

Tugas 3

8 Maret 2024

**LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Muhammad Dirga Juang Ikhsan
Stambuk : 13020220032
Kelas : A1

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2024**

1. BacaString.Java

Berikut adalah penjelasan singkat mengenai beberapa bagian utama dari kode tersebut:

1. **import java.io.BufferedReader;**: Mengimpor kelas `BufferedReader` dari paket `java.io`, yang digunakan untuk membaca input karakter dari sebuah aliran input.
2. **import java.io.IOException;**: Mengimpor kelas `IOException` dari paket `java.io`, yang diperlukan karena penggunaan operasi input/output (I/O) dapat menimbulkan pengecualian (exceptions) yang perlu ditangani.
3. **import java.io.InputStreamReader;**: Mengimpor kelas `InputStreamReader` dari paket `java.io`, yang digunakan untuk membaca input karakter dari sebuah aliran input byte (`System.in` dalam hal ini).
4. **import javax.swing.*;**: Mengimpor semua kelas yang ada dalam paket `javax.swing`, meskipun dalam kode ini tidak digunakan.
5. **public class BacaString { ... }**: Mendefinisikan kelas Java bernama `BacaString`.
6. **public static void main(String[] args) throws IOException { ... }**: Mendefinisikan metode utama (main) dari program. Metode ini memungkinkan program untuk dijalankan secara mandiri. `IOException` ditambahkan ke deklarasi metode karena operasi pembacaan input (`readLine()`) dapat menghasilkan pengecualian `IOException` yang perlu ditangani.
7. **BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));**: Membuat objek `BufferedReader` yang akan membaca input dari aliran input standar (`System.in`). Ini dilakukan dengan mengalirkan aliran input byte (`System.in`) ke dalam aliran karakter (`InputStreamReader`) yang kemudian diarahkan ke `BufferedReader`.
8. **System.out.print ("\nBaca string dan Integer: \n");**: Mencetak teks ke konsol, memberikan petunjuk kepada pengguna tentang apa yang harus dimasukkan.
9. **str = datAIn.readLine();**: Membaca satu baris input dari pengguna dan menyimpannya dalam variabel `str` menggunakan metode `readLine()` dari objek `BufferedReader`.

10. **System.out.print ("String yang dibaca : "+ str);**: Mencetak kembali string yang dibaca dari pengguna ke konsol.

Program tersebut sederhana, hanya membaca sebuah string dari pengguna dan mencetaknya kembali ke konsol.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac BacaString.java
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java BacaString

Baca string dan Integer:
masukkan sebuah string: Rumah
String yang dibaca : Rumah
```

2. ForEver.java

Berikut adalah penjelasan singkat mengenai beberapa bagian utama dari kode tersebut:

1. **public class ForEver { ... }:** Mendefinisikan kelas Java bernama ForEver.
2. **public static void main(String[] args) { ... }:** Mendefinisikan metode utama (main) dari program. Metode ini memungkinkan program untuk dijalankan secara mandiri.
3. **System.out.println("Program akan looping, akhiri dengan ^c");:** Mencetak teks ke konsol, memberikan petunjuk kepada pengguna tentang cara menghentikan program.
4. **while (true) { ... }:** Ini adalah loop while yang akan berjalan selamanya atau sampai dihentikan secara paksa. Pada setiap iterasi, program akan mencetak teks "Print satu baris" ke konsol.
5. **{ System.out.print ("Print satu baris\n"); }**: Bagian dari loop while yang mencetak teks "Print satu baris" ke konsol. Penambahan karakter \n diakhir bertujuan untuk membuat baris baru setelah teks tercetak.

Program ini secara efektif membuat loop tanpa akhir yang akan mencetak "Print satu baris" terus-menerus ke konsol sampai program dihentikan oleh pengguna dengan menekan kombinasi tombol ^c.

Output :

[illegible]

3. If1.java

1. **import java.util.Scanner;;** Mendeklarasikan impor kelas Scanner dari paket java.util. Ini memungkinkan program untuk menggunakan Scanner untuk membaca input dari pengguna.
2. **public class If1 { ... };** Mendefinisikan kelas Java bernama If1.
3. **public static void main(String[] args) { ... };** Mendefinisikan metode utama (main) dari program. Metode ini memungkinkan program untuk dijalankan secara mandiri.
4. **Scanner masukan = new Scanner(System.in);;** Membuat objek Scanner yang digunakan untuk membaca input dari pengguna melalui aliran input standar (System.in).

5. **int a;;** Mendeklarasikan variabel a sebagai integer untuk menyimpan nilai yang akan dimasukkan oleh pengguna.
6. **System.out.print ("Contoh IF satu kasus \n");** Mencetak teks ke konsol untuk memberikan informasi kepada pengguna tentang apa yang harus dilakukan.
7. **System.out.print ("Ketikkan suatu nilai integer : ");** Mencetak teks ke konsol untuk meminta pengguna untuk memasukkan nilai integer.
8. **a = masukan.nextInt();** Membaca sebuah nilai integer dari pengguna dan menyimpannya dalam variabel a menggunakan metode nextInt() dari objek Scanner.
9. **if (a >= 0) { System.out.print ("\nNilai a positif " + a); }** Ini adalah struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna lebih besar dari atau sama dengan nol, maka program akan mencetak pesan "Nilai a positif" diikuti dengan nilai a ke konsol.

Program ini sederhana, hanya mengecek apakah nilai yang dimasukkan oleh pengguna positif atau tidak, dan mencetaknya ke konsol sesuai dengan kondisi yang diberikan.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac If1.java
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 100

Nilai a positif 100
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>|
```

4. If2.java

Berikut adalah penjelasan singkat mengenai beberapa bagian utama dari kode tersebut:

1. **import java.util.Scanner;;** Mendeklarasikan impor kelas Scanner dari paket java.util. Ini memungkinkan program untuk menggunakan Scanner untuk membaca input dari pengguna.

2. **public class If2 { ... };** Mendefinisikan kelas Java bernama If2.
3. **public static void main(String[] args) { ... };** Mendefinisikan metode utama (main) dari program. Metode ini memungkinkan program untuk dijalankan secara mandiri.
4. **int a;** Mendeklarasikan variabel a sebagai integer untuk menyimpan nilai yang akan dimasukkan oleh pengguna.
5. **Scanner masukan = new Scanner(System.in);** Membuat objek Scanner yang digunakan untuk membaca input dari pengguna melalui aliran input standar (System.in).
6. **System.out.print ("Contoh IF dua kasus \n");** Mencetak teks ke konsol untuk memberikan informasi kepada pengguna tentang apa yang harus dilakukan.
7. **System.out.print ("Ketikkan suatu nilai integer :");** Mencetak teks ke konsol untuk meminta pengguna untuk memasukkan nilai integer.
8. **a = masukan.nextInt();** Membaca sebuah nilai integer dari pengguna dan menyimpannya dalam variabel a menggunakan metode nextInt() dari objek Scanner.
9. **if (a >= 0) { System.out.println ("Nilai a positif " + a); }** Ini adalah struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna lebih besar dari atau sama dengan nol, maka program akan mencetak pesan "Nilai a positif" diikuti dengan nilai a ke konsol.
10. **else { System.out.println ("Nilai a negatif " + a); }** Ini adalah bagian else dari struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna kurang dari nol, maka program akan mencetak pesan "Nilai a negatif" diikuti dengan nilai a ke konsol.

Program ini sederhana, hanya mengecek apakah nilai yang dimasukkan oleh pengguna positif atau negatif, dan mencetaknya ke konsol sesuai dengan kondisi yang diberikan.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :100
Nilai a positif 100
```

5. If3.java

Mari kita jelaskan bagian-bagian utama dari kode tersebut dengan penjelasan yang berbeda:

1. **import java.util.Scanner;**: Ini adalah pernyataan untuk mengimpor kelas Scanner dari paket java.util. Dengan menggunakan Scanner, program dapat membaca input dari pengguna.
2. **public class If3 { ... };** Mendefinisikan sebuah kelas Java yang disebut If3.
3. **public static void main(String[] args) { ... };** Ini adalah metode utama (main) dari program. Metode ini adalah titik masuk utama ketika program dijalankan.
4. **int a;**: Mendeklarasikan sebuah variabel bernama a yang akan digunakan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
5. **Scanner masukan = new Scanner(System.in);**: Membuat sebuah objek Scanner yang digunakan untuk membaca input dari pengguna melalui aliran input standar (System.in).
6. **System.out.print ("Contoh IF tiga kasus \n");**: Mencetak pesan ke konsol untuk memberikan informasi kepada pengguna tentang apa yang harus dilakukan.
7. **System.out.print ("Ketikkan suatu nilai integer :");**: Mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan nilai integer.
8. **a = masukan.nextInt();**: Membaca sebuah nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a.
9. **if (a > 0) { System.out.println ("Nilai a positif "+ a); };** Ini adalah kondisi pertama dalam struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna lebih besar dari 0, maka program akan mencetak pesan "Nilai a positif" diikuti dengan nilai a.
10. **else if (a == 0) { System.out.println ("Nilai Nol "+ a); };** Kondisi kedua dalam struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna sama dengan 0, maka program akan mencetak pesan "Nilai Nol" diikuti dengan nilai a.
11. **else { System.out.println ("Nilai a negatif "+ a); };** Bagian else dari struktur percabangan IF. Jika nilai a yang dimasukkan oleh pengguna kurang dari 0, maka program akan mencetak pesan "Nilai a negatif" diikuti dengan nilai a.

Dengan demikian, program ini memungkinkan pengguna untuk memasukkan nilai integer dan akan memberikan respons sesuai dengan kondisi nilai yang dimasukkan: positif, nol, atau negatif.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java KasusBoolean
true
benar
```

6. KasusBoolean.java

Mari kita jelaskan bagian-bagian utama dari kode tersebut:

1. **public class KasusBoolean { ... }:** Mendefinisikan sebuah kelas Java bernama KasusBoolean.
2. **public static void main(String[] args) { ... }:** Mendefinisikan metode utama (main) dari program. Metode ini adalah titik masuk utama ketika program dijalankan.
3. **boolean bool;:** Mendeklarasikan sebuah variabel bernama bool dengan tipe data boolean. