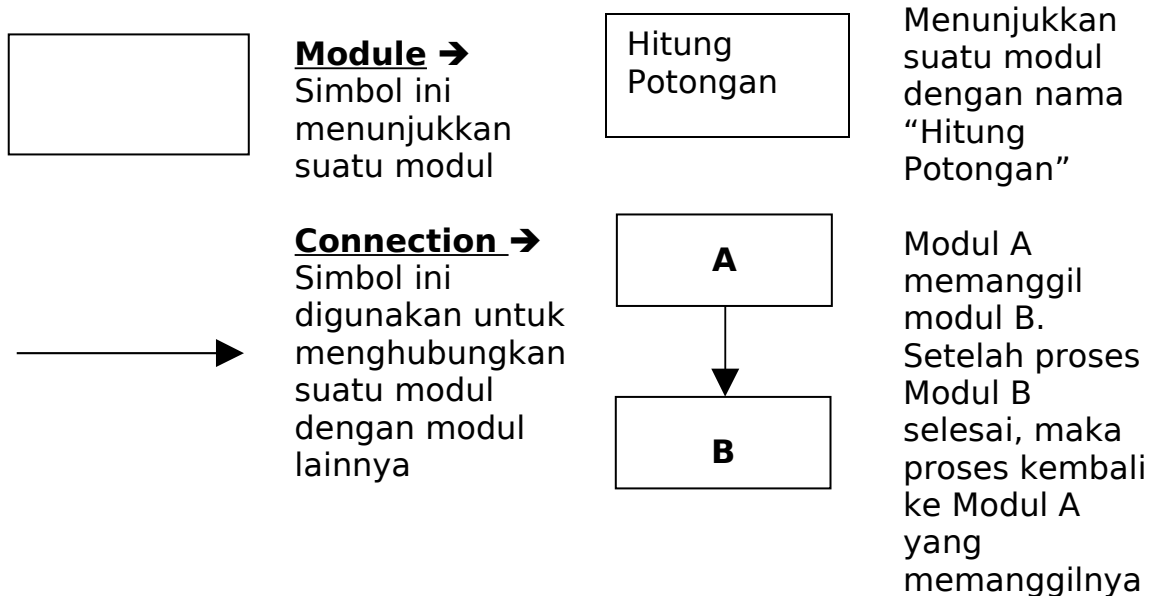


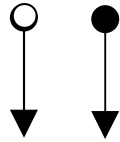
Bagan Tersrtuktur (*Structured Chart*)

I. Pendahuluan

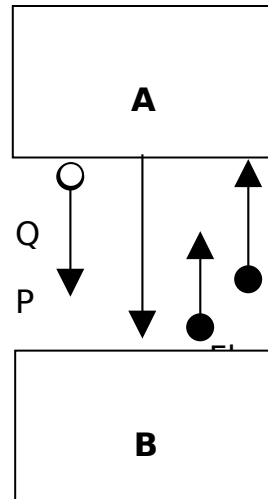
- Mirip dengan bagan berjenjang (*hierarchy chart*)
- Digunakan untuk mendefinisikan dan mengilustrasikan organisasi dari sistem informasi secara berjenjang dalam bentuk modul dan sub modul
- Bedanya : bagan tersrtuktur juga menunjukkan hubungan elemen data dan elemen kontrol antara hubungan modulnya.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

II. Simbol-simbol





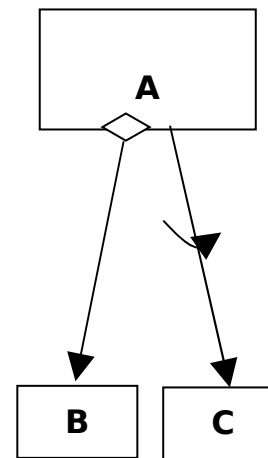
Couple → simbol ini menunjukkan suatu data atau elemen kontrol yang dikirimkan dari suatu modul ke modul lainnya. Panah dengan lingkaran kosong menunjukkan data yang dikirimkan sedangkan panah dengan lingkaran hitam menunjukkan elemen kontrol yang dikirimkan.



