

DAFTAR ISI

Kata Pengantar	i
Daftar Isi.....	ii
BAB 1 Pendahuluan.....	1
1.1 Tujuan.....	1
1.2 Program Salam	1
1.3 Memulai Java dengan NotePad	1
1.4 Memulai Java dengan Eclipse	2
1.5 Memulai Java dengan NetBeans	3
1.6 Latihan 1. Biodata	4
1.7 Latihan 2. Biodata 2.....	4
1.8 Menuliskan Komentar	4
1.9 Latihan 3. Menulis Komentar.....	5
BAB 2 Variabel dan Tipe Data	6
2.1 Tujuan.....	6
2.2 Variabel.....	6
2.3 Tipe Data.....	6
2.4 Tipe Data Angka	7
2.5 Latihan 1. Penjumlahan	8
2.6 Latihan 2. Pembagian	8
2.7 Tipe Data Teks	8
2.8 Latihan 3. Salam 2	8
2.9 Latihan 4. Kalimat	9
2.10 Tipe Data Boolean.....	9
2.11 Latihan 5. Gabungan Semua Tipe Data	9
BAB 3 Operator.....	11
3.1 Tujuan.....	11
3.2 Operator Pada Pemrograman Java	11
3.3 Operator Aritmatika	11
3.4 Latihan 1. Operator Aritmatika	11
3.5 Operator Increment dan Decrement	12

3.6	Latihan 2. Operator Increment dan Decrement.....	12
3.7	Operator Perbandingan	13
3.8	Latihan 3. Operator Perbandingan.....	13
3.9	Operator Logika.....	14
3.10	Latihan 4. Operator Logika.....	14
3.11	Latihan 5. Operator.....	15
BAB 4	Input dari Keyboard.....	16
4.1	Tujuan.....	16
4.2	Scanner	16
4.3	Latihan 1. Input dari Scanner	16
4.4	BufferedReader.....	17
4.5	Latihan 2. Input dari BufferedReader	17
4.6	JOptionPane	18
4.7	Latihan 3. Input dari JOptionPane.....	18
4.8	Latihan 4. Input Dari Keyboard	19
BAB 5	Percabangan.....	20
5.1	Tujuan.....	20
5.2	If.....	20
5.3	If-Else.....	20
5.4	Latihan 1. Kelulusan 1	20
5.5	If-Else-If.....	21
5.6	Latihan 2. Nilai Huruf	22
5.7	If Dalam If.....	22
5.8	Latihan 3. Kelulusan 2	23
5.9	Switch-Case.....	23
5.10	Latihan 4. Bobot Nilai Huruf.....	23
BAB 6	Perulangan.....	24
6.1	Tujuan.....	24
6.2	For.....	24
6.3	While-Do.....	24
6.4	Do-While.....	25
6.5	Perulangan Di Dalam Perulangan	25
6.6	Latihan 1. Deret Angka	25
6.7	Latihan 2. Deret Angka 2	25
6.8	Latihan 3. Kelulusan 3	26
6.9	Latihan 4. Deret Angka 3	26

BAB 7	Contoh-Contoh Kasus.....	28
7.1	Tujuan.....	28
7.2	Latihan 1. Gaji Pegawai.....	28
7.3	Latihan 2. Uang Kembalikan.....	28
7.4	Latihan 3. Tanda Silang.....	28
	Referensi.....	29

BAB 1

PENDAHULUAN

1.1 TUJUAN

Pada bab ini mahasiswa diharapkan mampu memahami bagaimana memulai membuat skrip program Java menggunakan beberapa aplikasi text editor ataupun IDE (Integrated Development Environment).

1.2 PROGRAM SALAM

Perhatikan skrip berikut ini.

Skrip 1-1 Salam.java

```
public class Salam {  
    public static void main(String[] args) {  
        System.out.println("Assalammu'alaikum");  
    }  
}
```

Program Skrip 1-1 adalah program untuk menampilkan kata “Assalammu’alaikum” pada output program. Sehingga ketika skrip tersebut dijalankan akan menghasilkan output sebagai berikut.

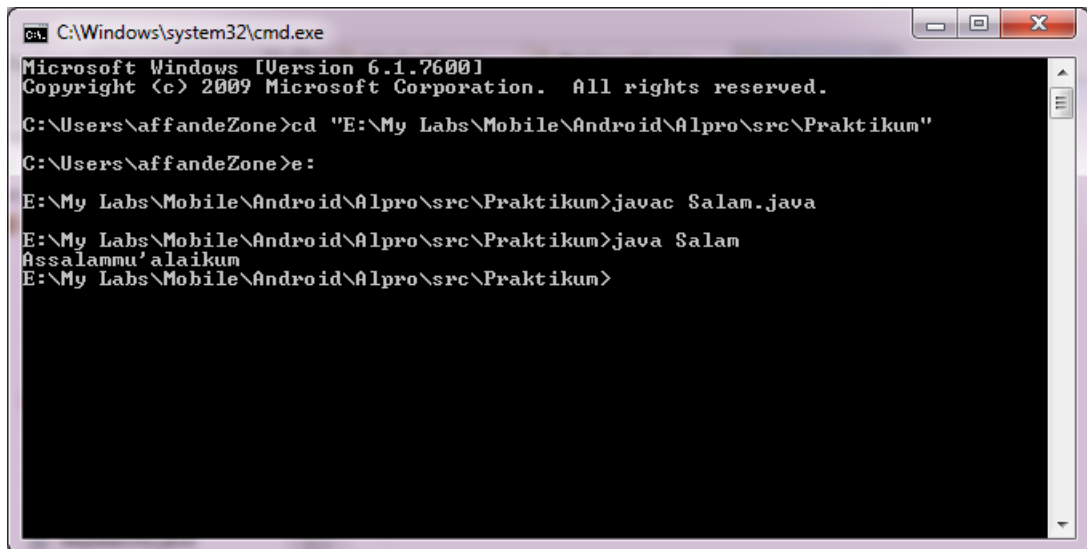
```
Assalammu'alaikum
```

Untuk menjalankan skrip tersebut, anda harus menyimpan skrip tersebut dalam bentuk file berekstensi java (*.java). Sedangkan nama file harus sama dengan nama class nya. Dengan begitu, skrip di atas harus disimpan dengan nama file **Salam.java**.

