

Modular Programming

Tujuan :

- Mengenal modul sebagai cara untuk memecahkan masalah.
- Menyajikan hirarki chart.
- Mengembangkan program menggunakan modul.

Modular programming:

- ↳ Memecahkan algoritma ke dalam algoritma yang lebih kecil / modul.
- ↳ Modul yang dibentuk mempunyai kesatuan tugas / fungsi maupun kesatuan proses / prosedur.
- ↳ Setiap modul harus mempunyai single entry dan single exit secara beruntun dari atas ke bawah / awal ke akhir modul.
- ↳ Memiliki main program dan sub program atau modul.

A. Definisi Masalah.

B. Menginventarisasi Struktur Kontrol.

C. Menyusun algoritma, menggunakan prinsip sequence, selection, repetition.

Contoh 10.1

Rancanglah algoritma untuk membaca tiga bilangan, kemudian susun bilangan tersebut dan cetak ketiga bilangan tersebut ke layar.

A. Diagram Definisi

Input	Proses	Output
bil_1	Baca bil_1, bil_2, bil_3.	bil_1
bil_2	Sort bil_1, bil_2, bil_3.	bil_2
bil_3	Cetak bil_1, bil_2, bil_3.	bil_3

B. Algoritma Pemecahan

```

Baca tiga bilangan
Baca bil_1, bil_2, bil_3
DO WHILE NOT (bil_1 = 0 AND bil_2 = 0
               AND bil_3 = 0)
    IF bil_1 > bil_2 THEN
        temp = bil_1
        bil_1 = bil_2
        bil_2 = temp
    ENDIF

```

```

IF bil_2 > bil_3 THEN
    temp = bil_2
    bil_2 = bil_3
    bil_3 = temp
ENDIF
IF bil_1 > bil_3 THEN
    temp = bil_1
    bil_1 = bil_3
    bil_3 = temp
ENDIF
Cetak bil_1, bil_2, bil_3
Baca bil_1, bil_2, bil_3
ENDDO
END.

```

Dari penyelesaian algoritma di atas dapat dibuat penyelesaian dengan modul sbb :

```

- Baca_tiga_bilangan
  Baca bil_1, bil_2, bil_3
  DO WHILE NOT (bil_1 = 0 AND bil_2 = 0
                AND bil_3 = 0)
    Sort_tiga_bilangan
    Cetak bil_1, bil_2, bil_3
    Baca bil_1, bil_2, bil_3
  ENDDO
- END
Sort_tiga_bilangan
IF bil_1 > bil_2 THEN
    temp = bil_1
    bil_1 = bil_2
    bil_2 = temp
ENDIF
IF bil_2 > bil_3 THEN
    temp = bil_2
    bil_2 = bil_3
    bil_3 = temp
ENDIF
IF bil_1 > bil_3 THEN
    temp = bil_1
    bil_1 = bil_3
    bil_3 = temp
ENDIF
END.

```