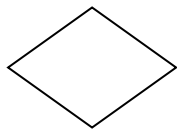
Modul A memanggil Modul B dan elemen data P dikirimkan dari modul A ke Modul B. Hasil proses dari modul B. mengirimkan data Q dari elemen control flag ke Modul A



Loop → Simbol ini menunjukkan suatu perulangan di dalam modul



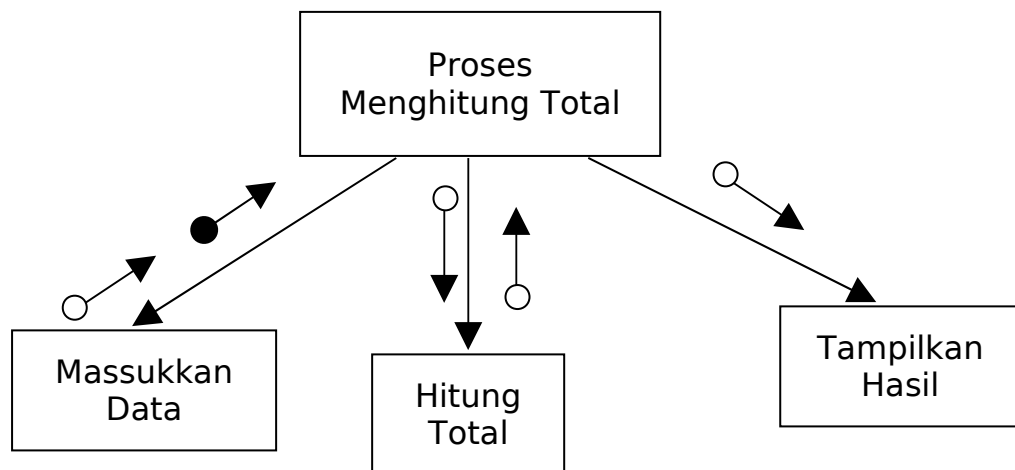
Modul A memanggil modul B bila kondisi yang diseleksi di modul terpenuhi. Modul A juga memanggil modul C berulang kali yang ditunjukkan oleh symbol perulangan.



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

III. Perulangan di bagan Terstruktur

- Ditunjukkan dengan simbol panah yg melingkar
- Banyaknya perulangan ditunjukkan oleh proses di modul letak perulangan tersebut terjadi.



Gambar 3.1 Proses Perulangan Di Bagan Terstruktur

- Modul-modul yang akan diproses berulang kali dari modul PROSES MENGHITUNG TOTAL adalah modul MASUKKAN DATA dan modul HITUNG TOTAL.
- Banyaknya perulangan yg dilakukan adl sampai dengan modul MASUKKAN DATA mengirim elemen kontrol HABIS yg berisi status bhw sudah tidak ada data lagi yg akan dimasukkan.
- Kemudian akan dilanjutkan dengan memanggil modul TAMPILKAN HASIL .
- Dalam bahasa BASIC proses tsb tampak sbb :

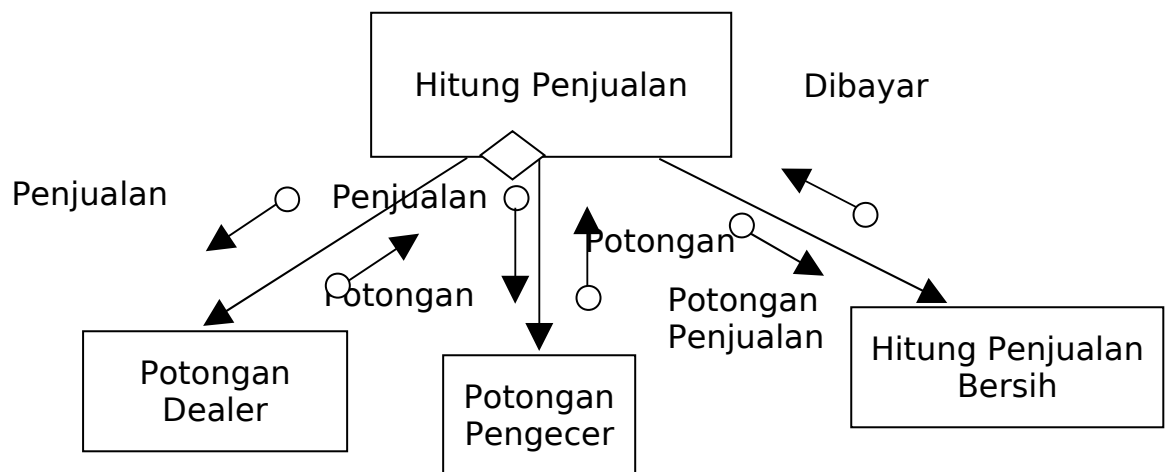
```
10    Proses Menghitung Total
20    HABIS$="T"
30    WHILE HABIS$="T" OR HABIS$="t"
40        GOSUB 100                'Masukkan Data
```

```
50          GOSUB 200          'Menghitung 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 suatu penyeleksian kondisi yg akan membuat keputusan ttg kegiatan-kegiatan yg akan dilakukan.

Contoh : Jika langganan adalah dealer Maka kerjakan modul Potongan-Untuk-Dealer Selainnya kerjakan modul Potongan-Untuk-Pengecer



Gambar 3.2. Proses perulangan di bagan terrstruktur

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

Dalam bahasa Pascal keputusan di modul HITUNG PENJUALAN tampak sbb :

