

RANGKUMAN DASAR PEMROGRAMAN MATERI DASPRO



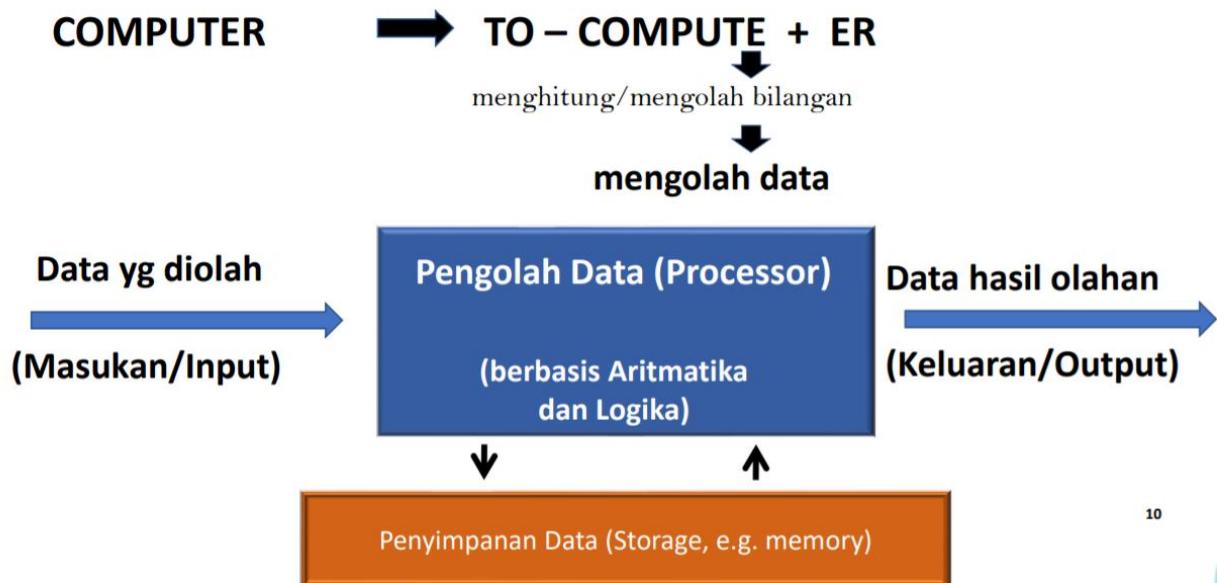
WISANG GERNIES RINEKSA PANGGALIH
2131710079

**PRODI D3 MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI**

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta Nomor 9, Jaimulyo, Kecamatan Lowokwaru, Kota Malang,
Jawa Timur
65141

Introduction - What Is a Computer?



Komputer adalah perangkat elektronik yang memanipulasi informasi atau data. Komputer mampu menyimpan, mengambil, dan mengolah data.

Algoritma : adalah susunan logis dan sistematis yang digunakan untuk memecahkan atau pun menyelesaikan suatu permasalahan tertentu.

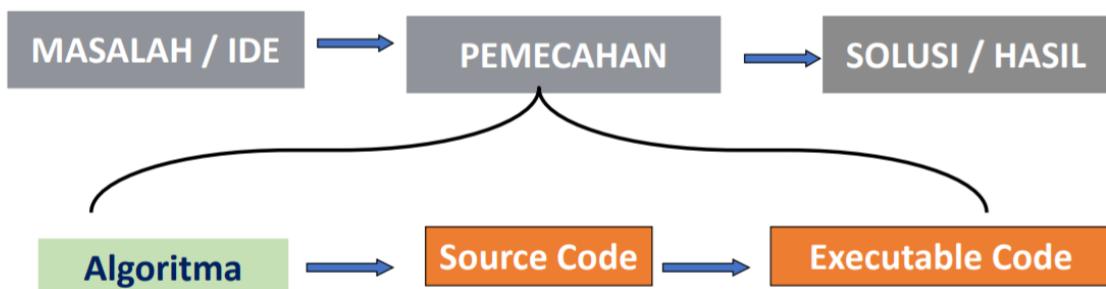
Algoritma...(1)

- Proses pengolahan data membutuhkan algoritma
- Algoritma : urutan langkah-langkah logis penyelesaian masalah yang disusun secara **sistematis** dan **logis**.
- Sistematis : urutan langkah-langkah yang disusun harus jelas, memiliki pola tertentu
- Logis : masuk akal, bisa dibaca dan diterima oleh akal
- Bisa dinilai benar salahnya

Algoritma...(2)

- **Pemrograman komputer**

- Algoritma diimplementasikan dalam **program** komputer
- Satu set instruksi atau langkah-langkah yang dijalankan dengan komputer untuk menyelesaikan suatu masalah.



Tahapan dalam menyelesaikan suatu masalah dalam komputer adalah. Jelas dan benar untuk menyelesaikan suatu permasalahan yang nantinya akan programmer ajarkan ke komputer dengan bahasa pemrograman. Bagan di atas menjelaskan tahap-tahap penyelesaian dalam sebuah sistem computer.

Penilaian Algoritma

- Hasilnya harus benar, sesuai dengan yang diinginkan
- Seberapa baik hasil dari algoritma tersebut
- Efisiensi algoritma → waktu dan memori

Dimungkinkan antara orang satu dengan yang lain, dari **masalah yang sama**, akan mendapatkan **algoritma yang berbeda**. Semuanya dianggap benar, asal hasil yang dikeluarkan sesuai dengan harapan. Namun demikian, diantara perbedaan algoritma tersebut, bisa dipilih mana yang lebih **efisien**

Struktur Dasar Algoritma

1. Struktur runtutan/sekuensial :

Digunakan untuk program yang pernyataannya sequential atau urutan.

2. Struktur pemilihan

Digunakan untuk program yang menggunakan pemilihan atau penyeleksian kondisi.

3. Struktur perulangan

Digunakan untuk program yang pernyataannya akan dieksekusi berulang-ulang.

Kriteria Algoritma

- *Input*
 - Terdapat inputan bernilai nol ataupun lebih banyak sebagai sumber yang berasal dari luar program.
- *Output*
 - Output minimal terdiri dari satu kuantitas hasil.
- *Definiteness*
 - Setiap instruksi yang diberikan harus jelas dan tidak ambigu.
- *Finiteness*
 - Jika kumpulan instruksi algoritma ditelusuri, maka tahapan algoritma akan berakhir setelah sejumlah langkah yang terbatas.
- *Effectiveness*
 - Setiap instruksi harus cukup mendasar sehingga mudah untuk dilakukan sesuai dengan kebutuhan

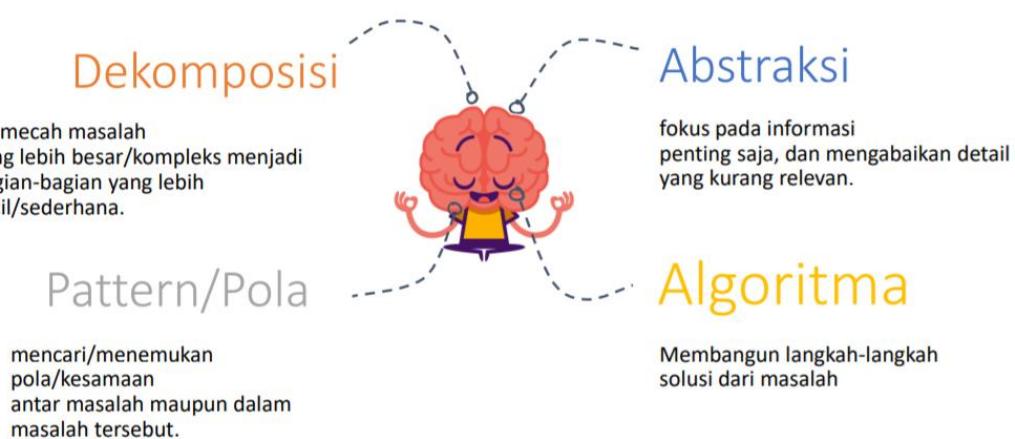
Contoh Penggunaan Algoritma

- **Masalah:** memasak nasi menggunakan rice cooker
- **Proses :**
 1. Menyiapkan beras , air, rice cooker, listrik.
 2. Mencuci **beras**
 3. Masukan **beras** ke **rice cooker** , menambahkan **air**.
 4. Memasang kabel **rice cooker** ke stop kontak **listrik**.
 5. Pilih mode cooking untuk memasak nasi.
 6. **Nasi** masak
- **Struktur : Runtutan**

Coputational Thinking

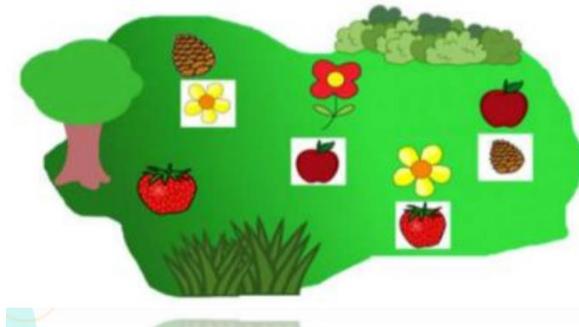
Computational Thinking (CT) adalah proses berpikir untuk memformulasikan persoalan dan solusinya, sehingga solusi tersebut secara efektif dilaksanakan oleh sebuah agen pemroses informasi (“komputer”, robot, atau manusia).

JITEKNIK NEGERI MALANG
Ada empat teknik kunci dalam
Computational Thingking

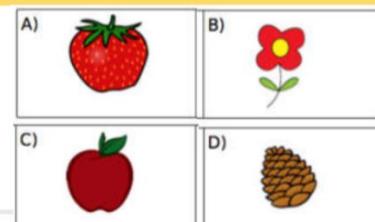


Contoh Permasalahan

- Keluarga Berang-berang sedang mempersiapkan acara Festival Makanan, dan mereka ingin memanggang kue kering; Kati akan membuat kue. Dia sangat memperhatikan urutan membuat kue dengan memasukkan bahan kue dengan urutan yang benar.
- Ketika dia berjalan ke taman, dia melihat ada secarik kertas pada setiap bahan kue yang akan digunakan. **Gambar pada kertas menjelaskan bahan kue yang harus ditambahkan pada urutan berikutnya.** Hanya ada satu bahan kue yang tidak memiliki kertas. Ilustrasi taman disamping seperti di bawah ini:



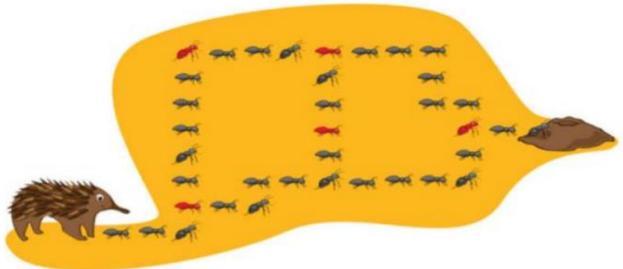
Bahan kue manakah yang harus dimasukkan pertama kali?



Bahan pertama yang perlu di masukkan yaitu, setelah itu masukkan appel, setelah apel di masukan, berikut masukkan buah pinus, selanjutnya masukkan bunga kuning, dan yang terakhir masukkan buah strawberi

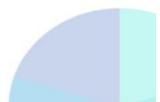
Contoh 2

Landak Etna (di pojok kiri bawah) ingin pergi ke bukit semut. Untuk pergi ke sana, ia harus mengumpulkan semut merah (yang pada gambar lebih besar). Bantulah Etna untuk menentukan jalannya dengan menyebutkan arahnya: Kanan (untuk ke kanan), Kiri (untuk ke kiri), Naik (untuk ke atas), Turun (untuk ke bawah).



Perintah mana yang harus diberikan kepada Bebras tersebut?

- Kanan, Naik, Kanan, Naik, Kanan, Naik, Kanan, Turun, Kanan, Turun, Kanan
- Kanan, Naik, Kanan, Turun, Kanan, Naik, Kanan
- Kanan, Naik, Kanan, Naik, Kiri, Turun, Kanan
- Kanan, Naik, Kanan, Turun, Kanan, Turun, Kanan



Jawaban dari permasalahan di atas yaitu B. Kanan, Naik, Kanan, Turun, Kanan , Naik, Kanan.

Studi Kasus

Contoh Studi Kasus Runtutan

- Berang - barang Bobi telah mengatur meja sarapan seperti yang ditunjukkan pada gambar. Terdapat taplak meja, pisau, piring, serbet, piring kecil-cangkir
- Buatlah algoritma yang digunakan Bobi untuk mengatur meja



Contoh

- Jawab :
- Input : taplak meja, pisau, piring, serbet, piring kecil-cangkir
- Proses :
 - Pasangkan taplak meja
 - Letakkan cangkir dan piring kecil
 - Letakkan serbet
 - Lektkakan piring
 - Diatas piring letakkan pisau
- Output : Meja sarapan yang telah ditata
- Data lain : -



Contoh Studi Kasus Runtutan

- Pak ahmad memiliki lahan berbentuk persegi dengan sisi 100m, di dalam lahan pak ahmad terdapat taman bunga berbentuk lingkaran. Berapakah luas lahan pak ahmad yang tidak ditanami bunga? Buatlah Algoritmanya

- Jawab :

- Input : Lahan dengan sisi = 100m

- Output : luas lahan yang tidak ditanami bunga

- Proses :

- Luas keseluruhan lahan (persegi)

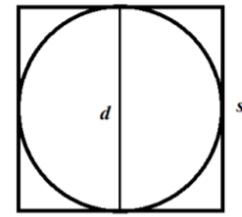
- $\bullet = \text{sisi} \times \text{sisi} = 10.000 \text{ m}^2$

- Luas ditanam bunga (lingkaran)

- $\bullet = \pi * r * r = 3.14 * 50 * 50 = 7850 \text{ m}^2$

- Luas yang tidak ditanami bunga = luas persegi – luas lingkaran

- $\bullet 10.000 - 7850 = 2.150 \text{ m}^2$



Contoh Studi Kasus Pemilihan

- Saat ujian akhir terdapat ketentuan bahwa jika mahasiswa memiliki nilai dibawah 70 maka mahasiswa tersebut mengikuti ujian ulang. Andi ternyata mendapatkan nilai 90. apakah Andi mengikuti ujian ulang?? Buatlah algoritmanya

- Jawab :

- Input : nilai mahasiswa

- Proses :

- Masukkan nilai mahasiswa

- Jika nilai mahasiswa < 70, maka mahasiswa tersebut harus mengikuti ujian ulang

- Jika nilai mahasiswa > 70, maka mahasiswa tersebut tidak perlu ujian ulang

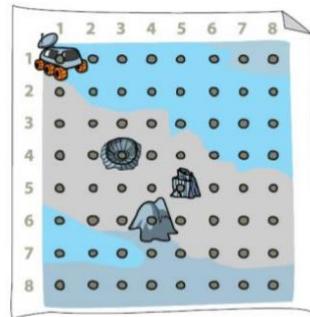
- Output ujian ulang / tidak

- Output : remidi / tidak

- Data Lain : -

Contoh Studi Kasus Perulangan

- Penjelajah bulan yang dikendalikan berang-berang bergerak dari satu arah ke yang lain menggunakan peta di bawah ini. mulai (1,1) sampai (8,8). Angka pertama adalah baris, yang kedua - adalah kolom jumlah. Dimanakah letak objek kawah, batu, dan gunung? Buatlah Algoritmanya :



Contoh Studi Kasus Perulangan

- Input : Peta (baris, kolom), jumlah baris, jumlah kolom
- Output : Posisi Gunung
- Proses :
 - Lakukan pengecekan secara berulang-ulang mulai dari baris pertama untuk kolom ke-1 sampai dengan ke-8
 - Jika menemukan objek simpan informasi posisi baris dan kolom
 - Lakukan pengecekan secara berulang-ulang mulai dari baris kedua untuk kolom ke-1 sampai dengan ke-8
 - Jika menemukan objek simpan informasi posisi baris dan kolom
 - Lakukan pengecekan secara berulang-ulang mulai dari baris ketiga untuk kolom ke-1 sampai dengan ke-8
 - Jika menemukan objek simpan informasi posisi baris dan kolom
 - Lakukan pengecekan secara berulang-ulang mulai dari baris keempat untuk kolom ke-1 sampai dengan ke-8
 - Jika menemukan objek simpan informasi posisi baris dan kolom
 - Lakukan perulangan sampai baris ke 8

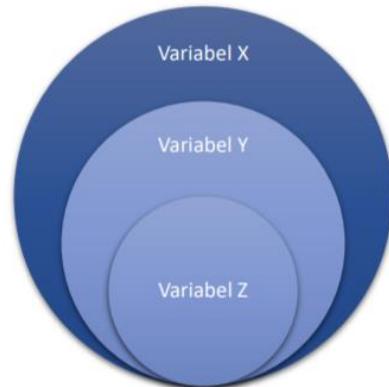
Variable, Tipe Data, Operator, Dan Input-Output

1. Variable

- Variable digunakan dalam bahasa pemrograman untuk menyimpan nilai sementara dimana untuk digunakan kembali nantinya.
- Variabel memiliki tipe data dan nama.
- tipe data mengindikasikan tipe dari nilai pada variabel tersebut.

2. Jenis Variable

- Variable lokal adalah variable yang hanya bisa dikenali pada sub program
- Variabel global adalah variable yang dapat dikenali pada keseluruhan program



3. Aturan Penulisan Variable

- Nama variable tidak boleh menggunakan kata kunci Java
- Nama variable boleh menggunakan huruf, angka(0-9), garis bawah(_), dan symbol dolar(\$), namun sebaiknya penggunaan
- Nama Variable sebaiknya menggunakan diawali huruf kecil
- Apabila nama variable lebih dari satu kata maka kata yang setelahnya diawali huruf besar.

Tipe Data

- Tipe data adalah jenis data yang ingin kita simpan di variabel.
- Tipe data dapat dikategorikan menjadi dua kelompok, yaitu
 1. *tipe data Primitif*
 2. *tipe data Referensi*.

Tipe Data Primitif

Jenis Data	Deskripsi	Ukuran	Minimum	Maksimum
boolean	true / false	1-bit		
char	Karakter Unicode	16-bit		
byte	Bilangan bulat	8-bit	-127	128
short	Bilangan bulat	16-bit	-32768	32767
int	Bilangan bulat	32-bit	-2147483648	2147483647
long	Bilangan bulat	64-bit	-9223372036854775808	9223372036854775807
float	Bilangan riil	32-bit	1.40129846432481707e-45	3.40282346638528860e+38
double	Bilangan riil	64-bit	4.94065645841246544e-324	1.79769313486231570e+308

Tipe Data Primitif adalah tipe data yang paling mendasar di pada Bahasa java, tipe data primitive sudah di definisika oleh java dan di beri nama dengan kata kunci, missal kata kunci int, long, char, dsb.

Casting tipe data

Casting adalah ketika kita ingin memberikan nilai dari tipe data primitive ke tipe data primitive yang lain

- Widening casting (otomatis) – mengubah tipe data dari yang ukurannya lebih kecil ke tipe data yang lebih besar

byte -> short -> char -> int -> long -> float -> double

Casting tipe data(2)

- Narrowing Casting (manual) – mengubah tipe data dari yang ukurannya lebih besar ke tipe data yang lebih kecil

double -> float -> long -> int -> char -> short -> byte

Contoh Casting tipe data

- Widening Casting(otomatis)

```
byte umur = 9;
double myDouble = umur;
System.out.println(umur);           // Outputs 9
System.out.println(myDouble);       // Outputs 9.0
```

- Narrowing Casting(manual)

```
double ipk = 3.78;
int myInt = (int) ipk;
System.out.println(ipk);    // Outputs 3.78
System.out.println(myInt);  // Outputs 3
```

Kegunaan ASCII ??

- ASCII adalah singkatan dari American Standard Code for Information Interchange.
- Sesuai dengan namanya, ASCII digunakan untuk pertukaran informasi dan komunikasi data.
- ASCII merupakan kode angka yang mewakili sebuah karakter.

USASCII code chart

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	0	0	0	1	0	1	0	1	0	0	1	0	1	0	1																																																																																																																														
0	0	0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p	1	!	1	A	Q	a	q	2	STX	DC2	"	2	B	R	b	r	3	ETX	DC3	#	3	C	S	c	s	4	EOT	DC4	\$	4	D	T	d	t	5	ENQ	NAK	%	5	E	U	e	u	6	ACK	SYN	8	6	F	V	f	v	7	BEL	ETB	'	7	G	W	g	w	8	BS	CAN	(8	H	X	h	x	9	HT	EM)	9	I	Y	i	y	10	LF	SUB	*	:	J	Z	j	z	11	VT	ESC	+	:	K	[k	{	12	FF	FS	.	<	L	\	l	l	13	CR	GS	-	=	M]	m	}	14	SO	RS	.	>	N	^	n	~	15	S1	US	/	?	O	-	o	DEL

LOGI INFORMASI
NIK NEGERI MALANG

Tipe data referensi

- Tipe data non-primitive dibuat berdasarkan kebutuhan programmer.
- Nilai bawaan non-primitive adalah null
- Pendeklarasian tipe data ini hampir sama dengan deklarasi pada tipe data primitif.
- Tipe data non-primitive diawali dengan huruf besar



Ciri khas tipe data referensi adalah kemampuannya menampung banyak nilai.

Pada tipe data primitif , nilai yang bisa ditampung Cuma 1 saja. Perhatikan contoh berikut ini:

Tipe Primitif :

int x = 9; (ada 1 nilai saja, yaitu angka 9)

char hurufku = "h"; (ada 1 nilai saja, yaitu huruf h)

Tipe Referensi :

String tulisan = "Aku Belajar Java"; (ada 16 nilai, termasuk spasi)

int[] daftar = { 1, 4, 9, 16, 25, 36, 49 }; (ada 7 nilai bertipe integer)

- Operator merupakan simbol yang biasa digunakan dalam menulis suatu pernyataan (*statement*) dalam bahasa pemrograman apapun. Operator akan melakukan suatu operasi terhadap operand sesuai dengan fungsinya.
- Contoh operasi antara lain penjumlahan, pengurangan, pembagian dan sebagainya.

Jenis Operator

1. Operator Aritmatika
2. Operator Increment dan Decrement
3. Operator Assignment
4. Operator Relasi
5. Operator Logika
6. Operator Bitwise

1. Operator Aritmatika

Arithmatic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika.

Arithmatic Operator	Description
+	plus
-	minus
*	point
/	divide
%	modulus



2. Operator Increment dan Decrement

Operator Increment dan Decrement digunakan untuk menaikkan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan hanya dapat digunakan pada variabel.

Operator	Use	Description
++	a++	Menaikan/menambah 1 nilai setelah operasi dilakukan
	++a	Menaikan/menambah 1 nilai sebelum operasi dilakukan
--	a--	Penurunan/mengurangi 1 nilai setelah operasi dilakukan
	--a	Penurunan/mengurangi 1 nilai sebelum operasi dilakukan



3. Operator Assignment

Operator assignment dalam Java digunakan untuk memberikan sebuah nilai ke sebuah variabel. Operator assignment hanya berupa '=' , shortcut assignment operator yang penting, yang digambarkan dalam tabel berikut :

Operator	Penggunaan	Ekuivalen Dengan
<code>+=</code>	<code>Op1 += Op2</code>	<code>Op1 = Op1 + Op2</code>
<code>-=</code>	<code>Op1 -= Op2</code>	<code>Op1 = Op1 - Op2</code>
<code>*=</code>	<code>Op1 *= Op2</code>	<code>Op1 = Op1 * Op2</code>
<code>/=</code>	<code>Op1 /= Op2</code>	<code>Op1 = Op1 / Op2</code>
<code>%=</code>	<code>Op1 %= Op2</code>	<code>Op1 = Op1 % Op2</code>
<code>&=</code>	<code>Op1 &= Op2</code>	<code>Op1 = Op1 & Op2</code>
<code>!=</code>	<code>Op1 != Op2</code>	<code>Op1 = Op1 ! Op2</code>
<code>^=</code>	<code>Op1 ^= Op2</code>	<code>Op1 = Op1 ^ Op2</code>
<code><<=</code>	<code>Op1 <<= Op2</code>	<code>Op1 = Op1 << Op2</code>
<code>>>=</code>	<code>Op1 >>= Op2</code>	<code>Op1 = Op1 >> Op2</code>
<code>>>>=</code>	<code>Op1 >>>= Op2</code>	<code>Op1 = Op1 >>> Op2</code>

4. Operator Relasi

Operator relasi dalam Java digunakan untuk menghasilkan nilai boolean yang sering digunakan untuk mengatur alur jalannya sebuah program.

Operator	Penggunaan	Deskripsi
<code>></code>	<code>Op1 > Op2</code>	Menghasilkan true jika Op1 lebih besar dari Op2
<code><</code>	<code>Op1 < Op2</code>	Menghasilkan true jika Op1 lebih kecil dari Op2
<code>>=</code>	<code>Op1 >= Op2</code>	Menghasilkan true jika Op1 lebih besar atau sama dengan Op2
<code><=</code>	<code>Op1 <= Op2</code>	Menghasilkan true jika Op1 lebih kecil atau sama dengan Op2
<code>==</code>	<code>Op1 == Op2</code>	Menghasilkan true jika Op1 sama dengan Op2
<code>!=</code>	<code>Op1 != Op2</code>	Menghasilkan true jika Op1 tidak sama dengan Op2



5. Operator Logika

Operator ini digunakan untuk ekspresi logik yang menghasilkan nilai boolean. Operator-operator yang digunakan adalah AND (&&), OR (||) dan NOT (!).

Operator	Deskripsi	Contoh
&&	and	x=6 y=3 (x < 10 && y > 1) hasil true
	or	x=6 y=3 (x==5 y==5) hasil false
!	not	x=6 y=3 !(x==y) hasil true

6. Operator Bitwise(2)

• Bitwise AND(&)

Hasil bit bernilai 1 ketika semua bit-bit bernilai 1, selain itu bernilai 0.

Contoh:

```
int a = 5;//0101
int b = 7;//0111
System.out.println(a&b); //output 5
//0101
//0111
//_____
//0101 -> 5
```

6. Operator Bitwise

Operator ini digunakan untuk melakukan manipulasi bit dari sebuah bilangan

• Bitwise OR(|)

Hasil bit bernilai 1 ketika salah satu bit-bit bernilai 1, selain itu bernilai 0.

Contoh:

```
int a = 5;//0101
int b = 7;//0111
System.out.println(a|b); //output 7
//0101
//0111
//_____
//0111 -> 7
```

6. Operator Bitwise(3)

- Bitwise XOR (^)

Nilai bit bernilai 1 ketika ada bit bernilai 1 dan 0, selain itu bernilai 0.

Contoh:

```
int a = 5; //0101
int b = 7; //0111
System.out.println(a^b); //output 2
//0101
//0111
//_____
//0010 -> 2
```

6. Operator Bitwise(4)

- Bitwise Complement (~)

Nilai bit yang berkebalikan, ketika nilai bit bernilai 1 maka menghasilkan 0 sedangkan bernilai 0 menghasilkan 1.

Contoh:

```
int a = 5; //0101
System.out.println(~a); //output -6
//0101
//_____
//1010 -> 10
~n=-(n+1)
~(-n)=(n-1)
```

Flowchart

Flowchart merupakan sebuah bagan dengan simbol-simbol tertentu yang digunakan untuk menjelaskan urutan proses dan hubungan antar proses lainnya pada sebuah program.

Simbol-simbol FlowChart

Symbol	Nama	Deskripsi
	Terminator	Simbol untuk permulaan (start) dan akhir (end) dalam sebuah proses
	Preparation	Simbol untuk mempersiapkan penyimpanan yang akan digunakan
	Input-output	Simbol yang menyatakan proses input dan output
	Flow Line (Garis Alir)	Arah aliran program

Simbol-simbol Flowchart

Symbol	Nama	Deskripsi
	Processing simbol	Simbol yang menunjukkan pemrosesan oleh komputer
	Percabangan	Simbol yang menunjukkan terdapat dua pilihan output
	On Page Connector	Penghubung bagian-bagian flowchart yang ada pada satu halaman
	Off Page Connector	Penghubung bagian-bagian flowchart yang berada pada halaman yang berbeda
	Predefined Process	Simbol untuk pelaksanaan suatu bagian (sub-program)/fungsi/prosedur

Penggunaan Inputan Pada Java

- Untuk membaca input dari keyboard yaitu dengan menggunakan library Scanner yang di import kedalam program java.
- Caranya adalah dengan menuliskan perintah **import java.util.Scanner** di baris paling atas dari kode program yang akan dibuat.
- Selanjutnya tuliskan perintah deklarasi scanner berikut ini didalam fungsi main() :

```
Scanner sc = new Scanner(System.in);
```

- Selanjutnya, tergantung dari jenis input yang akan dimasukkan, berupa bilangan bulat (int), bilangan koma (float/double), atau karakter (String).
 - Jika input berupa bilangan bulat, maka perintahnya adalah: **nextInt()**;
 - Jika input berupa bilangan koma, maka perintahnya adalah: **nextFloat()**;
 - Jika input berupa teks, maka perintahnya adalah: **nextLine()**;

Menampilkan Output Pada Java

Untuk menampilkan output ke layar, ada beberapa cara:

- System.out.print("Hello world");** Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik.
- System.out.println("Hello world");** Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik, sekaligus memberi perintah ganti baris di akhir kata/kalimat.
- System.out.println(panjang);** Perintah ini akan menampilkan isi variabel panjang ke layar. Perhatikan bahwa untuk menampilkan isi variabel, tidak perlu menggunakan tanda petik (").
- System.out.println("Panjang segi empat: " + panjang);** Perintah ini akan menampilkan kalimat "Panjang segi empat: " kemudian disambung dengan isi variabel panjang ke layar. Perhatikan untuk menyambung kalimat dengan isi variabel, digunakan tanda plus (+).