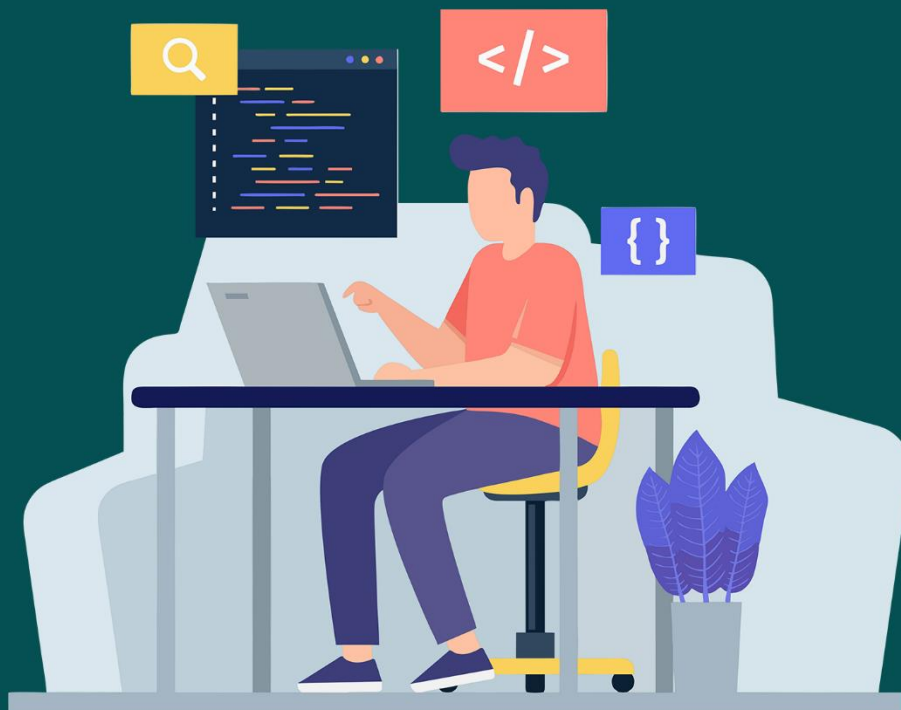


Modul Praktikum
Algoritma dan Pemrograman I

Prosedur dan Penggunaan Tiga File



TIM ASISTEN PEMROGRAMAN
ANGKATAN 12

Departemen Pendidikan Ilmu Komputer
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
2021



Pendahuluan

Dalam pembuatan suatu program yang **panjang dan kompleks**, sering kita jumpai suatu **bagian/potongan** program yang **digunakan di bagian lain** dalam program tersebut. Potongan program yang sering digunakan ini tentunya menjadi persoalan diantaranya program menjadi panjang karena beberapa hal yang sama ditulis berulang-ulang dan akan menimbulkan persoalan jika dilakukan proses modifikasi. Oleh karena itu, untuk mengatasinya kita harus menggunakan Subprogram.

Subprogram merupakan bagian dari program yaitu sebuah set instruksi yang dirancang untuk melakukan operasi yang sering digunakan dalam sebuah program. Didalam Subprogram, terdapat dua jenis yaitu Fungsi dan Prosedur.

Prosedur

Prosedur adalah **bagian** dari suatu program yang disusun secara **terpisah** untuk melakukan suatu **tugas khusus** atau **fungsi tertentu**.

Manfaat Menggunakan Prosedur dan Fungsi:

1. Dapat **memecahkan** program yang rumit dan besar menjadi program-program yang **lebih sederhana** atau **kecil**.
2. Untuk aktivitas yang dilakukan lebih dari satu kali / sering dilakukan **berulang-ulang**.
3. Meningkatkan kemampuan untuk menganalisis kesalahan, jika terjadi suatu kesalahan kita tinggal mencari **fungsi atau prosedur yang bersangkutan saja** dan tak perlu di seluruh program.

