Transformation Towards the Ultimate

Kampus Merdeka INDONESIA JAYA

TIF1101 – Dasar-Dasar Pemrograman

HO 03 - Mengenal C

Opim Salim Sitompul

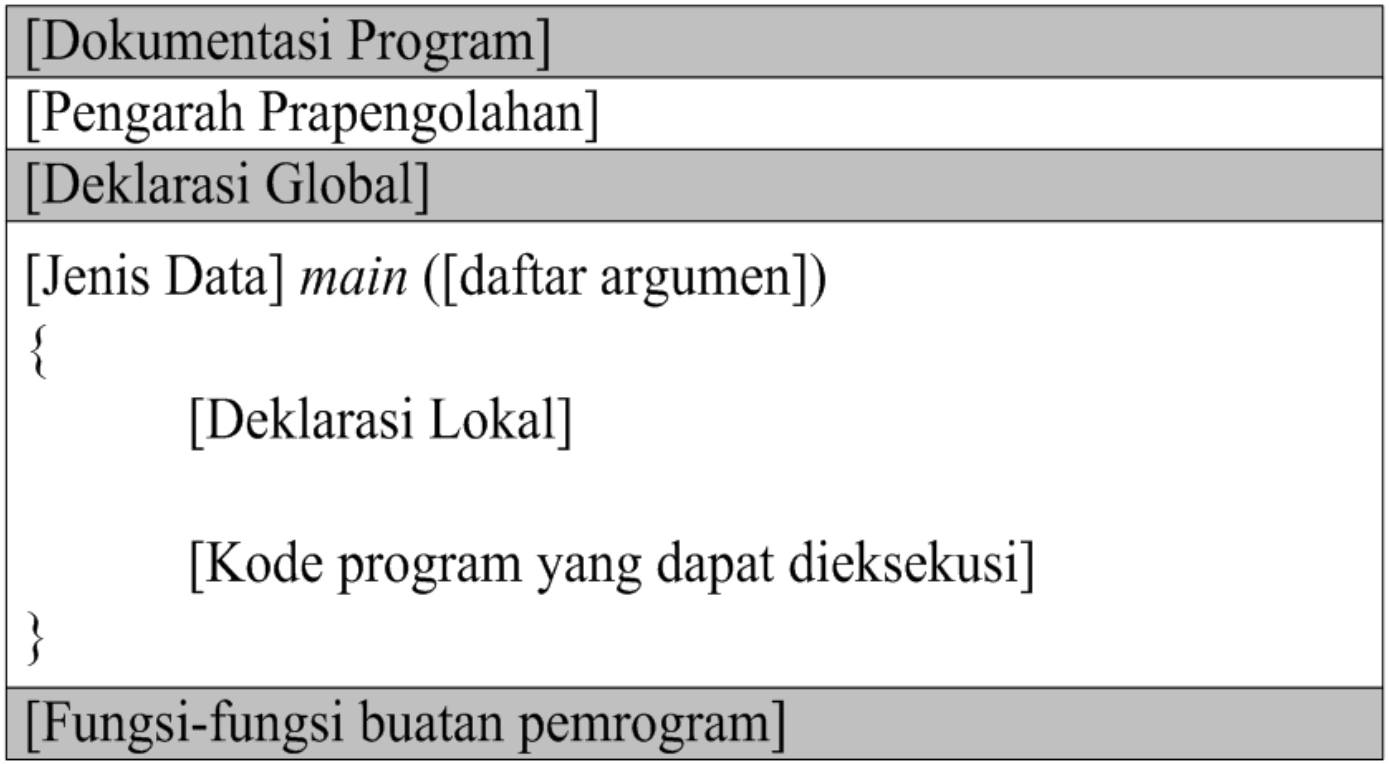
Department of Information Technology
Universitas Sumatera Utara

Outline

Pendahuluan

- Susunan Program C
  - Dokumentasi Program
  - Pengarah Prapengolahan
  - Deklarasi Global
  - Fungsi main()
  - Fungsi Buatan Pemrogram
- Pembatas
- Akhir Pernyataan
- Style Program

Susunan Program C



Gambar 1: Bagian-bagian Program C

Contoh Program C

```
1  /*****************************
2  *  Nama file: luasling.c
3  *  Menghitung luas lingkaran
4  *****************************/
5  #include <stdio.h>
6  #include <stdlib.h>
7  #define PI 3.14159
8  #define KUADRAT(x) ((x) * (x))
9
10 double luas;
11 double luas_lingkaran(float);
```

Contoh Program C

```
12 int main(int argc, char **argv)
13 {
14   float jari2;
15
16   if(argc < 2)
17       printf("PENGGUNAAN:\n\t%s jari-jari\n",
             argv[0]);
18   else
19   {
20      jari2 = atof(argv[1]);
21      luas = luas_lingkaran(jari2);
22      printf("Luas lingkaran berjari-jari %f
23                  = %f\n", jari2, luas);
24    }
25    return 0;
26 }
```

Contoh Program C

```
27 double luas_lingkaran(double r)
28 {
29   return (PI * KUADRAT(r));
30 }
```

Dokumentasi Program

- Sangat berguna untuk membantu memperjelas alur logika penyusunan program
- Program yang disusun akan lebih mudah dipahami orang lain
- Komentar-komentar yang dituliskan pada program tidak diproses oleh kompiler

Dokumentasi Program

- Komentar dapat dimulai dengan simbol dua-karakter yang terdiri dari garis miring dan asterisk (/*) dan diakhiri dengan asterisk dan garis miring (*/)
- Dapat diletakkan di mana saja di dalam program dan dapat mencakup lebih dari satu baris komentar
  - Pada awal program:
    - menjelaskan apa yang dilakukan oleh program
  - Pada bagian-bagian program yang lain:
    - memperjelas logika program.

Contoh Program C

```
1  /* Nama file: asctab.c
2     Mencetak tabel ASCII mulai dari nomor 0 s.d.
          255 */
3
4  #include <stdio.h>
5
6  int main()
7  {
8     int i;                  /* pencacah nomor ASCII */
9
10    /* Tampilkan seluruh karakter ASCII */
11    printf("TABEL ASCII\n");
```

Contoh Program C

```
12    for(i=0;i<256;i++)
13    {
14      printf("%d = %c\n", i, i);
15      /* berhenti tiap-tiap kelipatan 20 */
16      if(i%20 == 0)
17        getchar();
18    }
19
20    /* berhenti untuk 20 baris terakhir */
21    printf("Press <ENTER> ...\n");
22    getchar();
23
24    return 0;
25  }
```

Dokumentasi Program

- Hal-hal yang perlu diperhatikan:
  - Komentar hendaklah diberikan pada tempat-tempat yang bisa menimbulkan kekaburan pengertian.
  - Berikanlah komentar di tempat-tempat yang perlu saja
    - pada awal program untuk menjelaskan secara ringkas tujuan program
    - pada awal blok pernyataan untuk menjelaskan kumpulan tugas yang dilakukan
    - pada pernyataan keputusan untuk menjelaskan kondisi yang harus dipenuhi dalam pengambilan keputusan.
  - Komentar tidak boleh dibuat bertingkat, yaitu adanya komentar di dalam komentar, misalnya:
    /* Komentar awal /* Komentar lagi */ Komentar berikutnya */

Pengarah Prapengolahan

- Melakukan persiapan-persiapan yang diperlukan sebelum berkas program dikompilasi
- Berupa sebuah pengolah makro sederhana untuk memproses berkas program C sebelum kompiler membaca kode program.
- Di dalam program, pengarah prapengolahan diawali oleh karakter # yang dituliskan pada baris-baris pertama program.

Pengarah Prapengolahan

- Prapengolah dapat dibagi ke dalam tiga kelompok:
  - Penyisipan berkas (#include)
    - #include <stdio.h>
  - Pendefinisian makro (#define)
    - #define PI 3.141593
  - Pengarah kendali kompiler (#ifdef, #ifndef, dll).
    - #ifdef A
      /* Lakukan kompilasi apabila A telah didefinsikan */
      ...
      #endif
    - #ifndef A
      /* Lompati apabila A terdefisi; jika tidak lakukan kompilasi */
      ...
      #endif

Deklarasi Global

- Pada bagian deklarasi global terdapat:
  - Pendeklarasian variabel/Konstanta:
    - Semua variabel/konstanta yang dideklarasikan pada bagian ini akan dikenal oleh semua bagian program yang terdapat di bawahnya
    - **double** luas; /* deklarasi variabel */
      const double e = 2.718282; /* deklarasi konstata */
  - Prototipe fungsi:
    - **double** luas_lingkaran(**double**);

Deklarasi Global

- Deklarasi tentang akan digunakannya sebuah fungsi di dalam program, terdiri dari 3 bagian:
  - Jenis data yang dikembalikan oleh fungsi
  - Nama fungsi
  - Daftar argumen yang diberikan kepada fungsi:
    - jumlah argumen (arity)
    - jenis data masing-masing argumen

Fungsi *main*()

- Fungsi utama pada setiap program C dimana eksekusi keseluruhan program dimulai.
- Berapapun banyaknya fungsi yang terdapat dalam sebuah program C, *main*() adalah fungsi pertama yang akan dilaksanakan oleh kompiler.

Fungsi *main*()

- Di dalam fungsi *main*() terdapat:
  - Bagian Deklarasi Lokal
    - Variabel-variabel
    - Prototipe-prototipe fungsi
  - Kode program (pernyataan) yang dapat dieksekusi adalah kode-kode program yang merupakan baris-baris instruksi yang harus dilaksanakan oleh kompiler.
    - Fungsi input/output
    - Konstruksi runtunan, konstruksi keputusan, dan/atau konstruksi pengulangan

Contoh Fungsi main()

```
1  /*************************************
2   *  Nama file: kumulatif_ganjil.c *
3   *  Program menghitung kumulatif bilangan *
4   *************************************/
5  /* Program menghitung kumulatif bilangan */
6  #include <stdio.h>
7
8  int main()
9  {
10   int n, i, Jlh;
11
12   printf("Menghitung kumulatif bilangan.\n");
13   printf("Berikan nilai n: ");
14   scanf("%d", &n);
```

Contoh Fungsi main()

```
15    i = 1;
16    Jlh = 0;
17    printf("%d", i);
18    while(i < n)
19    {
20      Jlh = Jlh + i;
21      i = i + 2;
22      if(i < n)
23        printf("+%d", i);
24      else
25        printf("=");
26    }
27    printf("%d\n", Jlh);
28
29    return 0;
30  }
```

Fungsi Buatan Pemrogram

- Program C terdiri dari satu atau lebih unit yang disebut fungsi
- Fungsi merupakan sekelompok instruksi yang disusun untuk menyelesaikan suatu masalah
- Dapat digunakan kembali tanpa harus menuliskan kembali perincian untuk menyelesaikan masalah tersebut
- Berguna untuk memecah suatu problema besar menjadi bagian-bagian kecil yang lebih mudah diselesaikan
- Struktur fungsi buatan pemrogram sama saja seperti yang terdapat pada fungsi *main*()

Contoh Fungsi Buatan Pemrogram

```
1  /*************************************
2   *  Nama file: luasling.c            *
3   *  Program untuk menghitung luas lingkaran *
4   *************************************/
5  #include <stdio.h>
6  #include <stdlib.h>
7  #define PI 3.14159
8  #define KUADRAT(x) ((x) * (x))
9
10 double luas;
11 double luas_lingkaran(double);
```

Contoh Fungsi Buatan Pemrogram

```
12 int main(int argc, char **argv)
13 {
14   float jari2;
15
16   if(argc < 2)
17   {
18     printf("PENGGUNAAN:\n\t%s jari-jari\n", argv
           [0]);
19   }
20   else
21   {
22     jari2 = atof(argv[1]);
23     luas = luas_lingkaran(jari2);
```

Contoh Fungsi Buatan Pemrogram

```
24      printf("Luas lingkaran berjari-jari %f = %f\n
            ", jari2, luas);
25    }
26
27    return 0;
28  }
29
30  double luas_lingkaran(double r)
31  {
32    return (PI * KUADRAT(r));
33  }
```

Delimiter

- Delimiter (pembatas) adalah sederetan satu atau lebih karakter untuk menentukan batas antara bagian-bagian program yang terpisah, ekspresi matematika, dll.
- Delimiter menunjukkan awal atau akhir dari satu pernyataan tertentu, himpunan string atau badan fungsi.
- Dalam program C, delimiter meliputi:
  - Round bracket (parenthesis): ( )
  - Kurung kurawal (Curly bracket):
  - Escape sequence atau komentar: /* */, //
  - Tanda kutip ganda untuk membatasi string literal: " "
  - Tanda kutip tunggal untuk membatasi karakter: ‘ ‘

Akhir Pernyataan

- Setiap pernyataan (statement) dalam C diakhiri dengan titik koma (;) yang berperan untuk memberitahu kompiler akhir sebuah pernyataan.
- Carriage return yang diperoleh sewaktu menekan tombol <Enter> bukan penunjuk akhir pernyataan.
- C mengabaikan semua karakter yang disebut karakter-karakter whitespace, yaitu spasi, tabulator, dan carriage return (newline).

Style Program

- Keterbacaan (readability) merupakan unsur yang sangat penting dalam menyusun sebuah program karena dapat menggambarkan kerangka berpikir dan algoritma yang digunakan.
- Kemudahan penulisan program sangat besar dipengaruhi oleh sintaks (aturan penulisan) yang dapat digunakan.
- Sintaks program juga dapat mempermudah pengujian dan pemahamannya apabila pada suatu waktu program tersebut perlu dimodifikasi.

Style Program

- Untuk menunjang keterbacaan program ada beberapa hal yang perlu diperhatikan, di antaranya:
  - Gunakanlah nama variabel yang dapat menggambarkan isinya.
  - Berikanlah spasi, tabulator, dan baris kosong secukupnya.
  - Tulislah setiap pernyataan pada satu baris tersendiri.
  - Lengkapilah program dengan komentar secukupnya untuk memperjelas apa yang dilakukan oleh bagian-bagian program tertentu.

Style Program

- Contoh style program yang buruk:

```
1  /* Program 3.6
2  Nama file : sinus_u.c
3  Menghitung Sinus(x) (tanpa format)
4  Dengan rumus pendekatan:
5  sin(x) = x - x^3/3! + x^5/5! - x^7/7! + ...
6  di mana x dinyatakan dalam radian (pi radian =
7  180 derajat) */
8  #include <stdio.h>
9  #define JLH_SUKU 10
10 int tanda(int);
11 unsigned long faktorial(int);
12 double pangkat(double, int);
13 int main()
14 {
```

Style Program

```
15 double x = 0.0, sin_x;
16 int i, j, suku;
17 /* Tabel Sinus x untuk 10 nilai x */
18 printf("TABEL SINUS\nx\tSin(x)\n");
19 for(j=0; j<10; j++)
20 {
21 suku = 1;
22 sin_x = 0.0;
23 for(i=1; i<=JLH_SUKU; i+=2)
24 {
25 sin_x += tanda(suku) * ((pangkat(x,i))/(double)
      faktorial(i));
26 suku++;
27 }
```