

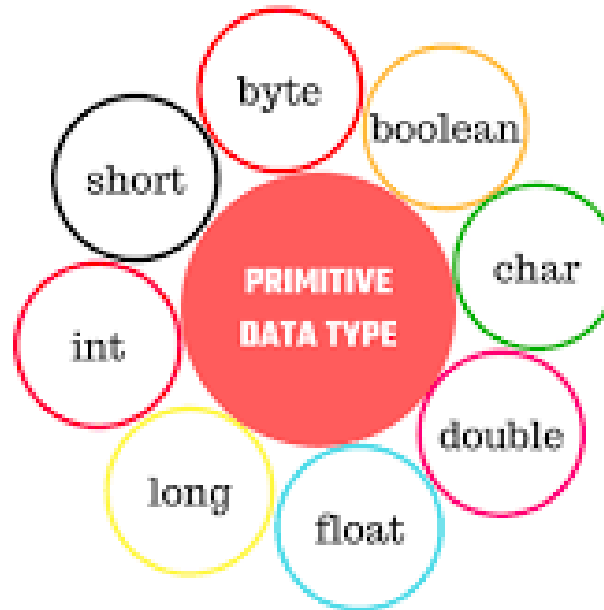
Type Data

# Tipe Data

- Tipe Data dapat dikelompokkan menjadi atas dua macam tipe yaitu :
  1. Tipe Dasar
  2. Tipe Bentuk.
- Tipe Dasar adalah tipe yang langsung dapat dipergunakan, sedangkan
- Tipe bentuk dibangun dari tipe dasar atau dari tipe bentuk lain yang sudah didefinisikan terlebih dahulu

# Tipe Data Dasar

1. bilangan logik,
2. bilangan bulat,
3. bilangan riil,
4. karakter.



# Bilangan Logik

- Bilangan logik hanya mengenal 2 macam nilai yaitu :
  - True ( Benar )
  - False ( Salah )
- Bilangan logik sering juga dinamakan sebagai bilangan Boolean. Istilah bilangan untuk bilangan logik ini karena kondisi Benar dapat dinyatakan dengan bilangan 1 dan Salah dinyatakan dengan bilangan 0.
- Tipe bilangan logik didefinisikan sbb :
  - Nama : boolean
  - Daerah : False, True
  - Misal : False, True
  - Operator : not, and, or, xor

# Bilangan Logik

- And hanya akan bernilai benar jika kedua-duanya benar
- Or hanya akan bernilai salah jika kedua-duanya salah
- Xor hanya bernilai salah jika kedua-duanya sama
- Contoh :
  - A = True
  - B = True
  - Not A = False
  - Not A and B = False
  - A Or Not B = True

# Operasi bilangan logik

- Operasi dengan bilangan logik menghasilkan bilangan logik.
- Jika a dan b adalah variable yang bertipe boolean, maka hasil operasi a dan b dengan operator boolean tersebut adalah seperti dalam tabel di bawah ini :

a	b
true	false
false	true

a	b	a and b	a or b	a xor b
True	True	True	True	False
True	False	False	True	True
False	True	False	True	True
False	False	False	False	False

# Bilangan Bulat

- Bilangan bulat adalah bilangan yang tidak mengandung bilangan pecahan dalam penyajiannya. Misalnya : 64, 94775, - 646, 0, dsb.
- Dalam algoritma, tipe bilangan bulat didefinisikan sbb :
- Nama : Integer
- Daerah : I
- Misal : 64, - 58, 38456, 0, 3444
- Operator Aritmatik
  - + tambah
  - - kurang
  - \* kali
  - / bagi
  - div hasil bagi
  - mod sisa bagi

# Bilangan Bulat

- Operator Relational / Perbandingan
  - $<$  lebih kecil
  - $\leq$  lebih kecil sama dengan
  - $>$  lebih besar
  - $\geq$  lebih besar sama dengan
  - $=$  sama dengan
  - $\neq$  tidak sama dengan



# Bilangan Bulat

- Ada 5 tipe data yang termasuk dalam kelompok integer ini. Kelima tipe data tersebut adalah :

Tipe	Batas Nilai	Ukuran dalam byte
Byte	0..255	1
Shortint	-128..127	1
Integer	-32.768..32.767	2
Word	0..65.535	2
Longint	-2.147.483.648.. 2.147.483.647	4

# Bilangan Bulat

- Contoh operasi aritmatika bilangan bulat :

- 13 + 10 hasil 23
- 86 - 16 hasil 70
- 2 \* 8 hasil 16
- 48 div 3 hasil 16
- 48 mod 3 hasil 0
- 45 mod 3 hasil 0
- 46 mod 3 hasil 1

# Bilangan Bulat

- Operasi aritmatik dengan operator relational menghasilkan nilai boolean (benar atau salah).
- Contoh operasi relasional :
  - 5    ☐    6    benar
  - 45   ☐    45   benar
  - 56   ☐    56   salah
  - 35   ☐    35   benar
  - 36   ☐    36   benar
  - 38   ☐    58   benar

# Bilangan Riil

- Bilangan riil adalah bilangan yang menggunakan pecahan dalam penyajiannya. Misal : 0.24, 3.78, 23.456, 345.567, dll
  - Tipe bilangan riil didefinisikan sbb :
    - Nama : real
    - Daerah : R
    - Misal : 0.      68.      37.4      67.45      0.3456      6E-6      -4.6
    - Operator :
- Operator Aritmatik
  - +              tambah
  - -              kurang
  - \*              kali
  - /              bagi

# Bilangan Riil

- Operator Relational / Perbandingan
- $<$  lebih kecil
- $\leq$  lebih kecil sama dengan
- $>$  lebih besar
- $\geq$  lebih besar sama dengan
- $\neq$  tidak sama dengan

# Bilangan Riil

- Semua bilangan riil harus mengandung ‘.’ ( titik ).
- Nilai 9 dianggap bilangan bulat, tetapi 9. dianggap bilangan riil.
- Bilangan riil dapat juga dituliskan dengan notasi E yang artinya perpangkatan sepuluh.
- Misalnya contoh 6E-6 artinya  $6 \times 10^{-6}$ .
- Bilangan Bulat ataupun bilangan Riil keduanya dinamakan tipe numerik.

# Bilangan Riil

- Ada 5 tipe data yang termasuk dalam kelompok real ini. Kelima tipe data tersebut adalah :

Type	Batas Nilai	Angka Signifikan	Ukuran dalam byte
Real	$2.9 \times 10E-39.. 1.7 \times 10E-38$	11 – 12	6
Single	$1.5 \times 10E-45.. 3.4 \times 10E-38$	7 – 8	4
Double	$5.0 \times 10E-324.. 1.7 \times 10E-308$	15 – 16	8
Extended	$1.9 \times 10E-4951.. 1.1 \times 10E-4932$	19 – 20	10
Comp	$-2E63 + 1.. -2E63 -1$	19 – 20	8

# Bilangan Riil

- Operasi aritmatik dengan salah satu operand-nya bertipe bilangan riil dinamakan operasi campuran dan menghasilkan nilai dalam daerah bilangan riil.

- Contoh operasi aritmatik bilangan riil :

- $7.5 + 4.7$  hasil 13.2
- $8.0 - 3.7$  hasil 4.3
- $9 / 4$  hasil 2.25
- $15 / 2.5$  hasil 6.0

- Seperti halnya pada tipe bilangan bulat, operasi relasional dengan bilangan riil menghasilkan nilai boolean.

- 

- Contoh operasi relasional :

- $0.067 < 34$  benar
- $9.70 \geq 5.6$  benar
- $8.76 \neq 8.76$  salah



# Bilangan Riil

- Operasi bilangan riil tidak mengenal operator “ = ” karena bilangan riil tidak dapat direpresentasikan secara benar oleh komputer. Misalnya  $1/3 = 0.3333\dots$  dengan angka 3 yang tidak pernah berhenti. Biasanya dalam perhitungan komputer dibulatkan sesuai dengan batasan jumlah digit dibelakang koma yang dipakai. Namun demikian,  $1/3$  tidaklah sama dengan  $0.3333$ .
- Dengan alasan ini maka operator “ = ” tidak digunakan untuk operasi bilangan riil.

# Karakter

- Karakter adalah data yang bertipe string dimana terdiri satu simbol saja. Misal : 0, 5, g, H, %, &, dll.
- Tipe Karakter didefinisikan sbb :
- Nama : char
- Daerah : ( 0,1, ..,9, a, b, c, .. ,z, A,B, .., Z, ' ', !, @, #, \$, %, dan karakter khusus lainnya. ( symbol ASCII )
- Misal : 'h', 'Y', '1', \* , dll.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9			
.	,	:	;	@	#	'	!	"	/	?	<	>
%	&	*	(	)		\$						

# Karakter

- Operator Relational / Perbandingan
  - < lebih kecil
  - <= lebih kecil sama dengan
  - > lebih besar
  - >= lebih besar sama dengan
  - = sama dengan
  - ≠ tidak sama dengan

# Karakter

- Seperti pada tipe bilangan bulat, karakter mempunyai keterurutan (successor / sesudah dan predecessor / sebelum ) yang ditentukan cara pengkodeannya di dalam komputer.
- Contoh operasi character :
  - A = a salah
  - F = F benar
  - K > B benar
  - Q < D salah

# TIPE DATA BENTUKAN

- Tipe bentukan dibangun dari beberapa elemen yang bertipe dasar atau dari tipe bentukan lain yang sudah didefinisikan terlebih dahulu.
- Operasi terhadap elemen bertipe dasar dilakukan seperti halnya pada tipe dasar. Tipe bentukan seringkali disebut juga tipe terstruktur . Tipe bentukan diberi nama oleh programmer.

# STRING

- String adalah deretan karakter dengan panjang tertentu. Tipe string didefinisikan sbb :
- Nama : string
- Daerah : deretan karakter yang didefinisikan pada daerah karakter
- Misal : 'BANDUNG', 'UNJANI', 'Jl. Terusan Jenderal Sudirman'
- Semua string harus diapit oleh tanda petik tunggal.

# STRING

a) Penyambungan      +

b) Relasional

< lebih kecil

$\leq$  lebih kecil sama dengan

> lebih besar

$\geq$  lebih besar sama dengan

$\neq$  tidak sama dengan

= sama dengan

# STRING

- Operator '+' disini bukanlah operator penjumlahan seperti pada tipe numerik ( integer atau real ). Operator '+' berarti penyambungan. Bila a dan b adalah peubah bertipe string, maka  $a + b$  sama dengan  $ab$ .
- Contoh :
- 'Teknik'+ 'INFORMATIKA' = 'TeknikINFORMATIKA'
- 'Teknik'+ '\_INFORMATIKA' = 'Teknik\_INFORMATIKA'
- 'xxx' + '\_yyy' + 'zzz' = 'xxx\_yyyzzz'
- '1' + '2' = '12'



# STRING

- Operator relasional, seperti pada karakter, menghasilkan nilai boolean (benar atau salah).
- Contoh :
- 'Efgh' = 'efg' salah
- 'saya' < 'SAYA' salah
- String yang terdiri dari angka dan huruf sering dinamakan dengan alfanumerik.

# TIPE BENTUKAN

- Tipe bentukan adalah tipe yang didefinisikan sendiri oleh pemrogram atau user defined data types
- Pembentukan disusun oleh satu atau lebih tipe dasar ada dua macam tipe bentukan:
  1. tipe dasar yang diberi nama dengan tipe baru dan
  2. tipe terstruktur

# Tipe dasar yang diberi nama dengan tipe baru

- kadang-kadang pemrogram ingin memberi nama Baru terhadap tipe dasar yang sudah dikenal alasan pemberian nama baru mungkin agar nama baru tersebut lebih akrab dan lebih mudah diinterpretasi oleh orang yang membaca teks algoritma kita dapat memberi nama baru untuk tipe dasar tersebut dengan kata kunci `typedef` nilai cara menulis konstanta dan operasi operasi terhadap tipe tersebut tidak berubah tetap sama dengan tipe dasar aslinya

- Deklarasi :

Type `BilanganBulat` : integer

P:`BilanganBulat`

# Tipe Terstruktur

- Disebut juga record/rekaman
- Sebuah record / rekaman disusun oleh satu atau lebih field. Tiap field berisi data dari tipe dasar tertentu atau tipe bentukan lain yang sudah didefinisikan terlebih dahulu.
- Nama record/rekaman ditentukan oleh programmer.
- Sebuah rekaman / record dengan tiga buah field jika digambarkan secara logic adalah sbb :

Field1	Field2	Field3
--------	--------	--------

# Tipe Terstruktur

- Contoh :
- Mendefinisikan tipe bentukan yang menyatakan data mahasiswa. Data mahasiswa terdiri atas NIM, Nama dan Usia.
- Misalkan tipe bentukan ini dinamakan MHS.
- Type MHS : record [ NIM : integer, nama : string, usia : integer ]
- Jika dideklarasikan M adalah variable bertipe MHS, maka cara mengacu tiap field pada record / rekaman M adalah :
  - M.Nim
  - M>Nama
  - M.usia

# Tipe Terstruktur

- Nama : MHS
- Daerah : sesuai dengan daerah masing-masing field
- Misal : [ 743658385, 'Uci Sanusi', 79 ], [ 649460485, 'Unang hermansyah, 34 ]
- Operator : Tidak ada operator untuk MHS tetapi kita dapat melakukan :
  - operasi integer terhadap MHS.NIM
  - operasi string terhadap MHS.nama
  - operasi real terhadap MHS.usia

# Nama

- Di dalam program, sesuatu yang diberi nama dapat berupa :
  - Variable ( Peubah )
  - Constant ( Tetapan )
  - Tipe seperti yang telah dijelaskan
  - Fungsi yang digunakan ( dijelaskan kemudian )
  - Prosedur yang digunakan ( dijelaskan kemudian )
- Dalam penulisan program, ada beberapa aturan penulisan nama yang harus dipenuhi.
  - Harus selalu diawali dengan huruf ( alfabet )
  - Huruf besar atau huruf kecil tidak dibedakan
  - Karakter penyusun nama hanya terdiri dari alfabet, angka, garis bawah( ' \_ ' ).
  - Tidak boleh ada spasi yang memisahkan string yang disusun.
  - Panjang nama tidak dibatasi.

# Nilai

- Nilai adalah nilai besaran dari tipe yang sudah dikenal. Nilai dapat berupa nilai yang dikandung oleh nama variable atau Constant ( tetapan ).
- Pengisian Nilai ke dalam Variable
- Nilai dapat diisikan ke dalam variable dengan cara Penugasan / assignment
- Penugasan (statemen) adalah mengisi sebuah Nilai pada variable secara langsung.
- Proses penugasan biasanya dalam algoritma dapat digambarkan dengan lambang ' $\leftarrow$ ' atau '='



# Nilai

- Nilai yang dimasukan ke dalam variable dapat juga di ambil dari variable yang lain selama tipe data yang dimasukan sama dengan tipe nama yang akan dimasukinya.
- Contoh Notasi Algoritma untuk penugasan :
- Nama  $\leftarrow$  tetapan ( Nilai tetapan diisikan ke dalam nama )
- Nama1  $\leftarrow$  nama2 (Nilai nama2 disalin ke nama1, nama1 sama dengan nama2)
- Nama  $\leftarrow$  ekspresi ( hasil perhitungan diisikan ke dalam nama )
- Atau
- Nama = tetapan ( Nilai tetapan diisikan ke dalam nama )
- Nama1 = nama2 (Nilai nama2 disalin ke nama1, nama 1 sama dengan nama2)
- Nama = ekspresi ( hasil perhitungan diisikan ke dalam nama )

# Nilai

- Input Nilai dari Input device
- Nilai untuk Nama variable dapat diisi dari peralatan input misalnya keyboard.
- Mengisi Nilai dari input device dinamakan operasi pembacaan data. Istilah baca ini timbul karena komputer seakan-akan membaca Nilai yang diberikan oleh pemakai.
- Dalam Algoritma perintah untuk pembacaan data dilakukan dengan perintah input.
- Contoh :
  - Input ( nilai1)

# Nilai

- Syaratnya adalah nilai1 adalah nama variable, tidak boleh nama tetapan / constant.
- Ketika perintah di atas dilaksanakan, komputer menunggu pemakai mengetikan Nilai nila1 dari input device.
- Bila diketikan angka 96 lalu diikuti dengan tombol <enter>, maka tempat di memori yang bernama nilai1 sekarang berisi 96.

# Ekspresi

- Suatu Nilai dipakai untuk proses transformasi menjadi keluaran yang diinginkan.
- Transformasi Nilai menjadi keluaran dilakukan melalui suatu perhitungan (komputasi). Cara perhitungan itu dinyatakan dengan suatu ekspresi.
- Suatu ekspresi terdiri dari operand dan operator. Operand adalah Nilai yang dioperasikan dengan operator tertentu. Operand dapat berupa nama variable atau constant / tetapan.
- Hasil perhitungan adalah Nilai dengan daerah yang sesuai dengan tipe operator yang dipakai.
- Ekspresi dikenal dalam dua macam : Ekspresi Numerik dan Ekspresi Bolean.

# Ekspresi Numerik

- Ekspresi Numerik adalah ekspresi yang baik operand- nya bertipe numerik dan hasilnya juga bertipe numerik.
- Misalkan diinisialisasikan / kamus sbb :

- Kamus

a, b, c : real

d : integer

e, f, g : integer

- Contoh Ekspresi Numerik :

axb

# Ekspresi Numerik

- Hasil perhitungan juga bertipe real. Pada ekspresi ini, operand-nya adalah a dan b sedangkan operatornya adalah “ x ” ( perkalian ). Bila hasil perhitungan disimpan ke dalam nama variable, maka nama variable haruslah bertipe sama dengan tipe hasil. Pengisian hasil ekspresi a x b ke dalam variable c melalui penugasan :
  - $c \leftarrow a \times b$
  - atau
  - $c = a \times b$

# Ekspresi Relasional

- Ekspresi relasional adalah ekspresi dengan operator `<`, `<=`, `>`, `>=`, `==` dan `!=`. Hasil ekspresinya selalu bernilai boolean.
- Inisialisasi / identifikasi kamusnya adalah :

Kamus

Ada, Ketemu, besar : boolean

x, y : integer

- Contoh :

Not ada            hasil false

ada and true    hasil true

x < 5            hasil false

ada or ( x = y ) hasil true

# Output Nilai

- Nilai yang disimpan di dalam memori dapat ditampilkan ke output device. Perintah penulisan Nilai adalah dengan perintah output.
- Contoh :
  - Output (nama1)
  - Output (constant)
  - Output (ekpresi)