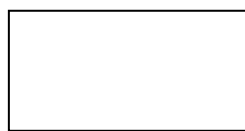


Bagan Terstruktur (*Structured Chart*)

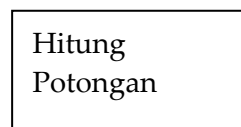
I. Pendahuluan

- Mirip dengan bagan berjenjang (*hierarchy chart*).
- Digunakan untuk mendefinisikan dan mengilustrasikan organisasi sistem informasi secara berjenjang dalam bentuk modul dan sub modul.
- Bedanya : bagan tersrtuktur juga menunjukkan hubungan elemen data dan elemen kontrol antara hubungan modulnya.
- Memberikan penjelasan sistem yang lengkap dipandang dari elemen data, elemen kontrol, modul, dan hubungan antar modulnya.

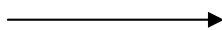
II. Simbol-simbol Dasar



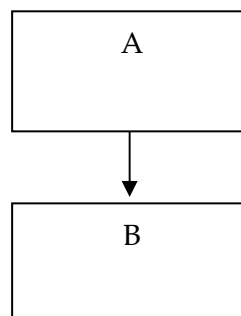
Module →
simbol ini
menunjukkan
suatu modul



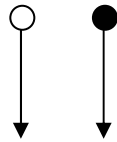
menunjukkan
suatu modul
dengan nama
"Hitung
Potongan"



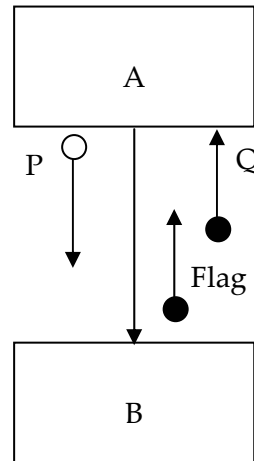
Connection →
simbol ini
digunakan
untuk
menghubungkan
suatu modul
dengan modul
lainnya.



Modul A
memanggil modul
B. Setelah proses
modul B selesai,
maka proses
kembali ke modul A
yang
memanggilnya.



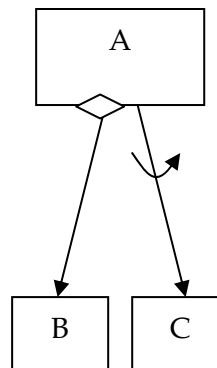
Couple → simbol ini menunjukkan suatu data atau elemen kontrol yang dikirimkan dari suatu modul ke modul lainnya. Panah dgn lingkaran kosong menunjukkan data yang dikirimkan sedangkan panah dengan lingkaran hitam menunjukkan elemen kontrol yang dikirimkan.



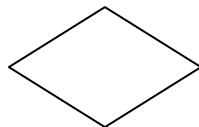
Modul A memanggil modul B dan elemen data P dikirimkan dari modul A ke modul B. Hasil proses dari modul B, mengirimkan data Q dan elemen kontrol flag ke modul A.



Loop → simbol ini menunjukkan suatu perulangan di dalam modul.



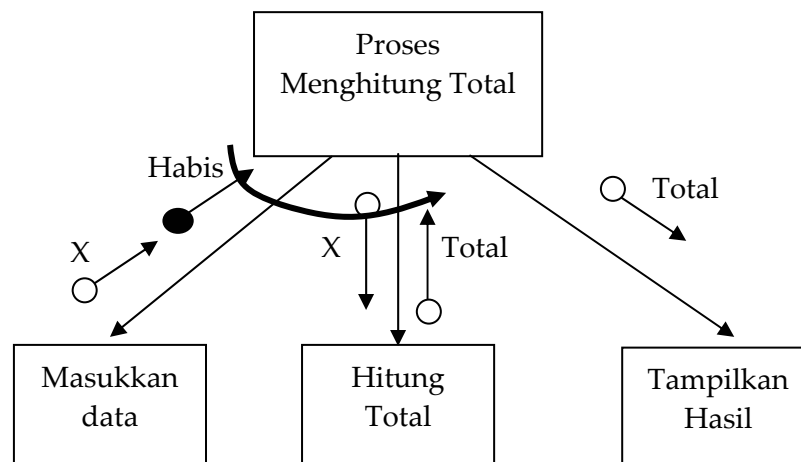
Modul A memanggil modul B bila kondisi yang diseleksi di modul terpenuhi. Modul A juga memanggil modul C berulang kali yang ditunjukkan oleh simbol perulangan.



Decision → simbol ini menunjukkan suatu penyeleksian kondisi di dalam modul.

III. Perulangan Di Bagan Terstruktur

- Ditunjukkan oleh simbol panah yg melingkar.
- Banyaknya perulangan tidak terlihat secara eksplisit dalam bagan terstruktur, namun ditunjukkan oleh proses yang terjadi di modul letak perulangan tersebut terjadi.



Gambar 7.1. Proses Perulangan Di Bagan Terstruktur

- Modul-modul yang akan diproses berulang kali dari modul PROSES MENGHITUNG TOTAL adalah modul MASUKKAN DATA dan HITUNG TOTAL.
- Banyaknya perulangan yang dilakukan adalah sampai dengan modul MASUKKAN DATA mengirimkan elemen kontrol HABIS yang berisi status bahwa sudah tidak ada data lagi yang akan dimasukkan.
- Kemudian akan dilanjutkan dengan memanggil modul TAMPILKAN HASIL .
- Dalam bahasa BASIC, proses tersebut tampak sebagai berikut :

```

10  Proses Menghitung Total
20  HABIS$="T"
30  WHILE HABIS$="T" OR HABIS$="t"
40      GOSUB 100                'Masukkan Data
50      GOSUB 200                'Menghitug Total
60  WEND
70  GOSUB 300                    'Tampilkan Hasil
80  END
90  '
100 'Subroutine Masukkan Data
110 INPUT "Nilai X " ; X
120 INPUT "Data sudah habis (Y/T)?" ; HABIS$
130 PRINT
140 RETURN
150 '
  
```

```

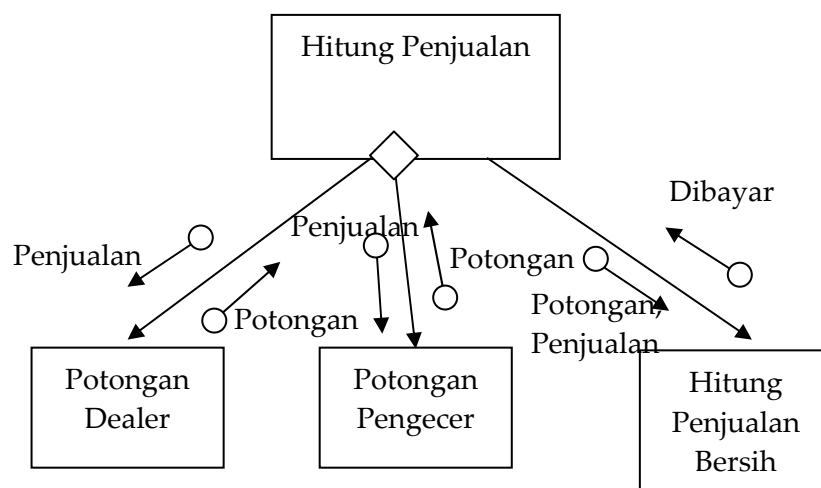
200  `Subroutine HITUNG TOTAL
210  TOTAL=TOTAL+X
220  RETURN
230  `
300  `Subroutine TAMPILKAN HASIL
310  PRINT
320  PRINT "Nilai Total = ";TOTAL
330  RETURN

```

IV. Keputusan di Bagan Tersruktur

Seringkali dalam suatu modul terdapat penyeleksian kondisi yang akan membuat keputusan mengenai kegiatan-kegiatan yg akan dilakukan.

Contoh : Jika langganan adalah dealer maka kerjakan modul Potongan-Dealer.
Selainnya kerjakan modul Potongan-Pengecer



Gambar 7.2. Proses Keputusan Di Bagan Terstruktur

Setelah mengerjakan modul Potongan-Untuk-Dealer atau Potongan-Untuk-Pengecer, maka proses akan kembali ke modul yang memanggilnya, yaitu modul HITUNG PENJUALAN dan dilanjutkan memanggil modul HITUNG PENJUALAN BERSIH.

Dalam bahasa Pascal keputusan di modul HITUNG PENJUALAN tampak sebagai berikut :

```
If JenisLangganan = 'Dealer' Then
    PotonganDealer(Penjualan,Potongan)
Else
    PotonganPengecer(Penjualan,Potongan);

HitungPenjualanBersih(Penjualan,Potongan,Dibayar);
Writeln('Penjualan bersih = ', Dibayar);
```

Sedang modul-modul yang lain tampak sebagai berikut :

```
Procedure PotonganDealer(Penjualan: real;var Potongan:real);
Begin
    Potongan := 0.25 * Penjualan;
End;

Procedure PotonganDealaer(penjualan : real; var Potongan
:real);
Begin
    Potongan := 0.10 * Penjualan;
End;

Procedure HitungPenjualanBersih(Penjualan,Potongan: real;
var Dibayar:real);
Begin
    Dibayar := Penjualan - Potongan'
End;
```

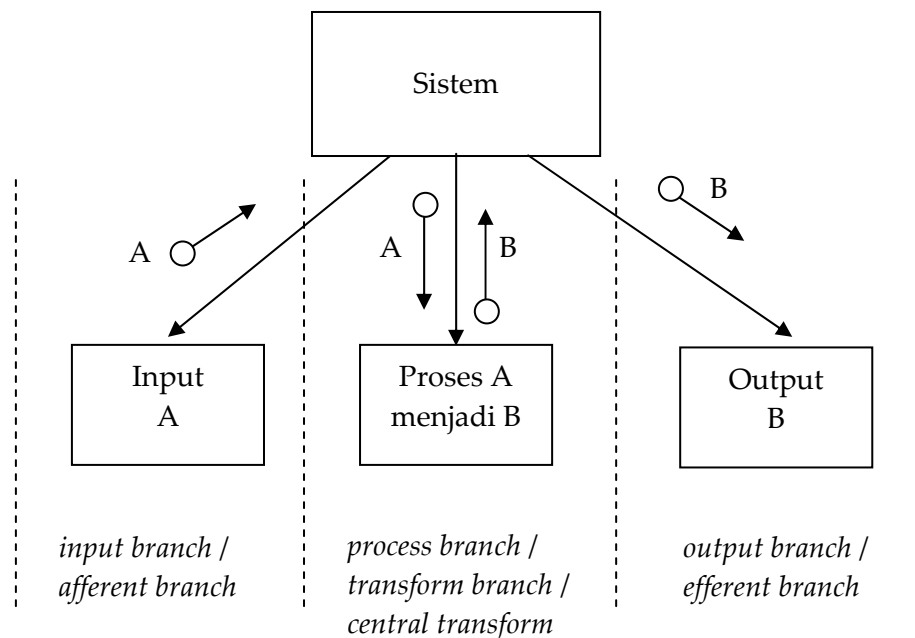
V. Model Bagan Terstruktur

Terdapat dua model bagan terstruktur, yaitu *transformed-centered* dan *transaction-centered*. Model suatu bagan terstruktur dapat berbentuk salah satu dari keduanya atau kombinasi. Model yang akan digunakan tergantung dari diagram arus data yg telah dibuat.

Transformed-centered

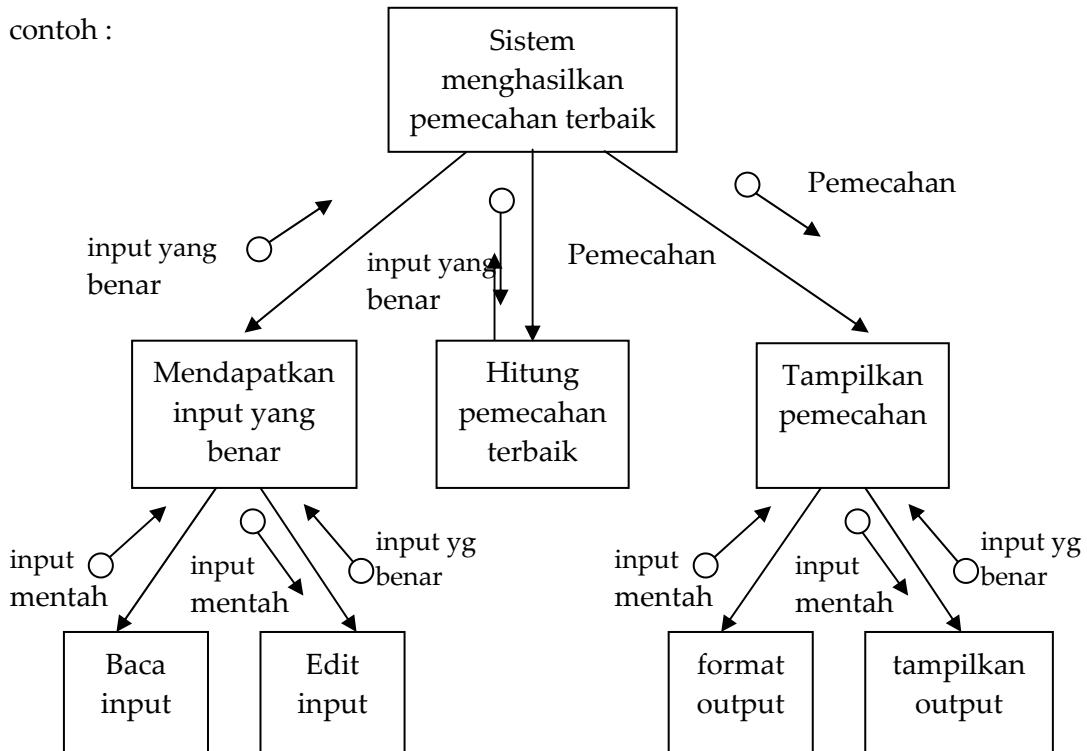
Model ini menggambarkan sistem dalam 3 cabang utama, yaitu :

- Cabang input (*input branch / afferent branch*) : cabang yang menerima input dan membentuk input ke dalam suatu status yang siap untuk diproses.
- Cabang proses (*process branch / transform branch / central transform*) : cabang yang melakukan fungsi utama sistem, yaitu memproses input yang dikirim dari cabang input.
- Cabang output (*output branch / efferent branch*) : cabang yang memformat data menjadi output.



Gambar 7.3 Model Dasar Bagan Terstruktur Transformed-centered

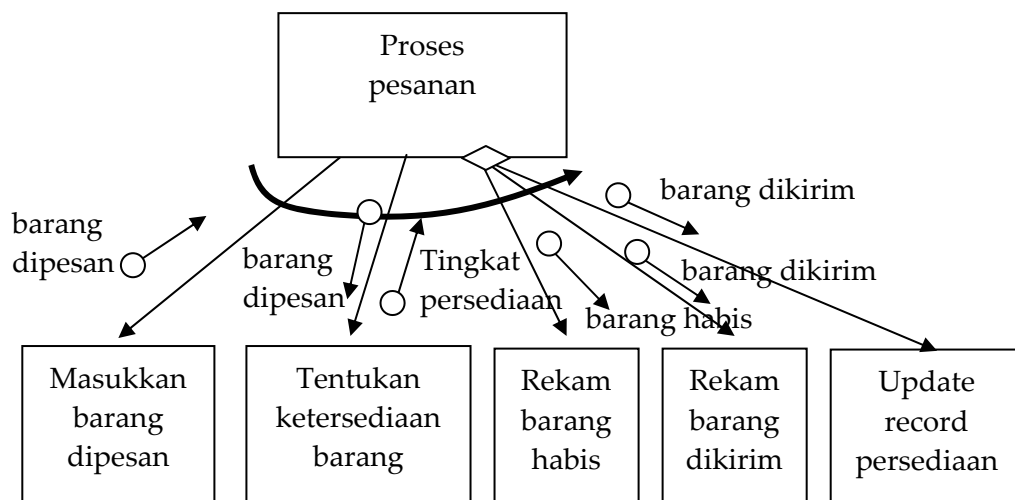
contoh :



Gambar 7.4 Contoh Bagan Terstruktur Transformed-centered

Transaction-centered

Seringkali diagram arus data menggambarkan suatu sistem yang menangani beberapa tipe transaksi yang mempunyai jalur yang berbeda. Diagram tersebut mungkin akan sulit dipilah-pilah berdasarkan transformasinya. Untuk diagram alur data tersebut, dapat dibuat bagan terstruktur model *transaction-centered*.



Gambar 7.5 Contoh Bagan Terstruktur Transaction-centered