

MODUL

ALGORITMA DAN PEMROGRAMAN



Disusun oleh :

Femi Dwi Astuti, S.Kom, M.Cs.

Indra Yatini Buryadi, S.Kom., M.Kom.

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

AKAKOM

YOGYAKARTA

2019

KATA PENGANTAR

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
MODUL 1 PENGANTAR ALGORITMA.....	1
MODUL 2 PSEUDOCODE DAN FLOWCHART	4
MODUL 3 TIPE DATA, VARIABEL, OPERATOR	12
MODUL 4 PENGENALAN JAVA DAN SEKUENSIAL	17
MODUL 5 SELEKSI IF	22
MODUL 6 SELEKSI SWITCH-CASE	28
MODUL 7 SELEKSI BERTINGKAT	33
MODUL 8 PERULANGAN WHILE.....	39
MODUL 9 PERULANGAN DO-WHILE.....	44
MODUL 10 PERULANGAN FOR	48
MODUL 11 PERULANGAN BERTINGKAT	52
MODUL 12 SELEKSI DALAM PERULANGAN	56
MODUL 13 PERULANGAN DALAM SELEKSI	60
MODUL 14 STUDI KASUS SEKUENSI, SELEKSI DAN ITERASI.....	64

MODUL 1

PENGANTAR ALGORITMA



CAPAIAN PEMBELAJARAN

Mahasiswa dapat membuat algoritma sederhana



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Text Editor (Microsoft Word, Notepad, dll)



DASAR TEORI

1. Pengertian Algoritma

'Algoritma' diambil dari nama belakang seorang tokoh matematikawan Persia bernama Muhammad ibn Musa al-Khuwarizmi (lahir tahun 730an, meninggal antara tahun 835 dan 850). Algoritma merupakan metode umum yang digunakan untuk menyelesaikan kasus-kasus tertentu. Algoritma adalah langkah – langkah logis tertentu untuk menyelesaikan suatu masalah. Guna algoritma adalah untuk membantu seseorang dalam menyelesaikan suatu masalah berdasarkan pada pola pikirnya masing-masing. Dalam menuliskan algoritma, dapat digunakan bahasa natural atau menggunakan notasi matematika, sehingga masih belum dapat dijalankan pada komputer.

Dalam kehidupan sehari-hari, kita sudah melakukan penyusunan algoritma untuk menyelesaikan permasalahan atau tantangan yang dihadapi. Misalnya algoritma memasak nasi, algoritma berangkat kuliah, algoritma masuk ke dalam suatu ruangan, algoritma menggunakan komputer, dll. Algoritma dapat digunakan untuk merepresentasikan suatu urutan kejadian secara logis dan dapat diterapkan di semua kejadian sehari-hari. Jadi algoritma adalah jembatan untuk mempermudah pemahaman alur kerja suatu proses.

2. Ciri – ciri algoritma:

1. Ada masukan (*input*).
2. Ada proses.

3. Ada keluaran (*output*).
4. Memiliki perintah yang jelas dan tidak membingungkan (*ambigu*).
5. Harus mempunyai tanda akhir (*stopping role*)

3. Sifat algoritma :

1. Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
2. Tidak tergantung pada suatu bahasa pemrograman.
3. Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.



PRAKTIK

1. Buatlah algoritma untuk memasak telur dadar

1. Siapkan mangkok untuk mengocok telur
2. Pecah dan masukkan telur ke dalam mangkok
3. Masukkan garam ke dalam mangkok
4. Kocok telur dalam mangkok
5. Nyalakan kompor
6. Siapkan wajan dan panaskan dikompor
7. Tuangkan minyak ke dalam wajan
8. Tuangkan telur yang sudah dikocok kedalam wajan
9. Masak telur hingga matang
10. Angkat telur
11. Matikan kompor

2. Buatlah algoritma menentukan apakah suatu bilangan termasuk genap atau ganjil

1. Masukkan sebuah bilangan bulat
2. Bagi bilangan yang sudah dimasukkan dengan angka 2
3. Jika menghasilkan sisa pembagian (0) maka bilangan tersebut genap
4. Jika menghasilkan sisa pembagian bukan (0) maka bilangan tersebut ganjil

3. Buatlah algoritma untuk menghitung luas segitiga

1. Masukkan alas
2. Masukkan tinggi
3. Hitung LuasSegitiga = $0,5 * \text{alas} * \text{tinggi}$
4. Tampilkan LuasSegitiga



LATIHAN

1. Buat algoritma dalam kalimat deklaratif untuk :

- a. Memasak mie instan
- b. Menghitung luas lingkaran
- c. Mendaftar mahasiswa baru di STMIK AKAKOM



TUGAS

1. Buatlah algoritma dalam kalimat deklaratif untuk melakukan daftar ulang di STMIK AKAKOM



REFERENSI

1. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, hal (3-5)
2. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, hal (21-37, 42-43)

MODUL 2

PSEUDOCODE DAN FLOWCHART



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat membuat algoritma dalam berbagai bentuk (pseudocode, flowchart)
2. Mahasiswa dapat menggunakan alat bantu visual seperti flowgorithm



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Text Editor (Microsoft Word, Notepad, dll)
2. Flowgorithm



DASAR TEORI

1. Penyajian Algoritma

Algoritma biasa disajikan dalam tiga bentuk yaitu :

- a. Kalimat deklaratif
- b. Pseudocode
- c. Flowchart

2. Pseudocode

Pseudocode adalah bentuk informal untuk mendeskripsikan algoritma yang mendekati struktur bahasa pemrograman tertentu. Tidak ada suatu standar untuk menyusun algoritma menggunakan pseudocode. Oleh karena *pseudocode* lebih cocok digunakan untuk menyusun algoritma dengan kasus yang besar dan kompleks, maka sangat dianjurkan kepada programmer pemula untuk mulai menggunakan pseudocode dalam menyelesaikan masalah. Tujuan dari penggunaan *pseudocode* adalah supaya :

1. lebih mudah dibaca oleh manusia
2. lebih mudah untuk dipahami

3. lebih mudah dalam menuangkan ide/hasil pemikiran

Seorang programmer yang ingin menerapkan algoritma tertentu, terutama yang kompleks atau algoritma baru, biasanya akan memulainya dengan membuat deskripsi dalam bentuk *pseudocode*. Setelah *pseudocode* tersebut jadi, maka langkah selanjutnya hanya tinggal menterjemahkannya ke bahasa pemrograman tertentu. *Pseudocode* ini biasanya disusun dalam bentuk yang terstruktur dengan pendekatan sekuensial (berurutan) dari atas ke bawah.

Contoh Pseudocode :

Pseudo code untuk menghitung jumlah dua buah bilangan :

Deklarasi

A, B : integer {input}

Jumlah : integer {output}

Deskripsi

Masukkan (A, B)

Jumlah \leftarrow A+B




Tampilkan (Jumlah)







3. Flowchart

Dalam membuat algoritma, diperlukan suatu mekanisme atau alat bantu untuk menuangkan hasil pemikiran mengenai langkah-langkah penyelesaian masalah yang sistematis dan terurut. Pada dasarnya untuk bisa menyusun solusi diperlukan kemampuan *problem-solving* yang baik. Oleh karena itu, sebagai sarana untuk melatih kemampuan tersebut terdapat sebuah *tool* (alat) yang dapat digunakan, yakni *flowchart*.

Secara formal, *flowchart* didefinisikan sebagai skema penggambaran dari algoritma. Tabel berikut menampilkan simbol-simbol yang digunakan dalam menyusun *flowchart*.

Tabel 1. Simbol-simbol dalam *flowchart*

	Terminator Sebagai simbol 'START' atau 'END' untuk memulai atau mengakhiri flowchart.
	Input/Output Digunakan untuk menuliskan proses menerima data atau mengeluarkan data
	Proses Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi aritmatika

	Conditional / Decision Digunakan untuk menyatakan proses yang membutuhkan keputusan
	Preparation Digunakan untuk memberikan nilai awal
	Arrow Sebagai penunjuk arah dan alur proses
	Connector (On-page) Digunakan untuk menyatukan beberapa arrow
	Connector (Off-page) Digunakan untuk menghubungkan flowchart yang harus digambarkan pada halaman yang berbeda. Biasanya pada simbol ini diberi label sebagai penanda, misalnya angka 1 atau A.
	Display Digunakan untuk menampilkan data ke monitor

Keuntungan menggunakan *flowchart* adalah penggunaan diagram untuk menggambarkan tahapan proses, sehingga lebih mudah dilihat dan dipahami. Namun demikian, flowchart juga memiliki kelemahan, yakni jika digunakan untuk menggambarkan proses atau algoritma untuk skala kasus yang besar, maka akan dibutuhkan banyak kertas

4. Flowgorithm

Flowgorithm merupakan aplikasi gratis yang dapat digunakan untuk mempelajari algoritma pemrograman komputer melalui diagram alir (flowchart). Aplikasi ini dikembangkan oleh **Devin Cook** di **Universitas Negeri Sacramento** pada Tahun 2017 lalu. Flowgorithm dapat diunduh di <http://www.flowgorithm.org/>

Contoh penyajian algoritma dalam kalimat deklaratif :

1. Masukkan nilai panjang dan simpan dalam variabel **Panjang**
2. Masukkan nilai lebar dan simpan dalam variabel **Lebar**
3. Hitung Luas dengan rumus: **Luas = Panjang * Lebar**
4. Tampilkan **Luas** Persegi Panjang

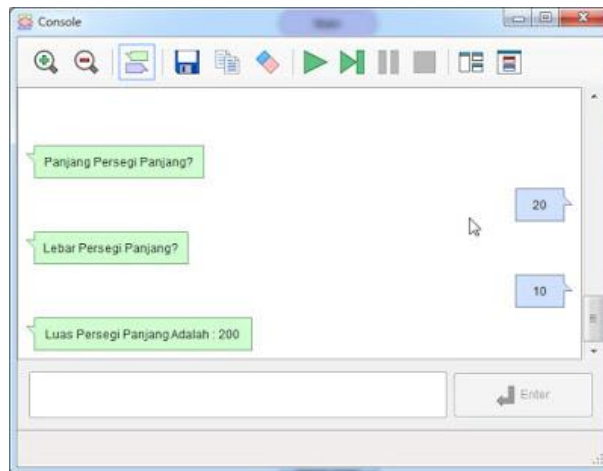
Contoh penyajian algoritma dalam flowchart dengan flowgorithm :

Jika kita menggambar flowchart di *Flowgorithm* kita bisa membuatnya menjadi seperti ini:



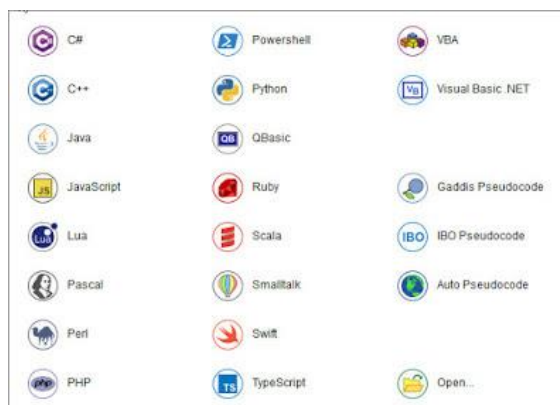
Gambar 1. Flowchart di Flowgorithm

Flowchart ini jika dijalankan maka akan ada beberapa pertanyaan tentang panjang dan lebar persegi panjang dan kita dapat menginput nilainya. Hasil lengkapnya terlihat pada gambar di bawah ini.



Gambar 2. Tampilan ketika Flowchart dijalankan

Flowgorithm juga dapat mengenerate flowchart menjadi source code program dan mendukung banyak bahasa pemrograman. Untuk bahasa pemrograman yang didukung diantaranya terlihat pada gambar berikut ini:



Gambar 3. Bahasa Pemrograman yang di dukung Flowgorithm

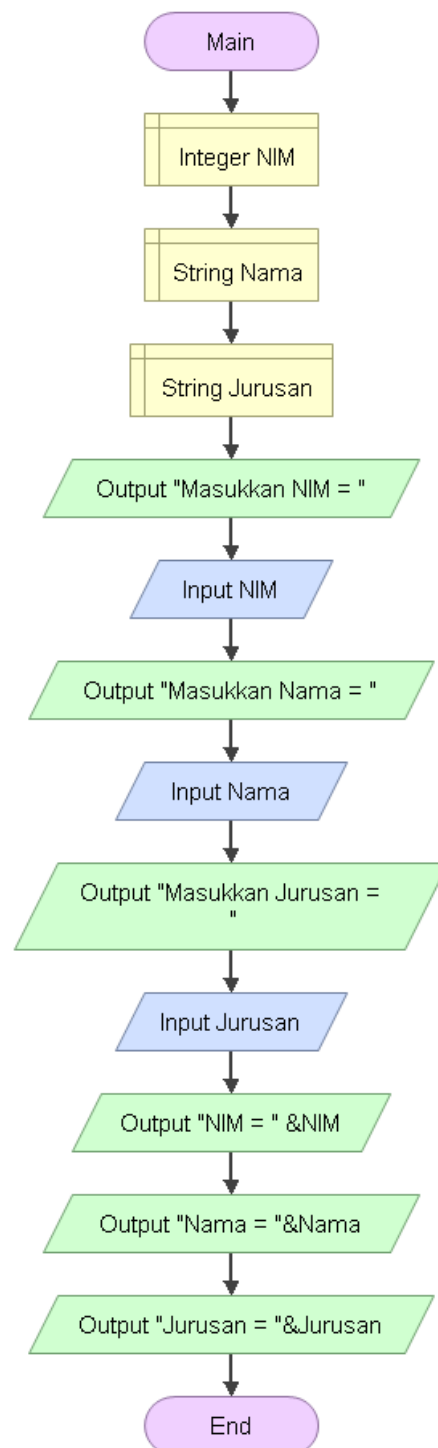


PRAKTIK

1. Buat pseudocode dan flowchart untuk menampilkan data pribadi

Pseudocode
Deklarasi NIM, Nama, Jurusan : String {input} NIM, Nama, Jurusan : String {output}
Deskripsi Masukkan (NIM, Nama, Jurusan) Tampilkan (NIM, Nama, Jurusan)

Flowchart



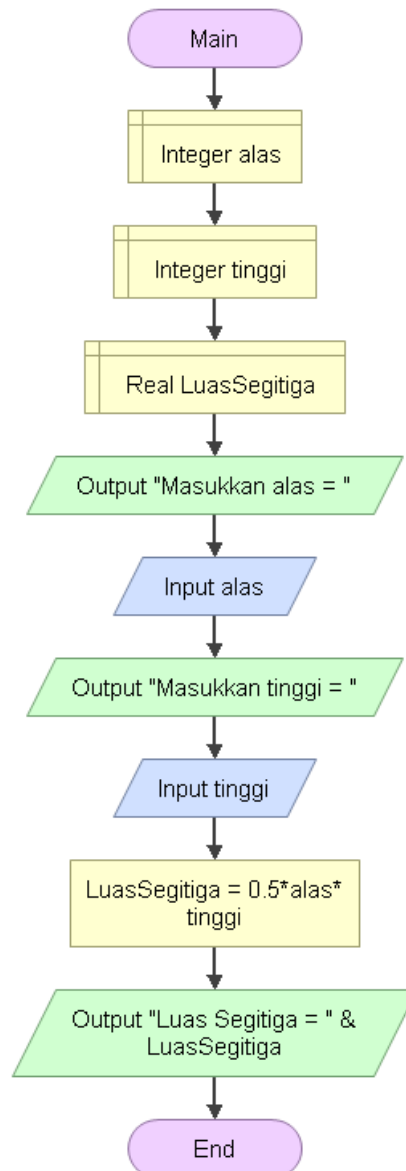
2. Sajikan praktik modul 1 nomor 3 (menghitung luas segitiga) dalam bentuk pseudocode dan flowchart

Pseudocode

Deklarasi alas, tinggi : integer {input}

LuasSegitiga : float {output}
Deskripsi
Masukkan (alas, tinggi)
 $\text{LuasSegitiga} \leftarrow 0,5 * \text{alas} * \text{tinggi}$
Tampilkan (LuasSegitiga)

Flowchart



LATIHAN

1. Modifikasi praktik 1 dengan menambahkan data alamat dan nomor telepon

2. Modifikasi praktik 2 untuk menghitung luas persegi panjang



TUGAS

1. Buat algoritma dan sajikan dalam bentuk pseudocode dan flowchart untuk menghitung angsuran pembayaran SPA dengan ketentuan :
Angsuran 1 : 40% dari SPA
Angsuran 2 : 30% dari SPA
Angsuran 3 : 20% dari SPA
Angsuran 4 : 10% dari SPA



REFERENSI

1. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, hal (843,847)
2. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, hal (45-53)

MODUL 3

TIPE DATA, VARIABEL, OPERATOR



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan berbagai macam tipe data
2. Mahasiswa dapat menggunakan variabel dan konstanta
3. Mahasiswa mampu menggunakan operator unary dan binary



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Text Editor (Microsoft Word, Notepad, dll)
2. Flowgorithm



DASAR TEORI

1. Tipe Data

Secara umum, tipe data terdiri dari :

1. Karakter

Himpunan karakter terdiri dari huruf, digit maupun simbol-simbol lainnya (termasuk spasi, karakter kontrol).

Contoh :

Huruf : A, a, B, b, C, c

Digit : 0, 1, 2, 3, 4, 5

Simbol dan lainnya : _ - + * dan sebagainya

2. Numeric

- a. **byte**: Tipe data byte adalah 8-bit integer bertanda *two's complement*. Tipe ini mempunyai nilai minimum -128 dan nilai maksimumnya adalah 127.
- b. **short**: Tipe data short merupakan integer 16 bit *two's complement* yang mempunyai nilai minimum -32,768 dan nilai maksimum 32,767.
- c. **int**: Tipe data int adalah integer 32 bit *two's complement*. Dia mempunyai nilai minimum -2,147,483,648 sedangkan nilai maksimumnya adalah 2,147,483,647 (inclusive).
- d. **long**: Tipe data long adalah integer 64 bit *two's complement*. Nilai minimumnya adalah -9,223,372,036,854,775,808 sedangkan nilai maksimumnya adalah 9,223,372,036,854,775,807 (inclusive). Gunakan tipe

data ini pada saat anda memerlukan jangkauan nilai yang lebih besar daripada yang bisa disajikan oleh int.

- **float**: Tipe data float merupakan *single-precision 32-bit IEEE 754 floating point*.
- **double**: Tipe data double adalah *double-precision 64-bit IEEE 754 floating point*.

Tabel 2.1. Batas nilai floating point

Parameter	float	double
N	24	53
K	8	11
E_{max}	+127	+1023
E_{min}	-126	-1022

3. **boolean**: Tipe data boolean hanya mempunyai dua nilai yang mungkin, yaitu true dan false. Gunakan tipe data ini untuk flag-flag sederhana untuk menjejak/menelusuri kondisi true atau false. Tipe data ini merepresentasikan satu bit informasi, tetapi ukurannya tidak didefinisikan dengan tepat.
4. **char**: Tipe data char adalah karakter Unicode 16 bit. Tipe data ini mempunyai nilai minimum '\u0000' (atau 0) dan nilai maksimum '\uffff' (atau 65,535).

2. Variabel dan konstanta

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah selama eksekusi berlangsung.

Konstanta adalah nilai yang tetap.

Contoh :

```
float jumlah;
jumlah = 10;
```

Konstanta bernama

Hal ini dapat dilakukan dengan menggunakan kata kunci const.

Contoh :

```
const float PHI = 3.14;
```

3. Operator

Operator adalah simbol khusus yang menyajikan operasi khusus pada satu, dua, atau tiga operand dan kemudian mengembalikan hasilnya. Operator-operator tersebut digunakan untuk membentuk ekspresi. Sebuah konstanta atau entitas dengan sendirinya merupakan ekspresi, demikian juga kombinasi dari konstanta dan variabel dengan operator. Suatu ekspresi yang diikuti dengan sebuah titik koma adalah sebuah pernyataan. Operator mempunyai sifat- sifat

Tabel 2.2 Sifat Operator

Sifat	Keterangan	Contoh
Unary	Operator yang hanya melibatkan 1 operand	-1
Binary	Operator yang hanya melibatkan 2 operand	1 + 2
Tenary	Operator yang hanya melibatkan 3 operand	1 + 2 * 2

Pada saat kita akan menggunakan operator harus mengetahui terlebih dahulu operator yang mana yang mempunyai presedence yang lebih tinggi. Operator di dalam tabel 2.1. dituliskan sesuai dengan presedence ordernya. Semakin ke bawah,

maka presedence-nya lebih rendah. Operator dengan presedence yang lebih tinggi dikerjakan lebih dulu dari pada operator dengan presedence yang lebih rendah. Operator yang ada di dalam baris yang sama mempunyai presedence yang sama. Pada saat operator dari presedence yang sama muncul di dalam ekspresi yang sama, harus diatur yang mana yang harus dikerjakan lebih dulu. Semua operator biner kecuali untuk operator pemberian dikerjakan dari kiri ke kanan. Operator pemberian dikerjakan dari kanan ke kiri.

Tabel 2.3 Precedence Operator

Operator	Precedence
<i>Postfix</i>	<i>expr++ expr--</i>
<i>Unary</i>	<i>++expr --expr +expr -expr ~ !</i>
Multiplikasi	* / %
Aditif	+ -
Pergeseran	<< >> >>>
Relasional	< > <= >= instanceof
Persamaan	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
Logika AND	&&
Logika OR	
<i>Ternary</i>	? :
Pemberian	= += -= *= /= %= &= ^= = <<= >>= >>>=

Operator Aritmatika

Tabel 2.4 Operator Aritmatika

Operator	Keterangan
+	Penjumlahan (tanda plus)
-	Pengurangan (tanda minus)
*	Perkalian
/	Pembagian
%	Sisa Pembagian

Operator *, / dan % mempunyai prioritas yang sama, tetapi lebih tinggi daripada + atau -



PRAKTIK

1. Buat algoritma sajian dalam bentuk pseudocode untuk menghitung nilai A yang diperoleh dari $A=B+C*D$

Deklarasi

B, C, D : integer {input}

A : integer {output}

Deskripsi
Deskripsi
Masukkan (B,C,D)
 $A \leftarrow B+C*D$
Tampilkan (A)

2. Modifikasi praktik 1 dimana C diperoleh dari $B*2$, Buat pseudocodenya

Deklarasi
B, D : integer {input}
A, C : integer {output}
Deskripsi
Masukkan (B, D)
 $C \leftarrow B*2$
 $A \leftarrow B+C*D$
Tampilkan (A)

3. Buat pseudocode dan flowchart untuk menghitung hasil modulo (sisa hasil bagi) dari dua buah bilangan

Deklarasi
Bil1, Bil2 : integer {input}
Hasil : integer {output}
Deskripsi
Masukkan (Bil1, Bil2)
 $Hasil \leftarrow Bil1 \% Bil2$
Tampilkan (Hasil)

4. Buat pseudocode dan flowchart untuk menghitung hasil konversi dari meter ke centimeter dan dari meter ke inci.

Deklarasi
m : float {input}
cm, inci : float {output}
Deskripsi
Masukkan (m)
 $cm \leftarrow m*100$
 $inci \leftarrow m*100/2,54$
Tampilkan (cm)
Tampilkan (inci)



LATIHAN

1. Modifikasi praktik 1 dengan membuat flowchartnya
2. Buat pseudocode dan flowchart untuk menjumlahkan, mengalikan, mengurangi dan membagi dua buah bilangan



TUGAS

1. Disampaikan oleh dosen pengampu dikelas



REFERENSI

1. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, hal ((41-70,93-99))

MODUL 4

PENGENALAN JAVA DAN SEKUENSIAL



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat menggunakan bahasa pemrograman JAVA
2. Mahasiswa dapat mengimplementasikan bahasa pemrograman untuk menyelesaikan masalah sederhana secara sekuensial



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

1. Pengenalan JAVA

Java adalah bahasa pemrograman multi platform. Java tidak menyediakan IDE khusus seperti halnya bahasa pemrograman yang lain. Pemrogram bisa menggunakan IDE yang *support* ke Java, misalnya Netbeans, Eclipse, TexPad, dan lain-lain. Editor teks bisa digunakan semisal Notepad. Jika editor yang digunakan tidak support Java, kita tinggal menyimpannya dalam ekstensi .java kemudian kompilasi dan menjalankannya menggunakan command prompt.

e. Pengenal (identifier)

Pengenal atau identifier adalah suatu nama yang bisa dipakai dalam pemrograman untuk menyatakan :

- variabel
- konstanta bernama
- tipe data
- fungsi
- label
- obyek

Contoh : noMhs; no_Mhs;

f. Kata Kunci

Pengenal sistem yang mempunyai makna khusus bagi kompilasi. Kegunaan dari golongan ini tidak dapat diubah.

Contoh :

case, char, const, do, else, for, return, void, while, dan lain-lain

g. Pernyataan Masukan Dan Keluaran Dalam Java

Dalam bahasa pemrograman Java, kita akan menggunakan pernyataan masukan dan keluaran. Pernyataan masukan adalah pernyataan untuk mendapatkan masukan dari keyboard. Sedangkan pernyataan keluaran adalah pernyataan untuk menampilkan sesuatu nilai ke layar.

Komponen keluaran

Untuk menampilkan ke layar secara tekstual, java mempunyai fasilitas output, yaitu dengan perintah

`System.out.print(hal yang akan ditampilkan, bisa lebih dari satu);`

untuk menampilkan ke layar dan setelah selesai tidak berpindah baris. Akibatnya perintah keluaran berikutnya akan ditampilkan pada baris yang sama. Dan perintah

`System.out.println(hal yang akan ditampilkan, bisa lebih dari satu);`

untuk menampilkan ke layar dan setelah selesai berpindah baris. Akibatnya perintah keluaran setelah itu akan ditampilkan pada baris berikutnya.

`System.out.println("Hasil penjumlahan "+hasil);`

`//menampilkan isi variabel hasil`

Komponen masukan

Untuk memasukkan sebuah nilai ke variabel yang sudah didefinisikan digunakan kelas Scanner (kelas ini disediakan mulai Java versi 1.5).

a. Input data bertipe Integer

Untuk menginputkan data dengan tipe integer digunakan method `nextInt()` di dalam kelas Scanner.

b. Input data bertipe String

Untuk menginputkan data dengan tipe integer digunakan method `next()` atau `nextLine()` di dalam kelas Scanner.

c. Input tipe yang lain

Untuk input data dari keyboard dengan tipe yang lain dan disediakan oleh kelas Scanner adalah sebagai berikut

- `nextBoolean` : input tipe boolean (true atau false saja)
- `nextShort` : input tipe short integer
- `nextLong` : input tipe long integer
- `nextFloat` : input tipe float
- `nextDouble` : input tipe double

2. Sekuensial

Sekuensi adalah pemrograman sederhana yang hanya dapat memecahkan masalah-masalah yang sederhana. Masalah yang dapat diselesaikan dengan sekuensi saja biasanya hanya satu masalah kecil yang berdiri sendiri. Pembahasan sebelumnya, sejauh ini menggunakan prinsip sekuensi. Disamping itu, pada bagian ini kita juga akan mempelajari bagaimana membuat pseudocode sebelum implementasi ke bahasa pemrograman.



PRAKTIK

1. Buka textpad, Ketikkan program untuk praktik 1, 2, 3, 4 yang ada di pertemuan 3

Praktik 1

```
1. public class Mod4_1
2. {
3.     public static void main(String args[])
4.     {
5.         int B,C,D,A;
6.         B=5;
7.         C=3;
8.         D=6;
9.         A = B+C*D;
10.    System.out.println("Hasil A = "+A);
11.    }
12. }
```

- a. Simpan dengan nama Mod4_1.java
- b. Tekan menu tools di text pad, kemudian pilih external tools, pilih compile java
- c. Tekan menu tools di text pad, kemudian pilih external tools, pilih run java application
- d. Amati outputnya, mengapa bisa demikian, jelaskan di laporan

Praktik 2

```
import java.util.Scanner;
public class Mod4_3
{
    public static void main(String args[])
    {
        Scanner masuk=new Scanner(System.in);
        int Bil1,Bil2,Hasil;
        System.out.print("Masukkan Bil 1 = ");
        Bil1=masuk.nextInt();
        System.out.print("Masukkan Bil 2 = ");
        Bil2=masuk.nextInt();
        Hasil=Bil1%Bil2;
        System.out.println("Hasil = "+Hasil);
    }
}
```

Amati outputnya, mengapa bisa demikian, jelaskan di laporan

Praktik 3

```
import java.util.Scanner;
public class Mod4_4
{
    public static void main(String args[])
    {
```

```

Scanner masuk=new Scanner(System.in);
int m;
float cm,inci;
System.out.print("Masukkan meter = ");
m=masuk.nextInt();
cm=m*100;
inci=m*100/2.54f;
System.out.println("Hasil konversi cm = "+cm);
System.out.println("Hasil konversi inci = "+inci);
}
}

```

Amati outputnya, mengapa bisa demikian, jelaskan di laporan

Praktik 4

Modifikasi praktik 1, dimana nilai B, C dan D dapat dimasukkan oleh user, jalankan dan amati outputnya, jelaskan di laporan



LATIHAN

1. Modifikasi praktik 1 dimana C diperoleh dari $B \times 2$
2. Buat program untuk menampilkan biodata dimana data dimasukkan oleh user. Data yang perlu disimpan dan ditampilkan adalah NIM, nama, tempat lahir, tanggal lahir, usia, no hp, e-mail, jurusan, alamat
3. Buat pseudocode dan program untuk menghitung luas lingkaran



TUGAS

1. Buat flowchart dan program untuk menghitung gaji dimana total gaji terdiri dari gaji pokok + tunjangan kinerja yang diperoleh dari 20% gaji pokok + tunjangan lain diperoleh dari 10% tunjangan kinerja
2. Buat pseudocode dan program untuk menghitung konversi dari celcius ke reamur dan celcius ke fahrenheit



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, hal (9-10,13-15,32-42)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, hal (15-18,33-36,71-76)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, hal (55,108)

MODUL 5

SELEKSI IF



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep seleksi if untuk menyelesaikan kasus yang sederhana



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Seleksi

Seleksi digunakan untuk mengarahkan suatu proses itu berjalan. Seleksi adalah Suatu program untuk mengambil keputusan berdasarkan suatu kondisi. Seleksi di Java memiliki dua macam bentuk yaitu : Seleksi dengan **if...else** dan **switch...case**. Suatu program komputer yang dibuat menggunakan algoritma tertentu pada umumnya memiliki perintah-perintah yang hanya akan dikerjakan pada situasi dan kondisi tertentu. Ini sangat mirip dengan kondisi yang ada di dunia nyata. Di dunia nyata biasanya kita juga memiliki pilihan-pilihan tertentu untuk melakukan sesuatu (apa yang harus dilakukan/akan kita kerjakan). Sebagai contoh, misalkan kita mungkin memiliki suatu pilihan sebagai berikut :

“Jika saya punya uang, maka saya akan membeli laptop”

Dari pernyataan di atas, tentunya kita **hanya** akan membeli laptop jika (dan hanya jika) kita memang punya uang. Java (juga hampir semua bahasa pemrograman lainnya) memiliki apa yang dinamakan sebagai **nilai kebenaran** (nilai boolean) yang nilainya benar (**TRUE**) atau salah (**FALSE**). Dengan demikian, dapat kita simpulkan dari pernyataan di atas, kata **hanya** akan membeli sepeda motor jika (dan hanya jika)

pernyataan “Saya punya uang” bernilai **TRUE** (atau dengan kata lain “Saya benar-benar punya uang”); jika saya tidak punya uang, atau (dalam konteks ekspresi boolean) pernyataan “Saya punya uang” bernilai **FALSE**, saya tidak akan membeli laptop.

Macam seleksi if

Pernyataan if dapat digunakan untuk mengambil suatu keputusan berdasarkan suatu kondisi.

Bentuk dari pernyataan ini ada tiga macam, yaitu :

1. If

Pernyataan if atau if tunggal ini hanya digunakan jika kondisinya hanya 1.

Bentuk umumnya adalah :

```
if(kondisi)
{
    Pernyataan;
}
```

Keterangan :

- Kondisi digunakan untuk menentukan pengambilan keputusan
- Pernyataan, berisi perintah-perintah dan akan dijalankan jika kondisi bernilai benar, jika kondisi salah, tidak ada yang dikerjakan

2. If-Else

Pernyataan if-else tidak berdiri sendiri tanpa adanya if, karena bentuk pernyataan else adalah merupakan suatu ingkaran dari pernyataan if. Pada dasarnya pernyataan if...else hanya dapat menangani 2 kemungkinan keadaan.

Bentuk umumnya adalah :

```
if(kondisi)
{
    Pernyataan1;
}
else
{
    Pernyataan2;
}
```

Keterangan :

- Kondisi digunakan untuk menentukan pengambilan keputusan
- Pernyataan1, berisi perintah-perintah dan akan dijalankan jika kondisi bernilai benar
- Pernyataan2, berisi perintah-perintah dan akan dijalankan jika kondisi bernilai salah

3. If-else if

Pada dasarnya pernyataan if...else if dapat menangani lebih dari 2 kemungkinan keadaan/kondisi.

Bentuk umumnya adalah :

```
if(kondisi1)
{
    Pernyataan1;
}
else if(kondisi2)
{
    Pernyataan2;
}
else if(kondisi3)
{
    Pernyataan3;
}
else
{
    Pernyataan4;
}
```

Keterangan :

- Kondisi1 digunakan untuk menentukan pengambilan keputusan
- Pernyataan1, berisi perintah-perintah dan akan dijalankan jika kondisi1 bernilai benar
- Pernyataan2, berisi perintah-perintah dan akan dijalankan jika kondisi2 bernilai benar
- Pernyataan3, berisi perintah-perintah dan akan dijalankan jika kondisi3 bernilai benar
- Pernyataan4, berisi perintah-perintah dan akan dijalankan jika kondisi1, kondisi2, kondisi3 bernilai salah.

Operator Relasi

Operator atau tanda operasi adalah suatu tanda atau simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi suatu nilai. Operator relasi adalah operator penghubung yang berupa benar atau salah. Operator Relasi selalu menghasilkan tipe data boolean **true** atau **false**, operator ini sering digunakan untuk mengecek sebuah kondisi dan diletakkan pada percabangan if.

Operator	Arti
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

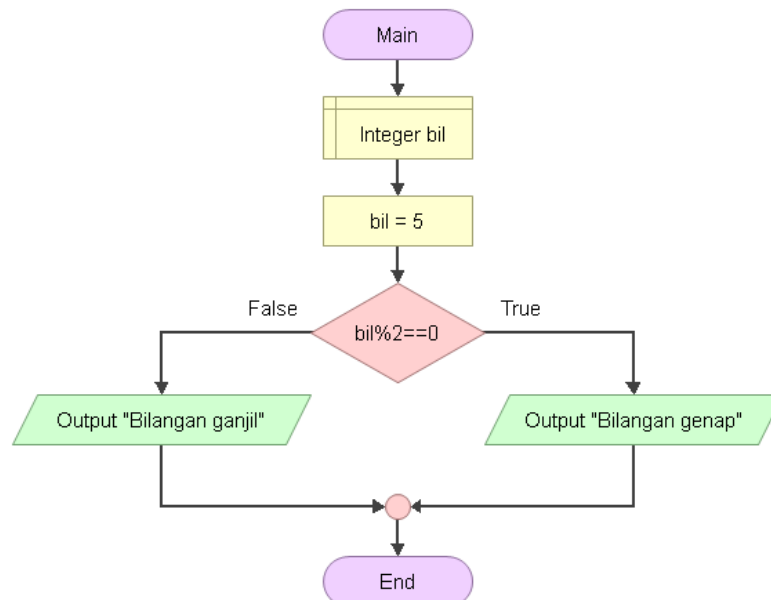


PRAKTIK

1. Buat program untuk menentukan apakah suatu bilangan merupakan bilangan 0
Program

```
public class modul5 {  
    public static void main(String[] args) {  
        int bil;  
        bil = 0;  
        if (bil==0) {  
            System.out.println("Bilangan 0");  
        }  
    }  
}
```

- a. Jalankan dan amati outputnya
 - b. Modifikasi praktik 1 dengan bilangan dapat dimasukkan oleh user
 - c. Uji dan amati output jika Bilangan yang dimasukkan adalah 0
 - d. Uji dan amati output jika Bilangan yang dimasukkan selain 0
2. Buatlah flowchart dan program untuk menentukan apakah suatu bilangan termasuk genap atau ganjil



Program

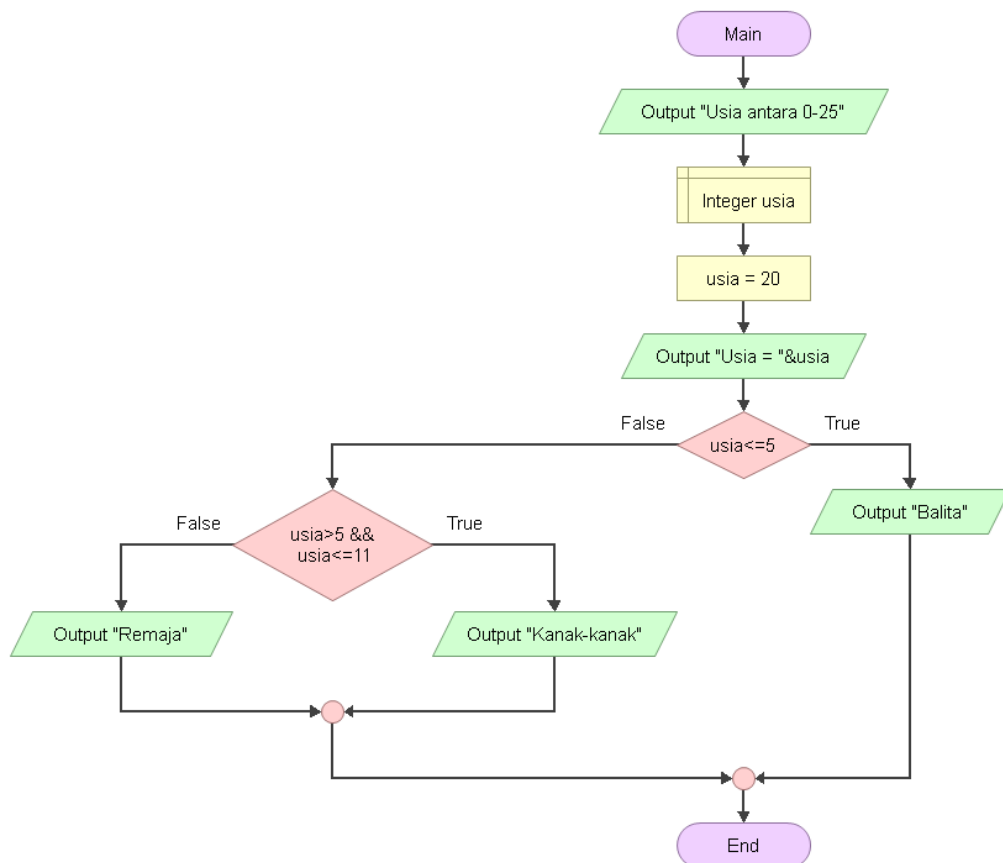
```
class modul5_1 {  
    public static void main(String[] args) {  
        int bil;  
        bil = 5;  
        if (bil % 2 == 0) {
```

```

        System.out.println("Bilangan genap");
    } else {
        System.out.println("Bilangan ganjil");
    }
}

```

- Jalankan dan amati outputnya
 - Modifikasi praktik 2 dengan bilangan dapat dimasukkan oleh user
 - Uji dan amati output jika Bilangan yang dimasukkan adalah 7
 - Uji dan amati output jika Bilangan yang dimasukkan selain 4
3. Buatlah flowchart dan ketikkan program untuk mengklasifikasikan usia seseorang antara usia 0 sampai 25.



Program

```

class modul5_2 {
    public static void main(String[] args) {
        System.out.println("Usia antara 0-25");
        int usia;
        usia = 20;
        System.out.println("Usia = " + usia);
        if (usia <= 5) {
            System.out.println("Balita");
        } else {

```

```

        if (usia > 5 && usia <= 11) {
            System.out.println("Kanak-kanak");
        } else {
            System.out.println("Remaja");
        }
    }
}

```

- Jalankan dan amati outputnya
- Modifikasi praktik 3 dengan menambahkan kondisi jika usia lebih dari 11 dan kurang dari atau sama dengan 25 disebut remaja dan jika lebih dari 25 disebut dewasa.



LATIHAN

- Buat program untuk menentukan predikat kelulusan berdasarkan IPK mahasiswa dengan ketentuan sebagai berikut :

IPK	Predikat Kelulusan
2,75 – 2,99	Memuaskan
3,00 – 3,50	Sangat memuaskan
3,51 – 4,00	Dengan pujian

Catatan : IPK dapat dimasukkan oleh user



TUGAS

- Buat flowchart dan program untuk menentukan bilangan terbesar dan terkecil dari 3 buah bilangan
- Buat program untuk menentukan apakah segitiga termasuk segitiga sama sisi, sama kaki atau segitiga sembarang berdasarkan panjang sisi-sisinya.



REFERENSI

- Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, hal (119-126,181-188)
- Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, hal (113-149)
- Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, hal (111-131)

MODUL 6

SELEKSI SWITCH-CASE



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep seleksi switch case untuk menyelesaikan kasus yang sederhana
2. Mahasiswa dapat mengubah dari satu bentuk seleksi ke bentuk seleksi yang lain



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

1. Seleksi Switch-Case

Selain menggunakan struktur *if*, Java juga mempunyai percabangan/seleksi dengan menggunakan *switch case*. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. Setiap ungkapan diungkapkan dengan sebuah nilai integral konstan, seperti sebuah nilai dengan tipe *byte*, *short*, *int* atau *char*.

Bentuk umum

```
switch (switch-expression)
{
    case value1: statement(s)1;
                  break;
    case value2: statement(s)2;
                  break;
    ....
    case valueN: statement(s)N;
```

```

        break;
    default: statement(s)-for-default;
}

```

Keterangan :

- **switch-expression** berisi ungkapan yang akan dicek valuenya. Harus ditulis diantara tanda (.....)
- **value1, value2** sampai **valueN** dilakukan pengecekan secara berurutan dimulai dari yang pertama, apabila ada value yang sesuai maka statement(s) yang mengikuti case dijalankan
- **break** ditemukan dari eksekusi pernyataan switch berakhir, berguna untuk menghentikan case. Switch-case tanpa break menyebabkan case berikutnya dieksekusi
- **default** hanya akan dijalankan jika value pada case tidak ada yang sesuai



PRAKTIK

1. Modifikasi program praktik 2 yang ada di modul pertemuan 5 untuk menentukan apakah bilangan termasuk genap atau ganjil menggunakan switch-case seperti di bawah

```

1. class modul6_1 {
2.     public static void main(String[] args) {
3.         int bil;
4.         bil = 5;
5.         switch(bil%2)
6.         {
7.             case 0:
8.                 System.out.println("Bilangan genap");
9.                 break;
10.            case 1:
11.                System.out.println("Bilangan ganjil");
12.                break;
13.            default:
14.                System.out.println("Salah bilangan");
15.        }
16.    }
17. }

```

- a. Jalankan dan amati hasilnya, jelaskan!
- b. Ubah bil=5 yang ada pada baris 4 menjadi bil=100, amati hasilnya, jelaskan!
2. Ketik program seperti dibawah untuk menampilkan jurusan berdasarkan kode jurusan

```

1. import java.util.Scanner;
2. class modul6_2 {
3.     public static void main(String[] args) {

```



```

4.      Scanner in=new Scanner(System.in);
5.      String kode;
6.      System.out.print("Masukkan kode jurusan = ");
7.      kode=in.next();
8.      switch(kode)
9.      {
10.         case "TI":
11.             System.out.println("Teknik Informatika");
12.             break;
13.         case "TK":
14.             System.out.println("Teknik Komputer");
15.             break;
16.         default:
17.             System.out.println("Salah masukkan kode jurusan");
18.     }
19. }
20. }

```

- Masukkan kode TI, jalankan dan amati hasilnya
- Masukkan kode TK, jalankan dan amati hasilnya
- Masukkan kode MI, jalankan dan amati hasilnya
- Modifikasi dengan menambahkan beberapa value di case untuk :
 Kode MI → Manajemen Informatika
 Kode SI → Sistem Informasi
 Kode KA → Komputerisasi Akuntansi
 Uji masing-masing value dan amati hasilnya