Format Dasar Prosedur

Struktur Deklarasi Prosedur:

```
void nama_prosedur(parameter);  
void nama_prosedur(tipe_data1 nama_parameter1, tipe_data2 nama_parameter2, ...);  
void sum(int a, int b);
```

void	Merupakan tanda bahwa perintah yang akan dijalankan merupakan sebuah prosedur.
nama_prosedur	Nama dari suatu prosedur yang dapat di eksekusi dan di panggil. Aturan penulisan nama prosedur sama seperti aturan pada penulisan nama variable
parameter	Parameter adalah variabel yang menyimpan nilai untuk diproses di dalam prosedur. Parameter berfungsi untuk menyimpan nilai yang akan diinputkan ke fungsi

Struktur Pemanggilan Prosedur:


```
nama_prosedur(nama_variabel);
```

nama_prosedur	Nama dari prosedur yang akan dipanggil.
nama_variabel	Variabel yang akan diproses di dalam prosedur

Catatan:

1. Nama variabel pada prosedur dan nama variabel saat pemanggilan tidak harus sama.

```
void sum(int a, int b);  
  
int main()  
{  
    /* code */  
    int num1 = 1, num2 = 2;  
    sum(num1, num2);  
  
    return 0;  
}
```

- 
2. Variabel dengan array tidak perlu dituliskan dengan alokasinya, cukup nama variabelnya saja. Kecuali prosedur memang hanya ingin mengambil satu nilai saja.

```
void prosedurSatu(int angka, char huruf)
{
    // isi
}

void prosedurArray(int angka[], char nama[])
{
    // isi
}

void prosedurArrayOfString(char nama[][50])
{
    // isi
}

int main()
{
    /* code */
    int angka[] = {12, 35, 55, 26, 12};
    char nama[][50] = {"Boba", "Bobi", "Techi", "Carla", "Lala"};

    prosedurSatu(angka[2], nama[4][2]);
    prosedurArray(angka, nama[4]);
    prosedurArrayOfString(nama);
    return 0;
}
```

Cara penulisan Prosedur:

1. Deklarasi dan Definisi Prosedur disatukan

```
1  #include <stdio.h>
2
3  // deklarasi dan definisi prosedur
4  void prosedur1()
5  {
6      // isi
7  }
8
9  void prosedur2()
10 {
11     // isi
12 }
13
14 void prosedur3()
15 {
16     // isi
17 }
18
19 int main()
20 {
21     /* code */
22
23     prosedur1(); // memanggil prosedu1
24     prosedur2(); // memanggil prosedu2
25     prosedur3(); // memanggil prosedu3
26     prosedur1(); // memanggil prosedu1
27
28     return 0;
29 }
30
```

2. Deklarasi dan Definisi Prosedur dipisah

```
1  #include <stdio.h>
2
3  // Deklarasi Prosedur
4  void prosedur1();
5  void prosedur2();
6  void prosedur3();
7
8  int main()
9  {
10     /* code */
11
12     prosedur1(); // memanggil prosedu1
13     prosedur2(); // memanggil prosedu2
14     prosedur3(); // memanggil prosedu3
15     prosedur1(); // memanggil prosedu1
16
17     return 0;
18 }
19
20 // definisi
21 void prosedur1()
22 {
23     // isi
24 }
25
26 void prosedur2()
27 {
28     // isi
29 }
30
31 void prosedur3()
32 {
33     // isi
34 }
35
```

Variable Scope

Variabel Scope (atau ruang lingkup variabel) adalah jangkauan kode program dimana perintah program **masih bisa mengakses** sebuah variabel. Dalam bahasa C, terdapat dua variable scope yaitu **variabel lokal** dan **variabel global**.

Variabel lokal adalah variabel yang hanya bisa diakses di dalam prosedur/fungsi itu sendiri. Sedangkan **variabel global** adalah variabel yang bisa diakses di semua prosedur/fungsi.

```
1  #include <stdio.h>
2
3  int n; // variabel global
4  void prosedur(int param);
5
6  int main()
7  {
8      /* code */
9      printf("%d", n); // akses variabel global
10
11     int i; // variabel lokal
12     for (i = 0; i < n; i++)
13     {
14         prosedur(i);
15     }
16     return 0;
17 }
18
19 void prosedur(int param)
20 {
21     printf("%d", n); // akses variable global
22
23     // paramater termasuk variabel lokal
24     printf("%d", param);
25
26     int i; // variabel lokal
27     for (i = 0; i < n; i++)
28     {
29         // statement
30     }
31 }
```

