5. Fungsi

S. Indriani L, M.T

Pengantar Fungsi

- Sebuah fungsi berisi sejumlah pernyataan yang dikemas dalam sebuah nama.
- Nama ini selanjutnya dapat dipanggil beberapa kali di beberapa tempat dalam program
- Tujuan pembuatan fungsi adalah:
  - ✓Memudahkan dalam mengembangkan program.
  - ✓Menghemat ukuran program.
- Format penulisan:

```
Tipe nama (argumen1, argumen2,…) pernyataan;
```

Contoh program untuk membuat tampilan sbb:



```
//contoh pembuatan dan pemanggilan fungsi

#include<iostream.h>
#include<iomanip.h>
#include<conio.h>

void garis();
void main()
{
   clrscr();
   garis();    //panggil fungsi garis()

   cout<<setiosflags(ios::left);            //Atur rata kiri
   cout<<setw(26)<<"J U D U L  B U K U"<<"PENGARANG"<<endl;

   garis();            //panggil kembali fungsi garis()
   cout<<setw(26)<<"Mastering Borland C++"<<"Tom Swan"<<endl;
   cout<<setw(26)<<"Turbo C++ By Example"<<"Greg Perry"<<endl;

   garis();    //panggil kembali fungsi garis()
}
void garis()
{
   int i;
   for (i=0;i<49;i++)
   {    cout<<'-'; } cout<<endl;
}
```

Prototipe Fungsi

- Deklarasi fungsi dikenal dengan sebutan prototipe fungsi.

  Prototipe ini berupa:

  - ✓ Nama fungsi
  - ✓ Tipe nilai balik fungsi
  - ✓ Jumlah dan tipe argumen

  dan diakhiri dengan titik-koma (;), sebagaimana pada pendeklarasian variabel.

- Sebagai contoh:
  - `long kuadrat(long l);`
  - `int maks3(int a, int b, int c);`
  - `double maks(double x, double y);`
  - `void garis();`

Penjelasan contoh:

- Contoh pertama:

  fungsi kuadrat () mempunyai argumen bertipe `long` dan nilai balik juga bertipe `long`.

- Contoh kedua:

  fungsi maks3() memiliki tiga buah argumen, masing-masing bertipe `int` dan nilai balik juga bertipe `int`.

- Contoh ketiga:

  fungsi maks() mempunyai 2 buah argumen, dengan masing-masing argumen bertipe `double`.

- Contoh keempat:

  fungsi garis() tidak memiliki argumen dan nilai baliknya tidak ada (`void`)

- Manfaat dari prototipe fungsi adalah untuk menjamin tipe argumen yang dilewatkan pada pemanggilan fungsi benar-benar sesuai.

  Tanpa adanya prototipe fungsi, amatlah mudah bagi pemogram untuk melakukan kesalahan tanpa sengaja dalam melewatkan argumen
- Misalnya, melewatkan argumen bertipe `float` pada fungsi yang semestinya meminta tipe data `int`. maka dengan adanya prototipe fungsi, kesalahan semacam itu akan terdeteksi oleh kompiler.

Definisi Fungsi

- Setiap fungsi yang dipanggil didalam program harus didefinisikan. Letaknya dapat dimana saja.
- Contoh:

```
long kuadrat (long l)
{
    return(l*l);
}
```

- Pada pendefinisian fungsi tidak terdapat tanda titik-koma. Ini berbeda dengan prototipe fungsi, yang selalu diakhiri dengan titik koma.
- Pernyataan return didalam fungsi digunakan untuk memberikan nilai balik fungsi.

Contoh penggunaan fungsi

```
//*contoh pembuatan dan pemanggilan fungsi dengan argumen
  bertipe long dan nilai balik juga bertipe long

#include<iostream.h>
#include<iomanip.h>
#include<conio.h>

long kuadrat(long l);    //prototipe fungsi

void main()
{
  clrscr(); //hapus layar

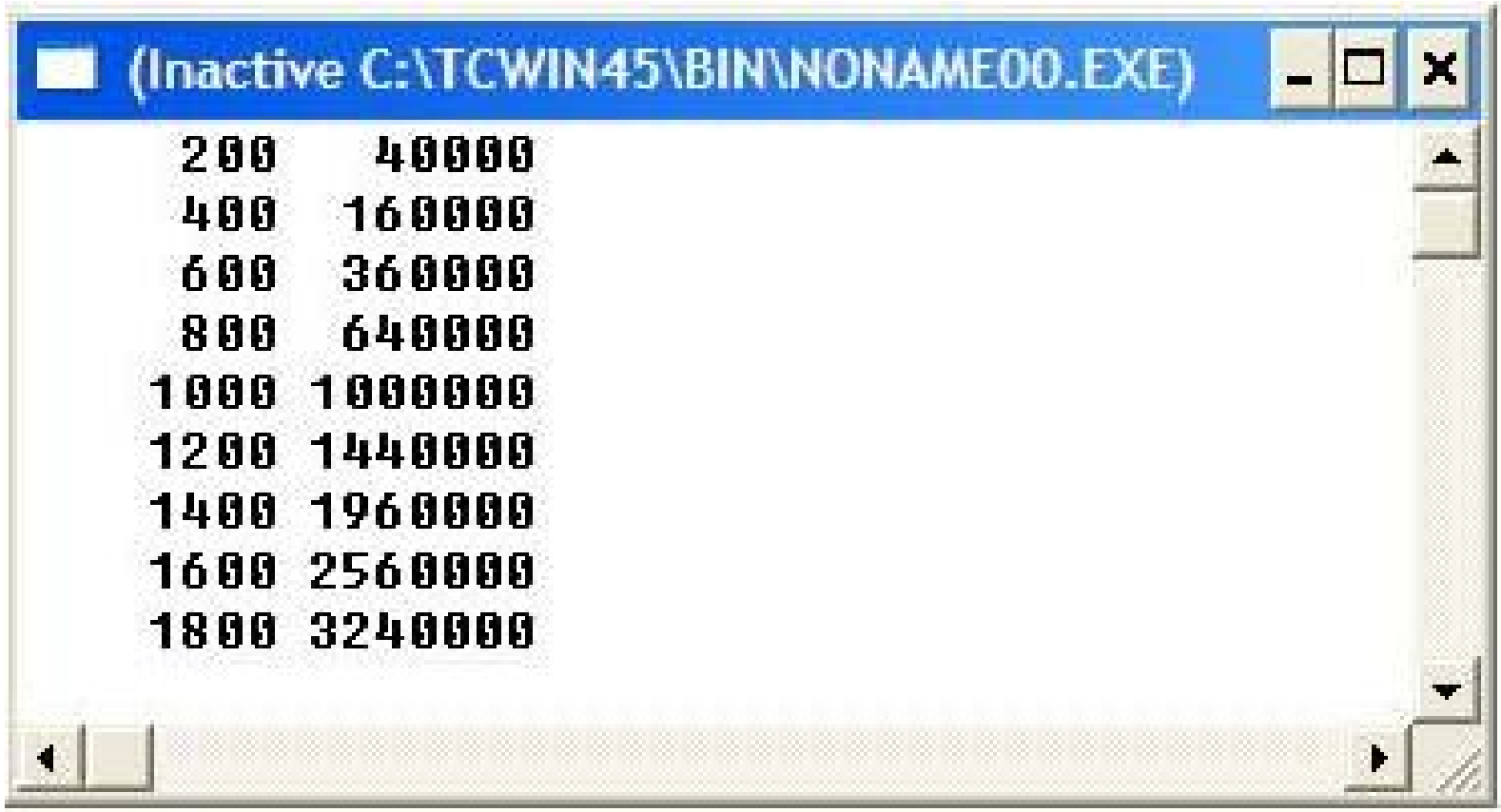
  for (long bil = 200;bil<2000;bil+=200)
  cout << setw(8) << bil
       << setw(8) << kuadrat (bil) << endl;

}

//definisi fungsi

long kuadrat(long l)
{
  return(l*l);
}
```

Hasil tampilan



Contoh penggunaan fungsi

```
//*contoh fungsi untuk memperoleh nilai terbesar diantara
  keduanya

#include<iostream.h>
#include<conio.h>

double maks(double x, double y);      //prototipe fungsi

void main()
{
  clrscr();

  cout << maks(65789,123)<<endl;
  cout << maks(123, 65789)<<endl;

}

//definisi fungsi

double maks(double x, double y)
{
  if (x>y)
  return(x);
      else
      return(y);
}
```

Fungsi tanpa nilai balik

- Adakalanya suatu fungsi tidak perlu memiliki nilai balik. Misalnya fungsi hanya dimaksudkan untuk menampilkan suatu keterangan saja. Pada fungsi seperti ini, tipe nilai balik fungsi yang diperlukan adalah void.
- Contoh:

```
void tampilkan_judul()
{
  cout<<“Universitas Komputer Indonesia”<<endl;
  cout<<“Jurusan Teknik Komputer”<<end;
  cout<<“Bandung<<“endl;
}
```

Lingkup variabel

- Ada variabel yang hanya dikenal di suatu fungsi dan tidak dikenal pada fungsi lain, namun ada juga variabel yang dapat diakses oleh semua fungsi.
- Jenis variabel berdasarkan kelas penyimpanannya, yang berkaitan dengan lingkup variabel, yaitu:

  ✓Variabel lokal

  ✓Variabel global

Semua variabel yang dibuat dalam suatu Function, akan bersifat variabel LOKAL.