```
If JenisLangganan = 'Dealer' Then
```

```
    PotonganUntukDealer(Penjualan,Potongan)
```

```
Else
```

```
PotonganUntukPengecer(Penjualan,Potongan);
```

```
HitungPenjualanBersih(Penjualan,Potongan,Dibayar);
```

```
WriteLn('Penjualan bersih = ', Dibayar);
```

Sedang modul-modul yg lain tampak sbb :

```
Procedure PotonganUntukDealer(Penjualan: real;var  
Potongan:real);
```

```
Begin
```

```
    Potongan := 0.25 * Penjualan;
```

```
End;
```

```
Procedure PotonganUntukDealaer(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 : Transformed-centered dan Transaction-centered.

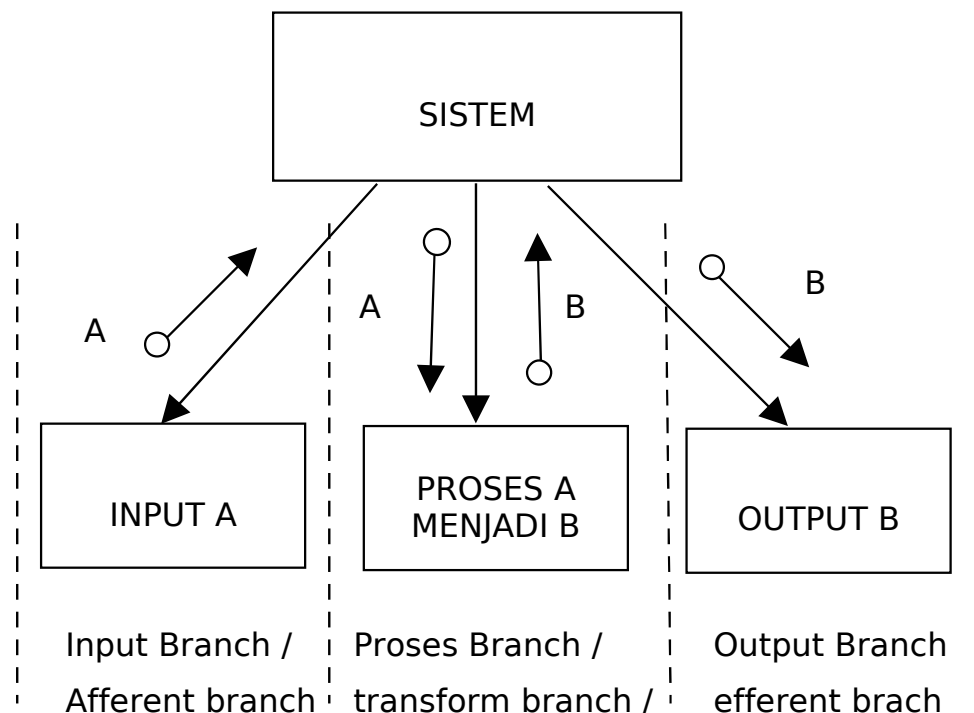
Dapat berbentuk salah satu atau kombinasi.

Model yg akan digunakan tergantung dari diagram arus data yg telah dibuat.

Transformed-centered

Model ini menggambarkan system 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 / centra transform) : Cabang yang melakukan fungsi utama system, yaitu memproses input yang dikirim dari cabang input
- Cabang Output (output branch / efferent branch) : cabang yang memformat data menjadi output.



Gambar 5.1 Model Dasar Bagan Terstruktur Transformed-centered

Transformed-centered

Seringkali diagram arus data menggambarkan suatu system 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.