

Pertemuan 3



Input Output & Operator dalam C

Mata Kuliah : Algoritma & Pemrograman
Dosen : Tessy Badriyah, SKom., MT., PhD.



Tujuan Pembelajaran

- Memahami Fungsi Input/Output dalam C
- Memahami Operator dalam Bahasa C yang terdiri dari:
 - Operator Aritmatik
 - Operator Increment dan decrement operators
 - Operator Penandaan (Assignment)
 - Operator Relasional
 - Operator Logika
 - Operator Bitwise
 - Operator Koma
 - Operator sizeof
 - Operator Ternary (?:)



C Input/Output

- Bahasa pemrograman C memiliki beberapa fungsi built-in yang digunakan untuk membentuk operasi input output.
- Yang paling umum adalah dengan menggunakan `printf()` dan `scanf()`.
- Fungsi `scanf()` membaca format input standart (keyboard) sedangkan fungsi `printf()` mengirim format output agar dapat ditampilkan pada layar (screen).



Contoh C Input/Output

```
#include <stdio.h>
```

```
//library ini diperlukan untuk  
//menjalankan fungsi printf()
```

```
int main()
```

```
{
```

```
    printf("C Programming");
```

```
    return 0;
```

```
}
```

Output :

```
C Programming
```



Bagaimana cara kerja program?

- Semua program C harus memiliki satu fungsi `main()`. Kode yang dapat dijalankan dimulai dari permulaan fungsi `main()`.
- `printf()` adalah fungsi library untuk mengirimkan output dalam bentuk yang terformat ke layar (screen). Fungsi `printf()` memerlukan deklarasi header file `"stdio.h"`.
- Disini, `stdio.h` adalah header file (standard input output header file), sedangkan `#include` adalah preprocessor directive untuk mendapatkan kode dari header file jika diperlukan. Ketika compiler mendapatkan suatu program menggunakan fungsi `printf()` akan tetapi tidak ditemukan header file `stdio.h`, maka akan dimunculkan pesan kesalahan.
- Pernyataan : `return 0;` merupakan perintah untuk keluar dari program, dengan kata lain adalah untuk mengakhiri suatu program.



Contoh Output bertipe Integer

```
#include <stdio.h>
int main()
{
    int testInteger = 5;
    printf("Bilangan = %d", testInteger);
    return 0;
}
```

Output :

Bilangan = 5



Contoh Output bertipe Integer

```
#include <stdio.h>
int main()
{
    int testInteger;
    printf("Masukkan bilangan integer: ");
    scanf("%d",&testInteger);
    printf("Bilangan = %d",testInteger);
    return 0;
}
```

Output :

```
Masukkan bilangan integer: 4
Bilangan = 4
```



Input/Output bertipe Float

```
#include <stdio.h>
int main()
{
    float f;
    printf("Masukkan bilangan:");
    // format %f digunakan untuk bilangan real/pecahan
    scanf("%f",&f);
    printf("Nilai = %f", f);
    return 0;
}
```

Output :

```
Masukkan bilangan:23.45
Nilai = 23.450000
```




Input/Output bertipe Karakter

```
#include <stdio.h>
int main()
{
    char chr;
    printf("Masukkan karakter: ");
    scanf("%c",&chr);
    printf("Anda memasukkan %c.",chr);
    return 0;
}
```

Output :

```
Masukkan karakter: g
Anda memasukkan g.
```

Format string %c digunakan untuk tipe data karakter.



Tentang kode ASCII

- Pada saat karakter dimasukkan, maka nilai numerik berupa ASCII value akan disimpan.

```
printf("Anda memasukkan karakter  
%c.\n",chr);
```

- Pada saat nilai tersebut dengan menggunakan format "%c" maka karakter dari yang dimasukkan akan ditampilkan.

```
printf("nilai ASCII value dari %c  
adalah %d.", chr, chr); \
```

- Pada saat 'g' dimasukkan, 103 akan disimpan ke dalam var1 dan bukan g.

```
Masukkan karakter: g
Anda memasukkan karakter g.
ASCII value of g is 103.
nilai ASCII dari 'g' adalah 103.
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char chr;
```

```
    printf("Masukkan karakter: ");
```

```
    scanf("%c",&chr);
```

```
    printf(" Anda memasukkan karakter  
%c.\n",chr);
```

```
    printf("nilai ASCII dari %c adalah %d.", chr,  
chr);
```

```
    return 0;
```

```
}
```



Operator dalam Bahasa C

- Operator adalah symbol yang beroperasi pada nilai atau variable. Contoh: operator '+' digunakan untuk membentuk operasi penambahan bilangan.
- Pemrograman Bahasa C memiliki berbagai macam operator yang akan membentuk suatu operasi tertentu.
- Operator dalam C yaitu:
 - Operator Aritmatik
 - Operator Increment dan decrement operators
 - Operator Penandaan (Assignment)
 - Operator Relasional
 - Operator Logika
 - Operator Bitwise
 - Operator Koma
 - Operator sizeof
 - Operator Ternary (?:)



Operator Aritmatik

- Operator aritmatik membentuk operasi aritmatika semisal penambahan, pengurangan dan perkalian pada nilai numerik atau konstanta.

Operator	Meaning of Operator
+	addition or unary plus
-	subtraction or unary minus
*	multiplication
/	division
%	remainder after division(modulo division)



Contoh Operator Aritmatik

```
// Contoh program yang menggunakan operator aritmatik
#include <stdio.h>
int main()
{
    int a = 9, b = 4, c;
    c = a+b;
    printf("a+b = %d \n", c);
    c = a-b;
    printf("a-b = %d \n", c);
    c = a*b;
    printf("a*b = %d \n", c);
    c = a/b;
    printf("a/b = %d \n", c);
    c = a%b;
    printf("Sisa hasil pembagian a dibagi dengan b adalah = %d \n", c);
    return 0;
}
```

Output :

$a+b = 13$

$a-b = 5$

$a*b = 36$

$a/b = 2$

Sisa hasil pembagian a dibagi dengan b adalah = 1



Operator Increment dan decrement

- Pemrograman C memiliki dua operator increment ++ dan decrement – untuk merubah nilai dari operand (konstanta maupun variabel) dengan 1.
- Operator Increment ++ akan menaikkan suatu nilai dengan 1 sedangkan operator decrement – akan menurunkan suatu nilai dengan 1.
- Kedua operator ini adalah unary operators, ini berarti hanya beroperasi pada operand tunggal saja.
- Pada program, operators ++ dan -- digunakan sebagai prefix.
- Kedua operators juga dapat digunakan sebagai postfix seperti a++ dan a--.

// Contoh program yang menggunakan operator increment dan decrement

```
#include <stdio.h>

int main()
{
    int a = 10, b = 100;
    float c = 10.5, d = 100.5;
    printf("++a = %d \n", ++a);
    printf("--b = %d \n", --b);
    printf("++c = %f \n", ++c);
    printf("--d = %f \n", --d);
    return 0;
}
```

Output :

```
++a = 11
--b = 99
++c = 11.500000
++d = 99.500000
```



Operator Penandaan (assignment)

- Operator penandaan (assignment operator) digunakan untuk menandai nilai dari suatu variable.
- Operator penandaan yang paling umum adalah operator =

Operator	Example	Same as
=	a = b	a = b
+=	a += b	a = a+b
-=	a -= b	a = a-b
*=	a *= b	a = a*b
/=	a /= b	a = a/b
%=	a %= b	a = a%b



Contoh program dengan operator penandaan

// Contoh program yang menggunakan operator penandaan (assignment operator)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a = 5, c;
```

```
    c = a;
```

```
    printf("c = %d \n", c);
```

```
    c += a; // c = c+a
```

```
    printf("c = %d \n", c);
```

```
    c -= a; // c = c-a
```

```
    printf("c = %d \n", c);
```

```
    c *= a; // c = c*a
```

```
    printf("c = %d \n", c);
```

```
    c /= a; // c = c/a
```

```
    printf("c = %d \n", c);
```

```
    c %= a; // c = c%a
```

```
    printf("c = %d \n", c);
```

```
    return 0;
```

```
}
```

Output:

c = 5

c = 10

c = 5

c = 25

c = 5

c = 0



Operator Relasional

- Operator relasional memeriksa relasi dari dua operands.
- Jika hasilnya adalah true, maka akan dikembalikan nilai 1 (return 1); tapi jika hasilnya adalah false, maka akan dikembalikan nilai 0 (return 0).
- Operator relasional digunakan dalam pencabangan dan perulangan.

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 returns 0
>	Greater than	5 > 3 returns 1
<	Less than	5 < 3 returns 0
!=	Not equal to	5 != 3 returns 1
>=	Greater than or equal to	5 >= 3 returns 1
<=	Less than or equal to	5 <= 3 return 0



Program Operator Relasional

// Contoh program yang menggunakan operator relasional

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a = 5, b = 5, c = 10;
```

```
    printf("%d == %d = %d \n", a, b, a == b); // true
```

```
    printf("%d == %d = %d \n", a, c, a == c); // false
```

```
    printf("%d > %d = %d \n", a, b, a > b); //false
```

```
    printf("%d > %d = %d \n", a, c, a > c); //false
```

```
    printf("%d < %d = %d \n", a, b, a < b); //false
```

```
    printf("%d < %d = %d \n", a, c, a < c);  
    //true
```

```
    printf("%d != %d = %d \n", a, b, a != b);  
    //false
```

```
    printf("%d != %d = %d \n", a, c, a != c);  
    //true
```

```
    printf("%d >= %d = %d \n", a, b, a >= b);  
    //true
```

```
    printf("%d >= %d = %d \n", a, c, a >= c);  
    //false
```

```
    printf("%d <= %d = %d \n", a, b, a <= b);  
    //true
```

```
    printf("%d <= %d = %d \n", a, c, a <= c);  
    //true
```

```
    return 0;
```

```
}
```

Output:

```
5 == 5 = 1  
5 == 10 = 0  
5 > 5 = 0  
5 > 10 = 0  
5 < 5 = 0  
5 < 10 = 1  
5 != 5 = 0  
5 != 10 = 1  
5 >= 5 = 1  
5 >= 10 = 0  
5 <= 5 = 1  
5 <= 10 = 1
```



Operator Logika

- Suatu ekspresi yang berisi operator logika akan mengembalikan nilai 0 atau 1 tergantung pada hasil ekspresinya true ataukah false.
- Operator logika umum digunakan pada percabangan dan perulangan

Operator	Meaning of Operator	Example
&&	Logial AND. True only if all operands are true	If c = 5 and d = 2 then, expression ((c == 5) && (d > 5)) equals to 0.
	Logical OR. True only if either one operand is true	If c = 5 and d = 2 then, expression ((c == 5) (d > 5)) equals to 1.
!	Logical NOT. True only if the operand is 0	If c = 5 then, expression ! (c == 5) equals to 0.



Contoh Operator Logika

```
// Program untuk mendemonstrasikan operator logika
#include <stdio.h>

int main()
{
    int a = 5, b = 5, c = 10, result;
    result = (a == b) && (c > b);
    printf("(a == b) && (c > b) hasilnya %d \n", result);
    result = (a == b) && (c < b);
    printf("(a == b) && (c < b) hasilnya %d \n", result);
    result = (a == b) || (c < b);
    printf("(a == b) || (c < b) hasilnya %d \n", result);
    result = (a != b) || (c < b);
```

```
    printf("(a != b) || (c < b) hasilnya %d \n",
result);

    result = !(a == b);
    printf("!(a == b) hasilnya %d \n", result);
    result = !(a != b);
    printf("!(a != b) hasilnya %d \n", result);
    return 0;
}
```

Output :

```
(a == b) && (c > b) hasilnya 1
(a == b) && (c < b) hasilnya 0
(a == b) || (c < b) hasilnya 1
(a != b) || (c < b) hasilnya 0
!(a != b) hasilnya 1
!(a == b) hasilnya 0
```



Operator Bitwise

- Selama komputasi, operator matematika seperti : penambahan, pengurangan, pembagian dan perkalian akan dikonversi ke dalam bit-level yang akan membuat pemrosesan menjadi semakin cepat.
- Operator Bitwise operators digunakan dalam Bahasa pemrograman C untuk membentuk operasi bit-level.

Operators	Meaning of operators
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right



Operator sizeof

- Operator sizeof merupakan unary operator yang akan mengembalikan ukuran dari data (konstanta, variable, array, struktur, dll).

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a, e[10];
```

```
    float b;
```

```
    double c;
```

```
    char d;
```

```
    printf("Size of int=%lu bytes\n",sizeof(a));
```

```
    printf("Size of float=%lu bytes\n",sizeof(b));
```

```
    printf("Size of double=%lu bytes\n",sizeof(c));
```

```
    printf("Size of char=%lu byte\n",sizeof(d));
```

```
    printf("Size of integer bertipe array yang memiliki 10 elemen =  
%lu bytes\n", sizeof(e));
```

