

Data Dictionary

(Kamus Data & Spesifikasi Proses)



Analisis dan Perancangan Sistem Informasi

Erik Kurniadi erik@uniku.ac.id

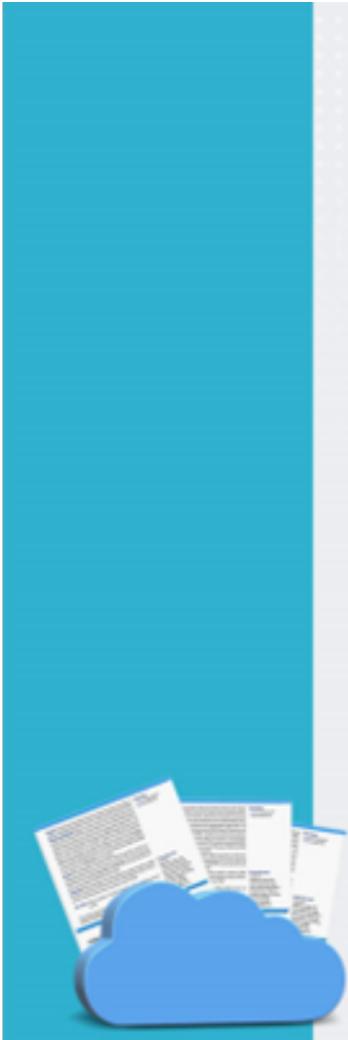
Data Disctionary (Kamus Data)



Data dictionary atau Kamus Data adalah suatu daftar terorganisasi tentang komposisi elemen data, aliran data, dan data store yang digunakan dalam DFD

Dibuat bersamaan dengan pembuatan DFD pada saat tahap analisis sistem berjalan.

Pengisian kamus data dilakukan setiap saat selama proses pengembangan berlangsung, ketika diketahui adanya data item atau saat diperlukan penambahan data item ke sistem



Kamus data berisi :

Elementary Data Element / Elementary Data

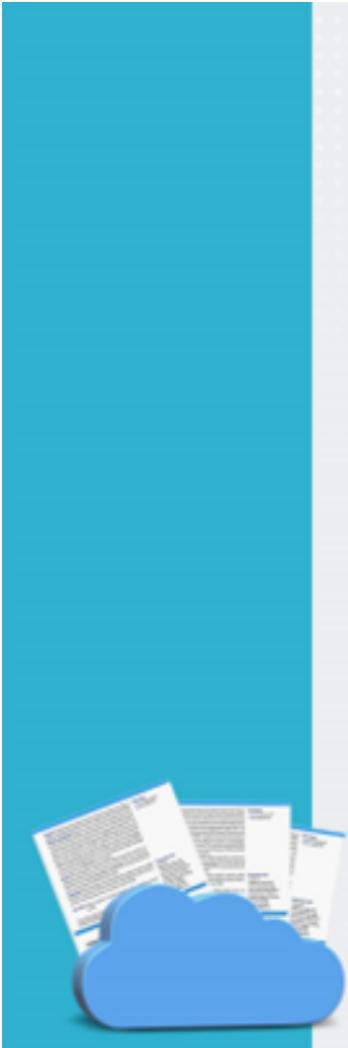
- yaitu elemen data tingkat rendah, tidak dapat diuraikan lebih lanjut, dapat berupa:
- data item, (NIM, Jns_kelamin, Lmt_kredit_pelanggan)
- Data literal, (NOSMOKE-MSG='Thank You for not smoking')

Group Data Element / Composite Data

- Merupakan gabungan elementary data elemen
- Contoh : Alamat terdiri dari : Nama_jalan, No_rumah, kota, kode_pos

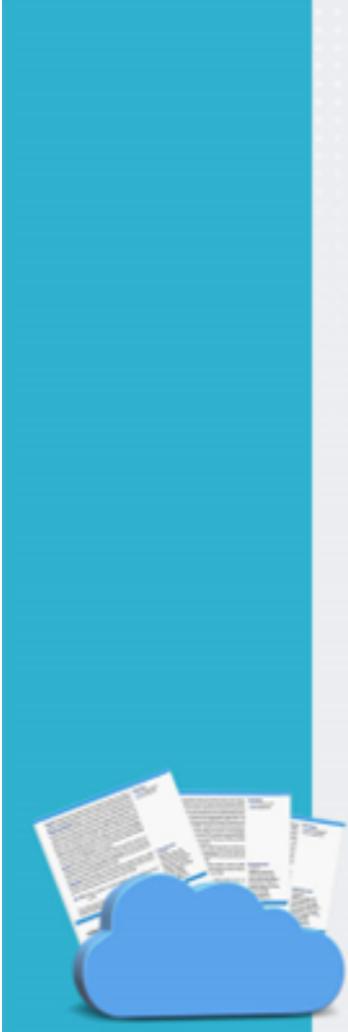
Data Store

Yaitu gabungan beberapa elementary data element atau group data element yang berhubungan



Simbol yang digunakan dalam kamus data

	Simbol	Keterangan
Assign	=	Terdiri dari. Cont: pelanggan = kode_pel.....
Concatenation	+	Penggabungan element data yang satu dengan yg lain Alamat= nama_jln+no+kota+kd_pos
Choice	[..]	Pilih satu dari beberapa alternatif Status_Rumah = [Rumah_sendiri sewa kontrak keluarga]
Iteration	{..}	Attribute Muncul berulang kali/ pengulangan elemen data. Cont :Karyawan = nama+Nama_Pasangan+ $^0\{nama_anak\}_n$
option	(...)	Menyatakan data tambahan, boleh ada boleh tidak. Cont : Karyawan = Nama+alamat+ (telpon)
Comment	*....*	Menyatakan penjelasan. Cont : Alamat=Jalan+Nomor+kota_kode_pos *alamat kantor*



Pada **simbol repetition** boleh dicantumkan **jumlah minimum dan maksimum** pengembangan data tersebut :

- Karyawan = nama+nama_pasangan+ $^0\{nama_anak\}_n$

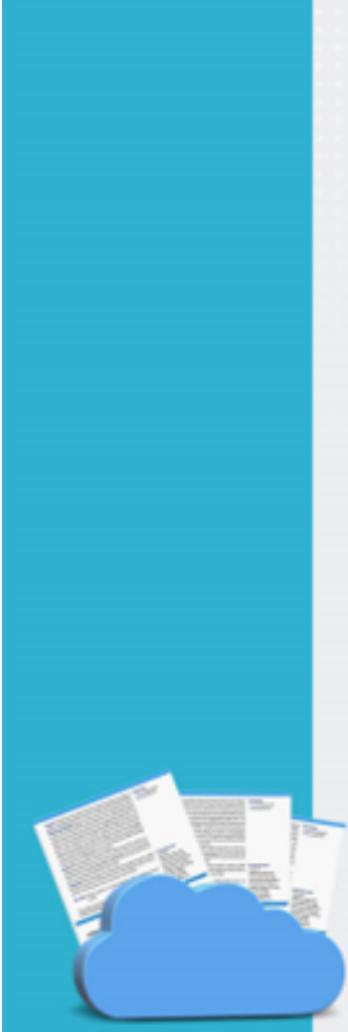
Candidate Key, menyatakan data elemen (satu atau beberapa) yang memiliki 'unique value' untuk setiap entry

- MST_MHS=NIM+Nama+Alamat+..+NopPen

NIM -> candidate key

Nama+Alamat -> candidate key

NopPen -> candidate key



Primary Key, menyatakan candidate key yang dipilih, ditandai dengan garis bawah atau @. Key hanya digunakan untuk data store.

- MST_MHS=NIM+Nama+Alamat+..+NopPen
Atau
- MST_MHS=@NIM+Nama+Alamat+..+NopPen



Data Tabel

Data Tabel kumpulan dari data yang terorganisasi berdasarkan key yang berada di dalam sistem.

