

# Pemrograman Dasar

Pertemuan IV

# Operator

- Pada ekspresi  $2 + 3$  :  
2 dan 3 disebut **operand**  
+ disebut **operator**
- Ditinjau dari jumlah operand yang dilibatkan, terdapat tiga macam operator :
  - **Operator Unary**  
Operator yang hanya melibatkan sebuah operand. Contoh :  $-1$
  - **Operator Binary**  
Operator yang melibatkan 2 buah operand. Contoh :  $2 + 3$
  - **Operator Tertiary**  
Operator yang melibatkan 3 buah operand. Contoh :  $a > 1 ? 3 : 2$   
  
maksudnya adalah :  
jika nilai  $a > 1$ , kembalikan 3  
jika tidak, kembalikan 2

# Daftar Operator Aritmatika

Operator	Prioritas	Makna	Contoh
-	1	Unary minus	-1
+	1	Unary plus	+1
*	2	Perkalian	$2 * 3 \rightarrow 6$ $2.0 * 3 \rightarrow 6.0$
/	2	Pembagian	$7 / 2 \rightarrow 3$ $7 / 2.0 \rightarrow 3.5$
%	2	Sisa pembagian	$7 \% 2 \rightarrow 1$ $8 \% 3 \rightarrow 2$ $8 \% 2 \rightarrow 0$
+	3	Penjumlahan	$2 + 3 \rightarrow 5$ $2 + 3.0 \rightarrow 5.0$
-	3	Pengurangan	$2 - 3 \rightarrow -1$ $2 - 3.0 \rightarrow -1.0$

# Daftar Operator Pembandingan

- Operator yang berguna untuk melakukan pembandingan antara **dua buah nilai**. Hasil pembandingan : 1 (True) atau 0 (False)
- Daftar Operator Pembandingan

Operator	Keterangan	Contoh Ekspresi	Hasil
>	Lebih besar dari	5 > 1	Benar (1)
>=	Lebih besar atau sama dengan	4 >= 4	Benar (1)
<	Lebih kecil dari	7 < 4	Salah (0)
<=	Lebih kecil atau sama dengan	6 <= 5	Salah (0)
!=	Tidak sama dengan	2 != 3	Benar (1)
==	Sama dengan	4 == 2	Salah (0)

# Daftar Operator Logika

- Operator yang digunakan untuk **membentuk suatu ekspresi pembandingan gabungan dari satu atau dua buah ekspresi pembanding**
- Pemakaian : Ekspresi Pembanding **Operator Logika** Ekspresi Pembanding
- Daftar operator logika

Operator	Keterangan	
&&	Operator “dan” atau AND.	True AND True ➔ TRUE True AND False ➔ FALSE False AND False ➔ FALSE
	Operator “atau” atau OR.	True AND True ➔ TRUE True AND False ➔ TRUE False AND False ➔ FALSE
!	Operator “tidak”	
?:	Operator berkondisi. Mirip if-else	

# Contoh Operator Logika

Ekspresi	Keterangan
<code>kar &gt;= 'A' &amp;&amp; kar &lt;= 'Z'</code>	Bernilai benar, hanya jika variabel <i>kar</i> berisi huruf kapital ('A' – 'Z')
<code>kode == 'a'    kode == 'A'</code>	Bernilai benar, hanya jika variabel <i>kode</i> berisi huruf 'a' atau 'A'
<code>! ( kar == 'A')</code>	Bernilai benar, hanya jika variabel <i>kar</i> berisi nilai selain 'A'

# Prioritas Operator

- Setiap operator dalam sebuah ekspresi memiliki prioritas pengerjaan yang berbeda-beda.
- Misal :  $1 + 4 * 2 = ?$
- Urutannya adalah :  $4 * 2 \rightarrow 8$ , kemudian ditambahkan 1, menjadi 9.
- Misal :  $5 * (5 + (6-2) + 1) ?$
- Hasilnya : 50
- $1 + 4 - 3$ . Nilainya adalah 2. Jika prioritasnya sama, maka dikerjakan **kiri  $\rightarrow$  kanan**
- $a = b = 5$ . Urutan pengerjaannya adalah **kanan  $\rightarrow$  kiri**.  
b diberi nilai 5, kemudian a diberi nilai sama dengan b (yaitu 5)
- Dalam C, juga terdapat operator *increment* dan *decrement*  

int a = 5;	int b = 7;
a--; <b>◀</b> nilai a = 4;	b++; <b>◀</b> nilai b = 8;

Operator	Prioritas
++ – (khusus yang berkedudukan sebagai awalan)	Tertinggi
- (Unary Minus)	
* / %	
+ -	Terendah

# Konversi Tipe Data

- Operasi perhitungan dalam C dilakukan antar operand dengan tipe data yang sama.
- Jika terdapat operand dengan tipe yang berbeda, C / C++ akan mengkonversi sehingga tipe datanya menjadi sama, dengan urutan prioritas sebagai berikut :
  - Long Double
  - Double
  - Float
  - (Char) akan dikonversi menjadi Int
  - Unsigned Long
  - Long
  - Int

- `int jumlah;`  
`float harga_per_unit;`  
`double harga_total;`

`harga_total = harga_per_unit * jumlah;`

maka jumlah akan dikonversi menjadi float, harga per\_unit tetap float  
hasil kali nya, dalam float, kemudian dikonversi ke double (sesuai harga\_total)

- Misal :
  - $0.00005 \times 8 = 0.00040$
  - $8 \times 0.5 = 4.0$
  - dll



# Contoh Kode Konversi

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

```
int main ()  
{  
    char kar = 'A';  
    int nilai;  
    nilai = (int) kar;  
    printf("Nilai ASCII %c : %d", kar, nilai);  
    return 0;  
}
```

```
// konversi.c
```

Bagaimana tampilannya ?

# Komentar

- Komentar atau “comment” biasa digunakan dalam source code untuk memberi keterangan kepada pembaca source code, atau bahkan untuk programernya sendiri
- Komentar bisa ditulis dalam multiple line :  
`/* lokasi komentarnya ada disini  
hingga disini */`
- Atau hanya dalam satu baris saja  
`// lokasi komentar cukup di baris ini saja`
- Misal :  
`printf (“\n”); // ini adalah perintah untuk pindah baris`

# Tipe Struct

- Dalam C dan C++ terdapat **tipe struct** yang dapat dipakai untuk menghimpun **sejumlah data yang berbeda beda**

```
• struct data_pegawai  
  
    {  
  
        int nip;  
  
        char nama[25];  
  
        long int gaji;  
  
    }
```

Dalam hal ini nip, nama dan gaji disebut sebagai **elemen struct** atau **field**

Setelah suatu tipe struct didefinisikan tipe tersebut dapat dipakai untuk mendeklarasikan suatu variabel

Caranya :

```
Struct data_pegawai peg
```

Dengan demikian, variable `peg` mempunyai komposisi `nip, nama, gaji`

mainStruct.cpp

```
1  #include <stdio.h>
2  #include <string.h>
3
4  struct data_pegawai
5  {
6      int nip;
7      char nama[25];
8      long int gaji;
9  };
10
11
12  int main()
13  {
14      struct data_pegawai peg;
15      peg.nip = 1789239;
16      strcpy(peg.nama, "Sekar Rinayung Jagad");
17      peg.gaji = 20000000;
18
19      printf("NIP   : %d\n", peg.nip);
20      printf("Nama  : %s\n", peg.nama);
21      printf("Gaji  : %ld\n", peg.gaji);
22
23      return 0;
24  }
```

# Output

```

NIP : 1789239
Nama : Sekar Rinayung Jagad
Gaji : 20000000
-----
Process exited after 0.01254 seconds with return value 0
Press any key to continue . . .
```

# Operasi Matematika

- C menyediakan berkas math.h untuk digunakan menyelesaikan fungsi terkait operasi matematika.
  - `sqrt (double x)` : akar pangkat kuadrat dari x
  - `pow (double x, double y)` :  $x^y$
  - `log (double x)` : logaritma dari x
  - `exp (double x)` :  $e^x$
  - `abs (long a)` : nilai absolut dari a. outputnya sesuai tipe input
    - `abs (int a)`
    - `abs (float a)`
    - `abs (double a)`
  - `tan (double x)` : nilai tangen dari x
  - `sin (double x)` : nilai sinus dari x
  - `cos (double x)` : nilai cosinus dari x
  - dll

# Contoh Operasi Matematika

- Program menghitung jarak lintas peluru

$$\text{Jarak} = 2 \times v_o^2 \times \sin \alpha \times \cos \alpha / g$$

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

```
int main ()
{
    double kecepatan, sudut, jarak;
    kecepatan = 32;
    sudut = 40;

    jarak = 2 * kecepatan * kecepatan * sin (sudut * 3.14 / 180) *
            cos (sudut * 3.14 / 180) / 9.8;
    printf ("Jarak = %f", jarak);
    return 0;
}
```

```
//peluru.c
```

# Operator Penambahan dan Pengurangan

- Pada C, anda bisa membuat pernyataan berikut :  
     $x = x + 1;$   
     $y = y - 1;$   
    Jika  $x$  awalnya 5, maka nilai  $x$  menjadi 6. Jika  $y$  awalnya 7, maka nilai  $y$  menjadi 6
- Selain dengan operasi di atas, penambahan atau pengurangan 1 angka bisa menggunakan operator ++ dan –
- ++ dan – bisa diletakkan di depan / belakang variabel. Hasilnya akan jauh berbeda

Ekspresi	Nilai x awalnya ..	Setelah ekspresi dieksekusi, hasilnya ..
$y = x++;$	5	$x = 6$ dan $y = 5$
$y = ++x;$	5	$x = 6$ dan $y = 6$
$y = x--;$	5	$x = 4$ dan $y = 5$
$y = --x;$	5	$x = 4$ dan $y = 4$



# Contoh Operator Penambahan dan Pengurangan

```
#include <stdio.h>
int main ()
{
    int x,y;
    x = 5;
    y = 8 - x++;
    printf("Contoh 1 : Nilai x = %d | y = %d \n", x, y);

    x = 5;
    y = 8 - ++x;
    printf("Contoh 2 : Nilai x = %d | y = %d\n", x, y);

    x = 5;
    y = 8 - x--;
    printf("Contoh 3 : Nilai x = %d | y = %d\n", x, y);

    x = 5;
    y = 8 - --x;
    printf("Contoh 4 : Nilai x = %d | y = %d\n", x, y);

    return 0;
}
```

# Operator Penugasan

- Operator penugasan berguna untuk memberikan nilai ke suatu variabel.

- Macam :

= Pemberian nilai

+= Penambahan bilangan

-= Pengurangan bilangan

\*= Pengalian bilangan

/= Pembagian bilangan

%= Pemerolehan sisa bagi bilangan

- Misal :

a = 2; b=3; c=4; d=5;

a += 2; ➡ a bernilai 4

b -= 1; ➡ b bernilai 2

c /= 2; ➡ c bernilai 2

d %=2; ➡ d bernilai 1

# Menangani Pemasukan Data

- Pada bahasa C, pemasukan data menggunakan header `stdio.h` dan perintah `scanf()`, dengan format :  
`scanf("string kontrol", argumen);`
- String kontrol adalah gabungan dari:
  - penentu format
  - karakter spasi-putih (spasi, newline, tab)
  - karakter bukan-spasi-putih

Kode	Keterangan
%c	Data yang diinput adalah sebuah karakter
%s	Data yang diinput adalah sebuah string
%i atau %d	Data yang diinput adalah sebuah integer
%e atau %f	Data yang diinput adalah sebuah bilangan real
%u	Data yang diinput adalah sebuah bilangan real tidak bertanda
l	Awalan untuk membaca data long int atau long doubl

# Contoh Penggunaan scanf( )

Pernyataan	Keterangan
scanf("%f", &jarak);	Perintah membaca sebuah bilangan real dan meletakkannya ke variabel jarak yang bertipe float
scanf("%d", &jumlah);	Perintah membaca sebuah bilangan integer dan meletakkannya ke variabel jumlah yang bertipe integer
scanf("%ld", &jum_penduduk);	Perintah membaca sebuah bilangan longint dan meletakkannya ke variabel jum_penduduk yang bertipe long integer

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

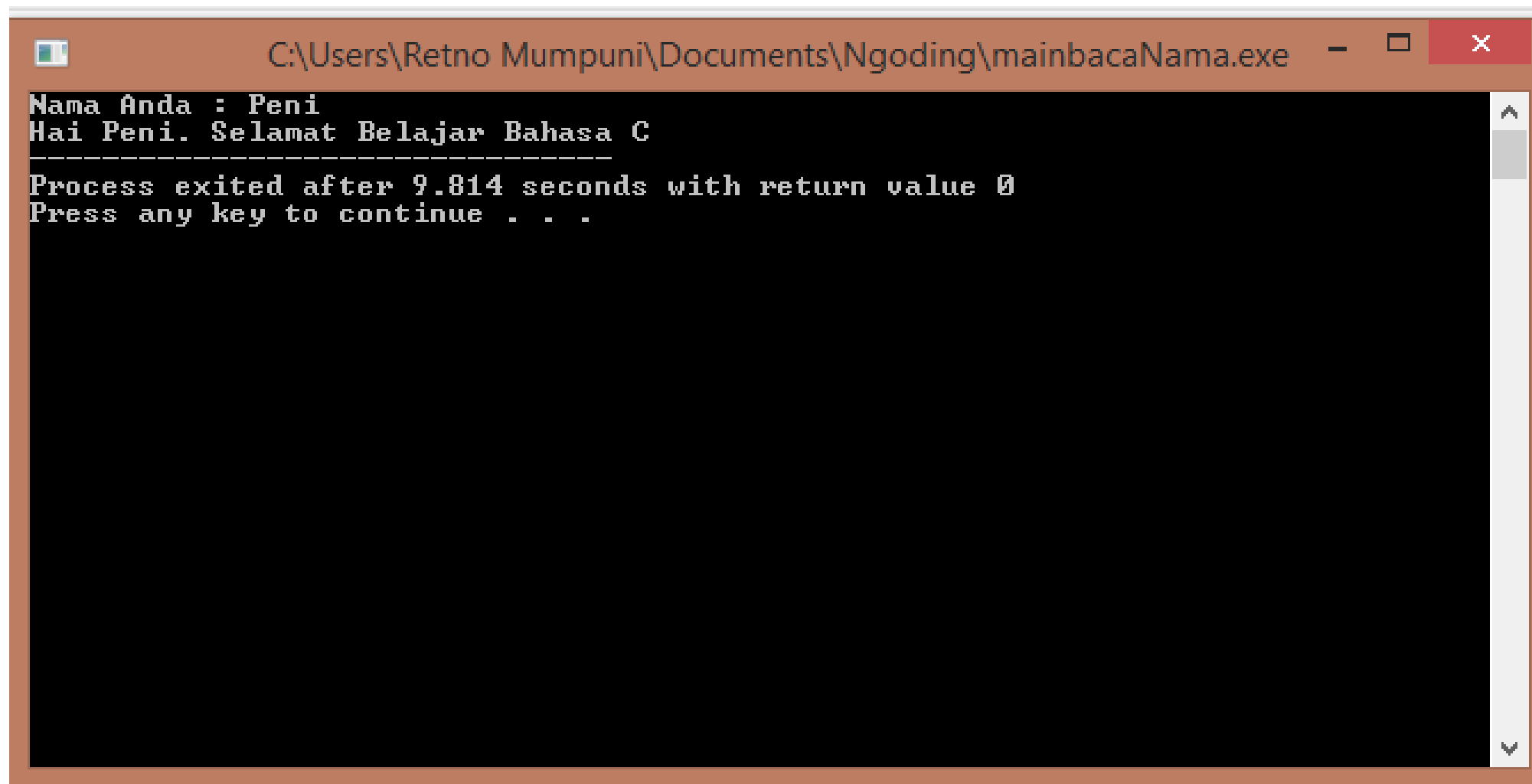
```
int main ( )
{
    double kecepatan, sudut, jarak;
    printf("Kecepatan: ");
    scanf("%lf", &kecepatan);
    printf("Sudut: ");
    scanf("%lf", &sudut);
    jarak = 2 * kecepatan * kecepatan * sin(sudut * 3.14 / 180) * cos(sudut * 3.14 / 180) / 9.8;
    printf("Jarak = %f", jarak);
    return 0;
}
//peluru2.c
```

# Fungsi gets

- C menyediakan fungsi bernama `gets()` untuk memasukkan data string dari keyboard
- `gets (variabel_string) ;`
- Bentuk diatas setara dengan
- `Scanf ("%s", variabel_string) ;`
- Prototipe fungsi **`gets()`** terdapat pada berkas header **`stdio.h`**

# bacanama.c

- `#include <stdio.h>`
- `int main()`
- `{`
- `char nama[35]`
- `printf( " nama anda : ");`
- `gets(nama);`
- `printf("Hai,%s. selamat belajar C\n",nama);`
- `return 0;`
- `}`



```
C:\Users\Retno Mumpuni\Documents\Ngoding\mainbacaNama.exe  
Nama Anda : Peni  
Hai Peni. Selamat Belajar Bahasa C  
-----  
Process exited after 9.814 seconds with return value 0  
Press any key to continue . . .
```

# Latihan Soal

- Apabila `kar` bernilai 'C', apakah hasil ekspresi berikut :
  - `kar == 'A'`
  - `kar >= 'C'`
  - `kar > 'D'`
  - `kar >= 'a' && kar <= 'z'`
  - `kar >= 'A' && kar <= 'Z'`
  - `kar == 'A' || kar == 'B' || kar == 'C'`
- Buatlah program C untuk membaca input 2 buah bilangan bulat dari keyboard user dan menampilkan hasil penjumlahan keduanya
- Buatlah program C yang membaca data dua buah jam (jam dan menit), kemudian hitunglah selisih kedua jam tersebut dalam menit. Bentuk dialognya :

Jam 1 :  
Menit 1 :  
Jam 2 :  
Menit 2 :

Selisih kedua jam tersebut adalah. .... menit