```
    return 0;
```

```
}
```

Output:

Size of int = 4 bytes

Size of float = 4 bytes

Size of double = 8 bytes

Size of char = 1 byte

Size of integer bertipe array yang memiliki 10 elemen = 40 bytes



Operator Ternary (?:)

- Operator ternary merupakan operator pencabangan (kondisional), yang bekerja dengan tiga operands.
- Cara penulisannya:
 - ekspresikondisional ? ekspresi1 : ekspresi2
- Operator kondisional bekerja sebagai berikut:
 - Jika ekspresikondisional bernilai true, maka akan didapatkan ekspresi1.
 - Jika ekspresikondisional bernilai false, maka akan didapatkan ekspresi2



Operator Ternary (?:)

```
#include <stdio.h>

int main(){
    char February;
    int days;
    printf("Jika tahun ini tahun kabisat, masukkan
    nilai 1, jika tahun ini bukan tahun kabisat,
    masukkan sembarang nilai selain 1 : ");
    scanf("%c",&February);

    // Jika (February == 'l') bernilai true, maka
    jumlah hari sama dengan 29.
    // If test condition (February == 'l') bernilai false,
    maka jumlah hari sama dengan 28.
    days = (February == '1') ? 29 : 28;

    printf("Number of days in February = %d",days);
    return 0;
}
```

Output :

Jika tahun ini tahun kabisat, masukkan nilai 1, jika tahun ini bukan tahun kabisat, masukkan sembarang nilai selain 1 : 1
Number of days in February = 29




Program untuk Hasil dan sisa pembagian

```
#include <stdio.h>
int main(){
    int pembagi, dibagi, hasil, sisa;
    pembagi, dibagi, hasil, sisa;
    printf("Enter pembagi: ");
    scanf("%d", &pembagi);
    printf("Enter dibagi: ");
    scanf("%d", &dibagi);
    // Computes hasil
    hasil = pembagi / dibagi;
    // Computes sisa
    sisa = pembagi % dibagi;
    printf("hasil = %d\n", hasil);
    printf("sisa = %d", sisa);
    return 0;
}
```

Output :

```
Enter pembagi: 25
Enter dibagi: 4
hasil = 6
sisa = 1
```



Program untuk menukar Bilangan menggunakan variable Temporary

```
#include <stdio.h>
int main()
{
    double bilanganPertama,
    bilanganKedua,
    temporaryVariable;
    printf("Masukkan bilangan
    pertama: ");
    scanf("%lf",
    &bilanganPertama);
    printf("Masukkan bilangan
    kedua: ");
    scanf("%lf",&bilanganKedua);
```

```
//Nilai dari bilanganPertama dimasukkan ke dalam temporaryVariable
    temporaryVariable = bilanganPertama;
    // Nilai dari bilanganKedua dimasukkan ke dalam bilanganPertama
    bilanganPertama = bilanganKedua;
    // Nilai dari temporaryVariable (yang berisi nilai awal dari
    bilanganPertama) dimasukkan ke dalam bilanganKedua
    bilanganKedua = temporaryVariable;
    printf("\nSetelah ditukar, bilangan pertama = %.2lf\n",
    bilanganPertama);
    printf("Setelah ditukar, bilangan kedua = %.2lf", bilanganKedua);
    return 0;
}
```

Masukkan bilangan pertama: 1.20
Masukkan bilangan kedua: 2.45

Setelah ditukar, bilangan pertama = 2.45
Setelah ditukar, bilangan kedua = 1.20



Program untuk menukar Bilangan tanpa menggunakan variable Temporary

```
#include <stdio.h>                                     // Proses penukaran

int main()
{
    double bilanganPertama, bilanganKedua;

    printf("Masukkan bilangan pertama: ");
    scanf("%lf", &bilanganPertama);

    printf("Masukkan bilangan kedua: ");
    scanf("%lf",&bilanganKedua);

    bilanganPertama = bilanganPertama - bilanganKedua;
    bilanganKedua = bilanganPertama + bilanganKedua;
    bilanganPertama = bilanganKedua - bilanganPertama;

    printf("\nSetelah ditukar, bilangan pertama = %.2lf\n",
    bilanganPertama);
    printf("Setelah ditukar, bilangan kedua = %.2lf", bilanganKedua);

    return 0;
}
```

Output :

Masukkan bilangan pertama: 10.25

Masukkan bilangan kedua: -12.5

Setelah ditukar, bilangan pertama = -12.5

Setelah ditukar, bilangan kedua = 10.25



Yang sudah dipelajari

- Fungsi Input/Output dalam program Bahasa C
- Penggunaan Keyword, Identifier, Variable, Konstanta, Tipe data dasar dalam program Bahasa C.



Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), *Design and Analysis of Computer Experiments*, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2nd edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6th edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall