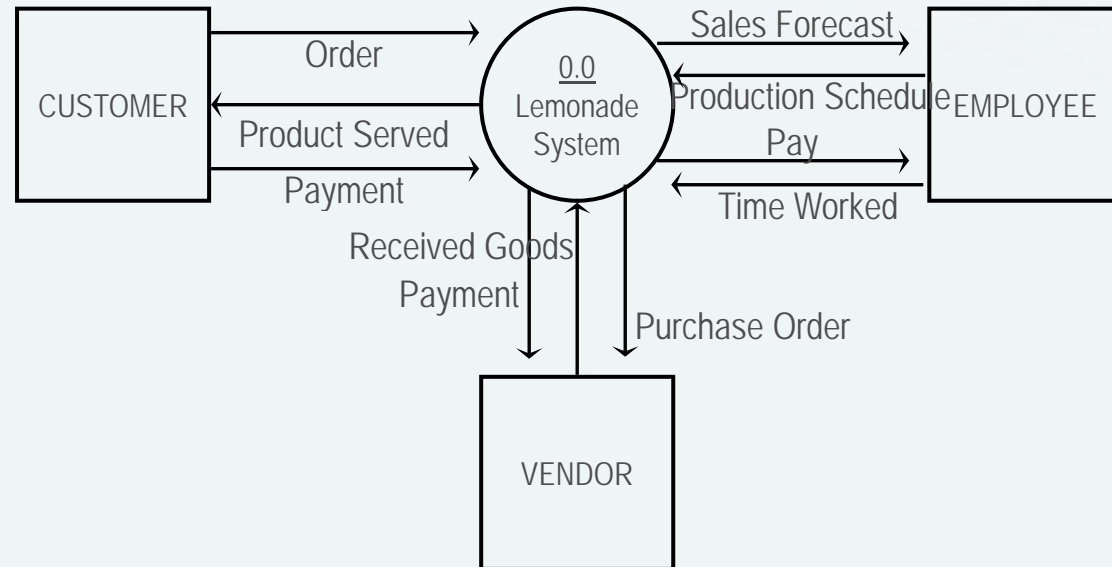
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

2. Membangun DFD Level Context (identifies sources and sink)

Context Level DFD



Membuat Data Flow Diagrams

Contoh

Membuat diagram level 0 identifikasi subsistem logika yang mungkin ada.

Customer Order
Serve Product
Collect Payment

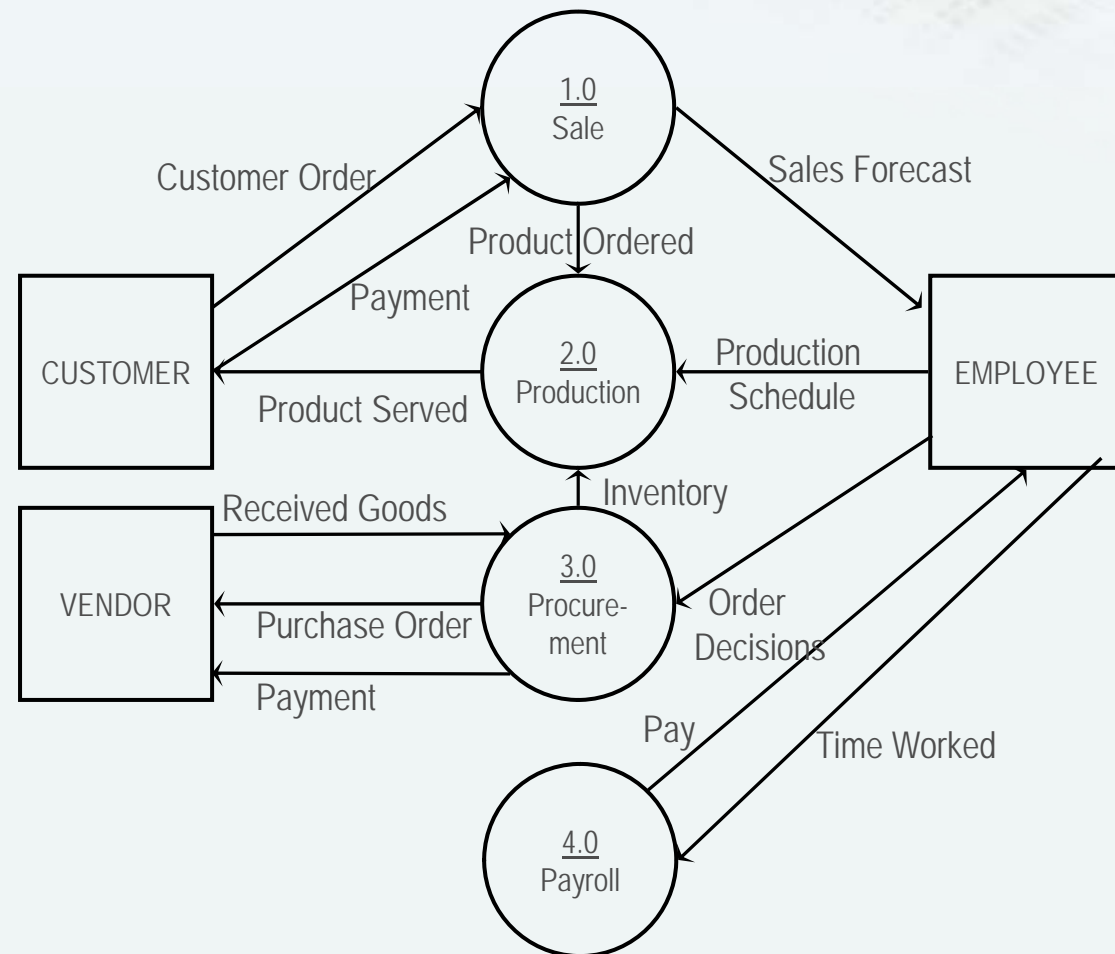
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

3. Membangun DFD Level 0 (identifies manageable sub processes)

Level 0 DFD



Membuat Data Flow Diagrams

Contoh

Membuat level 1 dekomposisi proses dalam level 0 dan identifikasi simpanan data.

Customer Order
Serve Product
Collect Payment

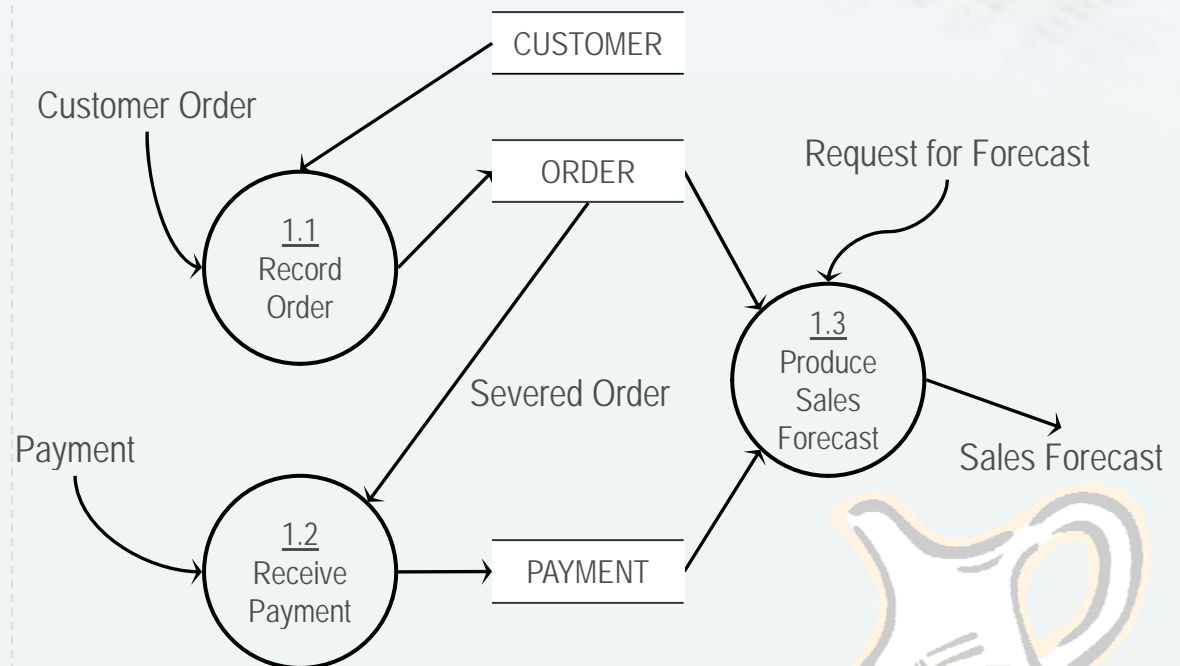
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

4. Membangun DFD Level 1- n (identifikasi alur data dan simpanan data)

Level 1 DFD



Membuat Data Flow Diagrams

Contoh

Membuat level 1 dekomposisi proses dalam level 0 dan identifikasi simpanan data.

Customer Order
Serve Product
Collect Payment

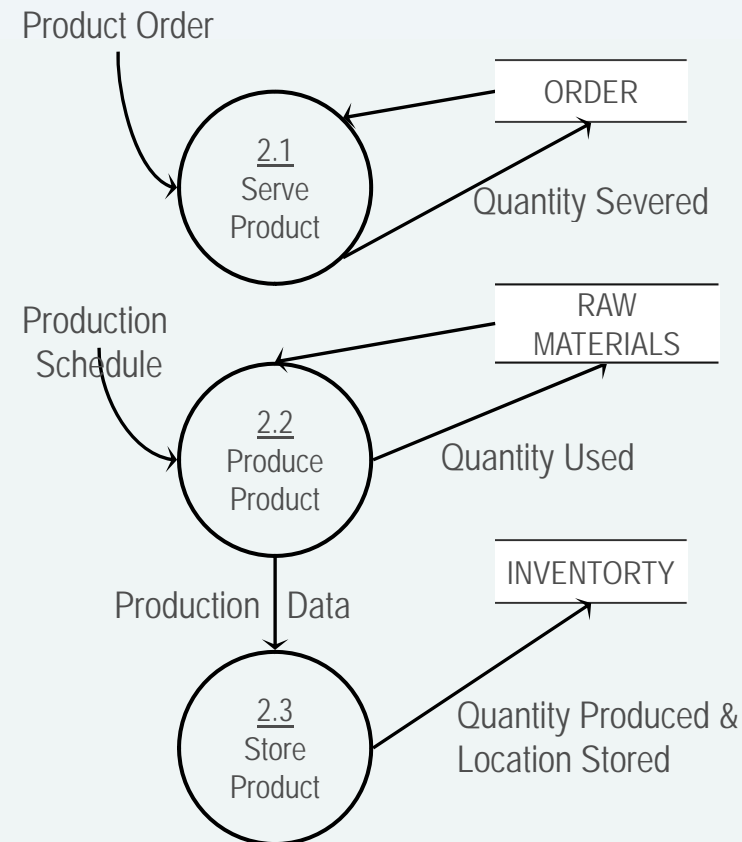
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

4. Membangun Level 1 (lanjutan)

Level 1 DFD



Membuat Data Flow Diagrams

Contoh

Membuat level 1 dekomposisi proses dalam level 0 dan identifikasi simpanan data.

Customer Order
Serve Product
Collect Payment

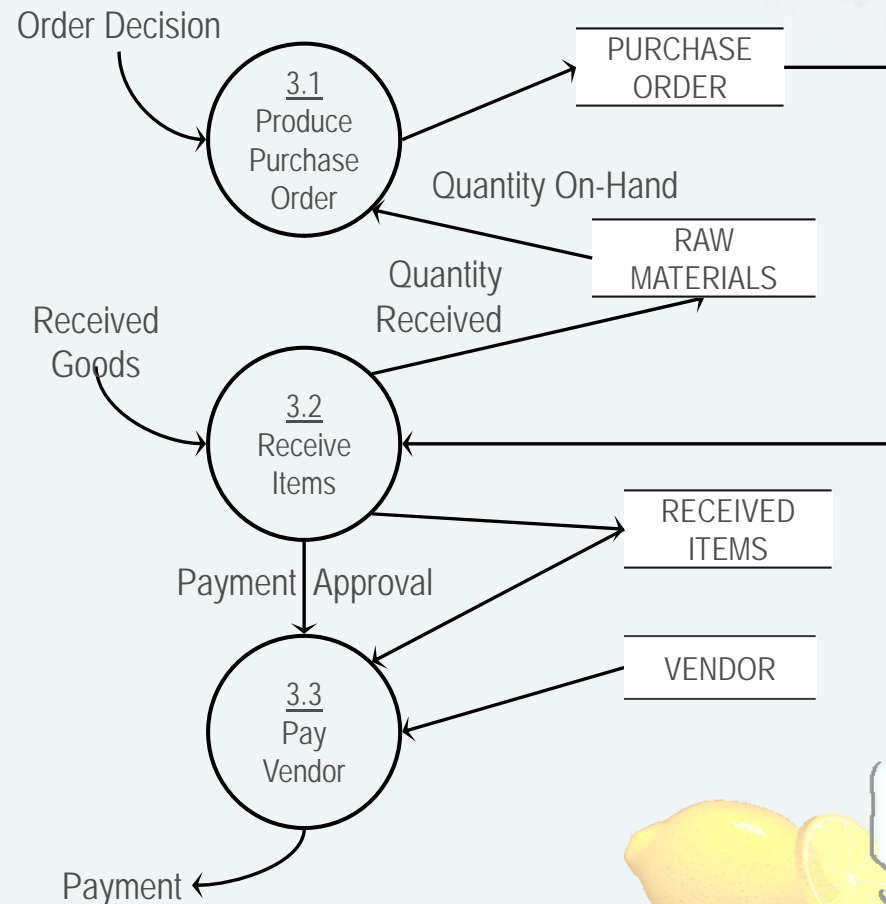
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

4. Membangun Level 1 (lanjutan)

Level 1 DFD



Membuat Data Flow Diagrams

Contoh

Membuat level 1 dekomposisi proses dalam level 0 dan identifikasi simpanan data.

Customer Order
Serve Product
Collect Payment

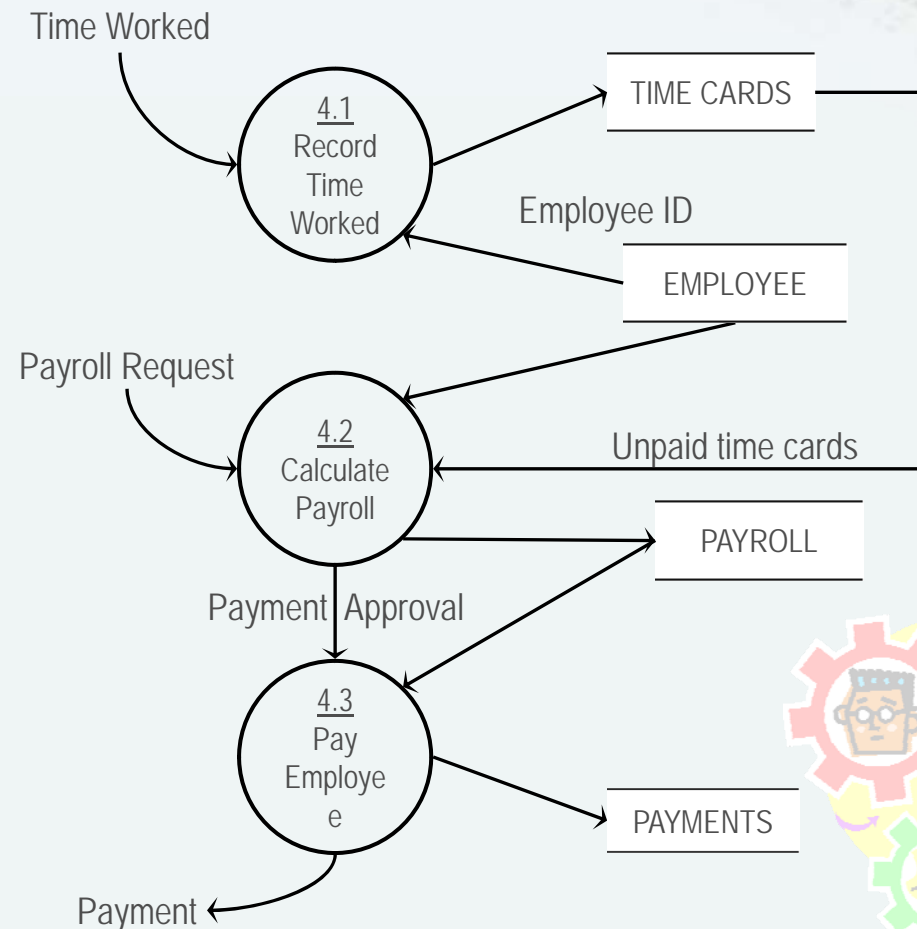
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

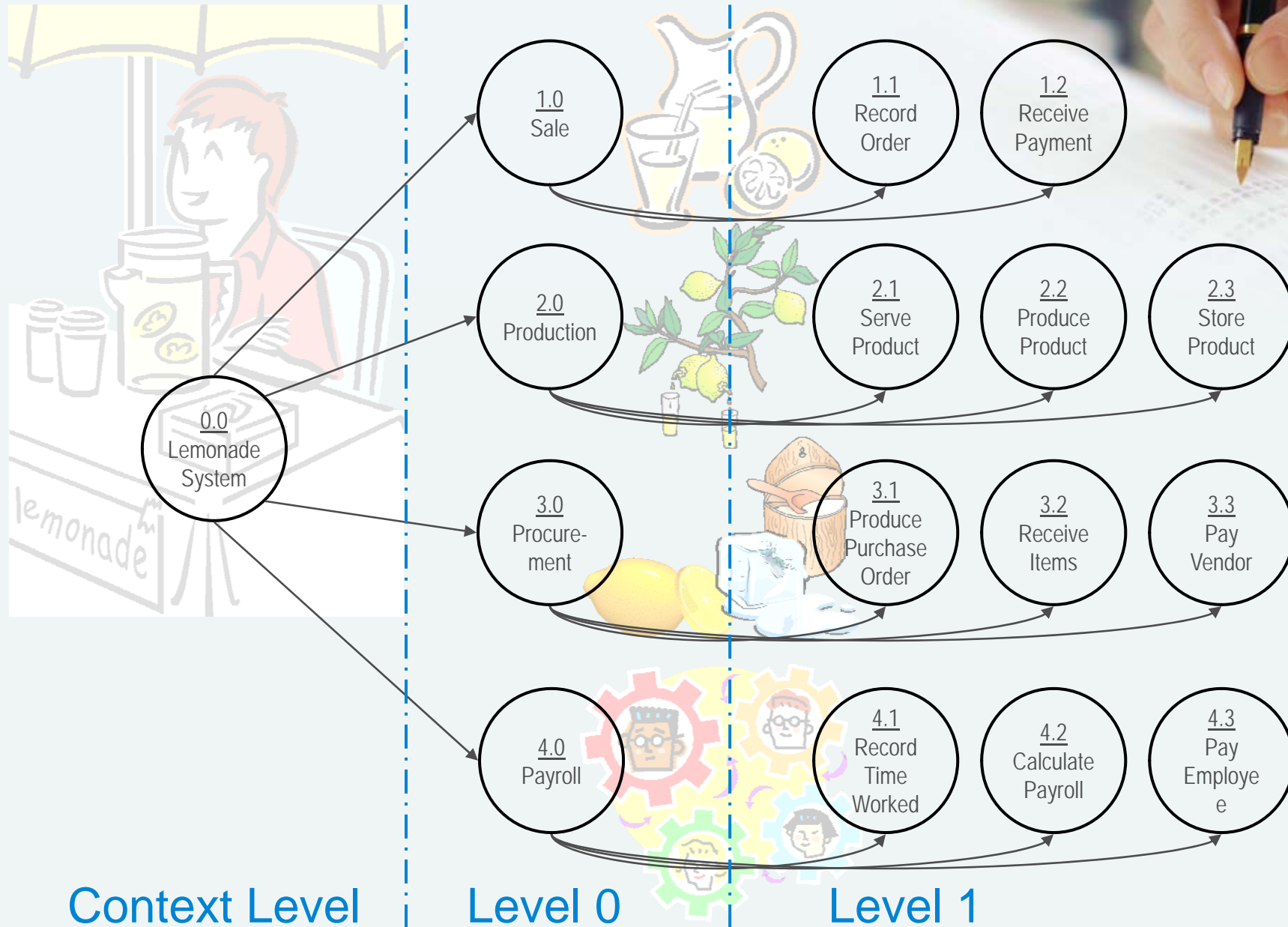
Pay for Labor

4. Membangun Level 1 (lanjutan)

Level 1 DFD



Dekomposisi Proses



Contoh Kasus 2

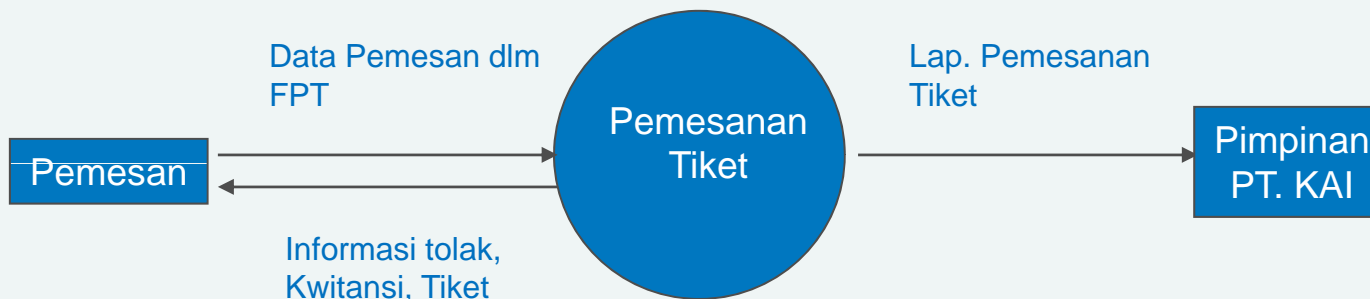
- Kasus: Sistem Pemesanan Tiket Kereta Api Pada PT. Kereta Api Indonesia (PT. KAI)
- Prosedur yang sedang berjalan:

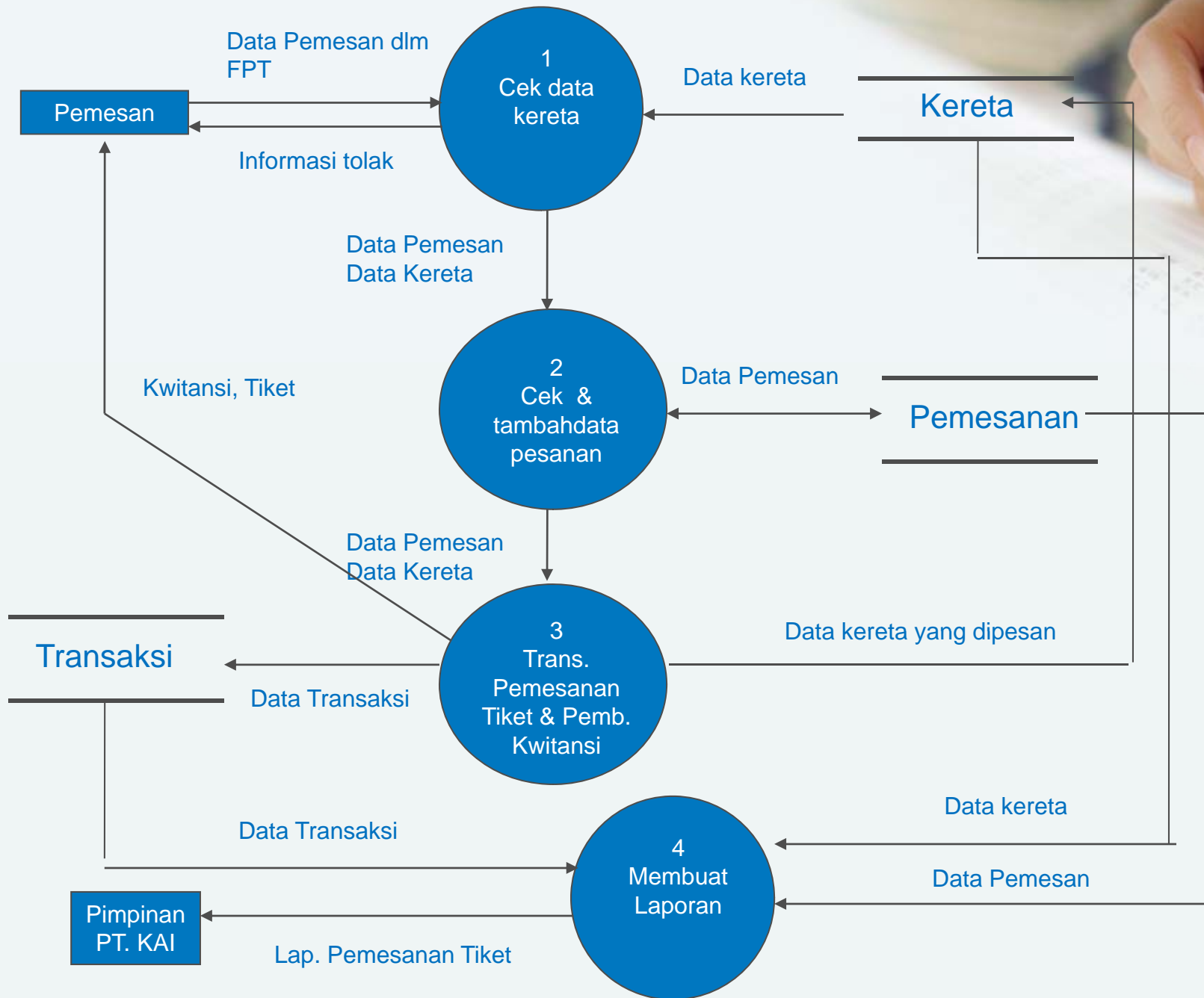
Pemesan mengisi formulir pesanan tiket yang telah disiapkan oleh Bagian Pemesanan. Adapun yang diisi di dalam formulir pesanan diantaranya adalah Nama pemesan, alamat, No. Telpon, nama kereta api, kelas, tgl.berangkat dan jumlah pesanan. Setelah formulir diisi dengan benar, formulir diserahkan ke bagian pemesanan. Data pesanan akan dientry ke dalam komputer dan disimpan di dalam file. Pembayaran dilakukan secara tunai. Setelah proses pembayaran selesai, maka pemesan mendapat bukti pembayaran berupa kwitansi dan bukti pembayaran tiket kereta api.



Contoh Kasus

- Prosedur yang diusulkan:
Prosedur yang diusulkan tidak jauh berbeda dengan prosedur yang telah berjalan sebelumnya, hanya pada prosedur yang diusulkan akan ditambah satu item yaitu No.ID pemesan berdasarkan urutan formulir data pesanan yang masuk ke Bagian Pemesanan.
- Diagram Konteks





Tugas Kelompok

Sebuah organisasi meminta tim anda untuk membuat sistem informasi. Buat tulisan yang menjelaskan tentang Perancangan sistem sebuah organisasi (jenis organisasi dan cakupan kasus bebas).

- Struktur Tulisan :
 - Cover
 - Kata Pengantar
 - Isi Tulisan
 - Daftar Pustaka (minimal 5)
- Setiap gambar dan tabel harus ada penjelasannya
- Isi tulisan:
 - Pendahuluan (organisasi yang akan dijelaskan bisnis prosesnya dan bisnis proses dari organisasi tsb)
 - Rancangan bisnis proses (DFD)
 - Rancangan Database (ERD dan Kamus Data)

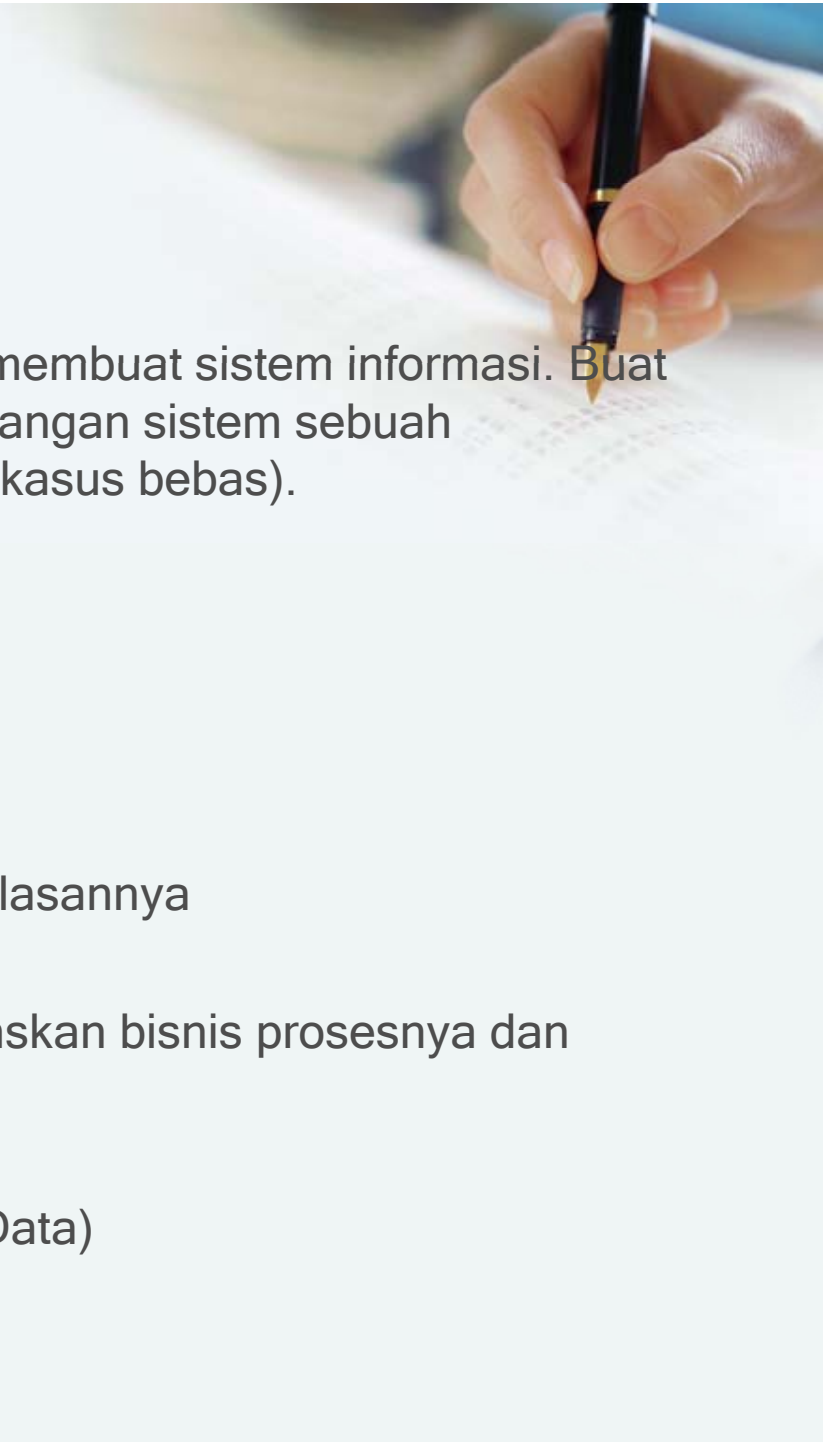




DIAGRAM JACKSON

PENDAHULUAN

- Ada dua tipe notasi untuk menggambarkan susunan komponen-komponen:
 - notasi grafik yang disebut dengan diagram Jackson, atau diagram struktur;
 - notasi yang non-grafik yang disebut dengan Struktur Text atau Schematic Logic.
- Diagram Jackson dan notasi-notasi struktur text untuk deret, iterasi dan seleksi diberikan pada gambar 1-3



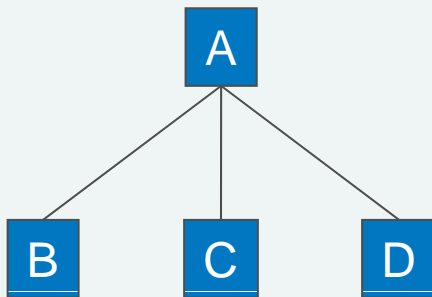
KOMPONEN DERET

A close-up photograph of a person's hand holding a black pen with a gold nib, writing on a white document. The document has some faint, illegible text on it. The background is blurred, showing what appears to be a desk or table.

- Deret mempunyai dua atau lebih bagian yang dapat terjadi secara bersamaan. Contohnya pada gambar 1 menunjukkan diagram Jackson dan notasi struktur text untuk sebuah komponen deret A yang terdiri dari B yang diikuti dengan C, dan D. Dengan perkataan lain, B, C, dan D adalah komponen-komponen dari A.

KOMPONEN DERET

Diagram Jackson



(a)

Struktur Teks

```
A Seq  
B;  
C;  
D;  
A End
```

(b)

Gambar 1. Penggambaran komponen deret

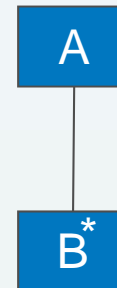
ITERASI

- Komponen dari sebuah iterasi mempunyai sebuah bagian yang terjadi *beberapa kali* atau *nol kali* untuk setiap kejadian. Pada gambar 2 tanda asteris di atas B menandakan bahwa komponen A mempunyai iterasi yaitu B; artinya, B dilaksanakan nol atau beberapa kali untuk setiap kejadian A secara berulang-ulang. Notasi struktur dari komponen A diberikan dalam tiga bentuk berbeda (i)-(iii) seperti pada gambar 2b.



ITERASI

Iteration



Struktur Text

Aitr A itr while <
~; kondisi >
A end ~;
 A end
(i) (ii)

A itr until <
kondisi >
~;
A end
(iii)

Gambar 2. Gambaran Komponen Iterasi

SELEKSI

- Suatu komponen seleksi mempunyai dua atau lebih bagian, dengan satu, dan hanya satu, sekali terjadi untuk setiap kejadian dari komponen seleksi.
- Pada gambar 3 diberikan sebuah gambaran dari diagram dan notasi struktur text dari komponen seleksi A. A mempunyai bagian-bagian B, C, dan D. Lingkaran dalam kotak B, C, dan D menunjukkan bahwa A adalah sebuah seleksi sedangkan B, C, dan D adalah komponen-komponennya.



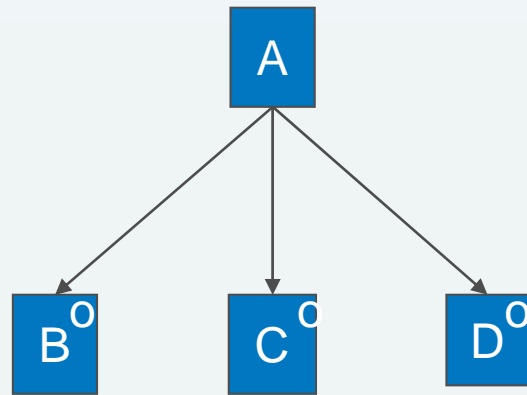
SELEKSI

- Ada juga kemungkinan untuk menunjukkan kondisi dari suatu seleksi dalam sebuah struktur teks biasa untuk keadaan seperti pada bagian (i) dan (iii) dari gambar 3b. Kasus khusus dari seleksi adalah "Seleksi Nol". Artinya "tidak melakukan apa-apa" dan digambarkan dengan `_`. Pada contoh gambar 4 diperlihatkan A hanya mempunyai satu komponen seleksi, yakni B.



SELEKSI

Selection



(a)

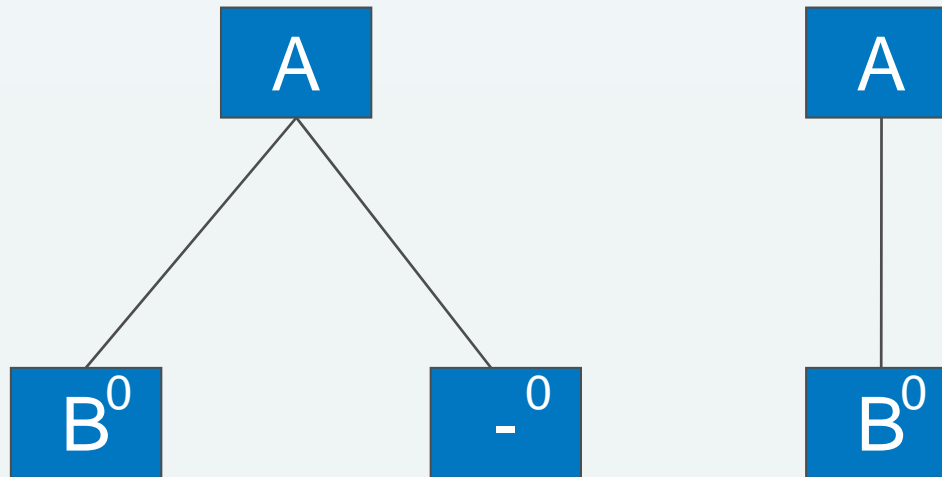
Struktur Teks

A sel	A sel<kondisi 1>	A sel (kondisi 1)
p;	p;	p;
a alt	A alt<kondisi 2>	A sel (kondisi2)
C;	C;	c;
A alt	A alt<kondisi 3>	A sel (else)
D';	D';	0;
A end	A end	A end
(i)	(ii)	(iii)

(b)

Gambar 3. Penggambaran Komponen Seleksi

SELEKSI

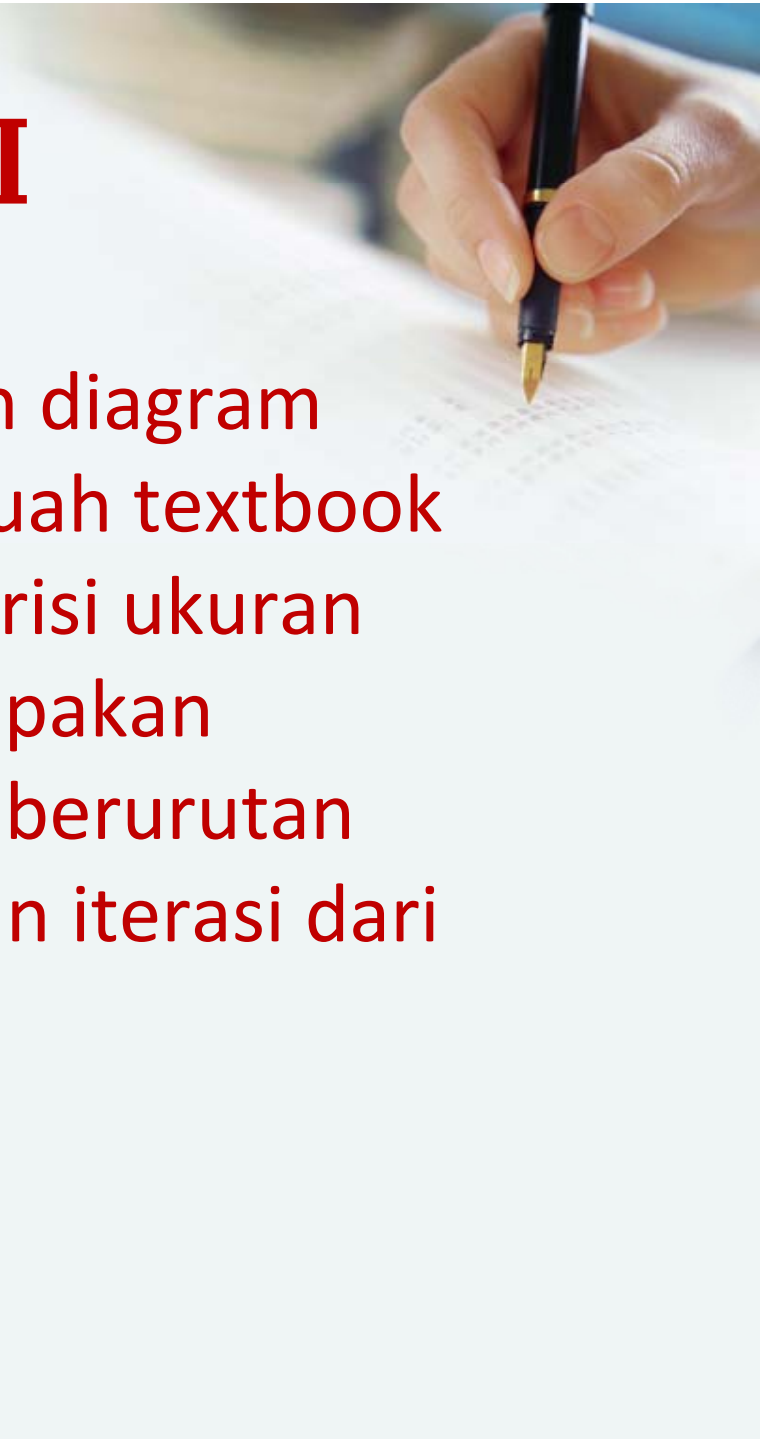


Gambar 4. Contoh Sebuah Seleksi Nol

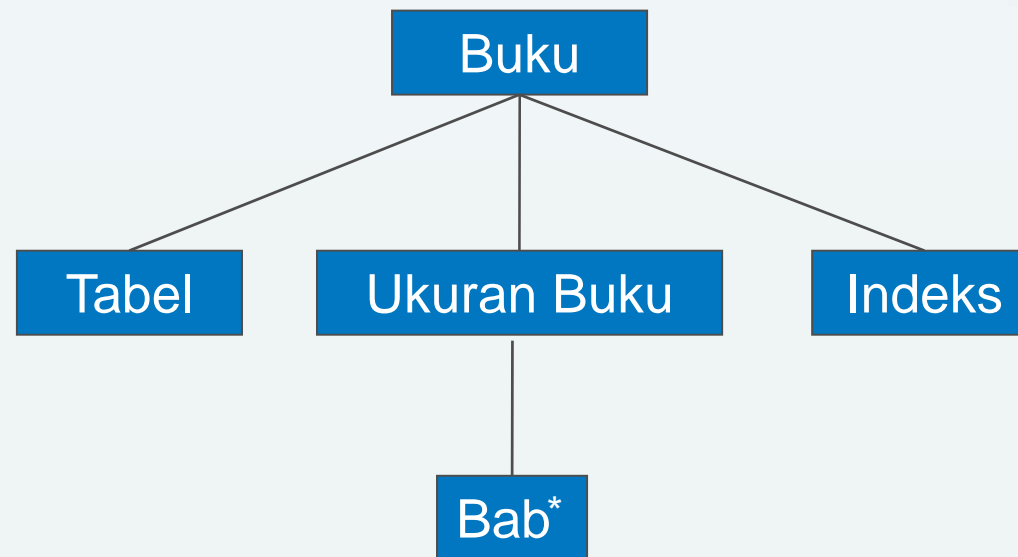


CONTOH ITERASI

- Contoh iterasi dalam sebuah diagram Jackson adalah struktur sebuah textbook dalam sebuah tabel yang berisi ukuran buku, dan indeks yang merupakan komponen-komponen yang berurutan sedangkan bab adalah bagian iterasi dari bentuk buku (gambar 5).



CONTOH ITERASI



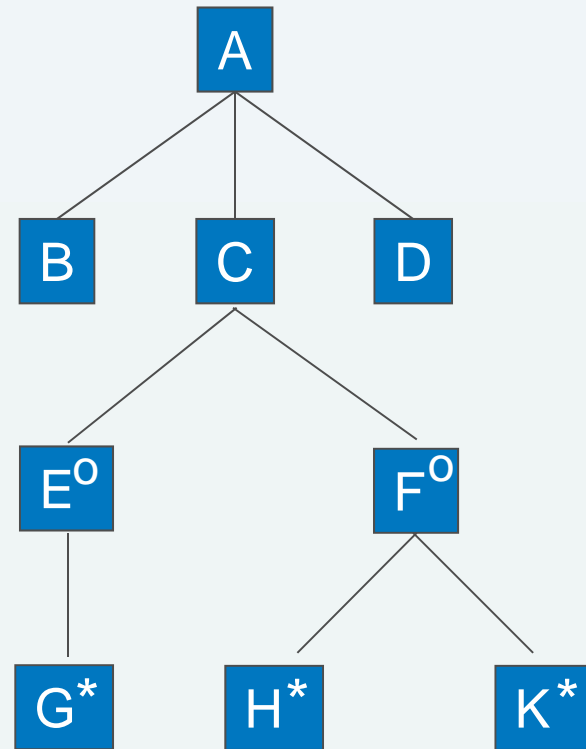
Gambar 5. Contoh Iterasi Dalam Diagram Jackson

CONTOH ITERASI

- Susunan komponen-komponen dalam hirarki diperlihatkan pada gambar 6. Pada gambar tersebut, A adalah suatu komponen deret, sedangkan B, C, dan D adalah bagian-bagian dari A. C adalah komponen seleksi; E dan F adalah bagian-bagiannya.
- E dan F adalah komponen-komponen iterasi.
- F juga merupakan suatu komponen deret dengan H dan K adalah bagian-bagiannya.



CONTOH ITERASI



Gambar 6. Struktur Dasar Dalam Diagram Jackson



W/O

WARNIER OR



PENDAHULUAN

- Metodologi pengembangan sistem ini pertama kali dikembangkan tahun 1970 oleh *Jean Domique Warnier* dari Paris. Kemudian dikembangkan lebih lanjut untuk desain sistem oleh *Ken Orr* dari Kansas
- Metodologi W/O menggunakan alat yang disebut dengan diagram W/O, yang secara selintas mirip dengan bagan berjenjang yang diputar.



OPERATOR DIAGRAM W/O

SIMBOL	ARTI
\oplus	XOR (exclusive OR) yaitu A atau B tetapi tidak kedua-duanya
+	OR (inclusive OR) yaitu A atau B atau kedua-duanya
/, *, -, +	Operator matematika
<u> </u> Proses	NOT



STRUKTUR DATA MENGUNAKAN DIAGRAM W/O

- Prinsip kunci dari metodologi W/O adalah desain dari struktur program yg tertulis dilengkapi dengan struktur datanya.
- Diagram W/O dapat menggambarkan struktur data yg berbentuk :
 1. Struktur Data Urut
 2. Struktur Data Repetisi
 3. Struktur Data Seleksi

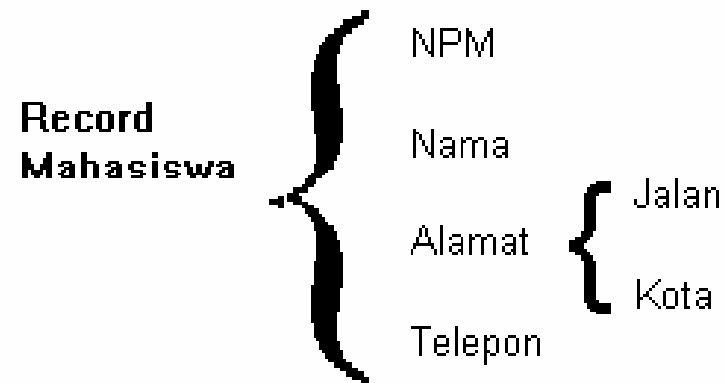


STRUKTUR DATA URUT

- Misalnya suatu record mahasiswa sbb :

		ALAMAT		
NPM	NAMA	JALAN	KOTA	TELEPON

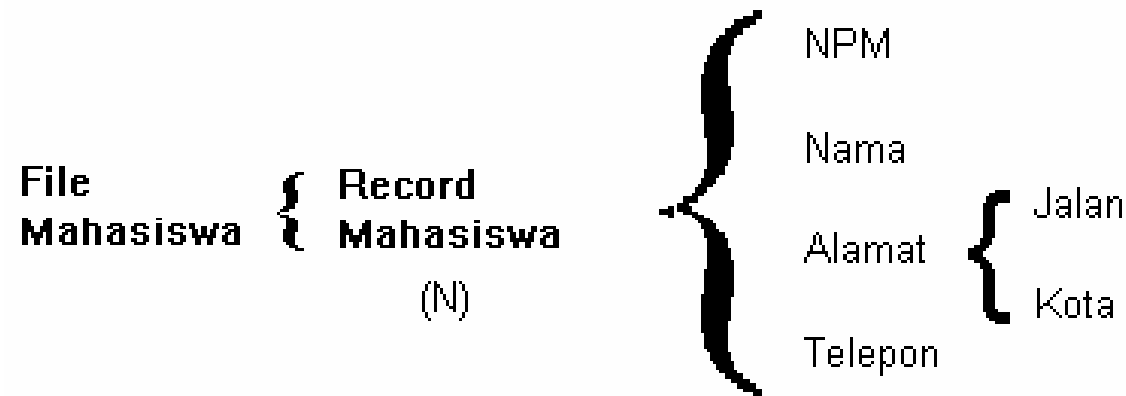
- Maka dapat digambarkan dengan diagram W/O:





STRUKTUR DATA REPETISI

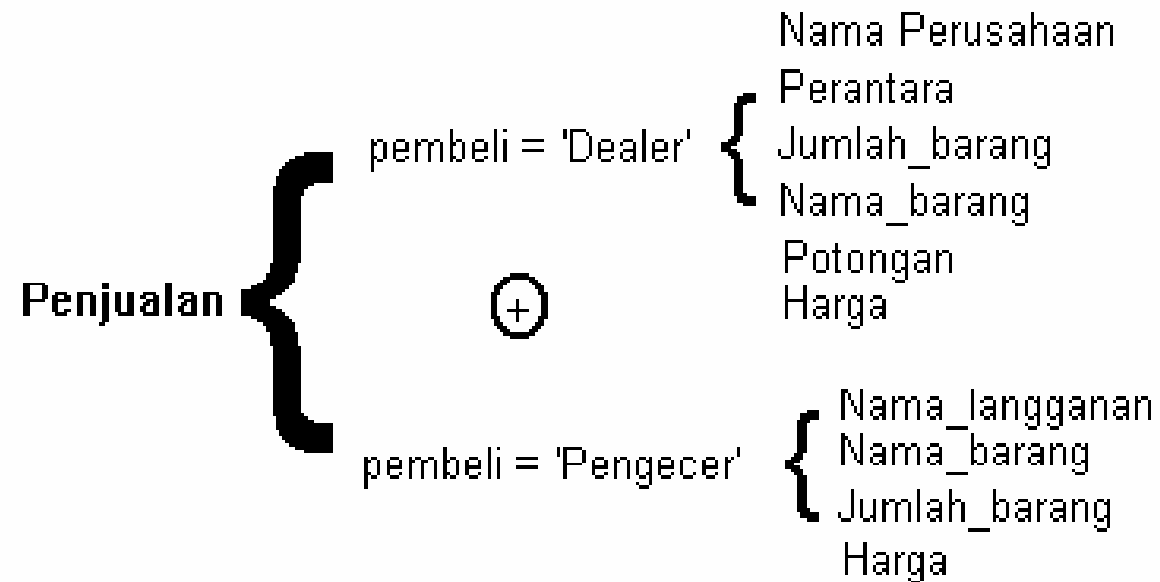
- Misalnya file mahasiswa terdiri dari n record,
- Maka dapat digambarkan dengan diagram W/O:





STRUKTUR DATA SELEKSI

- Dapat ditunjukkan oleh operator XOR sbb

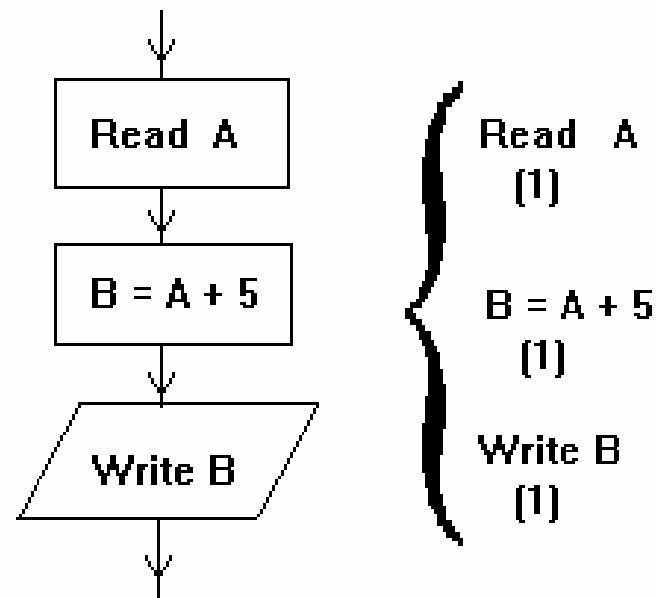




STRUKTUR PROSES PROGRAM MENGGUNAKAN DIAGRAM W/O

1. STRUKTUR PROSES URUT

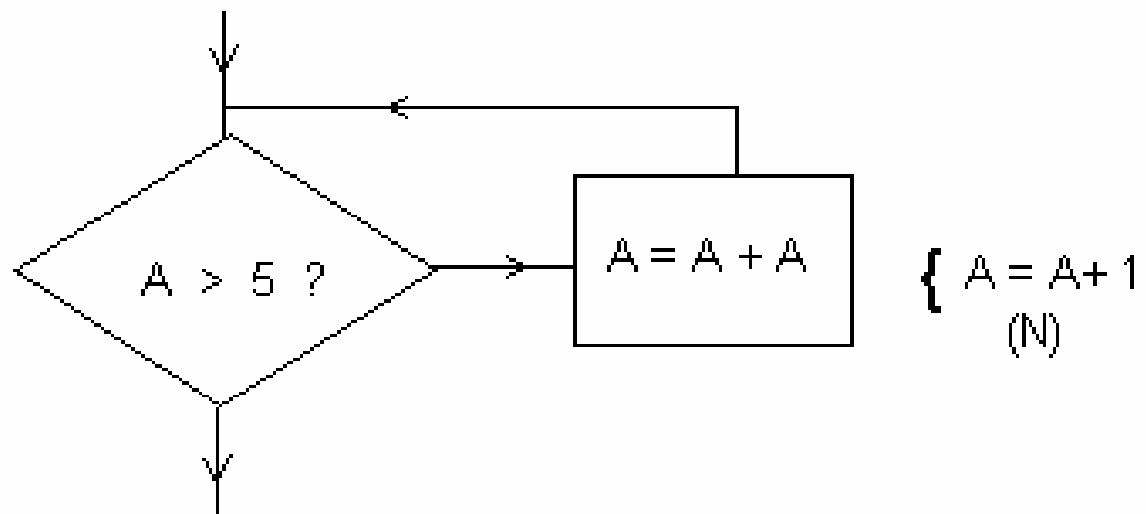
Struktur Proses Urut adalah masing - masing instruksi diproses urut satu dengan yang lainnya.



STRUKTUR PROSES PROGRAM MENGGUNAKAN DIAGRAM W/O

2. STRUKTUR PROSES REPETISI

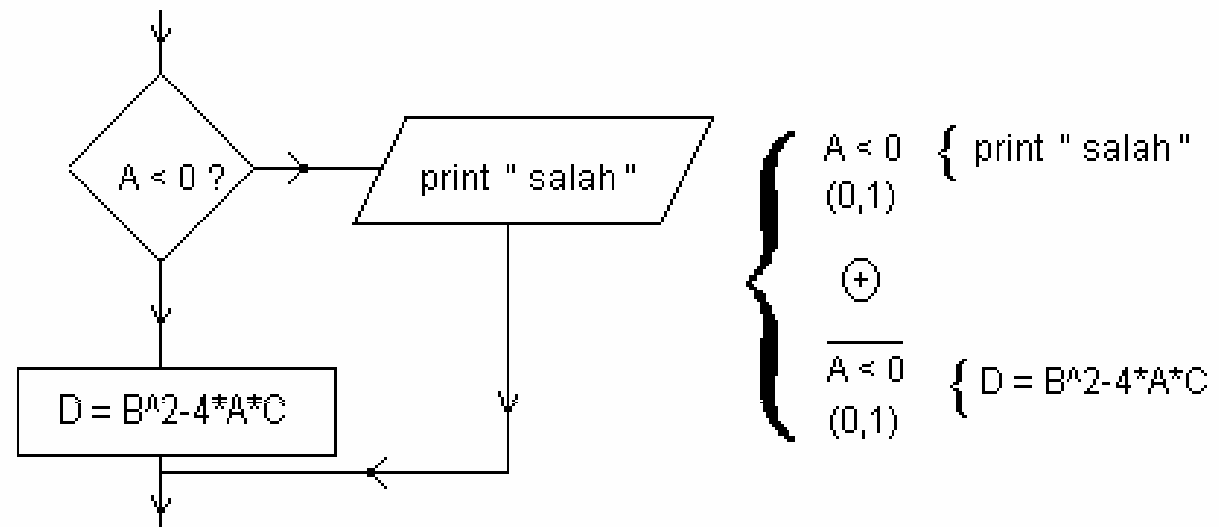
Struktur Proses Repetisi adalah struktur proses yg mengulang instruksi program berulang kali.



STRUKTUR PROSES PROGRAM MENGGUNAKAN DIAGRAM W/O

3. STRUKTUR PROSES SELEKSI

Struktur Proses Seleksi merupakan struktur proses yang menggunakan instruksi penyeleksian kondisi.

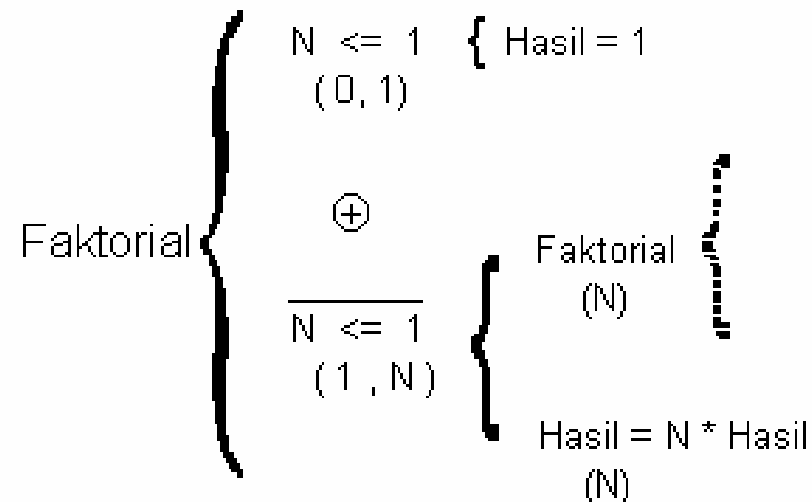




STRUKTUR PROSES PROGRAM MENGGUNAKAN DIAGRAM W/O

4. STRUKTUR PROSES REKURSI

Struktur Proses Rekursi adalah suatu proses dari suatu fungsi atau prosedur yang memanggil dirinya sendiri. Digambarkan dengan kurung kurawal (brace) yang terputus - putus.





STRUKTUR PROSES PROGRAM MENGGUNAKAN DIAGRAM W/O

- **Keterangan Simbol Diagram W/O tsb**

Simbol Angka diantara tanda kurung menunjukkan banyaknya proses

- * Untuk Urut maka instruksi biasanya dikerjakan 1 kali sehingga dituliskan (1).
- * Untuk Repetisi maka instruksi bisa dikerjakan sebanyak N kali maka dituliskan (N).
- * Untuk Seleksi disimbolkan dengan (0,1) yaitu suatu instruksi akan diproses 1 kali (1), atau tidak diproses (0).
- * Untuk rekursi bisanya disimbolkan dengan (1,N) , N untuk repetisinya