



Sub-Routine Function

(TIB26 – ALGORITMA PEMROGRAMAN)

Pertemuan 25, 26

Sub-CPMK

- Mahasiswa mampu membuat function dan mempergunakannya pada algoritma (C3, A3)

Materi

- Function
- Penempatan Return Value
- Function dengan passing Parameter

Perhatian

- Tidak disarankan copy-paste kode program dari presentasi ini, karena ada beberapa symbol yang dianggap sebagai Unicode oleh editor yang anda gunakan, sehingga akan dianggap sebagai symbol yang salah oleh compiler, sebaiknya diketik ulang saja



1.

Function

1.1 Function

- Function adalah sub-program yang memiliki nilai balikan / return value
- Function harus dideklarasikan dengan tipe data dari nilai balikan
- Syntax:

```
tipedata namafunction()  
{  
    // kode program  
    return variable_atau_data;  
}
```

1.2 Function (lanj.)

- Pencantuman tipe data pada deklarasi function memberi tahu compiler bahwa sub-routine tersebut memiliki nilai balikan sesuai dengan tipe data yang dicantumkan

1.3 Penulisan function

- Penulisan kode pada function sama dengan penulisan kode pada main program
- Penulisan function sebaiknya diletakkan di atas main program, sehingga dapat dikenali langsung oleh main program dan dapat dipanggil oleh main program

1.4 Pemanggilan Function

- Function dapat dipanggil oleh main program maupun oleh sub-program lainnya
- Function yang berada di bawahnya dapat mengenali procedure / function yang dicantumkan di atas nya
- Procedure yang berada dibawah function dapat mengenali dan memanggil function tersebut
- Function dapat pula mengenali dan memanggil procedure yang berada di atas nya

1.5 Notasi Flowchart Function

- Notasi Flowchart untuk function sama seperti notasi flowchart untuk main program
- Judul dan keterangan nama function serta tipe balikan perlu dicantumkan untuk menjelaskan flowchart tersebut adalah flowchart dari function dengan tipe data yang tercantum
- Nama function nantinya yang akan dipanggil pada flowchart utama

1.6 Pemanggilan function pada program utama

- Pemanggilan function pada program utama cukup dengan menuliskan variabel penampung dengan penugasan mencantumkan nama functionnya dan jika ada disertai parameteranya

- Contoh:

```
int contoh() {  
    return 100;  
}
```

```
int main() {  
    int a=contoh(); //pemanggilan function  
                    //untuk ditampung  
                    //pada variable a  
}
```

1.6 Pemanggilan function pada program utama (lanj.)

- Pemanggilan function pada program utama dapat juga untuk langsung dicetak ke layer console.

- Contoh:

```
int contoh()  
{  
    return 100;  
}
```

```
int main()  
{  
    cout<<contoh(); //pemanggilan function  
}
```



2.

Penempatan Return Value

2.1 Penempatan Return Value

- Penempatan return menentukan kapan function akan berhenti dan kembali ke program utama
- Ketika proses mendapatkan perintah return pada function, maka proses akan keluar dari function dan kembali ke program utama
- Return Value dapat ditempatkan pada akhir function maupun didalam pada percabangan

2.1 Penempatan Return Value (lanj.)

- Penempatan Return value pada awal function akan mengakibatkan kode program dibawah perintah return yang belum dieksekusi setelah proses mendapatkan perintah return, tidak akan dieksekusi

2.1 Penempatan Return Value (lanj.)

- Penempatan return value pada loop merupakan sesuatu yang tidak umum dan perlu dipertimbangkan dengan baik, karena ketika proses mendapatkan perintah return, proses akan keluar dari function dan kembali ke main program, sehingga tidak terjadi loop, jadi penempatan return value didalam loop sebaiknya disertai perintah percabangan, sehingga kendali progam akan menempatkan return value pada seleksi

2.2 Ekspresi return pada one way selection

- Ekspresi return pada oneway selection sebaiknya disertai perintah return pada sequence program setelah selection

- Contoh:

```
int contoh()  
{  
    If (a%2==0)  
    {  
        return a/2;  
    }  
    return a;  
}
```

2.3 Ekspresi return pada two way selection

- Ekspresi return pada oneway selection akan memberi kendali penuh pada program untuk memilih nilai balikan yang dihasilkan oleh function

- Contoh:

```
int contoh()  
{  
    If (a%2==0)  
    {  
        return a/2;  
    } else  
    {  
        return a;  
    }  
}
```



3.

Function dengan passing Parameter

3.1 Parameter pada Function

- Parameter dapat ditambahkan pada function
- Parameter harus dideklarasikan dengan tipe data yang tersedia
- Penulisan:

```
tipedata namafunction(tipedata par1, typedata par2, dst)
{
    //kode program
}
```

3.2 Perlakuan Parameter

- Sama seperti pada procedure, Parameter pada function diperlakukan sebagaimana layaknya sebuah variable, tetapi parameter hanya berlaku bagi function yang memiliki parameter tersebut, tidak dapat digunakan diluar function
- Contoh

```
int operasi(int a, int b;)
{
    int hasil;
    hasil = a + b;
    return hasil;
}
```

3.3 Contoh Function dengan passing parameter

```
int operasi(int a, int b)
{
    return a + b;
}
```

```
int main()
{
    clrscr();
    int hasil = operasi(15, 20);
    cout<<hasil;
    getch();
}
```

- function operasi dengan return value bertipe integer memiliki parameter a dan b yang masing-masing bertipe integer
- Parameter a dan parameter b dioperasikan pada tubuh function
- Pemanggilan function operasi melakukan penerusan parameter dari pemanggilannya pada main program dengan nilai 15 untuk parameter a dan 20 untuk parameter b
- Return Value ditampung pada variable hasil

3.4 Passing Parameter Array

- Array dapat diteruskan sebagai parameter dengan hanya menyebutkan nama arraynya
- Array dapat langsung dioperasikan pada return, dapat pula dioperasikan pada tubuh program

Contoh:

```
int c[5]={10, 20, 30, 40, 50};
```

```
int operasiarray(int a[5])  
{  
    return a[0] + a[1] + a[2] + a[3] + a[4];  
}
```

```
int main(void)  
{  
    clrscr();  
    cout<<operasiarray(c);  
    getch();  
}
```

Ringkasan

- Function adalah sub-program yang memiliki nilai balikan / return value
- Pencantuman tipe data sebagai tipe data pada deklarasi function memberi tahu compiler bahwa sub-routine tersebut memiliki nilai balikan bertipe data tersebut
- Penempatan perintah Return value menentukan kapan function berhenti dan kembali ke program utama

Ringkasan (lanj.)

- Penempatan return menentukan kapan function akan berhenti dan kembali ke program utama
- Ketika proses mendapatkan perintah return pada function, maka proses akan keluar dari function dan kembali ke program utama
- Return value pada percabangan memberikan kendali program untuk memilih nilai balikan yang dihasilkan oleh function



Terimakasih

TUHAN Memberkati Anda

Teady Matius Surya Mulyana (tmulyana@bundamulia.ac.id)