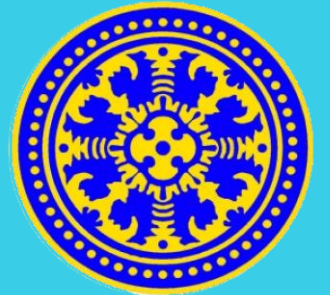


Algoritma dan Pemrograman

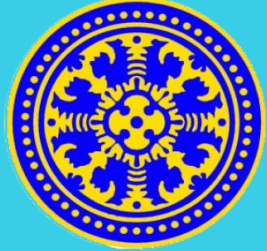


Topik 3 - Tipe Data, Operator dan Runtunan

I Dewa Made Bayu Atmaja Darmawan, S.Kom.M.Cs.

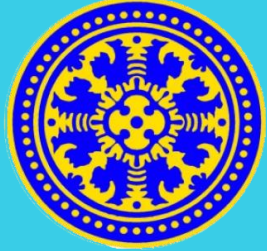
Kuliah Daring SPADA Indonesia

Capaian Pembelajaran



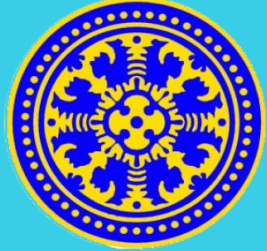
- Mahasiswa dapat mengenal dan membedakan tipe-tipe data dasar
- Mahasiswa dapat memahami penggunaan tipe-tipe data dasar dalam program
- Mahasiswa dapat memahami operator dan penggunaannya dalam program
- Mahasiswa dapat memahami konsep runtunan dalam program

Tipe Data Dasar



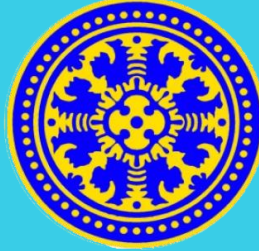
- Tipe data adalah himpunan nilai yang dapat dimiliki oleh sebuah data.
- Tipe data menentukan apakah sebuah nilai dapat dimiliki sebuah data atau tidak, serta operasi apa yang dapat dilakukan pada data tersebut.
- Tipe data dasar adalah tipe data yang dapat langsung digunakan.

Jenis Tipe Data Dasar



- Tipe data numerik terdiri atas angka/kumpulan angka serta dapat mengalami operasi perhitungan
- Tipe data kategorik dapat berupa angka maupun huruf namun tidak dapat mengalami operasi perhitungan.

Contoh Tipe Data Dasar



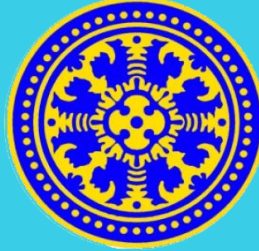
1. Integer/ bilangan bulat

- Merupakan tipe data dasar berupa bilangan yang tidak mengandung pecahan desimal. Tipe data ini juga memiliki urutan, sehingga dapat dibandingkan satu dengan lainnya.
- Tipe data integer tidak memiliki batasan, yaitu dari minus tak hingga hingga plus tak hingga.

Tabel 1. Tipe data integer

Tipe	Ukuran	Nilai
Short	8 bit	-128 .. 127
Int	16 bit	-32768 .. 32767
Long	32 bit	-2147483648 .. 2147483647

Contoh Tipe Data Dasar



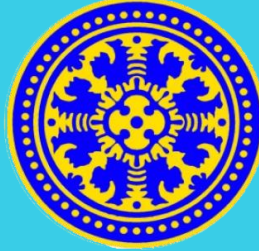
2. Real/ bilangan riil

- Real adalah tipe data dasar berupa bilangan yang memiliki pecahan desimal.
- Nilai dengan tipe data ini harus ditulis dengan sebuah titik sebagai pemisah bilangan utuh dan bilangan pecahannya.
- Digunakan untuk perhitungan yang melibatkan bilangan pecahan
- Tipe data real tidak memiliki batasan, yaitu dari minus tak hingga hingga plus tak hingga

Tabel 2. Tipe data real

Tipe	Ukuran	Nilai
float	32 bit	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$
Double	48 bit	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$

Contoh Tipe Data Dasar

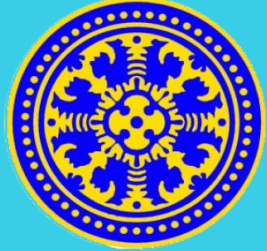


3. Char/ Karakter

- Char adalah tipe data dasar yang terdiri atas satu buah angka, huruf, tanda baca atau karakter khusus.
- Penulisan tipe data char diawali dan diakhiri dengan tanda kutip ganda.
- Terdapat sebuah karakter kosong yang disebut dengan null atau nil dan dituliskan sebagai “”.

Contoh char: “5” “A” “?” “+” “\$”

Contoh Tipe Data Dasar



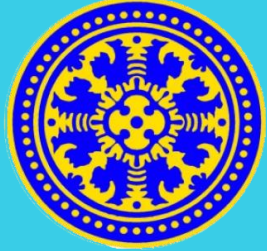
4. String

- String adalah tipe data dasar yang berupa kumpulan karakter dengan panjang tertentu.
- Penulisan sebuah string juga harus diawali dan diakhiri dengan tanda petik ganda
- String juga mengenal null yang dituliskan dengan “”

Contoh string:

- “BANDUNG”
- “ABC3456”
- “Lucu”
- “z”

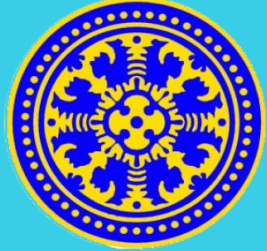
Contoh Tipe Data Dasar



5. Boolean/ bilangan logika

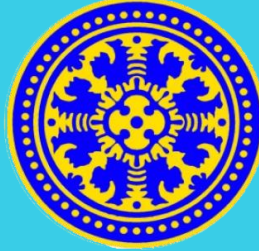
- Sebuah data boolean memiliki tepat dua buah kemungkinan nilai, direpresentasikan sebagai Benar dan Salah, atau True dan False, atau dapat juga dilambangkan dengan 1 dan 0.
- Digunakan untuk pemilihan dengan kondisi-kondisi tertentu, dimana program harus memilih aksi apa yang akan dijalankan dengan parameter tertentu.
- Tipe data ini paling sering digunakan untuk range yang memiliki dua buah nilai: lulus - tidak lulus, member - bukan member,

Variabel



- Variabel atau peubah adalah obyek yang nilainya dapat berubah-ubah dalam sebuah program.
- Pemrogram dapat memberikan nama pada sebuah variabel untuk mempermudah pemanggilan variabel tersebut di dalam program.
- Pada saat mendeklarasikan sebuah variabel, pemrogram harus menyebutkan nama variabel dan tipe data dari variabel tersebut

Contoh pendeklarasian variabel



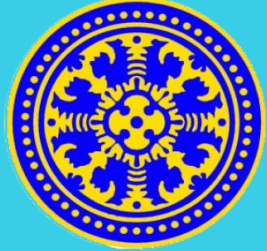
Contoh deklarasi variabel dalam pseudeocode :

```
1. KAMUS DATA {awal deklarasi variabel}
2.     x : integer
3.     nama: string
4.     TB : real
5.     jenisKelamin : char
6.     status : boolean
```

Contoh deklarasi variabel dalam program

```
1.  // Contoh Program C++
2.  #include <stdio.h>
3.  /* Program Utama */
4.  main() {
5.      printf("Selamat Datang");
6.      return 0;
7.  }
```

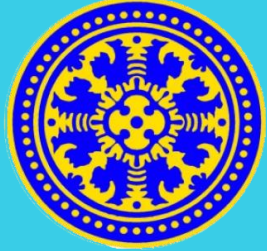
Contoh pendeklarasian variabel



Keterangan :