Penulisan Prosedur Dalam Satu File

```
1  #include <stdio.h>
2
3  // variable global
4  int min = 999999999, max = 0;
5
6  /* Deklarasi Prosedur */
7  void getMin(int angka[], int n);
8  void getMax(int angka[], int n);
9  void printUsername(char nama[]);
10
11 int main()
12 {
13     /* code */
14     int n;
15     char nama[50];
16     scanf("%s", nama);
17     scanf("%d", &n);
18     int angka[n];
19     int i;
20     for (i = 0; i < n; i++)
21     {
22         scanf("%d", &angka[i]);
23     }
24     /* Pemanggilan Prosedur */
25     getMin(angka, n);
26     getMax(angka, n);
27     printUsername(nama);
28     return 0;
29 }
```

```
31  /* Definisi Prosedur */
32
33  // Prosedur mencari nilai terkecil dari array angka
34  void getMin(int angka[], int n)
35  {
36      int i;
37      for (i = 0; i < n; i++)
38      {
39          if (angka[i] < min)
40          {
41              min = angka[i];
42          }
43      }
44  }
45
46  // Prosedur mencari nilai terbesar dari array angka
47  void getMax(int angka[], int n)
48  {
49      int i;
50      for (i = 0; i < n; i++)
51      {
52          if (angka[i] > max)
53          {
54              max = angka[i];
55          }
56      }
57  }
58
59  // Prosedur membuat username dari gabungan nama, nilai min, dan nilai max
60  void printUsername(char nama[])
61  {
62      printf("Username anda adalah: %s%d%d", nama, min, max);
63  }
```


Penggunaan Tiga File

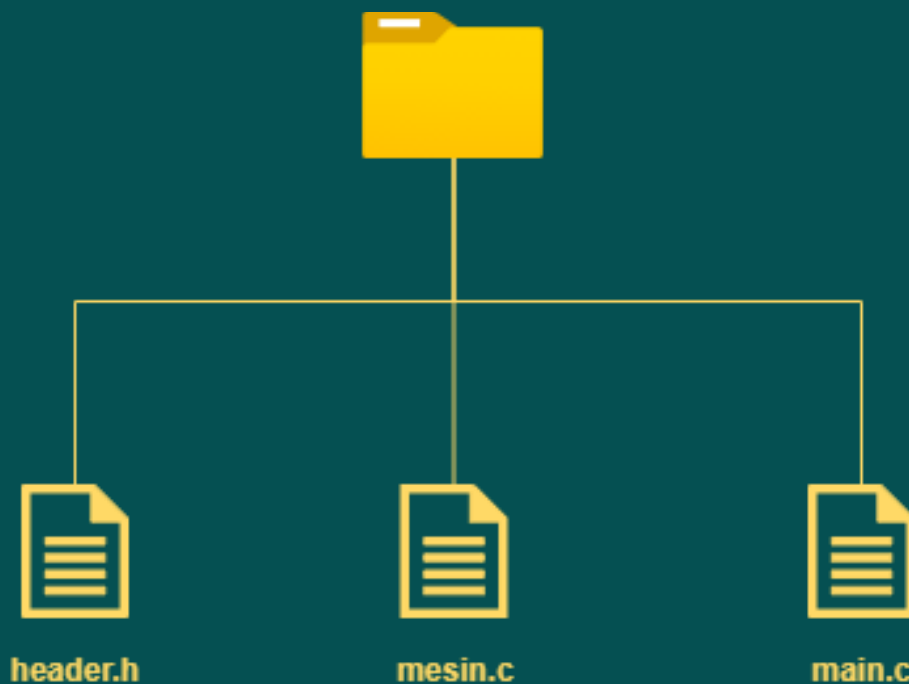
Tiga file tersebut terdiri dari:

- 1 Header
Tempat deklarasi library dasar bahasa C yang digunakan (stdio.h, string.h, malloc.h, dll), variabel global, typedef, serta daftar nama prosedur (deklarasi prosedur).
- 2 Mesin
Tempat untuk menuliskan kode mesin program yang terdiri dari prosedur, dan fungsi.
- 3 Main
Tempat untuk menuliskan kode utama program.

Cara kompilasi tiga file tersebut adalah:

```
gcc *.c -o namaOutput.exe
```

Ketiga file tersebut harus ditempatkan dalam satu folder.



Contoh:

```
▼ tiga_file
  C header.h
  C main.c
  C mesin.c
```

header.h

```
tiga_file > C header.h > ...
1  #include <stdio.h>
2
3  // variable global
4  int min = 999999999, max = 0;
5
6  /* Deklarasi Prosedur */
7  void getMin(int angka[], int n);
8  void getMax(int angka[], int n);
9  void printUsername(char nama[]);
10
```

mesin.c

```
tiga_file > C mesin.c > printUsername(char [])
1  #include "header.h"
2
3  /* Definisi Prosedur */
4
5  // Prosedur mencari nilai terkecil dari array angka
6  void getMin(int angka[], int n)
7  {
8      int i;
9      for (i = 0; i < n; i++)
10     {
11         if (angka[i] < min)
12         {
13             min = angka[i];
14         }
15     }
16 }
17
18 // Prosedur mencari nilai terbesar dari array angka
19 void getMax(int angka[], int n)
20 {
21     int i;
22     for (i = 0; i < n; i++)
23     {
24         if (angka[i] > max)
25         {
26             max = angka[i];
27         }
28     }
29 }
30
31 // Prosedur membuat username dari gabungan nama, nilai min, dan nilai max
32 void printUsername(char nama[])
33 {
34     printf("Username anda adalah: %s%d%d", nama, min, max);
35 }
```

main.c

```
tiga_file > C main.c > main()
1  #include "header.h"
2
3  int main()
4  {
5      /* code */
6      int n;
7      char nama[50];
8      scanf("%s", nama);
9      scanf("%d", &n);
10     int angka[n];
11     int i;
12     for (i = 0; i < n; i++)
13     {
14         scanf("%d", &angka[i]);
15     }
16     /* Pemanggilan Prosedur */
17     getMin(angka, n);
18     getMax(angka, n);
19     printUsername(nama);
20     return 0;
21 }
22
```

Pass By Value dan Pass By Reference

Buka terlebih dahulu <https://bit.ly/2DFgSX8>

Pass By Value, mengirimkan parameter berdasarkan nilai variabel asalnya yang akan dihubungkan dengan parameter pemanggil

Pass By Reference, mengirimkan parameter berdasarkan alamat dari nilai tertentu, maka dari itu bila ada nilai yang dirubah dari alamat asalnya maka akan terjadi juga terhadap nilai parameter yang dipanggil.

```
C value.c > ...
1  #include <stdio.h>
2
3  // parameter disertai tanda *, menandakan pass by reference
4  void addValue(int a, int *b)
5  {
6      *b = *b + a;
7  }
8
9  int main()
10 {
11     /* code */
12     int num1, num2;
13     scanf("%d%d", &num1, &num2);
14     printf("Nilai b sebelum prosedur dipanggil: %d\n", num2);
15
16     // parameter disertai tanda & untuk pass by reference
17     addValue(num1, &num2);
18     printf("Nilai b setelah prosedur dipanggil: %d\n", num2);
19
20     return 0;
21 }
22
```

Latihan

Buatlah program tersebut menggunakan Prosedur.

Buatlah sebuah program untuk menentukan predikat-predikat dari nilai yang diinputkan. Inputan terdiri dari nama dan nilai.

Ketentuan:

90 – 100 = A

80 – 89 = B

70 – 79 = C

60 – 69 = D

Dibawah 60 = E

Contoh Masukan:

```
5
Hirendel 91
Lala 84
Mushi 95
Jakard 23
Nana 84
```

Contoh Keluaran:

```
Predikat A:
-----
1. Hirendel
2. Mushi
-----
Predikat B:
-----
1. Lala
2. Nana
-----
Predikat C:
-----
Tidak ada data.
-----
Predikat D:
-----
Tidak ada data.
-----
Predikat E:
-----
1. Jakard
-----
```



Daftar Pustaka

Tim Asisten Praktikum Algoritma dan Pemrograman I Angkatan 11. (2020). *Perulangan For*. Modul Praktikum Algoritma dan Pemrograman I. Bandung: Universitas Pendidikan Indonesia.

<https://wisnuagung.github.io/mkdocs-fix/menu/php/prosedurFungsi/>

https://informatika.stei.itb.ac.id/~rinaldi.munir/PTI/2013-2014/KU1072_Subprogram_CPP_300913.pdf