



TIF1101 – Dasar-Dasar Pemrograman

HO 06 - Fungsi Input/Output

Opim Salim Sitompul

Department of Information Technology
Universitas Sumatera Utara

Outline

- 1 Pendahuluan
- 2 Fungsi Output
 - Fungsi Output Berformat
 - Output Karakter
 - Output String
- 3 Fungsi Input
 - Fungsi Input Berformat
 - Input Karakter
 - Input String

Pendahuluan

- Fungsi input dan output adalah fungsi-fungsi yang disediakan oleh kompiler berupa fungsi pustaka (*library function*) untuk menerima input dan menampilkan output pada alat I/O standard.
- Alat I/O standard tersebut terdiri dari keyboard (disebut *stdin*) dan layar monitor (disebut *stdout*).
 - Fungsi output standard:
printf(), putchar() dan puts()
 - Fungsi input standard:
scanf(), getchar() dan gets()

Pendahuluan

- Fungsi printf() dan scanf() disebut fungsi output/input berformat karena penentu format yang digunakan dalam tanda kurung memberitahu kompiler tentang bagaimana menampilkan/menerima input data yang diberikan.
- Lain halnya dengan fungsi gets() dan puts() yang dapat digunakan untuk menerima segala jenis input dan memberikan berbagai jenis output.

Pendahuluan

- Fungsi printf() dan scanf() diakses menggunakan sebuah berkas judul yang disebut stdio.h yang artinya standard input dan output.
- File ini terdiri dari prototipe-prototipe dari fungsi input dan output dan karenanya dapat digunakan dalam program manapun termasuk file itu sendiri.

Fungsi Output Berformat

- Output berformat - printf()
 - Mengubah, memformat dan mencetak argumennya pada alat output standar (*stdout*) di bawah pengendalian format yang ditentukan oleh penentu format.
 - Contoh:

```
printf("\n");  
printf("Selamat belajar, semoga berhasil.");  
printf("%5.1f = %5.1f/%5.1f\n", x, y, z);  
printf("Hasil perkalian = %d\n", hasil);
```

Fungsi Output Berformat

- Argumen yang diberikan pada fungsi printf() diapit oleh tanda kurung.
- Argumen dapat hanya berupa sebuah string, atau terdiri atas string dan satu atau lebih variabel, yang dipisahkan oleh operator koma.
- Bagian string dapat berupa untaian karakter biasa yang ditampilkan seperti apa adanya.
- Dapat pula disertai dengan penentu format berupa spesifikasi konversi, misalnya %d, dan/atau karakter escape, misalnya \n.

Fungsi Output Berformat

Penentu format	Jenis Data	Spesifikasi Konversi
%c	char	Karakter tunggal
%s	char[]	<i>String</i> , berupa konstanta atau <i>array</i> karakter
%d atau %i	int	Integer desimal bertanda
%f	float atau double	<i>Floating point</i> (notasi desimal)
%e atau %E	float atau double	<i>Floating point</i> (notasi ilmiah) menggunakan hurufe atau E
%n	<i>pointer</i> ke integer	Fungsi <i>printf()</i> akan menyimpan jumlah total karakter yang dicetak
%g atau %G	float atau double	<i>Floating point</i> (<i>double precision</i>)
%u	unsigned	Integer desimal tak bertanda
%x atau %X	unsigned	Integer heksadesimal tak bertanda menggunakan hurufkecil "abcdef" atau huruf besar "ABCDEF"
%o	unsigned	Integer oktal tak bertanda
p	<i>far pointer</i> ke void	Alamat dicetak dalam bentuk <i>segment</i> dan <i>offset</i> (SSSS:OOOO)

Gambar 1: Spesifikasi Format Output

Fungsi Output Berformat

- Spesifikasi konversi berfungsi untuk melakukan perubahan atas argumen variabel yang diberikan sesuai dengan jenis datanya.
- Spesifikasi konversi harus diawali dengan karakter % dan diakhiri dengan karakter konversi.
- Di antara karakter % dan karakter konversi dapat pula diberikan:
 - Bilangan yang menentukan lebar field minimum untuk bilangan integer.
 - Bilangan yang menyatakan jumlah maksimum karakter dalam sebuah string.
 - Dua buah angka yang dipisahkan oleh titik untuk sebuah nilai floating point.

Fungsi Output Berformat

- Contoh:
%8.2f, %5.1f, %+7.2f
- [][1][9][6][2][.][2][6] → 1962.26
[1][7][0][.][8] → 170.8
[][+][1][1][.][1][0] → 11.10

Fungsi Output Berformat

```
1  /* Nama file: oformat.c
2     Contoh penggunaan penentu format */
3  #include <stdio.h>
4
5  int main()
6  {
7     char str[] = "Selamat belajar, semoga
8         berhasil";
9     int i = 10, j = 100, k = 1000, l = -1000;
10    float x = 10.0, y = -100.0, z = 1000.0, t =
11        -15.55;
12    unsigned m = 0453, n = 0xf01b, o = 0xabfe, p
13        = 02544;
14    short q = 1000;
15    long r = 50000, s = 100000;
16    long double u = 145000.0L;
```

Fungsi Output Berformat

```

14 printf("%7s %8s %13s %9s\n", "Integer", "
    Float", "Oktal-Hexa", "Short-Long");
15 printf("% 5i\t%8.2f\t%#o\t%hd\n", i, x, m, q
    );
16 printf("%-5d\t%+8.2f\t%#x\t%ld\n", j, y, n,
    r);
17 printf("%+5d\t%E\t%#X\t%\#Ld\n", k, z, o, s)
    ;
18 printf("%5d\t%e\t%#o\t%#G\n", l, t, p, u);
19 printf("\n%-35s\n", str);
20
21 return 0;
22 }
```