1.3 MEMULAI JAVA DENGAN NOTEPAD

Berikut ini adalah langkah-langkah memulai Java dengan Notepad, NotePad++ atau JCreator (Windows):

1. Ketiklah Skrip 1-1 pada aplikasi NotePad, NotePad++ atau JCreator, kemudian simpan dengan nama **Salam.java**.
2. Buka **cmd.exe** lalu akses folder dimana anda menyimpan file Salam.java.
3. Compile dengan perintah **javac Salam.java**.
4. Jalankan hasil compile tadi dengan perintah **java Salam**.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

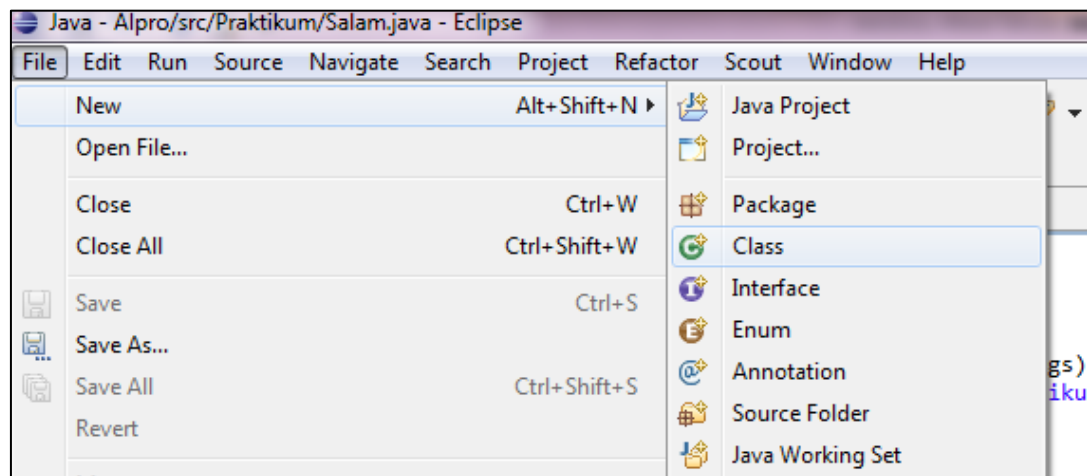
C:\Users\affandeZone>cd "E:\My Labs\Mobile\Android\Alpro\src\Praktikum"
C:\Users\affandeZone>:
E:\My Labs\Mobile\Android\Alpro\src\Praktikum>javac Salam.java
E:\My Labs\Mobile\Android\Alpro\src\Praktikum>java Salam
Assalamu'alaikum
E:\My Labs\Mobile\Android\Alpro\src\Praktikum>
```

Gambar 1-1 Menjalankan skrip dari cmd.exe

1.4 MEMULAI JAVA DENGAN ECLIPSE

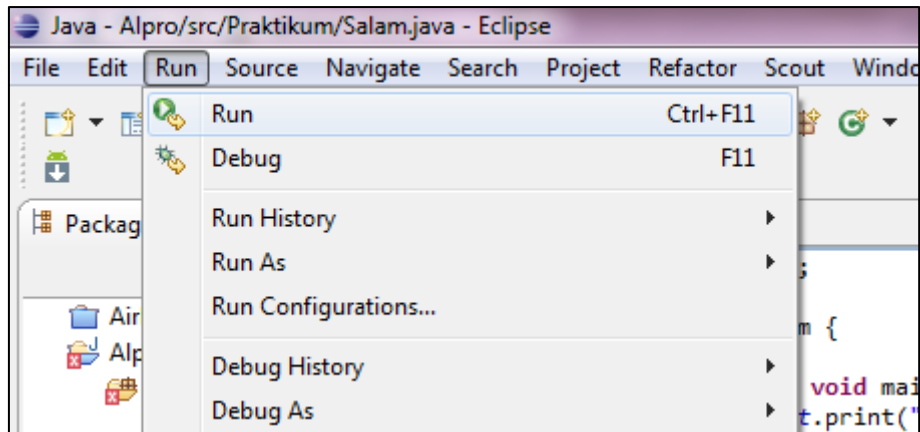
Berikut ini adalah langkah-langkah memulai Java dengan Eclipse:

1. Buatlah class baru pada menu File → New → Class dengan nama Salam.

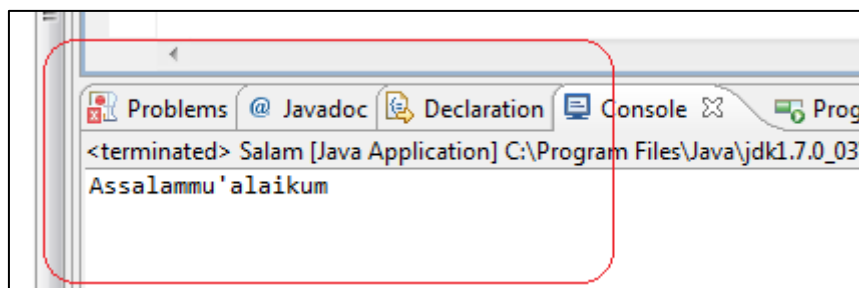


Gambar 1-2 Membuat class baru dari Eclipse

2. Ketiklah Skrip 1-1 pada class yang baru saja dibuat.
3. Jalankan skrip tersebut dengan klik menu Run → Run.



Gambar 1-3 Menjalankan skrip dari Eclipse

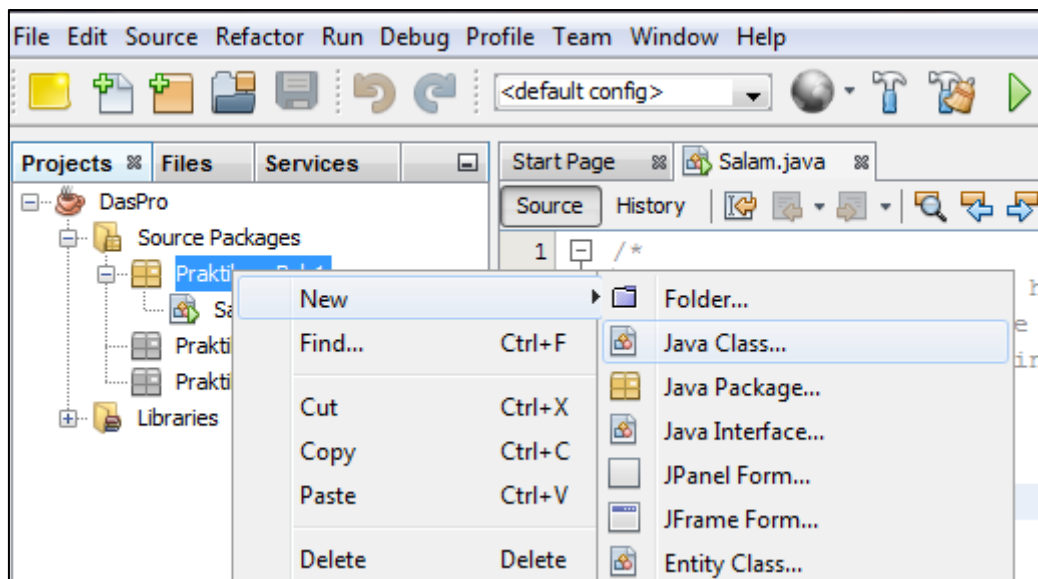


Gambar 1-4 Hasil output dari Eclipse

1.5 MEMULAI JAVA DENGAN NETBEANS

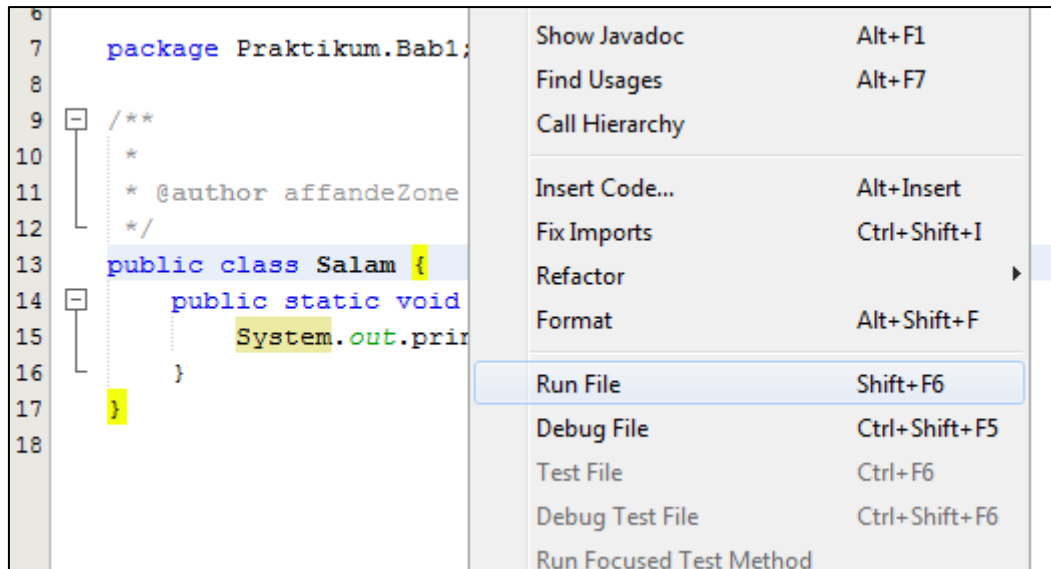
Berikut ini adalah langkah-langkah memulai java menggunakan NetBeans:

1. Buatlah class baru dengan klik kanan Package yang diinginkan, pilih New → Java Class lalu beri nama Salam, klik Finish.



Gambar 1-5. Membuat class baru pada NetBeans

2. Tulis Skrip 1-1 pada class yang baru saja dibuat.
3. Jalankan skrip tersebut dengan klik kanan, pilih Run File.



Gambar 1-6. Menjalankan skrip dari NetBeans

1.6 LATIHAN 1. BIODATA

Buatlah program Java yang menghasilkan output berupa biodata anda sendiri. Berikut contoh hasil output programnya.

```
Nama: Budi Santoso
NIM: 11351029999
TTL: Pekanbaru, 12 Juni 1993
Alamat: Jl. HR. Soebrantas km.30 No.252
Email: budisantoso@yahoo.com
Fb: /budisantoso
Tw: @budisantoso
```

1.7 LATIHAN 2. BIODATA 2

Buatlah program Java yang menghasilkan output berupa biodata anda sendiri. Berikut contoh hasil output programnya.

```
=====
| Nama      : Budi Santoso
| NIM       : 11351029999
| TTL       : Pekanbaru, 12 Juni 1993
| Alamat    : Jl. HR. Soebrantas km.30 No.252
| Email     : budisantoso@yahoo.com
| Fb        : /budisantoso
| Tw        : @budisantoso
|
|=====
```

1.8 MENULISKAN KOMENTAR

Menulis komentar pada pemrograman Java digunakan untuk memberikan catatan bagi programmer untuk suatu baris kode atau suatu kumpulan kode. Komentar juga bisa

digunakan untuk tidak menjalankan suatu baris kode tanpa menghapus baris kode tersebut. Ini karena apapun yang dijadikan komentar tidak akan diproses sedikitpun.

Ada dua cara menulis komentar pada Java, antara lain:

1. Komentar Satu Baris, untuk membuat komentar satu baris, tambahkan simbol // sebelum menulis komentar.

```
public class Salam {
    public static void main(String[] args) {
        // Ini adalah komentar
        System.out.println("Assalammu'alaikum");
    }
}
```

2. Komentar Lebih Dari Satu Baris, tambahkan simbol /* di awal komentar dan akhiri dengan simbol */ di akhir komentar.

```
public class Salam {
    public static void main(String[] args) {
        /* Ini adalah komentar yang
        Yang bisa dibuat lebih dari
        Satu bari */
        System.out.println("Assalammu'alaikum");
    }
}
```

1.9 LATIHAN 3. MENULIS KOMENTAR

Perhatikan skrip pada latihan 1 (Biodata) dan latihan 2 (Biodata 2) sebelumnya, buatlah komentar pada setiap baris kodenya dengan komentar yang menjelaskan tentang fungsi dan proses masing-masing baris kodenya.

BAB 2

VARIABEL DAN TIPE DATA

2.1 TUJUAN

Materi pada bab ini disajikan supaya mahasiswa mampu mengenal dan memahami tentang variabel, cara membuat variabel dan fungsi variabel pada bahasa pemrograman. Mahasiswa juga diharapkan mampu mengenal berbagai tipe data yang bisa disimpan pada suatu variabel dan bagaimana perbedaan fungsi dari masing-masing tipe data.

2.2 VARIABEL

Variabel merupakan tempat/wadah untuk menyimpan suatu nilai sesuai untuk diproses dan diolah oleh skrip program lainnya. Nilai pada variabel dapat disimpan sementara, dipanggil kembali, dihitung, dibandingkan dan dikonversi sesuai dengan karakteristik tipe data dari variabel tersebut.

Perhatikan skrip berikut ini.

Skrip 2-1 SalamNama.java