Definisi data tabel :

- Satu Dimensi

File_MHS = {NIM+Nama+Alamat+telpon}

NIM	Nama	Alamat	Telpon
0211311	Joni	Mawar 21	455456
0211312	Jojon	Melati 22	454557
0211313	Jontor	Mangga 23	568544

– Dua Dimensi

File_MHS = {Jurusan+{NIM+Nama+Alamat
+telpon}}

Jurusan	NIM	Nama	Alamat	Telpon
Manajemen	0211311	Joni	Mawar 21	455456
	0211312	Jojon	Melati 22	454557
	0211313	Jontor	Mangga 23	568544

Jurusan	NIM	Nama	Alamat	Telpon
Teknik	0211211	Oon	Pisang 21	445545
	0211212	Oneng	Duren 22	446654
	0211213	Ombo	Duku 23	448875

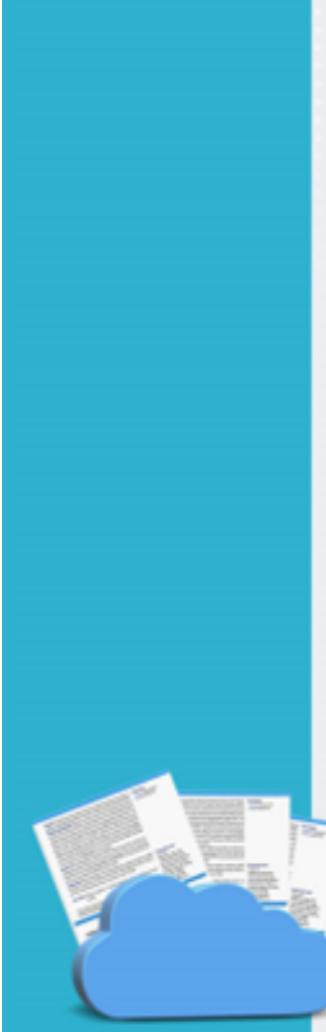


- Tiga Dimensi

File_MHS = {PTS + {Jurusan+{NIM+Nama+Alamat+telpon}}}

PTS	Jurusan	NIM	Nama	Alamat	Telpon
STIKI	Teknik	0211211	Odon	Pisang 21	445545
		0211212	Oneng	Duren 22	446654
		0211213	Ombo	Duku 23	448875





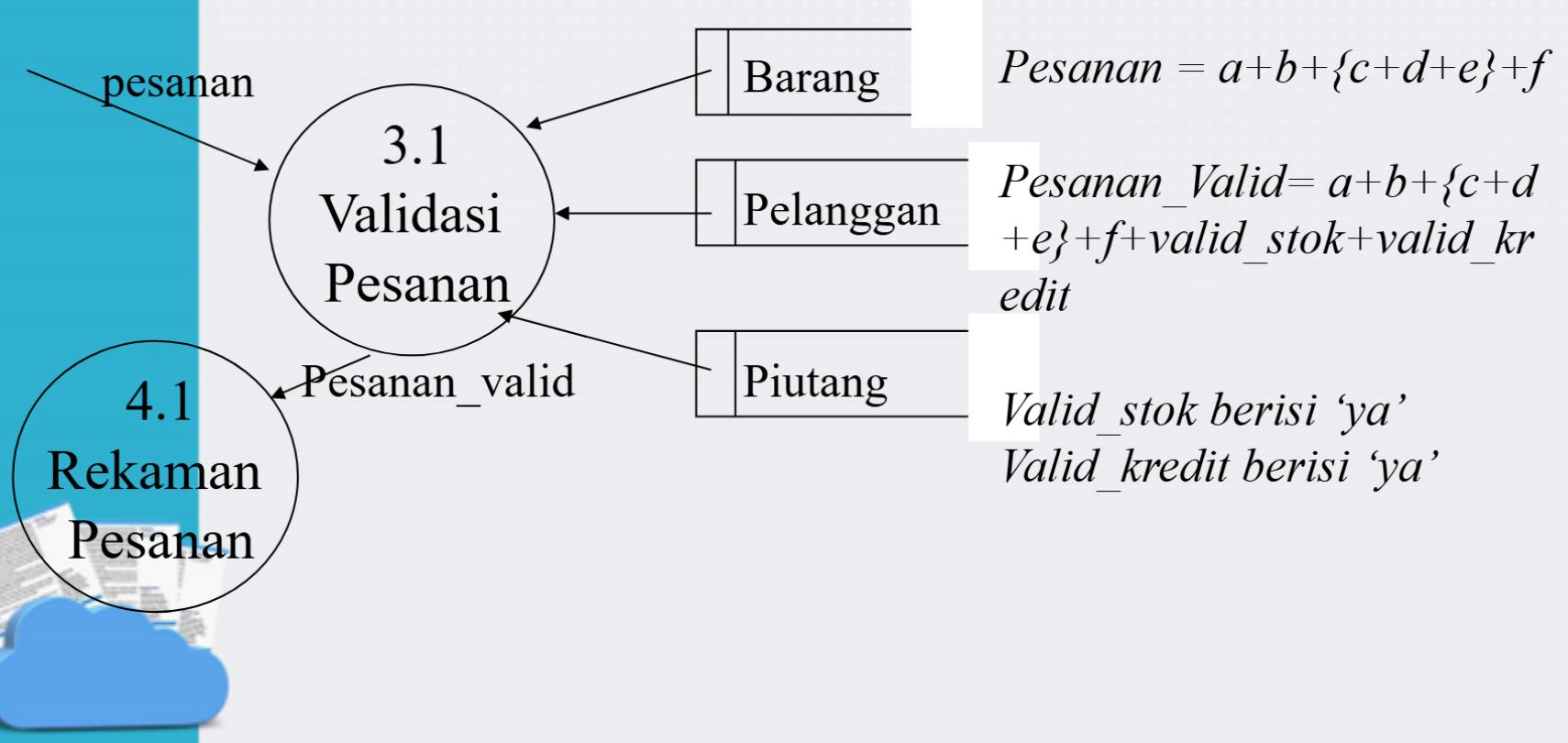
Alias, menyatakan nama lain dari suatu **data elemen** atau **data store** yang sebenarnya sama dengan data elemen atau data store yang telah ada

Terjadi karena kurang koordinasi antara beberapa analis sistem

- Analis sistem yang 1 menggunakan Employee
- Analis sistem yang 2 menggunakan karyawan
- Tetapi keduanya sama

Penggunaan alias sebaiknya dihindari !

Aliran data harus dikamuskan :



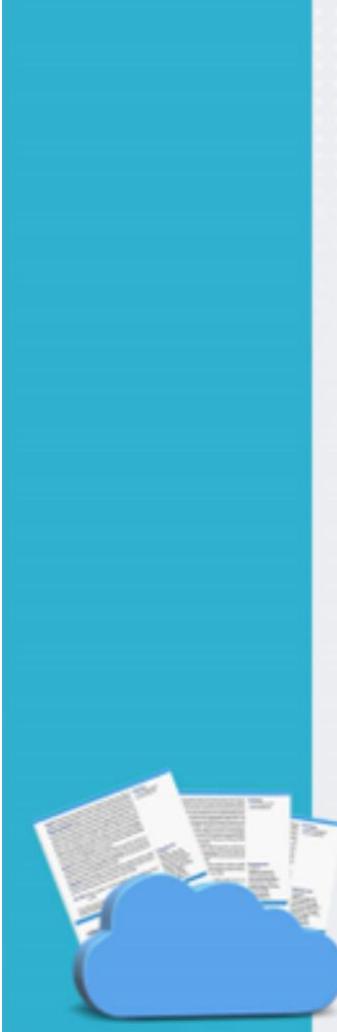
Proses Spesification

Setiap proses (bubble) pada DFD harus memiliki 'proses spesification'. => untuk mengetahui apa yang terjadi didalam proses tersebut.

Spesifikasi proses menjadi pedoman bagi programmer dalam membuat program (coding)

Metode yang dipakai untuk menggambarkan proses pada top level berbeda dengan metode yang digunakan untuk proses yang ada pada level yang paling bawah. Pada level atas dapat menggunakan kalimat deskriptif, pada level yang paling bawah (functional primitive) membutuhkan spesifikasi yang lebih terstruktur.





Metode yang digunakan dapat berupa:

- Narasi, uraian proses dalam bentuk 'cerita'
- Bhs Indonesia/Inggris yang terstruktur
- Decision Tree (Pohon keputusan)
- Decision Table (Table Keputusan)

Narasi

Penjelasan proses dalam bentuk kalimat:

'untuk setiap pembelian dari pelanggan harap dip eriksa: apabila batas kredit dilampaui maka periks a histori pembayarannya. Apabila historinya kurang baik maka ditolak. Bila historinya baik & pembeliannya tidak melebihi 1 jt maka minta persetujuan manajer, tetapi bila pembelian diatas 1 jt maka dit olak'



Bhs. Indonesia Terstruktur

Instruksi proses dengan bhs.Indonesia:

- Jika maka
- (seperti aturan logika pemrograman)
 - Buka file
 - Baca File
 - Baca File Berdasarkan ...
 - Tulis File ...
 - Tutup File ...
 - Hitung ...
 - Cetak ke layar ...
 - Tampil ke layar ...

Instruksi tidak multak (kaku), yang penting dapat dimengerti.



Contoh Bahasa Indonesia

Modul Validasi_Penjualan

Jika saldo_piutang > limit_kredit

Maka

Jika histori_bayar = buruk maka tolak

Selain itu

jika nilai_barang <= 1 juta maka minta acc manager

selain itu

tolak

akhir jika

akhir jika

selain itu

pembelian_dilayani

Akhir jika

Akhir Modul





Structured English

Instruksi proses dengan bhs.inggris

- If then else
- While do
- Repeat until
- Do case
 - Open file
 - Read file
 - Write file
 - Rewrite file
 - Searching file For ...
 - Locate file .. For ...
 - Print
 - Display
 - Input ..
 - Calculate ..
 - Compute ...

Decision Table

kondisi

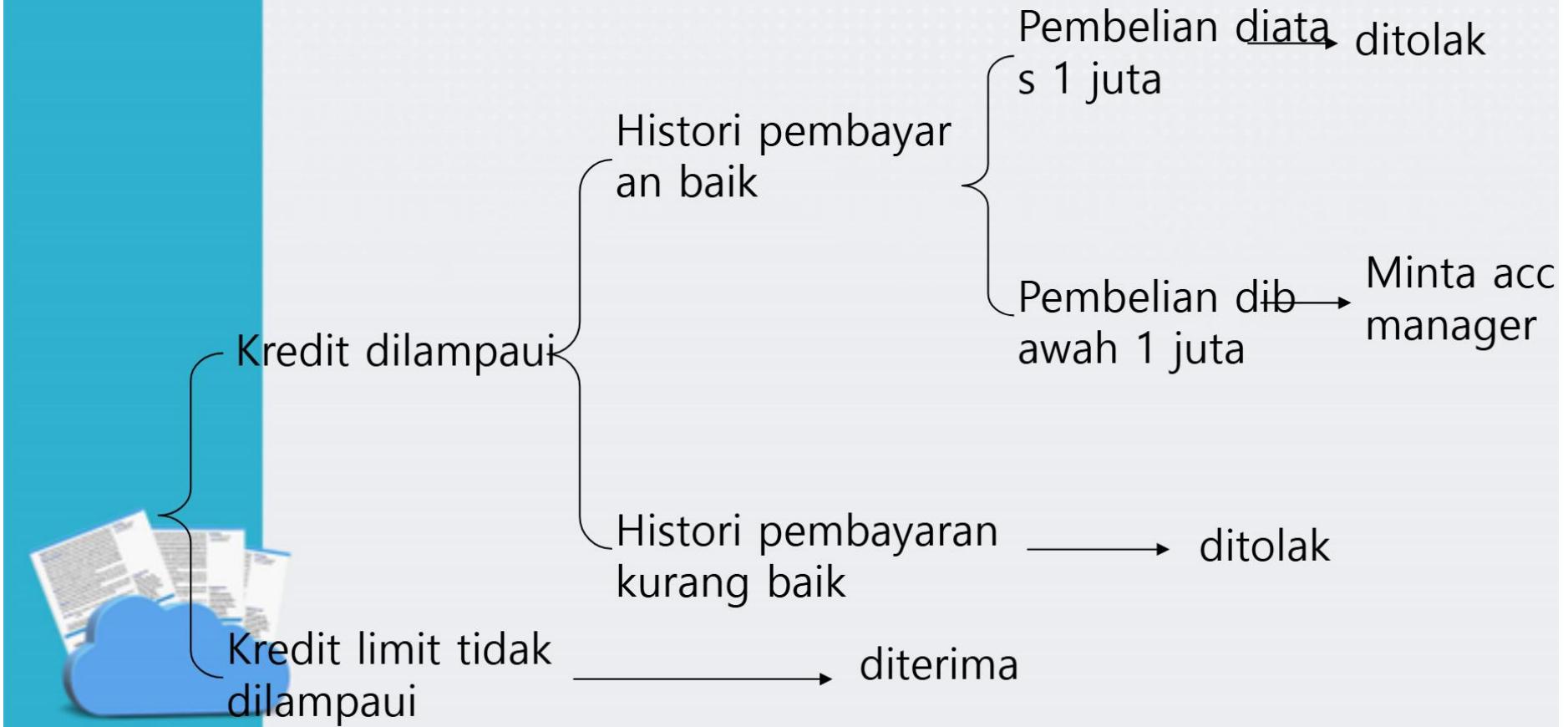
aksi

aturan

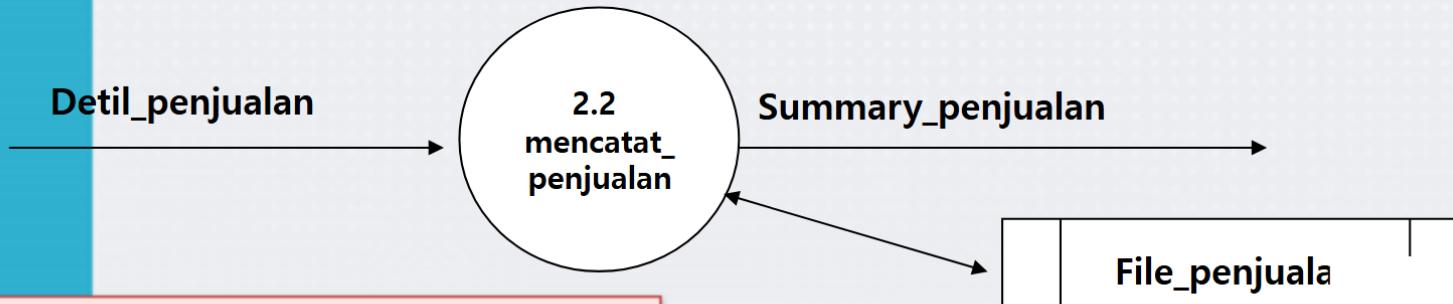
Kredit limit dilampaui	Y	Y	Y	Y	N	N	N	N
Histori pembayaran baik	Y	Y	N	N	Y	Y	N	N
Pembelian diatas 1 juta	Y	N	Y	N	Y	N	Y	N
Pembelian diterima					X	X	X	X
Tolak		X		X	X			
Minta persetujuan manager			X					



Decision Tree



Contoh spesifikasi proses



Detail_Penjualan :
Cust+part_no+Qty_sold+Unit_price

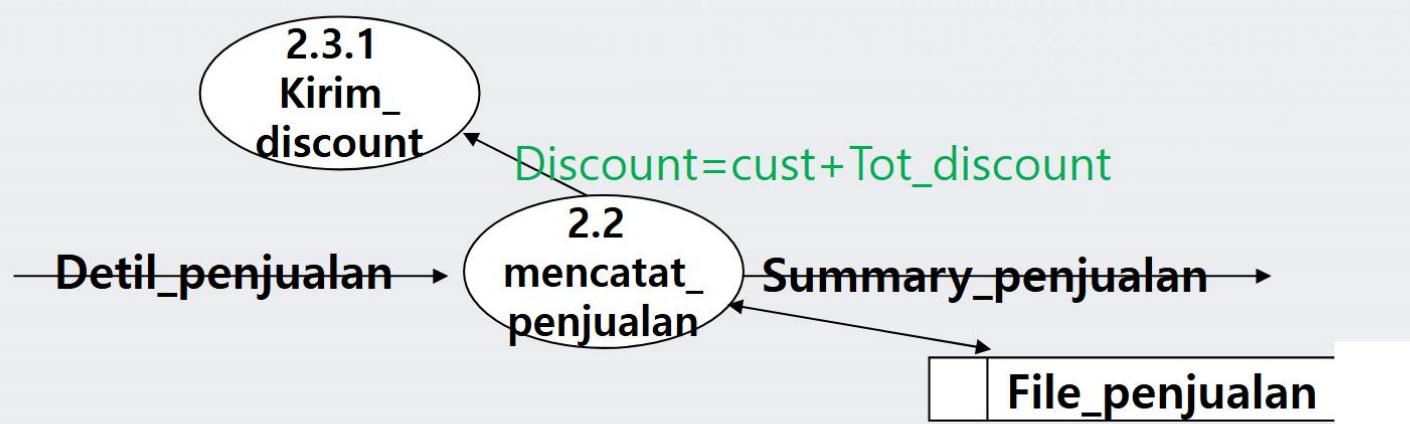
File_penjualan :
Part_no+Total_Qty+Total_value

Summry_penjualan :
cust+sale_value



MODULE mencatat_penjualan
receive'detil penjualan'
get record in 'file penjualan' for part_no in detail_penjualan
Total_Qty=Total_Qty+sold_Qty
Sale_Value=Qty_Sold+Unit_Price
Total_Value=Total_Value+Sale_Value
Write 'File Penjualan' record
Send 'summary Penjualan'
ENDMODULE

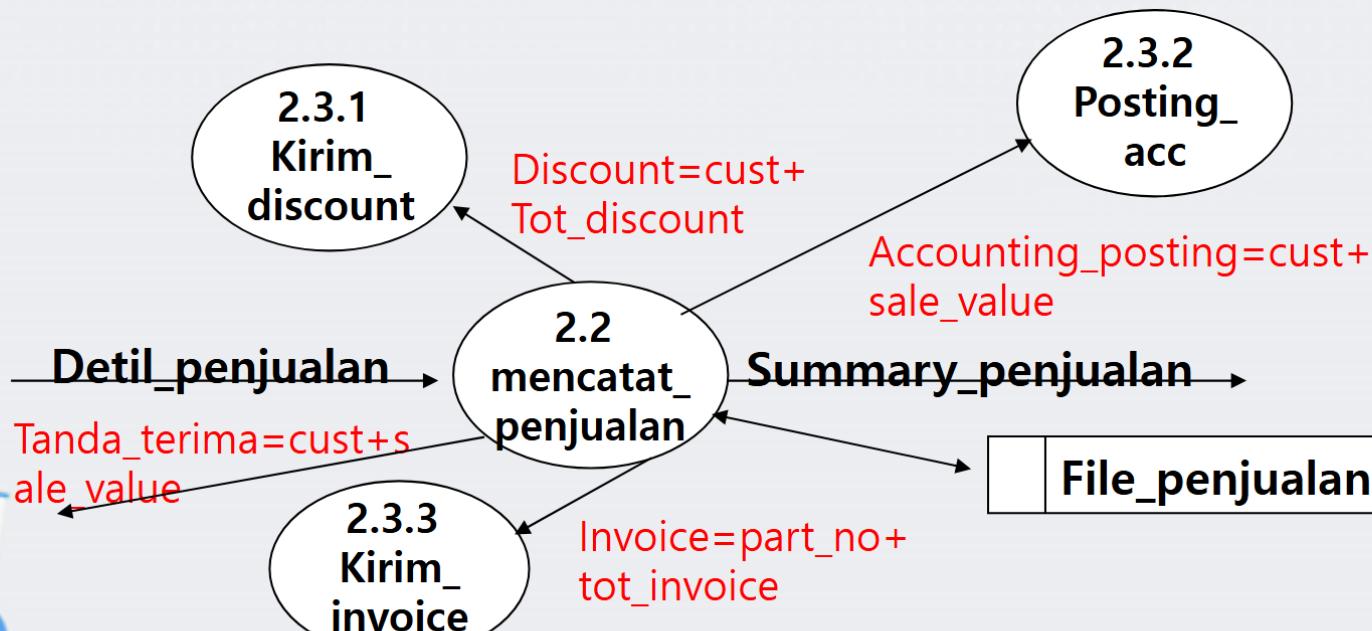
Pada contoh sebelumnya ditambah kondisi : bila penjualan diatas >5 juta maka diberi discount sebesar 10%





```
MODULE mencatat_penjualan
receive'detil penjualan'
get record in 'file penjualan' for part_no in detail_penjualan
    Total_Qty=Total_Qty+sold_Qty
    Sale_Value=Qty_Sold+Unit_Price
    Total_Value=Total_Value+Sale_Value
    Write 'File Penjualan' record
    Send 'summary Penjualan'
    If sale_value>5 juta
        discount = sale_value * 0.10
        send 'discount'
    endif
ENDMODULE
```

Pada contoh sebelumnya ditambah kondisi lagi: bila pembelian dengan tunai atau cek akan diberi tanda terima sedangkan untuk pembelian dengan kredit akan dikirim invoice dan bila pembelian menggunakan fasilitas "account" akan diposting. Discount hanya diberikan untuk fasilitas 'account'



Terima Kasih

