

Mata Kuliah Pemrograman

Pengenalan Pemrograman C

Oleh I Putu Arya Dharmaadi, S.T., M.T.

Dosen Jurusan Teknologi Informasi

Universitas Udayana

<http://aryadharmaadi.com>

Senin, 05
Oktober 2020

0



```
#include <stdio.h>
int main(){
    int a;
    int b;
    printf("Masukkan Angka pertama : ");
    scanf("%d",&a);
    printf("Masukkan Angka kedua : ");
    scanf("%d",&b);
    if (a > b){
        printf("Angka pertama lebih besar dari angka kedua");
    }else if (a == b){
        printf("Angka pertama sama besar dengan angka kedua");
    }else{
        printf("Angka kedua lebih besar dari angka pertama");
    }
    return 0;
}
```

Memanggil library

Blok pembuka

Deklarasi variabel

Mencetak kalimat

Meminta input user

Pengecekan kondisi

Aksi yg dilakukan

Kembalian wajib

Blok penutup



MENULIS KOMENTAR

- Komentar adalah blok yang tidak dieksekusi oleh compiler.
- Komentar biasanya digunakan sebagai dokumentasi penjelasan kode
- Dalam bahasa C, disediakan 2 cara untuk menulis komentar.
 - Cara 1: Jika komentar hanya sebaris, maka diawali dengan double slash //
 - Cara 2: Diawali dengan slash bintang /* dan diakhiri dengan */



CONTOH KOMENTAR PADA PROGRAM

```
1  /* Program Latihan */
2  /* Dibuat oleh Arya Dharmaadi
3   Prodi TI           */
4
5 #include <stdio.h>
6
7 int main(){
8     int nomor;                      // pembuatan variabel
9     printf("Masukkan nomor anda: "); // mencetak kalimat ke layar
10    scanf("%d",&nomor);            // meminta input dari user
11    printf("Halo peserta nomor %d\n",nomor); // mencetak nomor urut ke layar
12    return 0;                      // akhir program
13 }
```



CASE SENSITIVITY BAHASA C

- Case sensitivity → peka terhadap perbedaan huruf besar dan kecil
- Artinya, menulis dengan huruf besar akan dianggap berbeda dengan menggunakan huruf kecil
- Maka dari itu, biasakan menulis kode dengan huruf kecil



CONTOH ERROR KARENA CASE SENSITIVITY

```
1  /* Program Latihan */  
2  /* Dibuat oleh Arya Dharmaadi  
3   Prodi TI */  
4  
5  #Include <stdio.h>                                invalid preprocessing directive  
6  
7  Int main(){                                         unknown type name 'Int'  
8    int Nomor;  
9    printf("Masukkan nomor anda: ");      implicitly declaring library function 'printf' w...  
10   scanf("%d",&nomor);      use of undeclared identifier 'nomor'  implicitly declarin...  
11   printf("Halo peserta nomor %d\n",nomor);      use of undeclared identifier 'nomor'  
12   return 0;  
13 }
```



WHITE SPACE

- White space → karakter yang berperan sebagai pemisah antar keyword atau identifier
- Contoh karakter white space → spasi, tab, newline (enter), dll
- White space bisa diisi sebanyak mungkin, asalkan code tetap rapi dan enak dibaca



```
1  /* Program Latihan */  
2  /* Dibuat oleh Arya Dharmaadi  
3   Prodi TI */  
4  
5  #include <stdio.h>  
6  
7  △ intmain() {  
8  △     △ type specifier missing  
9  △ }  
10 int  
11 nomor; // pembuatan variabel  
12  
13 printf("Masukkan nomor anda: "); // mencetak kalimat ke layar  
14  
15 scanf("%d",&nomor); // meminta input dari user  
16  
17 printf("Halo peserta nomor %d\n",nomor); // mencetak nomor urut ke layar  
18  
19 return 0; // akhir program  
20 }
```



CONTOH WHITE SPACE YANG RAPI

```
1  /* Judul      : Program Latihan          */
2  /* Deskripsi   : Program mencetak nomor urut peserta dari      */
3  /*               nomor yang diberikan          */
4  /* Developer   : I Putu Arya Dharmaadi      */
5  /* Tanggal     : 27 September 2020          */
6  /* Versi       : 1.0.0                      */
7
8 #include <stdio.h>
9
10 int main(){
11     int nomor;                         // pembuatan variabel
12     printf("Masukkan nomor anda: ");    // mencetak kalimat ke layar
13     scanf("%d",&nomor);                // meminta input dari user
14     printf("Halo peserta nomor %d\n",nomor); // mencetak nomor urut ke layar
15     return 0;                          // akhir program
16 }
```

WHITE SPACE UNTUK MEMPERTEGAS BLOK DI DALAM BLOK

- White space, khususnya karakter TAB, umumnya digunakan untuk memperjelas batas-batas antar blok kode
- Blok kode umumnya diawali dengan karakter { dan diakhiri dengan }
- Jadi, baris kode setelah karakter { diberikan TAB agar baris kode lebih menjorok ke dalam



CONTOH PENGGUNAAN TAB

```
8 | #include <stdio.h>
9 |
10| int main(){
11|     int nomor;           // pembuatan variabel
12|     printf("Masukkan nomor anda: ");    // mencetak kalimat ke layar
13|     scanf("%d",&nomor);      // meminta input dari user
14|
15|     if (nomor>0){          // pengecekan nomor yg diinput
16|         /* jika nomor lebih dari nol, maka akan mencetak pesan berikut */
17|         printf("Halo peserta nomor %d\n",nomor);
18|     }
19|
20|     return 0;              // akhir program
21| }
```



TERIMA KASIH