Output Karakter

- Menampilkan satu karakter ke layar.
- Contoh:
char ch = 'A';
...
putchar(ch);

Output Karakter

▶ outchar.c

```
1  /* Namafile: outchar.c
2     Menampilkan output karakter ke layar */
3  #include <stdio.h>
4
5  int main()
6  {
7     char ch;
8
9     printf("Berikan sebuah karakter: ");
10    scanf("%c", &ch);
11    printf("Karakter yang diberikan adalah: ");
12    putchar(ch);
13
14    return 0;
15 }
```

Output String

- Menampilkan untaian karakter ke layar

- Contoh:

```
char str[] = "Fasilkom-TI USU Medan"; /* Ini OK */
```

```
char kalimat[80];
```

```
...
```

```
puts(str);
```

- kalimat = "Tahun 2018\n"; /* ERROR */

- strcpy(kalimat, "Tahun 2018\n"); /* Ini OK */

```
...
```

```
puts(kalimat);
```


Output String

```
1 /* Namafile: outstring1.c
2    Menampilkan string ke layar */
3
4 #include <stdio.h>
5
6 int main()
7 {
8     char str[]="Fasilkom-TI USU Medan"; /*OK*/
9
10    puts(str);
11
12    return 0;
13 }
```

Output String

```
1 /* Namafile: outstring2.c
2     Menampilkan string ke layar */
3 #include <stdio.h>
4 #include <string.h>
5
6 int main()
7 {
8     char kalimat[80];
9
10    /*kalimat = "Tahun 2018\n"; */ /* ERROR */
11    strcpy(kalimat, "Tahun 2018\n");
12    puts(kalimat);
13
14    return 0;
15 }
```

Fungsi Input Berformat)

- Men-scan input yang diberikan melalui alat input standard (keyboard (stdin)) sesuai dengan penentu format yang diberikan.
- Meletakkan nilai yang diberikan tersebut pada alamat variabel yang bersesuaian.
- Contoh:
`scanf("%d", &x);`
`scanf("%f %f %d", &x, &y, &z);`
`scanf("%c", &ch);`
`scanf("%s", str);`

Fungsi Input Berformat

```
1 /* Namafile: CthIO.c
2     Input/Output Berformat */
3 #include <stdio.h>
4
5 int main()
6 {
7     int usia;
8
9     printf("Berapa usia Anda? ");
10    scanf("%d",&usia);
11    if(usia>=17)
12        printf("Usia %d berhak memilih\n", usia);
13    else
14        printf("Belum berhak memilih\n");
15    return 0;
16 }
```

Fungsi Input Berformat

Penentu Format	Spesifikasi Konversi
%c	Karakter tunggal
%s	<i>String</i> karakter
%d	Integer desimal bertanda
%i	Integer desimal, heksa desimal, atau oktal
%f	<i>Floating point</i> (notasi desimal)
%e, %E, %f, %g, atau %G	<i>Floating point</i> (notasi ilmiah)
%u atau %U	Integer desimal tak bertanda
%x	Integer heksa desimal tak bertanda
%o	Integer oktal tak bertanda
%p	Angka heksa desimal dalam bentuk "SSSS:OOOO" menggunakan huruf besar
%n	Sebenarnya bukan format baca. Argumen yang bersesuaian dengan format ini adalah <i>pointer</i> ke integer. Sebelum mengembalikan nilai, fungsi <i>scanf()</i> akan menyimpan di dalam integer ini, jumlah total karakter yang telah dibaca.
l	Prefiks yang digunakan bersama %d, %u, x, %o untuk menyatakan long integer (contoh: %ld). Juga digunakan bersama dengan variabel <i>floating point</i> untuk menyatakan double , bukan float .
h	Prefiks untuk membaca bilangan short integer.

Gambar 2: Spesifikasi Format Input

Input Karakter

- Menerima input satu karakter tunggal dari alat input standar (stdin)
- Contoh:
char ch;
...
ch = getchar();

Input Karakter

```
1 /* Namafile: inputchar.c
2    Menerima input karakter */
3 #include <stdio.h>
4
5 int main()
6 {
7     char ch;
8
9     printf("Berikan sebuah karakter: ");
10    ch = getchar();
11    ch = ch ^ 32;
12    printf("Karakter Anda: %c", ch);
13
14    return 0;
15 }
```

Input String

- Menerima input untaian karakter dari alat input standar.
- Di lingkungan Windows:
- Contoh:

```
char str[80];
...
gets(str);
```


Input String

```
1  /* Nama file: gets.c
2     Menggunakan fungsi gets() */
3  #include <stdio.h>
4
5  int main()
6  {
7     char str[80];
8
9     printf("Berikan sebuah kalimat: ");
10    gets(str);
11    printf("Anda menulis: %s\n", str);
12
13    return 0;
14 }
```

Input String

- Digunakan di lingkungan Linux:

- Contoh:

```
int bytes_read;
```

```
int nbytes = 100;
```

```
char *my_string;
```

```
...
```

```
my_string = (char *) malloc (nbytes + 1);
```

```
bytes_read = getline (&my_string, &nbytes);
```

Fungsi Input String

```
1  /* Nama file: getline.c
2     Program contoh penggunaan getline */
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  int main()
7  {
8     int bytes_read;
9     int nbytes = 100;
10    char *my_string;
11
12    printf("Tulis satu baris teks.");
13    /* Dua baris ini merupakan inti program */
14    my_string = (char *) malloc (nbytes + 1);
15    bytes_read = getline (&my_string, &nbytes);
```

Fungsi Input String

```
16  if (bytes_read == -1)
17  {
18      puts ("ERROR!");
19  }
20  else
21  {
22      printf("Saudara tulis %s\n", my_string);
23  }
24
25  return 0;
26 }
```