```
public class SalamNama {
    public static void main(String[] args) {
        String nama = "Affandes";
        System.out.println("Assalammu'alaikum " + nama);
    }
}
```

Pada Skrip 2-1 dapat dilihat bahwa `nama` merupakan sebuah variabel yang diberi label `nama`. Variabel ini menyimpan teks yaitu "Affandes". Karena variabel tersebut menyimpan teks, maka tipe data variabel tersebut adalah `String`.

Skrip di atas menjelaskan bahwa variabel nama dapat dipanggil kembali pada baris kode berikutnya, sehingga ketika program dijalankan, program akan menampilkan tulisan "Assalammu'alaikum Affandes".

2.3 TIPE DATA

Tipe data merupakan karakteristik sebuah variabel. Karakteristik ini akan menentukan fungsi variabel itu nantinya. Ada tipe data yang menyimpan data angka, ada juga tipe data yang menyimpan data teks. Perbedaan yang jelas antara tipe data angka dan teks adalah, tipe data angka dapat dihitung menggunakan operasi matematika, sedangkan tipe data teks tidak bisa dihitung.

Pada Skrip 2-1 di atas, String adalah tipe data teks, sehingga nilai yang disimpan di dalam variabel tersebut adalah teks, meskipun diisi dengan nilai “1000” atau “seribu” tetap dianggap sebagai teks yang tidak bisa dihitung.

Tipe data bukan hanya tentang menyimpan jenis data angka atau teks, namun juga tentang kapasitas penyimpanannya yang berbeda-beda. Sebagai contoh, tipe data angka Byte hanya bisa menyimpan angka dari -128 sampai 127, sedangkan tipe data Short dapat menyimpan angka -32768 sampai 32767 begitu seterusnya. Untuk tipe-tipe data lainnya dijelaskan pada sub bab berikutnya.

Ada 2 jenis tipe data pada pemrograman Java, yaitu:

1. Tipe data primitif, yaitu tipe data yang paling dasar pada Java. Tipe data ini terdiri dari 3, yaitu :
 - a. Tipe data angka, tipe data yang menyimpan data angka.
 - b. Tipe data karakter, tipe data yang menyimpan 1 karakter teks.
 - c. Tipe data boolean, tipe data yang hanya bernilai `True` atau `False`.
2. Tipe data turunan, yaitu tipe data yang dibuat dari satu atau lebih tipe data primitif. Pada umumnya tipe data ini berbentuk class.

Pada modul ini tidak membahas tentang tipe data turunan, kecuali tipe data String. Pembahasan tipe data String digabungkan dengan tipe data karakter.

2.4 TIPE DATA ANGKA

Tipe data ini dapat menyimpan nilai angka dengan kapasitas tertentu. Sehingga variabel yang menggunakan tipe data ini tidak bisa menyimpan teks ataupun karakter di dalamnya. Salah satu fungsi tipe data angka adalah untuk menyimpan data angka yang dapat dihitung dengan operasi matematika seperti penjumlahan, pengurangan, perkalian dan sebagainya.

Ada beberapa jenis tipe data angka, antara lain:

1. Bilangan Bulat (Integer), merupakan tipe data untuk menyimpan bilangan bulat. Tipe data ini terdiri dari beberapa jenis sesuai dengan kapasitas penyimpanannya.

Tipe	Keyword	Kapasitas	Rentang Nilai
Byte	byte	1 byte	-128 sampai 127
Short Integer	short	2 byte	-32768 s/d 32767
Integer	int	4 byte	-2147483648 s/d 2147483647
Long Integer	long	8 byte	-9223372036854775808 s/d 9223372036854775807

2. Bilangan Berkoma (Decimal), merupakan tipe data untuk menyimpan bilangan berkoma. Tingkat presisi dan kapasitas nilai yang dapat disimpan terdiri dari 2 tipe.

Tipe	Keyword	Kapasitas	Rentang Nilai
Float	float	4 byte	-3.4×10^{38} to 3.4×10^{38}
Double	double	8 byte	-1.8×10^{308} to 1.8×10^{308}

2.5 LATIHAN 1. PENJUMLAHAN

Perhatikan skrip berikut ini. Lengkapilah skrip berikut ini sehingga program berjalan dengan benar.

```
public class Penjumlahan {
    public static void main(String[] args) {
        int nilaiA = 1500;
        int nilaiB = 75;
        // Lengkapilah kode pada baris ini
        System.out.println("Nilai A = " + nilaiA);
        System.out.println("Nilai B = " + nilaiB);
        System.out.println("Jumlah = " + jumlah);
    }
}
```

2.6 LATIHAN 2. PEMBAGIAN

Perhatikan skrip berikut ini. Lengkapilah skrip berikut ini sehingga program berjalan dengan benar.

```
public class Pembagian {
    public static void main(String[] args) {
        int a = 20;
        int b = 8;
        // Lengkapilah kode pada baris ini
        System.out.println("A = " + a);
        System.out.println("B = " + b);
        System.out.println("A dibagi B = " + hasilBagi);
    }
}
```

2.7 TIPE DATA TEKS

Tipe data teks adalah tipe data yang menyimpan data dalam bentuk teks. Data teks tidak hanya terdiri dari huruf saja, angka yang disimpan ke variabel bertipe teks akan dianggap sebagai teks sehingga tidak bisa digunakan untuk menghitung.

Ada beberapa jenis tipe data teks, antara lain:

1. Char, tipe data yang hanya menyimpan 1 karakter teks saja.
2. String, tipe data yang menyimpan rangkaian teks paling banyak $2^{31}-1$ karakter atau sekitar 2GB teks.

2.8 LATIHAN 3. SALAM 2

Perhatikan skrip berikut. Perbaikilah skrip berikut ini sehingga menghasilkan output sesuai dengan output yang diinginkan.

```
public class Salam2 {
    public static void main(String[] args) {
        // Buat karakter
        char c1 = 'S';
    }
}
```

```

char c2 = 'A';
char c3 = 'L';
char c4 = 'A';
char c5 = 'M';
// Print
System.out.println(c1+c2+c3+c4+c5);
System.out.println(c5+c4+c3+c2+c1);
    }
}

```

Berikut adalah output yang diinginkan.

```

SALAM
MALAS

```

2.9 LATIHAN 4. KALIMAT

Perhatikan skrip berikut. Perbaikilah skrip berikut ini sehingga menghasilkan output sesuai dengan output yang diinginkan.

```

public class Kalimat {
    public static void main(String[] args) {
        // Buat variabel
        String subjek = "Saya";
        String prediket = "menulis";
        String objek = "program Java";
        String keterangan = "hari ini";
        // Print kalimat S+P+O+K
        System.out.println(subjek+prediket+objek+keterangan);
    }
}

```

Berikut adalah output yang diinginkan.

```

Saya menulis program Java hari ini

```

2.10 TIPE DATA BOOLEAN

Tipe data boolean merupakan tipe data yang hanya menyimpan nilai `True` atau `False`. Tipe data ini digunakan untuk keperluan percabangan. Percabangan lebih lanjut akan dibahas pada bab 5.

2.11 LATIHAN 5. GABUNGAN SEMUA TIPE DATA

Tulislah program berikut ini. Kemudian, berikan komentar untuk setiap baris kode program yang ada. Komentar yang dibuat berisi penjelasan singkat masing-masing baris kode.

```

public class TrueFalse {
    public static void main(String[] args) {
        String nama = "Budi";
        int nilai = 54;
    }
}

```

```
boolean lulus = false;
System.out.println("Nama : "+nama);
System.out.println("Nilai : "+nilai);
System.out.println("Status : "+lulus);
}
}
```

BAB 3 OPERATOR

3.1 TUJUAN

Pada bab ini mahasiswa diharapkan mampu mengenal dan memahami jenis-jenis operator pada pemrograman Java beserta cara penggunaannya.

3.2 OPERATOR PADA PEMROGRAMAN JAVA

Pada pemrograman Java terdapat beberapa jenis operator, antara lain: operator aritmatika, operator perbandingan, operator logika dan sebagainya.

3.3 OPERATOR ARITMATIKA

Berikut ini adalah operator aritmatika yang bisa digunakan pada Java.

Operator	Contoh	Keterangan
+	n1 + n2	Penjumlahan
-	n1 - n2	Pengurangan
*	n1 * n2	Perkalian
/	n1 / n2	Pembagian
%	n1 % n2	Sisa Pembagian

3.4 LATIHAN 1. OPERATOR ARITMATIKA

Buatlah skrip berikut ini. Analisa dan perhatikan baris kode apa saja yang tidak berjalan sesuai dengan hasil yang diinginkan atau jelaskan mengapa hasil hitungannya berbeda. Kemudian perbaiki menjadi kode yang benar.

```
public class Aritmatika {
    public static void main(String[] args) {
        // Buat variabel
        int a = 12;
        int b = 8;
        int c = 5;
        // Hitung
        int hasil1 = a + b - c;
        int hasil2 = a * b / c;
        int hasil3 = a + b * c;
        int hasil4 = a + b / c;
        int hasil5 = (a + b) * c;
        int hasil6 = (a - b) * c;
        // Print
        System.out.println(hasil1);
        System.out.println(hasil2);
        System.out.println(hasil3);
        System.out.println(hasil4);
        System.out.println(hasil5);
    }
}
```

```

        System.out.println(hasil6);
    }
}

```

3.5 OPERATOR INCREMENT DAN DECREMENT

Berikut ini adalah operator increment dan decrement pada Java.

Operator	Contoh	Keterangan
++	++n1	n1 ditambahkan 1 sebelum dievaluasi.
++	n1++	n1 ditambahkan 1 setelah dievaluasi.
--	--n1	n1 dikurangi 1 sebelum dievaluasi.
--	n1--	n1 dikurangi 1 setelah dievaluasi.

3.6 LATIHAN 2. OPERATOR INCREMENT DAN DECREMENT

Buatlah skrip berikut ini. Analisa hasil output dari program tersebut. Kemudian jelaskan mengapa terjadi perbedaan hasil outputnya?

```

public class IncreDecre {
    public static void main(String[] args) {
        // Buat variabel awal
        int a = 0;
        int b = 0;
        int c = 9;
        int d = 9;
        // Print ke 1
        System.out.println("Print ke 1");
        System.out.println("A = " + a++);
        System.out.println("B = " + ++b);
        System.out.println("C = " + c--);
        System.out.println("D = " + --d);
        // Print ke 2
        System.out.println("Print ke 2");
        System.out.println("A = " + a++);
        System.out.println("B = " + ++b);
        System.out.println("C = " + c--);
        System.out.println("D = " + --d);
        // Print ke 3
        System.out.println("Print ke 3");
        System.out.println("A = " + a++);
        System.out.println("B = " + ++b);
        System.out.println("C = " + c--);
        System.out.println("D = " + --d);
        // Print ke 4
        System.out.println("Print ke 4");
        System.out.println("A = " + a++);
        System.out.println("B = " + ++b);
        System.out.println("C = " + c--);
        System.out.println("D = " + --d);
    }
}

```

3.7 OPERATOR PERBANDINGAN

Berikut ini adalah operator perbandingan yang bisa digunakan pada Java.

Operator	Contoh	Keterangan
>	n1 > n2	n1 Besar dari n2
>=	n1 >= n2	n1 Besar sama dengan n2
<	n1 < n2	n1 Kecil dari n2
<=	n1 <= n2	n1 Kecil sama dengan n2
==	n1 == n2	n1 Sama dengan n2
!=	n1 != n2	n1 Tidak sama dengan n2

3.8 LATIHAN 3. OPERATOR PERBANDINGAN

Buatlah skrip berikut ini. Analisa dan perhatikan hasil outputnya. Jelaskan mengapa hasil outputnya seperti itu?

```
public class OperatorPerbandingan {
    public static void main(String[] args) {
        // Buat variabel
        int a = 10;
        int b = 8;
        int c = 12;
        int d = 5;
        // Bandingkan
        boolean tes1 = a > b;
        boolean tes2 = b < c;
        boolean tes3 = (a * 2) >= (b + c);
        boolean tes4 = (a / d) <= (c - a);
        boolean tes5 = (a - d) == d;
        boolean tes6 = (a - b) != (b - d);
        boolean tes7 = a > 12;
        boolean tes8 = b <= 7;
        boolean tes9 = c == 10;
        boolean tes10 = d != 5;
        // Print
        System.out.println("Tes ke 1 = " + tes1);
        System.out.println("Tes ke 2 = " + tes2);
        System.out.println("Tes ke 3 = " + tes3);
        System.out.println("Tes ke 4 = " + tes4);
        System.out.println("Tes ke 5 = " + tes5);
        System.out.println("Tes ke 6 = " + tes6);
        System.out.println("Tes ke 7 = " + tes7);
        System.out.println("Tes ke 8 = " + tes8);
        System.out.println("Tes ke 9 = " + tes9);
        System.out.println("Tes ke 10 = " + tes10);
    }
}
```


3.9 OPERATOR LOGIKA

Berikut ini adalah operator logika pada Java.

Operator	Contoh	Keterangan
AND	B1 && b2 B1 & b2	Bernilai True jika b1 dan b2 bernilai True.
OR	B1 b2 B1 b2	Bernilai False jika b1 dan b2 bernilai False.
XOR	B1 ^ b2	Bernilai True jika b1 ≠ b2
NOT	!b1	Kebalikan dari b1

3.10 LATIHAN 4. OPERATOR LOGIKA

Buatlah skrip berikut ini. Analisa dan perhatikan hasil outputnya. Jelaskan mengapa hasil outputnya seperti itu?

```
public class OperatorLogika {
    public static void main(String[] args) {
        // Buat variabel
        boolean b1 = 5 < 10;
        boolean b2 = 9 > 11;
        boolean b3 = !b1;
        boolean b4 = !b2;
        // Tes Logika AND
        boolean TesAND1 = b1 & b2;
        boolean TesAND2 = b1 && b2;
        boolean TesAND3 = b1 && b3;
        boolean TesAND4 = b1 && b4;
        // Tes Logika OR
        boolean TesOR1 = b1 | b2;
        boolean TesOR2 = b1 || b2;
        boolean TesOR3 = b1 || b3;
        boolean TesOR4 = b1 || b4;
        // Tes Logika XOR
        boolean TesXOR1 = b1 ^ b2;
        boolean TesXOR2 = b1 ^ b3;
        boolean TesXOR3 = b1 ^ b4;
        // Print
        System.out.println("Tes AND 1 = " + TesAND1);
        System.out.println("Tes AND 2 = " + TesAND2);
        System.out.println("Tes AND 3 = " + TesAND3);
        System.out.println("Tes AND 4 = " + TesAND4);
        System.out.println("Tes OR 1 = " + TesOR1);
        System.out.println("Tes OR 2 = " + TesOR2);
        System.out.println("Tes OR 3 = " + TesOR3);
        System.out.println("Tes OR 4 = " + TesOR4);
        System.out.println("Tes XOR 1 = " + TesXOR1);
        System.out.println("Tes XOR 2 = " + TesXOR2);
        System.out.println("Tes XOR 3 = " + TesXOR3);
    }
}
```

3.11 LATIHAN 5. OPERATOR

Buatlah skrip program yang menyelesaikan masalah berikut ini.

1. Jika ditentukan $n1 = 10$ dan $n2 = 15$. Buatlah skrip program untuk mengevaluasi dan menampilkan hasil evaluasi berikut ini:
 - a. $h1 =$ penjumlahan $n1$ dan $n2$, lalu dikalikan dengan penjumlahan $n2$ dan $n1$.
 - b. $h2 =$ sisa bagi dari $n1$ dengan 4, kemudian dikalikan dengan $n2$.
2. Perhatikan soal no.1, jika anda sudah mendapatkan nilai $h1$ dan $h2$, maka buat skrip program yang mengevaluasi dan menampilkan hasil evaluasi berikut ini:
 - a. $b1 = (h1 \text{ besar sama dengan } h2)$
 - b. $b2 = (h2 \text{ besar sama dengan } h1)$
 - c. $b3 = (h1 \text{ Mod } 4) == (++h2 \text{ Mod } 5)$
 - d. $b4 = (b1 \text{ XOR } b3) \text{ AND } (b2 \text{ OR } b3)$
 - e. $b5 = b2 \text{ OR } (b3 \text{ AND } (b2 \text{ XOR } b1))$

BAB 4 INPUT DARI KEYBOARD

4.1 TUJUAN

Pada bab ini mahasiswa diharapkan mampu mengetahui dan mencoba mengambil input data pada keyboard sebagai input pada skrip program yang dibuat.

4.2 SCANNER

Class Scanner merupakan class yang dapat mengambil data yang diketikkan pada keyboard untuk disimpan ke dalam variabel pada program Java. Untuk menggunakan Scanner ini, anda harus mengimpor library `java.util.Scanner` agar class Scanner bisa digunakan. Berikut kode untuk mengimpor library Scanner.

```
import java.util.Scanner;
```

Kode tersebut ditulis pada baris awal sebelum nama class Java. Setelah anda mengimpor library Scanner, buatlah objek dari Scanner dengan kode berikut.

```
Scanner s1 = new Scanner(System.in);
```

Pada kode di atas, `s1` merupakan nama objek/variabel dari class Scanner yang baru saja dibuat. Anda bisa menggunakan nama apa saja untuk Scanner. Kemudian objek/variabel tadi digunakan untuk mengambil inputan dari keyboard dengan kode berikut.

```
int nilai1 = s1.nextInt();
```

Kode `s1.nextInt()` akan menghasilkan angka yang ditulis pada keyboard. Sedangkan `int nilai1` digunakan sebagai variabel untuk menyimpan data angka yang dihasilkan `s1.nextInt()`.

4.3 LATIHAN 1. INPUT DARI SCANNER

Buatlah skrip di bawah ini. Apa yang terjadi jika anda memasukkan angka atau teks?

```
import java.util.Scanner;

public class InputScanner {
    public static void main(String[] args) {
        // Buat scanner
        Scanner s1 = new Scanner(System.in);
        // Ambil nilai dari keyboard
        System.out.print("Masukkan nilai1 = ");
        int nilai1 = s1.nextInt();

        System.out.print("Masukkan nilai2 = ");
        int nilai2 = s1.nextInt();
    }
}
```

```
        // Hitung dan Print
        int jumlah = nilai1 + nilai2;
        System.out.println("Jumlah = " + jumlah);
    }
}
```

4.4 BUFFEREDREADER

Class `BufferedReader` merupakan class yang dapat membaca data dari keyboard untuk disimpan pada sebuah variabel. Untuk menggunakan `BufferedReader`, tulis kode berikut pada baris paling awal program anda.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

Selanjutnya buatlah objek/variabel dari `BufferedReader` dengan kode berikut.

```
BufferedReader br1 = new BufferedReader(new
InputStreamReader(System.in));
```

Setelah itu anda bisa mengambil dan menyimpan input dari keyboard ke variabel yang diinginkan dengan kode berikut.

```
try {
    String nama = br1.readLine();
} catch (IOException ex) {
    System.out.println("Error pada saat input data");
}
```

Kode untuk mengambil input keyboard harus ditulis pada blok `try-catch` untuk menghindari terjadi error.

4.5 LATIHAN 2. INPUT DARI BUFFEREDREADER

Buatlah skrip di bawah ini. Apa yang terjadi jika anda memasukkan angka atau teks?

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class InputBR {

    public static void main(String[] args) {
        // Buat BufferedReader
        BufferedReader br1 = new BufferedReader(new
InputStreamReader(System.in));

        try {
            // Ambil data nama dari keyboard
            System.out.print("Nama = ");
            String nama = br1.readLine();
```

```

        // Ambil data umur dari keyboard
        System.out.print("Umur = ");
        int umur = Integer.parseInt(br1.readLine());
        // Print
        System.out.println("Assalamu'alaikum " + nama + ".
        Apa kabar? Tidak terasa sekarang anda sudah " + umur + " tahun.");

        } catch (IOException ex) {
            System.out.println("Error pada saat input data");
        }
    }
}

```

4.6 JOPTIONPANE

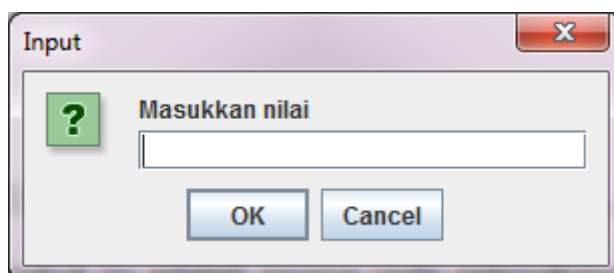
JOptionPane merupakan alternatif menerima input keyboard berbasis dialog. Untuk menggunakan class ini, impor class JOptionPane dengan kode berikut.

```
import javax.swing.JOptionPane;
```

Setelah itu gunakan kode berikut untuk menyimpan input dari keyboard ke variabel.

```
String nilai = JOptionPane.showInputDialog("Masukkan nilai");
```

Ketika program dijalankan, maka akan tampil dialog untuk meminta input dari keyboard.



4.7 LATIHAN 3. INPUT DARI JOPTIONPANE

Buatlah skrip di bawah ini. Apa yang terjadi jika anda memasukkan angka atau teks?

```

import javax.swing.JOptionPane;

public class InputJOP {
    public static void main(String[] args) {
        // Input Nama
        String nama = JOptionPane.showInputDialog("Masukkan nama
        anda!");
        int umur =
        Integer.parseInt(JOptionPane.showInputDialog("Masukkan umur
        anda!"));
        // Print
        System.out.println("Hai " + nama + ", apa kabar? Tidak
        terasa sekarang anda sudah " + umur + " tahun.");
    }
}

```

4.8 LATIHAN 4. INPUT DARI KEYBOARD

Buatlah skrip program pada latihan terdahulu yang menggunakan input dari keyboard, baik menggunakan Scanner, BufferedReader maupun JOptionPane. Berikut skrip program terdahulu yang harus anda ubah input nilainya dari keyboard.

1. Bab 1 Latihan 2. Biodata 2
2. Bab 2 Latihan 1. Penjumlahan
3. Bab 3 Latihan 5. Operator

BAB 5

PERCABANGAN

5.1 TUJUAN

Pada bab ini mahasiswa diharapkan mampu memahami struktur percabangan if-else dan swit-case dan cara menggunakannya.

5.2 IF

If digunakan untuk membuat suatu kode yang hanya berjalan pada kondisi tertentu. Misalnya suatu nilai akan dikalkulasikan jika nilai tersebut lebih besar dari 50. Berikut contoh kodenya

```
if(nilai >= 50) {  
    nilai = nilai + poin;  
}
```

Kode di atas adalah contoh penggunaan kode `if` untuk memproses nilai jika `nilai` tersebut bernilai lebih besar atau sama dengan 50. Ini ditunjukkan dengan ekspresi logika yang dibuat sebagai syarat `if`.

5.3 IF-ELSE

Kode if-else digunakan untuk membuat percabangan kode yang jalan sesuai dengan kondisi yang diinginkan. Percabangan pada if-else hanya untuk 2 percabangan. Sebagai contoh apabila nilai mahasiswa lebih besar dari 55 maka mahasiswa tersebut mendapat prediket lulus untuk mata kuliah yang dimaksud, jika tidak maka tidak lulus. Berikut ini contoh kodenya.

```
if(nilai > 55) {  
    System.out.println("Anda lulus ");  
} else {  
    System.out.println("Anda tidak lulus ");  
}
```

Kode di atas jika dijalankan akan memproses salah satu baris kode yang dibuat, sesuai dengan berapa nilai yang diinputkan.

5.4 LATIHAN 1. KELULUSAN 1

Lengkapi skrip di bawah ini sehingga menghasilkan output yang diinginkan.

```
import java.util.Scanner;
```

```

/**
 *
 * @author affandeZone
 */
public class Kelulusan1 {
    public static void main(String[] args) {
        // Buat Scanner
        Scanner inp = new Scanner(System.in);

        // Ambil Nama
        System.out.print("Masukkan nama anda = ");
        String Nama = inp.next();

        // Ambil NIM
        System.out.print("Masukkan NIM = ");
        String nim = inp.next();

        // Ambil Nilai
        System.out.print("Masukkan Nilai = ");
        int nilai = inp.nextInt();

        // Print
        ...
        ...
        ...
    }
}

```

Berikut adalah output yang diinginkan.

```

Masukkan nama anda = Budi Santoso
Masukkan NIM = 2012141
Masukkan Nilai = 78
Selamat anda lulus.

```

Berikut alternatif output yang diinginkan

```

Masukkan nama anda = Budi Santoso
Masukkan NIM = 2012141
Masukkan Nilai = 55
Anda tidak lulus.

```

5.5 IF-ELSE-IF

Kode if-else-if digunakan untuk membuat percabangan lebih dari 2 percabangan. Misalkan prediket untuk IPK dibawah 3,00 adalah “Cukup”, IPK antara 3,00 sampai 3,49 adalah “Memuaskan”, IPK antara 3,50 sampai 3,74 adalah “Sangat Memuaskan” dan IPK 3,75 ke atas adalah “Cum Laude”. Berikut contoh kode untuk kasus tersebut.

```

if(nilai < 3.00) {
    System.out.println("Cukup");
} else if(nilai < 3.49) {
    System.out.println("Memuaskan");
} else if(nilai < 3.74) {

```



```

        System.out.println("Sangat Memuaskan ");
    } else {
        System.out.println("Cum Laude ");
    }
}

```

Salah satu baris kode `System.out.println` di atas hanya akan berjalan salah satu di antara beberapa baris kode sesuai dengan kondisi yang terpenuhi. Kondisi tersebut ditentukan dari ekspresi logika yang dievaluasi pada kode `if`.

5.6 LATIHAN 2. NILAI HURUF

Buatlah skrip program yang menerima input `nilai` berupa data integer dan menghasilkan output berupa nilai huruf dengan ketentuan sebagai berikut.

Rentang nilai input	Output
0 sampai 55	E
56 sampai 65	D
66 sampai 75	C
76 sampai 85	B
86 sampai 100	A
Kecil dari 0 atau besar dari 100	Error

5.7 IF DALAM IF

Kode `if` dapat digunakan di dalam blok `if` lainnya, ini disebut `nested if`, yaitu `if` bersarang atau `if` di dalam `if`. `if` dalam `if` ini digunakan untuk kasus-kasus yang dibutuhkan menggunakan model `if` ini. Misalkan menentukan percabangan berdasarkan 2 atau lebih nilai pembandingan.

Sebagai contoh kriteria lulus seleksi untuk siswa jurusan IPA adalah 8,5 dan IPS 7,5. Apabila tidak memenuhi kriteria tersebut maka tidak lulus. Berikut contoh kode dalam Java.

```

if(jurusan == "IPA") {
    if(nilai >= 8.5) {
        System.out.println("Anda lulus ");
    } else {
        System.out.println("Anda tidak lulus ");
    }
} else if(jurusan == "IPS") {
    if(nilai >= 7.5) {
        System.out.println("Anda lulus ");
    } else {
        System.out.println("Anda tidak lulus ");
    }
}
}

```

5.8 LATIHAN 3. KELULUSAN 2

Buatlah skrip program yang menerima input `nama` berupa String, `jenis_kelamin` berupa String dan `tinggi_badan` berupa integer. Skrip akan menghasilkan output “Selamat Anda Lulus” atau “Anda gagal” dengan ketentuan sebagai berikut.

Jenis_Kelamin	Tinggi_Badan	Output
Laki-Laki	0 sampai 170	Anda Gagal
Laki-Laki	171 sampai 230	Selamat Anda Lulus
Perempuan	0 sampai 160	Anda Gagal
Perempuan	161 sampai 210	Selamat Anda Lulus

5.9 SWITCH-CASE

Switch-case digunakan untuk membuat kode percabangan lebih dari 2. Kode ini merupakan kode alternatif selain if-else-if. Misalkan kode 1 adalah IPA, kode 2 adalah IPS dan kode 3 adalah BHS. Berikut kode dalam Java.

```
switch(kode) {
    case 1:
        System.out.println("IPA"); break;
    case 2:
        System.out.println("IPS"); break;
    case 3:
        System.out.println("BHS"); break;
}
```

5.10 LATIHAN 4. BOBOT NILAI HURUF

Buatlah skrip yang mengkonversi nilai huruf (A, B, C, D, E) menjadi bobot angka (4, 3, 2, 1, 0). Inputan berupa teks, dan output berupa angka.

BAB 6

PERULANGAN

6.1 TUJUAN

Pada bab ini mahasiswa diharapkan mampu mengenal dan memahami fungsi dari kode perulangan for dan while.

6.2 FOR

For adalah kode yang digunakan untuk menjalankan serangkaian kode secara berulang-ulang. Pada kode for ini terdapat beberapa komponen yang dicantumkan, antara lain: (1) inisialisasi, (2) kondisi, (3) perubahan nilai, (4) statement yang diulang. Berikut ini format sintaks untuk kode for.

```
for(inisialisasi; kondisi; perubahan_nilai) {  
    statement  
}
```

Berikut ini adalah contoh skrip untuk mencetak tulisan “Hello” sebanyak 10 kali.

```
for(int a = 0; a < 10; a++) {  
    System.out.println("Hello");  
}
```

6.3 WHILE-DO

Kode while-do merupakan kode alternatif untuk melakukan perulangan selain for. Cara kerjanya sama, namun sintaks (aturan penulisan) yang berbeda. Berikut sintaks while-do.

```
inisialisasi;  
while(kondisi) {  
    statement  
    perubahan_nilai  
}
```

Berikut ini adalah contoh skrip untuk mencetak tulisan “Hello” sebanyak 10 kali.

```
int a = 0;  
while(a < 10) {  
    System.out.println("Hello");  
    a++;  
}
```

6.4 DO-WHILE

Kode do-while merupakan kode while-do dengan sintaks yang berbeda. Cara kerja do-while relatif sama dengan while-do. Berikut sintaks untuk do-while.

```
inisialisasi;
do {
    statement
    perubahan_nilai
} while(kondisi);
```

Berikut ini adalah contoh skrip untuk mencetak tulisan “Hello” sebanyak 10 kali.

```
int a = 0;
do {
    System.out.println("Hello");
    a++;
} while(a < 10);
```

6.5 PERULANGAN DI DALAM PERULANGAN

Kode perulangan dapat dibuat di dalam perulangan lainnya, baik for, while-do maupun do-while. Sama halnya dengan kode if, kode for dan kode perulangan lainnya dapat dijalankan pada kode if atau for lainnya dengan baik.

Berikut contoh kode for di dalam for.

```
for(int a = 0; a < 5; a++) {
    for(int b = 0; b < 5; b++) {
        System.out.print(a);
    }
}
```

Kode di atas akan menghasilkan output berikut ini.

```
0000011111222223333344444
```

6.6 LATIHAN 1. DERET ANGKA

Buatlah skrip menggunakan Perulangan untuk mencetak deret angka seperti berikut ini.

1. Output : 10 9 8 7 6 5 4 3 2 1
2. Output : 1 3 5 7 9 11 13 15 17 19
3. Output : 2 4 6 8 10 12 14 16 18 19
4. Output : 11 13 16 20 25 31 38 46 55 65

6.7 LATIHAN 2. DERET ANGKA 2

Buatlah skrip menggunakan Perulangan dan Percabangan untuk mencetak deret angka seperti berikut:

1. Output : 1 -2 3 -4 5 -6 7 -8 9 -10

2. Output : 11 2 13 4 15 6 17 8 19 10

3. Output : 1 10 2 9 3 8 4 7 5 6

6.8 LATIHAN 3. KELULUSAN 3

Buatlah program yang sama seperti pada Bab 5 Latihan 3 (Kelulusan 2) yang dapat mampu digunakan untuk 5 orang sekaligus. Gunakan perulangan untuk mengulang skrip yang ada pada skrip sebelumnya.

6.9 LATIHAN 4. DERET ANGKA 3

Buatlah skrip untuk mencetak deret angka seperti berikut:

1. Deret Segitiga 1

```
1
22
333
4444
55555
```

2. Deret Segitiga 2

```
1
12
123
1234
12345
```

3. Deret Segitiga 3

```
12345
1234
123
12
1
```

4. Deret Segiempat 1

```
123456
234567
345678
456789
```

5. Deret Segiempat 2 (Perkalian)

```
1    2    3    4    5
2    4    6    8    10
3    6    9    12   15
4    8    12   16   20
5    10   15   20   25
```

6. Deret Segiempat 3 (Penjumlahan)

2	3	4	5	6					
3	4	5	6	7					
4	5	6	7	8					
5	6	7	8	9					
6	7	8	9	10					

BAB 7 CONTOH-CONTOH KASUS

7.1 TUJUAN

Pada bab ini diharapkan mahasiswa mampu memahami berbagai kasus yang melibatkan beberapa materi-materi sebelumnya dan membuat skrip program yang dapat menyelesaikan kasus-kasus yang disediakan.

7.2 LATIHAN 1. GAJI PEGAWAI

Buatlah skrip program yang mampu menghitung gaji 5 orang pegawai dengan ketentuan sebagai berikut:

1. Inputan program terdiri dari nama_pegawai, jumlah_kehadiran sebanyak 5 kali.
2. Gaji dihitung berdasarkan jumlah_kehadiran, yaitu 50.000,- per 1 hari hadir.
3. Apabila jumlah_kehadiran < 20 hari, maka gaji dipotong sebesar 10% dari gaji yang diperoleh.

7.3 LATIHAN 2. UANG KEMBALIAN

Buatlah skrip program yang mampu menghitung berapa lembar pecahan uang kembalian pada sebuah transaksi di toko swalayan. Berikut adalah ketentuannya.

1. Inputan program berupa angka uang kembalian, misalnya 37.500.
2. Sistem menghitung dan menampilkan lembaran pecahan uang kembalian, misalnya :

```
1 lembar 20.000
1 lembar 10.000
1 lembar 5.000
1 lembar 2.000
1 lembar 500
```

3. Pecahan yang tersedia antara lain: 100.000, 50.000, 20.000, 10.000, 5.000, 1.000, 500, 200, 100.

7.4 LATIHAN 3. TANDA SILANG

Buatlah skrip program yang mampu mencetak output seperti berikut ini.

```
@===@
=@=@=
==@==
=@=@=
@===@
```

REFERENSI

1. Barry, Paul dan Griffiths, David. *Head First Programming*. United States of America. O'Reilly Media. 2009.
2. Burd, Barry. *Beginning Programming with Java™ For Dummies®*, ed 2. Indiana. Wiley Publishing. 2005.
3. Cadenhead, Rogers dan Lemay, Laura . *Sams Teach Yourself Java™ in 21 Days*, ed 5. United States of America. Sams Publishing. 2007.
4. Cadenhead, Rogers. *Sams Teach Yourself Java™ in 24 Hours*, ed 6. United States of America. Sams Publishing. 2012.
5. Darwin, Ian F. *Java™ Cookbook™*, ed 2. United States of America. O'Reilly Media. 2004.
6. Ergül, Özgür. *Guide to Programming and Algorithms Using R*. London. Springer-Verlag. 2013.
7. Niemeyer, Patrick dan Leuck, Daniel. *Learning Java*, ed 4. United States of America. O'Reilly Media. 2013.
8. Wang, Wallace. *Beginning Programming For Dummies®*, ed 3. Indiana. Wiley Publishing. 2004.