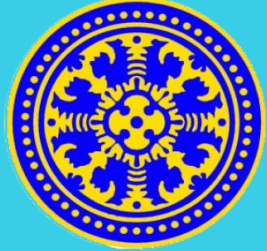
- Tanda doublelash (//) adalah sebagai komentar, artinya baris tersebut tidak akan dieksekusi oleh program. Berlaku hanya untuk satu baris
- Tanda /* */ juga merupakan komentar dimana komentar dituliskan diantara tanda /* dan */. Tanda ini digunakan untuk penulisan komentar lebih dari satu baris
- Kode #include <stdio.h>, yang diawali dengan tanda crash (#) dapat kita sebut dengan preprocessor directive, merupakan perintah-perintah untuk memberitahukan kepada compiler untuk melakukan berbagai macam definisi seperti menggunakan (include) file librari misalnya stdio.h, karena didalam file tersebut mengandung beberapa fungsi yang akan digunakan didalam program.

Contoh pendeklarasian variabel



- Instruksi main() dimana pada program tersebut merupakan fungsi utama atau program utama. Merupakan penanda awal dari eksekusi sebuah program. Untuk awal instruksi ditandai dengan kurung kurawal.
- Printf("Selamat Datang") merupakan instruksi untuk mencetak tulisan "Selamat Datang" kelayar.
- Return 0 merupakan nilai kembali dari fungsi utama yaitu nilainya adalah 0
- Setiap instruksi pada program harus diakhiri dengan tanda semicolon (;).

Panduan Pemrogram dalam Penamaan Variabel



Beberapa panduan yang biasa diacu pemrogram dalam penamaan variabel, antara lain:

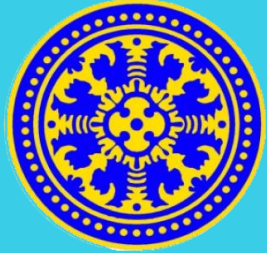
- Huruf pertama pada nama variabel menunjukkan tipe data dari variabel.

Contoh: diawali dengan 'c' untuk variabel char, 'i' untuk integer, 's' untuk string, dan seterusnya.

- Nama variabel harus cukup jelas menunjukkan tujuan penggunaan variabel tersebut.

Contoh: sNama adalah variabel string untuk menyimpan nama, cJenisKelamin adalah variabel char untuk menyimpan jenis kelamin, bStatus adalah variabel boolean untuk menyimpan status.

Panduan Pemrogram dalam Penamaan Variabel



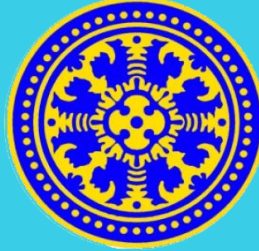
- Nama variabel tidak boleh mengandung spasi kosong atau karakter khusus ! @ # \$ % ^ & * () { } [] ' " ; : < > , . / ? | dan \. Beberapa pemrogram menggunakan '_' untuk memisahkan kata di nama variabel.

Contoh: cJenis_kelamin, sNama_orang_tua, iNilai_akhir

- Cara lain untuk memisahkan kata dalam nama variabel adalah dengan memberikan huruf besar di awal tiap kata.

Contoh: cJenisKelamin, sNamaOrangTua, iNilaiAkhir

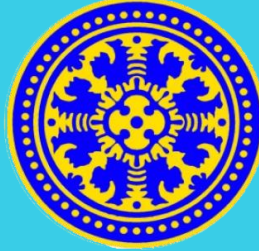
Contoh Program



- Contoh program untuk memberikan nilai pada sebuah variabel :

```
1. #include <stdio.h>
2. main() {
3.     int lA,lB;
4.     String NamaA, NamaB;
5.     // Pengisian secara Langsung
6.     lA = 20;
7.     NamaA = "Joko Handono";
8.     // Pengisian dengan Inputan
9.     scanf("%i",&lB);
10.    scanf("%s",&NamaB);
11.    // Menampilkan Kelayar
12.    printf("Nilai lA : %i",lA);
13.    printf("Nilai lB : %i",lB);
14.    printf("Nilai NamaA : %s",NamaA);
15.    printf("Nilai NamaB : %s",NamaB);
16. }
```

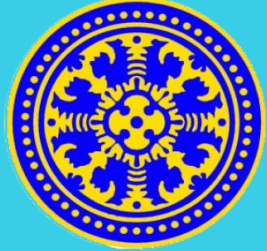

Konstanta



- Pada variabel, nilai yang disimpan dapat berubah-ubah selama program dijalankan. Sedangkan pada konstanta, nilai yang disimpan tetap dan tidak dapat diubah sejak dideklarasikan hingga program berakhir.
- Cara penulisan konstanta didalam program, di tulis dengan diawali dengan tanda crash (#) kemudian diikuti dengan define, selanjutnya nama konstantanya dan selanjutnya nilainya dan ditulis diluar program utama setelah pendeklarasian librari namespace.

```
1. #include <stdio.h>
2. #define iMaxs 100
3. #define fPi 3.14159
4. #define sSapa 'Hello'
5. #define newLine '\n'
6. main() {
7.     ...
8. }
```

Operator



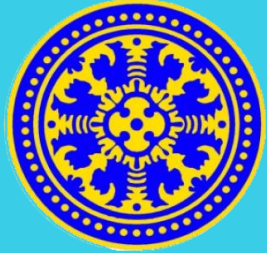
- Operator adalah pengendali operasi yang akan dilakukan pada beberapa operan sehingga membentuk sebuah ekspresi.
- Dalam sebuah ekspresi terdapat sebuah operator yang diapit dua operan.

Contohnya pada ekspresi:

$$x + y$$

Dimana x dan y adalah operan, sedangkan '+' adalah operatornya

Macam - macam operator



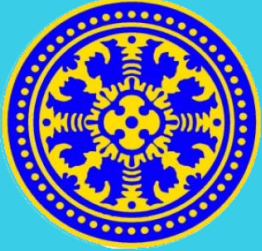
1. Operator aritmatik

- Operator ini membentuk perhitungan aritmatik. Kedua operan dari operasi aritmatik ini dapat berupa nilai integer atau real.
- Sebuah operator aritmatik tidak dapat diterapkan pada dua bilangan dengan tipe data yang berbeda.
- Output dari operasi aritmatik akan memiliki tipe data yang sama dengan tipe data kedua operannya. Misalnya, jika sebuah bilangan integer dijumlahkan dengan bilangan integer lainnya maka outputnya adalah bilangan integer juga

Tabel 3. Operator aritmatik

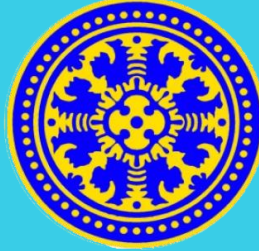
Lambang	Deskripsi	Contoh
+	Penjumlahan	$x = y + z$
-	Pengurangan	$x = y - z$
*	Perkalian	$x = y * z$
/	Pembagian	$x = y / z$
%	Modulo (sisanya)	$x = y \% z$

Contoh Program



```
1  //Program Aritmatik
2  /* IS:Tersedia dua buah bilangan integer
   FS:Hasil Modulo duabuah bilangan */
3  #include <stdio.h>
4
5  main () {
6      // Deklarasi Variabel
7      int iTambah;
8
9      int iAngka1, iAngka2;
10     printf("Masukan Bilangan Pertama : ");
11     scanf("%i", iAngka1);
12     printf("Masukan Bilangan Kedua : ");
13     scanf("%i", iAngka2);
14     // Penjumlahan
15     iTambah = iAngka1 + iAngka2;
16     printf("Hasil Penjumlahan %i + %i = %i",
17           iAngka1, iAngka2, iTambah);
18     return 0;
19 }
```

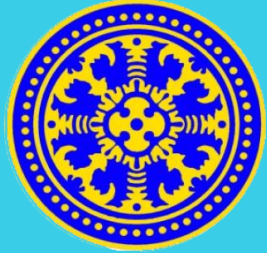

Contoh Program



- Program di atas akan mengembalikan nilai hasil penjumlahan sesuai dengan inputan. Misalnya pada inputan pertama kita masukan 10 dan yang kedua kita masukan 23 maka hasilnya adalah 33. Outputnya adalah:

```
Masukan Bilangan Pertama : 10  
Masukan Bilangan Kedua   : 23  
Hasil Penjumlahan 10 + 23 = 33
```

Macam - macam operator



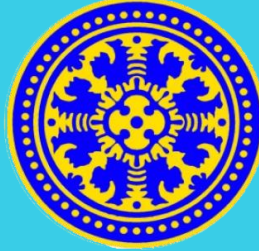
2. Operator Assignment

- Dalam pemrograman bahasa C++, Operator ini digunakan memasukan nilai kedalam sebuah variabel, tanpa menghilangkan atau mengosongkan nilai variabel sebelumnya.

Tabel 4. Operator relasional

Lambang	Deskripsi	Contoh
+=	Menambahkan	x += 1
-=	Mengurangkan	x -= 1
*=	Mengalikan	x *= 2
/=	Membagi	x /= 2
%=	Mem-mod	x %= 2

Macam - macam operator

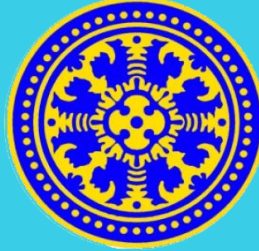


3. Increase and decrease

- Penulisan ini dilambangkan dengan ++ (Increase) dan -- (decrease). Operator ini berfungsi untuk menaikkan atau menurunkan satu satuan nilai pada sebuah variabel.
- Ada dua macam penulisan operator ini, yaitu simbol dapat ditulis sebelum nama variabel dan setelah variabel.

1	B = 3;	1	B = 3;
2	A = ++B;	2	A = B++;
3	// A = 4, B = 4	3	// A = 3, B = 4

Macam - macam operator



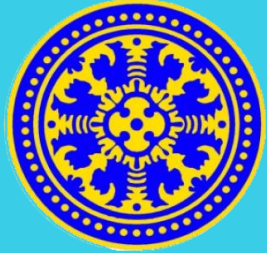
4. Operator relasional

- Operator ini membandingkan dua operan dan hasilnya berupa nilai boolean (BENAR atau SALAH).
- Operasi relasional dapat dilakukan pada dua nilai dengan tipe data yang sama: tipe data integer, riil, char, string, maupun boolean.

Tabel 4. Operator relasional

Lambang	Deskripsi	Contoh
<code>==</code>	Sama dengan	<code>x == y</code>
<code>!=</code>	Tidak sama dengan	<code>x != y</code>
<code>></code>	Lebih dari	<code>x > y</code>
<code><</code>	Kurang dari	<code>x < y</code>
<code>>=</code>	Lebih dari atau sama dengan	<code>x >= y</code>
<code><=</code>	Kurang dari atau sama dengan	<code>x <= y</code>

Macam - macam operator



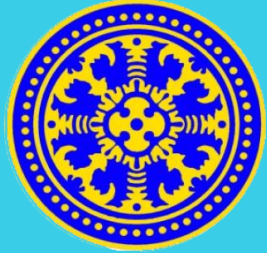
- Operator logika
- Operator logika adalah operator yang digunakan untuk mengkombinasikan hasil ekspresi yang mengandung operator relasional.

Tabel 5. Operator logika

Lambang	Deskripsi	Contoh
&&	And / Dan	$x > 7 \ \&\& \ x = y$
 	Or / Atau	$x \neq y \ \ x > 3$
!	Not / Tidak	$! (x > y)$

- Pola penggunaan operator logika adalah:
ekspresi1 OPERATOR ekspresi2

Macam - macam operator



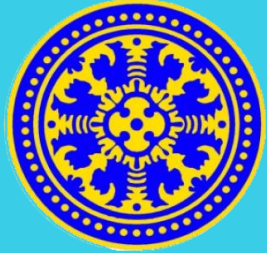
- Output dari penggunaan operator AND dan OR adalah sebagai berikut:

Tabel 6. Output operator logika

ekspresi1	ekspresi2	kombinasi dengan	
		AND	OR
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

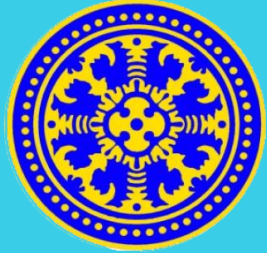
- Pola yang mudah untuk mengingat output kedua operator logika tersebut adalah: True AND True = True, False OR False = False.

Macam - macam operator



- Beberapa contoh penggunaan operator logika:
 - $(x > 7) \ \&\& \ (x = y)$
Jika ternyata nilai x adalah 8 dan y adalah 5, maka
 $(8 > 7) \ \&\& \ (8 = 5)$
True AND False
False (output operasi)
 - $(x \neq y) \ || \ (x > 3)$
Jika ternyata nilai x adalah 4 dan y adalah 4, maka
 $(4 \neq 4) \ || \ (4 > 3)$
False OR True
True (output operasi)

Macam - macam operator



- NOT ($x > y$)

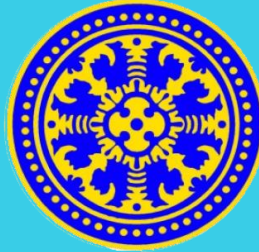
Jika ternyata nilai x adalah 3 dan y adalah 3, maka

NOT ($3 > 3$)

NOT (False)

True (output operasi)

Urutan Operasi

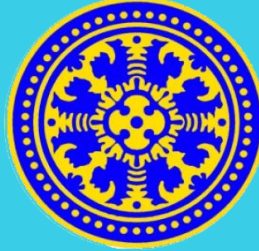


- Sebuah ekspresi mungkin terdiri atas beberapa operasi sekaligus. Misalnya:

$$iHasil = x * 2 \% 2 > y \&\& (x != 3)$$

- Untuk menentukan operasi mana yang dilakukan terlebih dahulu daripada operasi lainnya, setiap operator memiliki level urutan
- Operator yang memiliki level lebih tinggi (ditunjukkan dengan angka yang semakin kecil) akan dioperasikan terlebih dahulu dibandingkan operator lain yang levelnya lebih rendah.
- Sedangkan pada operator-operator yang berada pada level yang sama, operasi dilakukan secara berurutan dari kiri ke kanan. Hal ini disebut asosiativitas

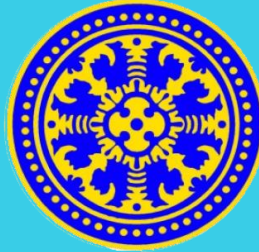
Urutan Operasi



Tabel 7. Urutan operasi

Operator	Deskripsi	Asosiativitas	Level Urutan
()	Tanda kurung		1
!	Logika NOT		2
*	Perkalian		
/	Pembagian	Kiri ke kanan	3
%	Modulo		
+	Penjumlahan	Kiri ke kanan	4
-	Pengurangan		
<	Kurang dari	Kiri ke kanan	5
<=	Kurang dari/ sama dengan		
>=	Lebih dari/ sama dengan		
>	Lebih dari		
=	Sama dengan	Kiri ke kanan	6
!=	Tidak sama dengan		
&&	Logika AND	Kiri ke kanan	7
	Logika OR	Kiri ke kanan	8

Urutan Operasi



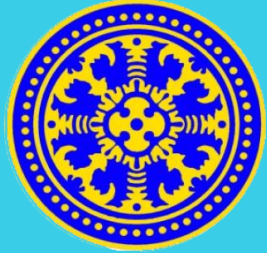
- Misalnya pada ekspresi berikut ini:

`iHasil = x * 2 % 2 > y && (x != 3)`

- Jika inputannya adalah `x = 5` dan `y = 3` maka urutan pengerjaannya adalah:

Level	Pengerjaan
1	<code>iHasil ← x * 2 % 2 > y && (5 != 3)</code> <code>iHasil ← x * 2 % 2 > y && True</code>
3	<code>iHasil ← 5 * 2 % 2 > y && True</code> <code>iHasil ← 0 > y && True</code>
5	<code>iHasil ← 0 > 3 && True</code> <code>iHasil ← False && True</code>
7	<code>iHasil ← False</code>

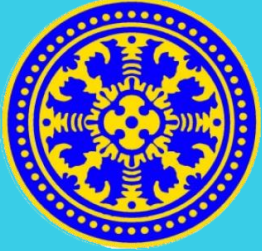
Runtunan



- Secara umum, program akan dibaca dan dieksekusi secara berurutan baris demi baris. Misalnya pada algoritma berikut ini:

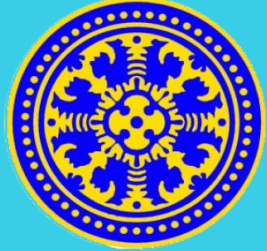
```
1.  Algoritma Runtunan;  
2.  {IS:Tersedia empat bilangan yang akan  
    dioperasikan  
    FS:Mengoutputkan dua bilangan setelah  
    dioperasikan }  
3.  Kamus data  
4.    a,b,c,d : integer  
5.  BEGIN  
6.    a ← 3  
7.    b ← 2  
8.    c ← a * b  
9.    a ← 5
```


Contoh Program



```
1. // Program Runtunan;  
2. /*IS:Tersedia empat bilangan yang akan  
   dioperasikan  
   FS:Menampilkan dua bilangan setelah  
   dioperasikan */  
3. #include <stdio.h>  
4.  
5. main () {  
6.     int a,b,c,d;  
7.     a = 3;  
8.     b = 2;  
9.     c = a * b;  
10.    a = 5;  
11.    d = a + b;  
12.    printf("Nilai C : %i",c);  
13.    printf("Nilai D : %i",d);  
14. }
```

Urutan Operasi

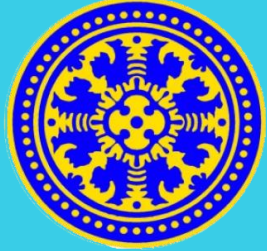


- Jika Program dijalankan, maka hasil keluaran program adalah seperti berikut :

Nilai C : 6

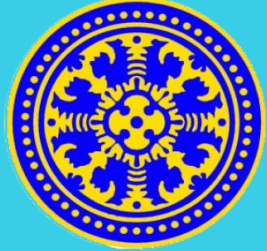
Nilai D : 7

Rangkuman

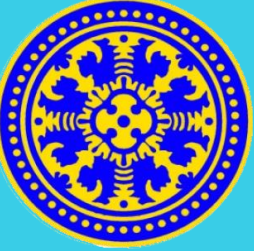


1. Tipe data dasar adalah tipe data yang dapat langsung digunakan, dan memiliki ukuran tertentu sesuai dengan tipe data masing-masing. Beberapa tipe data dasar: integer, real, char, string, dan boolean.
2. Variabel adalah obyek yang nilainya dapat berubah-ubah dalam sebuah program, sedangkan konstanta memiliki nilai yang tetap sejak dideklarasikan hingga program berakhir.
3. Pada saat mendeklarasikan sebuah variabel, pemrogram harus menyebutkan nama variabel dan tipe data dari variabel tersebut.
4. Operator adalah pengendali operasi yang akan dilakukan pada beberapa operan sehingga membentuk sebuah ekspresi. Tiga macam operator dalam pemrograman: operator aritmatik, operator relasional, dan operator logika.

Rangkuman



5. Penggunaan beberapa operator sekaligus dalam sebuah ekspresi mengakibatkan perlunya menentukan urutan operasi yang diatur dalam level urutan operator.
6. Operator dengan level yang lebih tinggi akan dieksekusi lebih dahulu daripada operan dengan level yang lebih rendah. Operator-operator yang memiliki level yang sama akan dieksekusi menurut urutan penulisan (dari kiri ke kanan).



Pertanyaan?