- Contoh:

```
#include<iostream.h>
void CETAK();
void main()
{
  CETAK();
}

void CETAK()
{
  int A, B, T;
  A=5; B=2;
  T=A+B;
  cout<<T;
}
```

Variabel LOKAL dan variabel GLOBAL, CALLING Function, dan CALLED Function.

- Contoh:

```
#include<iostream.h>
void CETAK();
void main()
{
  int A, B, T;
  A=5; B=2;
  T=A+B;
  CETAK();
}
void CETAK()
{
  cout<<T;
}
```

Semua variabel yang ada dalam fungsi utama main(), juga bersifat LOKAL, hanya dikenal dalam fungsi main() saja, tidak dikenal didalam fungsi lain. Error akan terjadi karena fungsi CETAK tidak mengenal variabel T, karena dalam fungsi CETAK tidak mengenal variabel T, karena dalam fungsi CETAK tidak ada perintah menyiapkan variabel T, seperti int T;

Sehingga programnya menjadi:

```
#include<iostream.h>
#include<conio.h>

void CETAK();
int T;
void main()
{
  int A, B;
  A=5; B=2;
  T=A+B;
  CETAK();
}
void CETAK()
{
  cout<<T;
}
```

Dari contoh program disamping, main program atau main function disebut CALLING Function, yaitu program yang meng-CALL, dan Fungsi CETAK disebut CALLED Function karena merupakan fungsi yang di CALL.

```
#include<iostream.h>
void TAMBAH();
int A,B,T;
void main()
{
  A=5; B=2;
  TAMBAH();
  cout<<T;
}
void TAMBAH()
{
  T=A+B;
}
```

- Dalam fungsi main () dan juga dalam fungsi TAMBAH(), tidak disiapkan variabel, sehingga semua variabel diambil dari variabel GLOBAL.
- Dalam fungsi main() ada disiapkan variabel A dan B, sehingga yang diisi dengan 5 dan 2 adalah A dan B yang ada dalam fungsi main(), variabel local yang dimilikinya sendiri.
- Fungsi TAMBAH() tidak menyiapkan variabel A, B, dan T, sehingga variabel A, B, dan T diambil dari variabel GLOBAL. Dalam kelompok variabel GLOBAL memang ada variabel A,B, dan T, sehingga proses dalam fungsi TAMBAH() tidak error, hanya isi variabel A, B, dan T yang ada dalam kelompok variabel GLOBAL dan nilainya tidak diketahui (berisi nilai interger sembarang)

Passing variabel (passing by Value)

- Variabel Parameter dan Variabel Argument

  Calling function dapat mengirimkan (passing) suatu nilai ke Called Function. Bila nilai yang dikirim adalah nilai atau data yang akan diproses., maka pengiriman nilai tersebut disebut passing by value. Bila nilai yang dikirim berupa nilai pointer (bukan data) yang merefer ke suatu data, maka pengiriman tersebut disebut passing by reference.

Contoh-contoh berikut mengenai passing by value:

```
#include<iostream.h>

void CETAK(int T);
void main()
{
  int A, B, T;
  A=5; B=2;
  T=A+B;
  CETAK(T);
}
void CETAK(int T)
{
  cout<<T;
}
```

Nama variabel argument boleh berbeda dengan nama variabel parameter.

```
#include<iostream.h>
void CETAK(int T);
void main()
{
  int A, B, T;
  A=5; B=2;
  T=A+B;
  CETAK(T);
}
void CETAK(int X)
{
  cout<<X;
}
```

Referensi

- Pada C++ referensi digunakan untuk memberikan nama alias dari variabel.
- Bentuk deklarasinya:

```
int &ref=nama_variabel;
```

```
// contoh program referensi

#include<iostream.h>
#include<conio.h>

void main()
{
  int i;
  int &r=i;
  clrscr();

  i=0;
  cout<<"i="<<i<<endl;
  cout<<"r="<<r<<endl;

  r=55;
  cout<<"i="<<i<<endl;
  cout<<"r="<<r<<endl;
}
```



Latihan

- Buatlah program dengan menggunakan fungsi untuk menampilkan tampilan pada layar sbb:

```
==========================
MATA KULIAH        DOSEN
==========================
PBO                S. INDRIANI, L
ALGORITMA          SRI NURHAYATI
==========================
```

Pekerjaan Rumah

- Buat program penggunaan dari :
  - variabel lokal
  - variabel global!

- PR dikerjakan berkelompok, maksimum 2 orang, berupa listing program dan hasil tampilannya, dikertas A4