- Perhatikan algoritma untuk menampilkan jumlah hari berdasarkan kode bulan
 - Definisikan/masukan variabel kode_bulan
 - jika kode_bulan = 1, 3, 5, 7, 8, 10, 12
 - tampilan jumlah hari = 31
 - jika kode_bulan = 4, 6, 9, 11
 - tampilan jumlah hari = 30
 - jika kode_bulan = 2
 - tampilan jumlah hari = 28 atau 29
 - jika kode_bulan yang dimasukan salah
 - tampilkan komentar "Salah memasukan kode bulan"

Berdasarkan algoritma tersebut, ketik program dibawah untuk implementasinya

```

1. import java.util.Scanner;
2. public class CaseBulan
3. {
4.     public static void main(String args[])
5.     {
6.         Scanner masuk = new Scanner(System.in);
7.         int kode_bulan;
8.         System.out.print("Masukkan kode bulan : ");
9.         kode_bulan=masuk.nextInt();
10.        switch (kode_bulan) {
11.            case 1:case 3:case 5: case 7: case 8:case 10:case 12:
12.                System.out.println("Jumlah hari = 31 hari");
13.                break;

```

```

14.         case 4:case 6: case 9: case 11:
15.             System.out.println("Jumlah hari = 30 hari");
16.             break;
17.         case 2:
18.             System.out.println("Jumlah hari = 28 atau 29 hari");
19.             break;
20.     default:
21.         System.out.println("Salah masukan kode bulan");
22.         break;
23.     }
24. }
25. }

```

- a. Jalankan dan amati hasilnya jika kode bulan diisi 2
 - b. Jalankan dan amati hasilnya jika kode bulan diisi 9
 - c. Jalankan dan amati hasilnya jika kode bulan diisi 5
 - d. Jalankan dan amati hasilnya jika kode bulan diisi 20
4. Modifikasi praktik 3 dengan **menghilangkan** keyword **break** yang ada pada baris ke 13 lalu lakukan :
- a. masukkan kode bulan 5. Amati hasilnya, mengapa bisa demikian?
 - b. Masukkan kode bulan 9. Amati hasilnya, mengapa bisa demikian?



LATIHAN

1. Modifikasi praktik 3 menjadi bentuk seleksi if



TUGAS

1. Buat flowchart dan program dengan switch-case untuk menghitung luas bangun ruang berdasarkan pilihan user dengan ketentuan :
 Jika user memilih 1, maka akan menghitung luas persegi panjang
 Jika user memilih 2, maka akan menghitung luas lingkaran



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, hal (119-126,181-188)
2. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, hal (131)

MODUL 7

SELEKSI BERTINGKAT



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep seleksi bertingkat untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Seleksi pernah dipelajari pada pertemuan modul 5 dan 6 baik menggunakan if maupun switch-case. Seleksi bertingkat dapat diartikan sebagai seleksi di dalam seleksi. Beberapa bentuk kemungkinan seleksi bertingkat dua dapat dilihat seperti di bawah :

Bentuk 1	Bentuk 2	Bentuk 3
<pre>if(kondisi1) { Pernyataan1; } else { if(kondisi2) Pernyataan2; else Pernyataan3; }</pre>	<pre>if(kondisi1) { if(kondisi2) Pernyataan1; else Pernyataan2; } else { Pernyataan3; }</pre>	<pre>if(kondisi1) { if(kondisi2) Pernyataan1; else Pernyataan2; } else { if(kondisi3) Pernyataan3; else Pernyataan4; }</pre>

Keterangan :

Bentuk 1

- ❖ Jika kondisi 1 bernilai Benar, maka akan dikerjakan Pernyataan 1,
- ❖ Jika kondisi 1 bernilai Salah, maka akan mengerjakan bagian else. Bagian else akan dikerjakan dimulai dengan pengecekan kondisi 2, jika kondisi 2 bernilai Benar, maka Pernyataan 2 akan dikerjakan, jika kondisi 2 bernilai Salah, maka Pernyataan 3 yang akan dikerjakan.

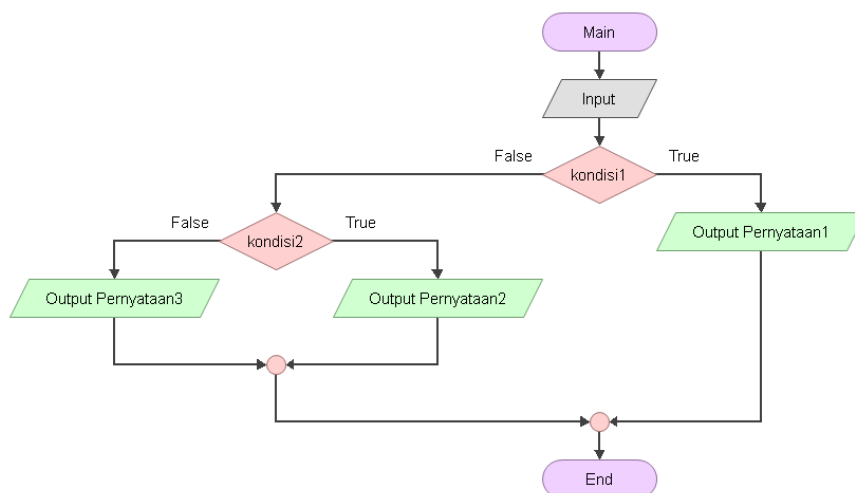
Bentuk 2

- ❖ Jika kondisi 1 bernilai Benar, maka akan dikerjakan statement yang ada didalam if yaitu dimulai dari pengecekan kondisi 2, jika kondisi 2 bernilai Benar, maka Pernyataan 1 akan dikerjakan, tetapi jika kondisi 2 bernilai Salah, maka Pernyataan 2 yang akan dikerjakan.
- ❖ Jika kondisi 1 bernilai Salah, maka Pernyataan 3 yang akan dikerjakan.

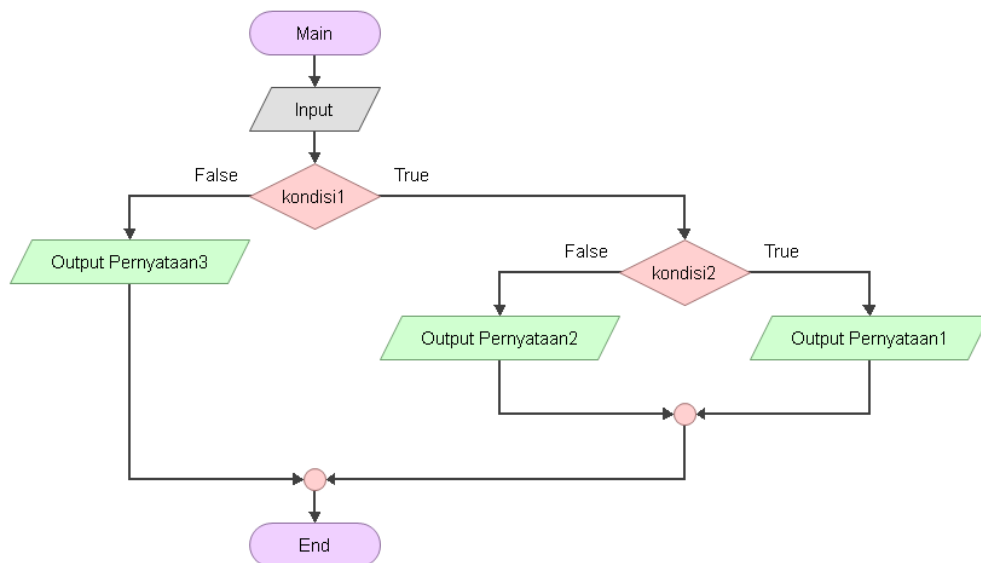
Bentuk 3

- ❖ Jika kondisi 1 bernilai Benar, maka akan dikerjakan statement yang ada didalam if yaitu dimulai dari pengecekan kondisi 2, jika kondisi 2 bernilai Benar, maka Pernyataan 1 akan dikerjakan, tetapi jika kondisi 2 bernilai Salah, maka Pernyataan 2 yang akan dikerjakan.
- ❖ Jika kondisi 1 bernilai Salah, maka akan mengerjakan bagian else. Bagian else akan dikerjakan dimulai dengan pengecekan kondisi 3, jika kondisi 3 bernilai Benar, maka Pernyataan 3 akan dikerjakan, tetapi jika kondisi 3 bernilai Salah, maka Pernyataan 4 yang akan dikerjakan.

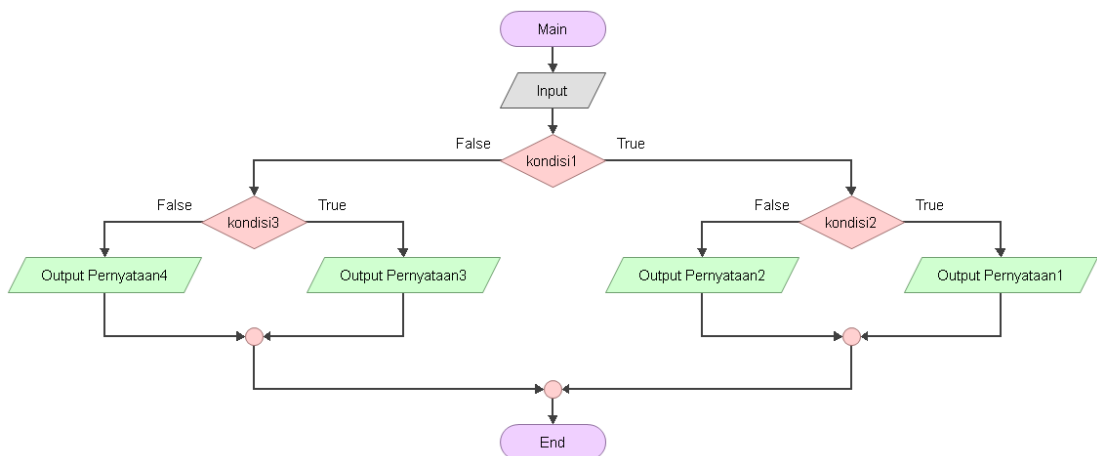
Flowchart Bentuk 1



Flowchart Bentuk 2



Flowchart Bentuk 3



Selain bentuk 1, bentuk 2, dan bentuk 3, masih ada bentuk-bentuk seleksi bertingkat yang lain, misalnya, if-else if didalam if, if di dalam switch, switch di dalam if, dan lain-lain.



PRAKTIK

1. Ketik program di bawah

```
1. import java.util.Scanner;  
2. public class modul7_1{
```

```

3. public static void main (String arg[]){
4.     Scanner in=new Scanner(System.in);
5.     int nilai;
6.     System.out.print("Masukkan angka bulat (0 - 100) ");
7.     nilai=in.nextInt();
8.     if (nilai>=60)
9.     {
10.         if (nilai>=80)
11.             System.out.println("Nilaimu bagus sekali ");
12.         else
13.             System.out.println("Nilaimu bagus ");
14.     }
15.     else
16.         System.out.println("Nilaimu kurang ");
17. }
18. }

```

- Jalankan program, masukkan nilai 70, amati hasilnya, jelaskan!
- Jalankan program, masukkan nilai 90, amati hasilnya, jelaskan!
- Jalankan program, masukkan nilai 50, amati hasilnya, jelaskan!

2. Modifikasi praktik 1, dengan ketentuan :

Jika nilai < 60, maka ada proses seleksi lagi yaitu :

Jika nilai >= 30 maka akan ditampilkan nilaimu kurang

Jika nilai < 30 maka akan ditampilkan nilaimu jelek

3. Ketik program di bawah

```

1. import java.util.Scanner;
2. public class modul7_3{
3.     public static void main (String arg[]){
4.         Scanner in=new Scanner(System.in);
5.         Scanner masuk=new Scanner(System.in);
6.         String pil, jenis;
7.         System.out.println("Hitung persegi/lingkaran");
8.         System.out.println("=====");
9.         System.out.print("masukkan pilihan : ");
10.        pil=in.next();
11.        switch(pil)
12.        {
13.            case "persegi":
14.                int sisi;
15.                System.out.print("masukkan sisi : ");
16.                sisi=masuk.nextInt();
17.                System.out.print("luas/keliling : ");
18.                jenis=in.next();
19.                switch(jenis)
20.                {
21.                    case "luas":
22.                        int luas=sisi*sisi;
23.                        System.out.println("Luas persegi : "+luas);
24.                        break;
25.                    case "keliling":
26.                        int kel=4*sisi;
27.                        System.out.println("Keliling persegi : "+kel);
28.                        break;

```

```

29.         default :
30.             System.out.println("Salah masukkan jenis");
31.         }
32.         break;
33.         case "lingkaran":
34.             double jari;
35.             System.out.print("masukkan jari-jari : ");
36.             jari=masuk.nextDouble();
37.             System.out.print("luas/keliling : ");
38.             jenis=in.next();
39.             switch(jenis)
40.             {
41.                 case "luas":
42.                     double luasl=3.14*jari*jari;
43.                     System.out.println("Luas lingkaran : "+luasl);
44.                     break;
45.                 case "keliling":
46.                     double kell=2*3.14*jari;
47.                     System.out.println("Luas lingkaran : "+kell);
48.                     break;
49.                 default :
50.                     System.out.println("Salah masukkan jenis");
51.             }
52.         break;
53.         default :
54.             System.out.println("Salah masukkan pilihan");
55.     }
56. }
57. }

```

- Jalankan program diatas dengan menguji beberapa kemungkinan pilihan maupun jenis
- Hilangkan keyword break yang ada di baris ke 32 kemudian uji dengan memasukkan pilihan “persegi” dan menghitung “keliling”, amati yang terjadi, mengapa bisa demikian



LATIHAN

- Buat program dengan if bertingkat untuk menampilkan harga mobil/motor berdasarkan pilihan yang dimasukkan oleh user dengan ketentuan :
 Pilih 1, jika pilihan mobil dan ada pilihan selanjutnya apakah Jazz (170 jt), Brio (120 jt), Mobilio (170 jt)
 Pilih 2, jika pilihan motor dan ada pilihan selanjutnya Vario(16 jt), Beat (14 jt), Vixion(20 jt)



TUGAS

1. Buat program dengan switch bertingkat untuk menampilkan besaran SPA yang harus dibayar untuk kuliah di STMIK AKAKOM berdasarkan jenjang dan jurusan yang dipilih dengan ketentuan :

TK,KA,MI (D3)	TI(S1)	SI(S1)
10.000.000	13.000.000	12.000.000



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 8

PERULANGAN WHILE



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep perulangan while untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah.

Karakteristik while() adalah:

1. Dilakukan pengecekan kondisi terlebih dahulu sebelum dilakukan perulangan. Jika kondisi yang dicek bernilai benar (true) maka perulangan akan dilakukan.
2. Blok statement tidak harus ada. Struktur tanpa statement akan tetap dilakukan selama kondisi masih true.

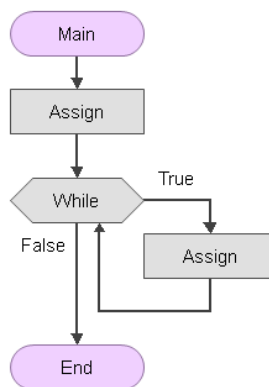
Bentuk umum :

while (ungkapan) Pernyataan

Keterangan :

- bagian pernyataan akan dieksekusi selama ungkapan dalam **while** bernilai benar.
- Pengujian terhadap ungkapan pada **while** dilakukan sebelum bagian pernyataan.
- Kemungkinan pernyataan pada **while** tidak dijalankan sama sekali, jika ketemu kondisi yang pertama kali bernilai salah.

Flowchart perulangan while dapat dilihat seperti gambar berikut :



Catatan :

Pernyataan perulangan dengan while akan selalu dikerjakan jika ungkapan selalu benar. Oleh karena itu, kita harus membuat kondisi suatu saat ungkapan bernilai salah agar perulangan berakhir.



PRAKTIK

1. Ketik program di bawah

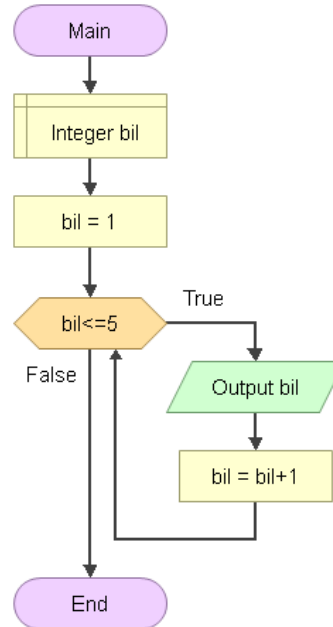
```
1. import java.util.Scanner;
2.     public class UlangWhile1
3.     {
4.         public static void main(String args[])
5.         {
6.             Scanner masuk = new Scanner(System.in);
7.             int bil;
8.             bil=1;
9.             while (bil<=5) {
10.                 System.out.println(bil);
11.                 bil=bil+1;
12.             }
```

```

13.         }
14.     }

```

Buat flowchart untuk program diatas seperti berikut :



- Modifikasi praktik 1 dengan mengubah pernyataan **bil=1** yang ada pada baris 8 menjadi **bil=5**, dan pernyataan **while(bil<=5)** yang ada di baris ke 9 dengan **while(bil>=1)** dan **bil=bil+1** pada baris 11 menjadi **bil=bil-1**, amati hasil outputnya, kenapa bisa demikian, jelaskan !
- Buat program untuk menampilkan tulisan STMIK AKAKOM dan buat suatu pernyataan jika tulisan tersebut bisa ditampilkan selama jawaban True (Ya) dan akan di hitung jumlah yang di tampilkan

```

import java.util.Scanner;
public class modul8_3 {
    public static void main(String[] args) {
        boolean running = true;
        int counter = 0;
        String jawab;
        Scanner scan = new Scanner(System.in);
        while( running ) {
            System.out.println("STMIK AKAKOM");
            System.out.print("Tampilkan Tulisan lagi [ya/tidak]> ");
            jawab = scan.nextLine();
            // cek jawabannya, kalau ya maka berhenti mengulang
            if( jawab.equalsIgnoreCase("tidak") ){
                running = false;
            }
            counter++;
        }
        System.out.println("Anda sudah melakukan perulangan sebanyak "
+ counter + " kali");
    }
}

```

```
}  
}
```

- a. Simpan dan jalankan
 - b. Ujilah dengan mengisi **ya** sebanyak **2** kali , amati hasilnya
 - c. Lanjutkan menguji dengan mengisi **tidak**, amati hasilnya
4. Buat program dengan while untuk mencetak bilangan genap dari 0 sampai 10



LATIHAN

1. Modifikasi praktik 4 agar bilangan genap yang dicetak dimulai dan diakhiri menurut keinginan user.



TUGAS

1. Hitunglah jumlah dan rata-rata bilangan positif, dimana banyaknya data ditentukan dari data yang dimasukan dengan keyboard, contoh hasil output yang diinginkan sbb:

```
Banyaknya data : 4  
Data ke-1 : 3  
Data ke-2 : 5  
Data ke-3 : 2  
Data ke-4 : 6  
Rata-rata : 4.0  
Jumlah : 16.0  
Press any key to continue . . .
```

2. Buatlah algoritma, flowchat dan program untuk menghasilkan deret suatu bilangan, yang banyaknya data (n) di inputkan.

Jika n =3 menghasilkan deret 1 4 9

Jika n =4 menghasilkan deret 1 4 9 16



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)

2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 9

PERULANGAN DO-WHILE



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep perulangan do-while untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Perintah ini sangat mirip dengan perintah while, perbedaannya adalah bahwa pada perintah while pemeriksaan ekspresi logika dilakukan di awal, jadi jika ekspresi bernilai false maka pengulangan tidak akan dijalankan sama sekali, namun pada perintah do – while ini pemeriksaan dilakukan di akhir kalang (loop) sehingga walaupun sejak awal ekspresi akan bernilai false maka minimal kalang (loop) akan dieksekusi sekali. Seperti pada perintah while proses pengulangan akan berjalan jika kondisi yang diperiksa pada while masih bernilai benar dan pengulangan akan dihentikan jika kondisinya sudah bernilai salah.

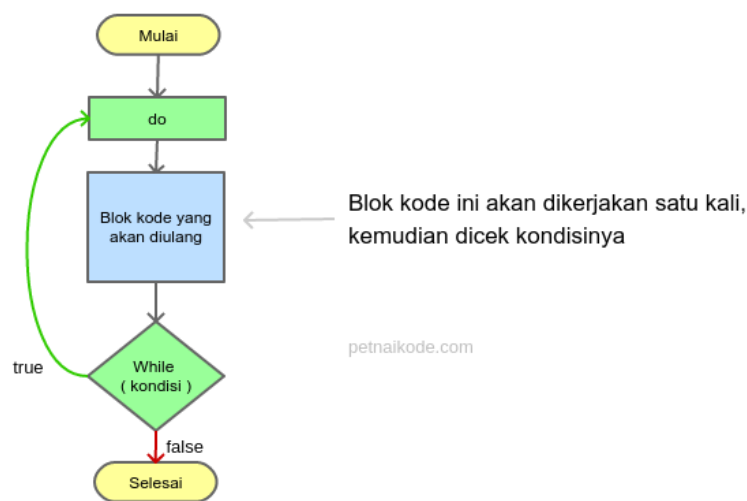
Sintaks penulisannya sebagai berikut :

```
do
{
    Pernyataan;
} while(ungkapan);
```

Keterangan :

- Bagian pernyataan dijalankan secara berulang sampai ungkapan bernilai salah.
- Pengujian ungkapan dilakukan setelah bagian pernyataan, maka pada pernyataan **do ... while** minimal akan dijalankan sekali, karena begitu masuk ke blok perulangan, tidak ada cek kondisi tetapi langsung mengerjakan pernyataan.

Flowchart perulangan while dapat dilihat seperti gambar berikut :



PRAKTIK

1. Tuliskan program berikut ini :

```
1. import java.util.Scanner;
2.     public class UlangDo1
3.     {
4.         public static void main(String args[])
5.         {
6.             Scanner masuk = new Scanner(System.in);
7.             int bil;
8.             bil=1;
9.             do {
10.                System.out.println(bil);
11.                bil+=2;
12.            } while (bil<=10);
13.        }
14.    }
```


- a. Amati hasil outputnya !
- b. Modifikasi dengan mengganti bil=1 yang ada di baris 8 menjadi bil=20, amati kembali hasil outputnya, mengapa bisa demikian?

2. Salinlah program berikut

```
1. import java.util.Scanner;
2. public class UlangDo
3. {
4.     public static void main(String args[])
5.     {
6.         Scanner masuk = new Scanner(System.in);
7.         int bil;
8.         bil=1;
9.         do {
10.             System.out.println(bil);
11.             bil++;
12.         }
13.         while (bil<=5);
14.     }
15. }
16.
```

3. Modifikasi praktik 2 agar output yang ditampilkan menjadi 5 4 3 2 1
4. Buat program untuk menampilkan tulisan “STMIK AKAKOM” sebanyak yang diinginkan user



LATIHAN

1. Buatlah program untuk mencetak konversi suhu dari celcius ke fahrenheit mulai dari 20 sampai 30
2. Modifikasi praktik 4 dimana jika user memasukkan kata “ya”, maka akan terus ditampilkan, dan program akan berhenti menampilkan tulisan “STMIK AKAKOM” jika user sudah memasukkan kata “tidak”.



TUGAS

1. Ubah tugas yang ada di modul pertemuan 8 menjadi bentuk do-while



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 10

PERULANGAN FOR



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep perulangan FOR untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

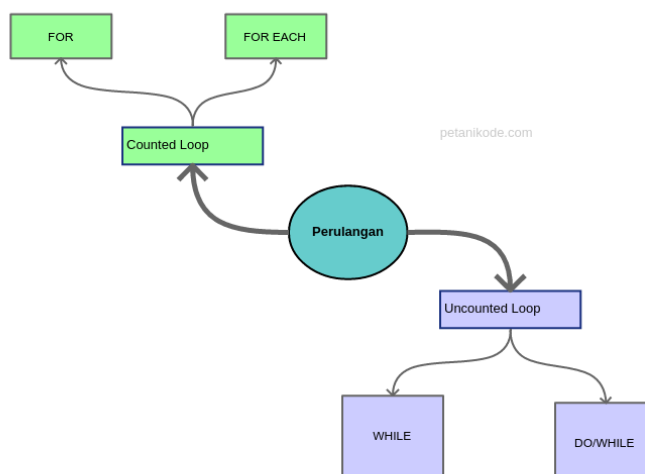
Perintah pengulangan ini digunakan untuk mengulangi suatu baris perintah tertentu dengan jumlah pengulangan yang sudah diketahui sebelumnya, selain itu perintah pengulangan ini dilengkapi dengan pencacah (counter). Perintah for mempunyai tiga buah argumen. Argumen yang pertama merupakan deklarasi dan pemberian nilai awal variabel, yaitu variabel yang digunakan sebagai pencacah (counter), argumen yang kedua (di tengah) pada perintah for ini merupakan suatu kondisi (ekspresi logika) yang diuji, apabila menghasilkan nilai true maka perulangan akan dilakukan (dilanjutkan) sedangkan jika false maka pengulangan akan dihentikan. Argumen yang ketiga, yaitu yang paling kanan merupakan perintah yang akan dieksekusi setiap kali perintah dalam perulangan dijalankan. Pada umumnya argument ketiga ini berupa perintah untuk menaikkan (increment) atau menurunkan (decrement) nilai variabel pencacah yang diinisialisasi pada argument pertama. Nilai variabel penghitung akan secara otomatis bertambah atau berkurang tiap kali sebuah pengulangan dilaksanakan tergantung perintah yang ditulis pada argumen ini. Sintaks penulisan perintah for sebagai berikut :

Bentuk umum :

```
for(nilai_awal; ekspresi_logika; penambahan/penurunan)
{
    Pernyataan;
}
```

Argumen pertama bisa berupa variable, biasanya bertipe int dan langsung diberi nilai awal, apabila nama variabel sudah dideklarasikan pada baris sebelumnya maka variabel tersebut dapat langsung digunakan dalam perintah for ini dan akan tetap dikenal meskipun keluar dari kalang (loop) yaitu blok perulangan, namun jika dideklarasikan di dalam perintah for, maka identifier variabel tersebut hanya dikenal di dalam kalang dan tidak akan dikenal diluar kalang (setelah perintah for selesai). Argumen yang kedua merupakan ekspresi logika, biasanya berupa perbandingan nilai (memakai operator hubungan/ perbandingan) dari variabel yang dideklarasikan pada argument pertama, sedangkan argument yang terakhir berupa perintah increment atau decrement dari variabel yang dideklarasikan pada argument pertama memakai operator unary ++ atau -- . Perulangan dalam pemrograman dibagi menjadi dua jenis:

1. **Counted loop:** Perulangan yang jumlah pengulangannya terhitung atau tentu.
2. **Uncounted loop:** Perulangan yang jumlah pengulangannya tidak terhitung atau tidak tentu.



Counted loop terdiri dari perulangan *For* dan *For each*. Sedangkan *Uncounted loop* terdiri dari perulangan *While* dan *Do/While*



PRAKTIK

1. Tuliskan program di bawah

```
1. public class UlangFor
2. {
3.     public static void main (String args[])
4.     {
5.         int bil;
6.         for (bil=1;bil<=5;bil++)
7.             System.out.println(bil);
8.     }
9. }
```

- Jalankan dan amati hasilnya
- Ubah `bill++` menjadi `bil--`, amati hasilnya

2. Tuliskan program di bawah

```
1. public class UlangFor2
2. {
3.     public static void main (String args[])
4.     {
5.         int bil;
6.         for (bil=5;bil>=1;bil--)
7.             System.out.println(bil);
8.     }
9. }
```

- Jalankan dan amati hasilnya
- Ubah `bil>=1` menjadi `bil<=1`, jalankan dan amati hasilnya
- Dari hasil modifikasi praktik 2 b, modifikasi kembali menjadi bentuk `do-while` jalankan kembali dan amati hasilnya. Apakah hasilnya sama? jelaskan!

3. Gabungkan kedua program tersebut agar mendapat 2 keluaran urutan ascending dan urut descending

```
1. public class contoh_for
2. {
3.     public static void main (String[] args)
4.     {
5.         System.out.println ("Pengulangan Dengan For Ascending");
6.         for (int bil = 1; bil <= 5; ++bil)
7.             System.out.println ("Indeks Naik ke = "+bil);
8.         System.out.println ("Pengulangan Dengan For Descending");
9.         for (int bil = 5; bil > 0; --bil)
10.            System.out.println ("Indeks Naik ke = "+bil);
11.     }
12. }
```



LATIHAN

1. Buat program dengan for untuk menampilkan bilangan kelipatan 5 dari 10 sampai 50
2. Modifikasi latihan 1 dimana batas awal dan akhir dimasukkan oleh user



TUGAS

1. Buat program untuk menjumlahkan deret bilangan bulat positif yang lebih kecil dari 20. Contoh output :

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Jumlah:190
Press any key to continue . . .
```

2. Modifikasi latihan nomor 2 dengan menambahkan hasil penjumlahan dari deret kelipatan 5 yang sudah diperoleh



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 11

PERULANGAN BERTINGKAT



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep perulangan bertingkat dua untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Perulangan adalah model pengembangan sistem yang bersifat dinamis dalam artian setiap tahapan proses pengembangan sistem dapat diulang jika terdapat kekurangan atau kesalahan. Setiap tahapan pengembangan sistem dapat dikerjakan berupa ringkasan dan tidak lengkap, namun pada akhir pengembangan akan didapatkan sistem yang lengkap pada pengembangan sistem.

Di dalam komputer/pemrograman, **iterasi** adalah sifat tertentu dari algoritma atau program komputer di mana suatu urutan atau lebih dari langkah algoritmik dilakukan di loop (perulangan) program.

Perulangan bersarang/bertingkat/nested loop adalah apabila pada blok statement **perulangan** terdapat **perulangan** lagi, jadi dapat dikatakan perulangan di dalam perulangan. perulangannya bisa 2 atau lebih.

Sebuah program mengizinkan blok pengulangan di dalam blok pengulangan lainnya, dan tidak membatasi jenis pengulangan apa yang boleh berada di dalam pengulangan lainnya, misalnya di dalam blok pengulangan for terdapat pengulangan while, atau

didalam pengulangan while terdapat pengulangan for. Beberapa bentuk perulangan bertingkat dapat dilihat sebagai berikut :

Bentuk 1
<pre>for(ungkapan1;ungkapan2;ungkapan3) { for(ungkapan4;ungkapan5;ungkapan6) { Statement; } }</pre>
Bentuk 2
<pre>ungkapan 1 while(ungkapan2) { for(ungkapan4;ungkapan5;ungkapan6) { Statement; } ungkapan3 }</pre>
Bentuk 3
<pre>ungkapan 1 do { Ungkapan4; while(ungkapan5) { Statement; Ungkapan6; } ungkapan3; } while (ungkapan2);</pre>

Selain bentuk-bentuk diatas, masih banyak bentuk-bentuk yang lain.



PRAKTIK

1. Tuliskan praktik yang harus dikerjakan mahasiswa disini

```
1. public class Perulanganbersarang{
2.     public static void main(String[] args) {
3.         int bilangan1;
4.         int bilangan2;
5.         for (bilangan1=1; bilangan1<=3; bilangan1++) {
6.             for (bilangan2=1; bilangan2<=2; bilangan2++) {
7.                 System.out.println "[" + bilangan1 + "]" + "[" + bilangan2 + "]");
8.             }
9.         }
10.    }
```



```
11. }  
12.
```

2. Modifikasi praktik 1 dengan menggantinya menjadi perulangan while dalam while
3. Ketik program berikut

```
1. import java.util.Scanner;  
2. public class Latihan {  
3.     public static void main(String[] args){  
4.         Scanner input = new Scanner(System.in);  
5.         System.out.print("Masukan Jumlah Baris: ");  
6.         int j_baris = input.nextInt(); //Menentukan Jumlah Baris  
7.  
8.         for(int baris=1; baris<=j_baris; baris++){  
9.  
10.            for(int s=1; s<=baris; s++){  
11.                System.out.print("* ");  
12.            }  
13.            Sstem.out.println();  
14.        }    }  
15.
```

jalankan program tersebut dengan memberi masukan 5 , 10 dan 15 apa hasil keluaran dari masing-masing masukan tersebut, analisis dan jelaskan.



LATIHAN

1. Tulislah sebuah program yang menggunakan loop for untuk menghasilkan keluaran sebagai berikut :

```
1  
12  
123  
1234  
12345
```



TUGAS

1. Buatlah program menggunakan looping while yang membentuk baris angka seperti berikut

```
1  
1 2  
1 2 3
```

1 2 3 4
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 12

SELEKSI DALAM PERULANGAN



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep perulangan while untuk menyelesaikan kasus



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Pada dasarnya sebuah program dieksekusi secara runtut dari mulai statement yang pertama kali dibaca dilanjutkan dengan statement yang dibaca berikutnya.

Tetapi alur pemrosesan itu bisa diubah dengan menggunakan **seleksi dan perulangan** sehingga memungkinkan sebuah program menjalankan tugas yang lebih kompleks.

Alur pemrosesan dimulai dari bagian utama program.

- * Seleksi dan iterasi/perulangan dapat digabungkan dengan dua kemungkinan, yang pertama seleksi dalam perulangan dan yang kedua adalah perulangan dalam seleksi (akan dibahas pada pertemuan ke-13), gambaran sederhana dari model yang pertama adalah:

```
for(ungkapan1;ungkapan2;ungkapan3)
{
    if(kondisi)
    {
```

```

        Statement;
    }
}

```

Keterangan

- * Dalam model ini, statement baru akan dijalankan jika kondisi dalam if bernilai benar. Statement akan terus dijalankan selama ungkapan2 dalam for masih bernilai benar.



PRAKTIK

1. Ketik dan amati hasil output dari program berikut ini

```

1. public class ContohBreak {
2.     public static void main(String args[])
3.     {
4.         int [] numbers={10,20,30,40,50};
5.         for(int x:numbers)
6.         {
7.             if(x==30)
8.             {
9.                 break;
10.            }
11.            System.out.print(x);
12.            System.out.print("\n");
13.        }
14.    }
15.}

```

2. Ketik dan amati hasil output dari program berikut ini

```

1. public class ContohBreakWhile {
2.     public static void main(String args[])
3.     {
4.         int bil=1;
5.         while(bil<=10)
6.         {
7.             if(bil>5)
8.             break;
9.             System.out.println(bil);
10.            bil++;
11.        }
12.    }
13.}

```

3. Cobalah program dibawah, jalankan dan amati hasilnya

```

1. import java.util.Scanner;
2. public class SeleksiDalamPerulangan {
3.     public static void main(String args[])
4.     {
5.         int JPB=0, JM=0, jumlah=0;
6.         int jawab=1;
7.         System.out.println("Kategori barang : ");
8.         System.out.println("1. Pecah belah : ");
9.         System.out.println("2. Makanan : ");
10.        Scanner masuk=new Scanner(System.in);
11.        while(jawab!=0)
12.        {
13.            System.out.println("Masukkan Kategori barang (1,2): ");
14.            int kategori=masuk.nextInt();
15.            if(kategori==1)
16.            {
17.                JPB++;
18.            }
19.            else
20.            {
21.                JM++;
22.            }
23.            System.out.println("Barang habis ? (0=ya,1=tidak) ");
24.            jawab=masuk.nextInt();
25.        }
26.        System.out.println("");System.out.println("");
27.        System.out.println("Data yang dimasukkan ");
28.        System.out.println("Jumlah Pecah belah = "+JPB);
29.        System.out.println("Jumlah makanan = "+JM);
30.    }
31.}

```



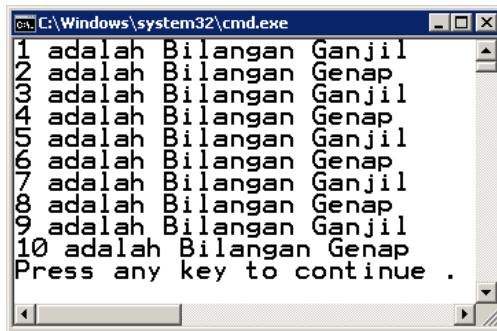
LATIHAN

1. Modifikasi praktik 1 dengan mengubah bentuk perulangan for menjadi while dan do-while, amati hasilnya, jelaskan dalam laporan



TUGAS

1. Buatlah program untuk menghasilkan output sebagai berikut :



```
C:\Windows\system32\cmd.exe
1 adalah Bilangan Ganjil
2 adalah Bilangan Genap
3 adalah Bilangan Ganjil
4 adalah Bilangan Genap
5 adalah Bilangan Ganjil
6 adalah Bilangan Genap
7 adalah Bilangan Ganjil
8 adalah Bilangan Genap
9 adalah Bilangan Ganjil
10 adalah Bilangan Genap
Press any key to continue .
```



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 13

PERULANGAN DALAM SELEKSI



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat menyelesaikan kasus dengan menggabungkan konsep perulangan dalam seleksi



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Seperti yang telah dijelaskan pada modul 12 bahwa Seleksi dan iterasi/perulangan dapat digabungkan dengan dua kemungkinan, yang pertama seleksi dalam perulangan dan yang kedua adalah perulangan dalam seleksi. Pada modul 13 ini, akan dibahas model yang kedua yaitu perulangan dalam seleksi. gambaran sederhana dari model ini salah satunya adalah :

```
if(kondisi)
{
    for(ungkapan1;ungkapan2;ungkapan3)
    {
        Statement;
    }
}
```

Dalam model ini, perulangan baru akan dijalankan jika kondisi dalam if bernilai benar. Statement akan terus dijalankan selama ungkapan2 dalam for masih bernilai benar.



PRAKTIK

1. Amati output dari 2 program berikut ini :

Program pertama

```
1. import java.util.Scanner;
2. public class For_if1 {
3.     public static void main(String args[]) {
4.         Scanner masuk = new Scanner(System.in);
5.         int nilai, i;
6.         System.out.print(" Masukan angka bulat (0 100) = ");
7.         nilai=masuk.nextInt();
8.         if (nilai>=60) {
9.             for (i=0; i<=5; i++) {
10.                 System.out.println( "Kamu hebat...!");
11.             }
12.         }
13.         else {
14.             for (i=0; i<=5; i++) {
15.                 System.out.println("Belajar lagi ya....");
16.             }
17.         }
18.     }
19. }
```

Program kedua

```
1. import java.util.Scanner;
2. public class For_if2
3. {
4.     public static void main(String args[])
5.     {
6.         Scanner masuk = new Scanner(System.in);
7.         int nilai, i;
8.         System.out.print("Masukan angka bulat = ");
9.         nilai=masuk.nextInt();
10.        if (nilai>5) {
11.            for (i=nilai; i>=1; i--) {
12.                System.out.print(i);
13.            }
14.        }
15.        else {
16.            for (i=nilai; i<=10; i++) {
17.                System.out.print(i);
18.            }
19.        }
20.    }
21. }
```


2. Cobalah program berikut

```
1. import java.util.Random;
2. import java.util.Scanner;
3. public class for_if5{
4.     public static void main (String arg[]){
5.         String prinsip;
6.         int i, pil, acak;
7.         Scanner masuk=new Scanner(System.in);
8.         Random rd=new Random();
9.         System.out.println("    M e n u");
10.        System.out.println("1. Bilangan Random");
11.        System.out.println("2. Semboyan Hidup");
12.        System.out.print("Masukan pilihan anda = ");
13.        pil=masuk.nextInt();
14.        if (pil == 1){
15.            System.out.println(" angka keberuntunganmu adalah =");
16.            for(i=1;i<=3; i++){
17.                acak = rd.nextInt(20);
18.                System.out.println(acak);
19.            }
20.        }
21.        else {
22.            System.out.print("Masukan prinsip hidup anda = ");
23.            prinsip=masuk.next();
24.            System.out.println("Prinsipmu adalah : " +prinsip);
25.        }
26.    }
27. }
28.
```

Ada dua menu pilihan dari program di atas, apa hasil outputnya jika anda memilih Menu 1 ? apakah hasilnya akan sama dengan teman di sebelah kiri atau kanan anda. Dan apa hasil outputnya jika anda memilih menu 2



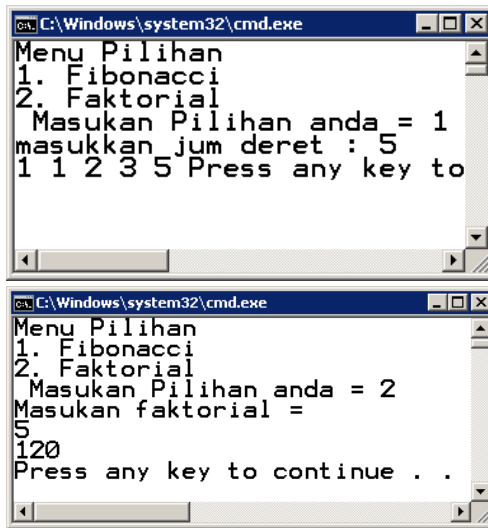
LATIHAN

1. Soal latihan akan disampaikan dosen dalam kelas



TUGAS

1. Buatlah program dalam suatu menu untuk menghitung bilangan Fibonacci dan Faktorial seperti gambar berikut :



```
C:\Windows\system32\cmd.exe
Menu Pilihan
1. Fibonacci
2. Faktorial
Masukan Pilihan anda = 1
masukkan jum deret : 5
1 1 2 3 5 Press any key to

C:\Windows\system32\cmd.exe
Menu Pilihan
1. Fibonacci
2. Faktorial
Masukan Pilihan anda = 2
Masukan faktorial =
5
120
Press any key to continue . .
```



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)

MODUL 14

STUDI KASUS

SEKUENSI, SELEKSI DAN ITERASI



CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan konsep Sekuensi, seleksi dan iterasi untuk menyelesaikan kasus yang sederhana
2. Mahasiswa dapat mengubah dari satu bentuk seleksi ke bentuk seleksi yang lain begitu juga dalam perulangan



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK
3. Flowgorithm



DASAR TEORI

Teori mengenai sekuensi, seleksi dan iterasi dapat dilihat pada modul pertemuan sebelumnya.



PRAKTIK

Kerjakan soal-soal berikut :

1. Suatu rangkaian yang tersusun atas 3 resistor yang di paralel, buatlah diagram alir/flowchart yang meminta nilai R1, R2, R3 dari keyboard untuk menampilkan nilai R dengan rumus :

$$R = 1/(1/R1+1/R2+1/R3)$$

2. Buat program untuk menghitung nilai R berdasarkan kasus praktik 1
3. Buatlah flowchart dan program untuk menghitung nilai rata-rata dari minimal 5 data yang dimasukan menggunakan keyboard
4. Modifikasi praktik 3 agar jumlah data yang akan dihitung rata-ratanya bisa fleksibel sesuai keinginan user



LATIHAN

1. Buatlah program untuk menampilkan Segitiga bintang sebagai berikut :

```

*****
*****
*****
***
*
```



TUGAS

1. Buatlah program untuk menghitung akar-akar persamaan kuadrat (diskriminan) $ax^2+bx+c=0$ dimana memiliki akar penyelesaian berupa x_1 dan x_2 , dengan ketentuan sebagai berikut :
 1. Jika diskriminan lebih besar daripada nol, kedua nilai x yang menjadi solusi persamaan tersebut berupa bilangan real
 2. Jika diskriminan sama dengan nol, kedua nilai x yang menjadi solusi persamaan tersebut berupa bilangan kembar
 3. Jika diskriminan lebih kecil dari pada nol, kedua nilai x yang menjadi solusi persamaan tersebut berupa bilangan kompleks

Penentuan X_1 dan X_2 berdasarkan rumus :

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



REFERENSI

1. Deitel P., Dietel H., 2015, *Java How to Program Tenth edition*, Deitel & Associates, Inc., Prentice Hall, New Jersey, Hal (119-126,181-188)
2. Rosa A. S., 2018, *Logika Algoritma dan Pemrograman Dasar*, Modula, Bandung, Hal (113-149)
3. Abdul Kadir, 2012, *Algoritma & Pemrograman Menggunakan Java*, CV. Andi Offset, Yogyakarta, Hal (111-155)