

The background is a dark navy blue. It features several large, overlapping, semi-transparent geometric shapes in various colors: bright green, cyan, magenta, orange, and blue. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to be layered on top of others. The overall aesthetic is modern and digital.

# **METHOD (SUB PROGRAM)**

# PENGERTIAN

- Sub program/method adalah suatu bagian dari program yang digunakan untuk menjalankan suatu tugas tertentu dan letaknya terpisah dari bagian program yang menggunakannya.
- Sub program akan dipanggil dalam program hanya pada saat dibutuhkan

# KELEBIHAN SUB PROGRAM

1. Program menjadi lebih mudah dimengerti, karena proses penyelesaian masalah dipecah-pecah menjadi lebih sederhana
2. Program menjadi lebih independen, karena suatu sub masalah diselesaikan dalam sub program yang terpisah sehingga tidak akan mempengaruhi perintah pada sub program yang lain
3. Reusability (dapat dipakai kembali), karena sub program yang pernah dibuat dapat dipakai kembali di program yang berbeda sehingga pada pembuatan program berikutnya semakin mudah

# KELEBIHAN SUB PROGRAM

4. Program lebih efisien dan ukurannya lebih kecil, karena penulisan baris program dapat ditulis 1 kali saja dalam sub program dan jika dibutuhkan tinggal memanggil sub program tersebut
5. Program lebih konsisten, karena jika ada penulisan baris program yang sama secara berulang maka programmer tidak harus mengingat kembali posisi baris program yang ditulis sebelumnya
6. Perubahan kode program dapat dilakukan dengan cepat, karena perubahan hanya dilakukan pada bagian/sub program yang salah/perlu perbaikan saja tanpa harus menelusuri seluruh program

# JENIS SUB PROGRAM

Secara garis besar sub program dibagi menjadi :

## 1. **Procedure**

Adalah sub program yang jika dipanggil akan melakukan suatu proses tertentu tetapi tidak ada hasil di dalam dirinya ATAU tidak memberikan nilai balik ke program yang memanggilnya

## 2. **Function**

Adalah sub program yang setelah dipanggil akan menyimpan nilai tertentu di dalam dirinya ATAU memberikan nilai balik ke program yang memanggilnya

# PROCEDURE / PROSEDUR

Syntax prosedur :

```
void namaProsedur(parameter)  
{  
    ... instruksi yang dikerjakan ...  
}
```

Kata kunci void karena prosedur tidak memberikan nilai balik

# PROCEDURE / PROSEDUR

Contoh prosedur :

```
public class Coba {  
    static void informasi () {  
        System.out.print("ALGO");  
    }  
    public static void main (String[] args)  
    {  
        informasi();  
    }  
}
```

# FUNCTION / FUNGSI

Syntax prosedur :

```
tipe namaFungsi(parameter)  
{  
    ... instruksi yang dikerjakan ...  
    return nilai;  
}
```

- ✓ Kata kunci *return* untuk mengembalikan nilai yang disimpan dalam fungsi ke program pemanggilnya
- ✓ *Tipe* adalah sembarang tipe data, yang merupakan tipe data dari nilai balik fungsi.
- ✓ *Nilai* bisa berupa data, ekspresi atau variabel



# FUNCTION / FUNGSI

Contoh fungsi :

```
public class Coba {  
    static int C ()  
    {  
        int C = 22+7;  
        return C;  
    }  
    public static void main (String[] args)  
    {  
        System.out.print("HASIL = " +C());  
    }  
}
```

# PARAMETER SUB PROGRAM

- Dalam banyak hal, sebuah sub program membutuhkan data yang akan diolah dari luar (dari luar sub program, entah itu dari program utama atau dari sub program lain). Data masukan ini disebut sebagai **parameter** atau **argumen**.
- Parameter digunakan untuk pertukaran informasi antara fungsi dengan titik dimana fungsi tersebut dipanggil.  
Parameter ada 2 jenis :
  1. **Parameter formal**, dideklarasikan dalam fungsi
  2. **Parameter aktual**, disertakan pada waktu pemanggilan fungsi.

# PARAMETER SUB PROGRAM

Parameter formal dan aktual saling berpasangan, sehingga jumlah dan tipe data kedua parameter ini **HARUS** sama. Tapi untuk nama variabel parameter boleh tidak sama (dan dianjurkan untuk dibedakan).

Ketika dipanggil, parameter aktual menggantikan parameter formal.

# VARIABEL LOKAL - GLOBAL

**Variabel Global**, adalah variabel yang dideklarasikan oleh program utama (sehingga variabel ini dikenal oleh seluruh bagian program, baik itu program utama atau fungsi-fungsi yang lain).

**Variabel Lokal**, adalah variabel yang dideklarasikan oleh fungsi (hanya dikenal didalam fungsi tersebut).

Jika variabel hanya digunakan dalam fungsi, maka sebaiknya dideklarasikan sebagai variabel lokal dan usahakan memakai variabel global seminim mungkin.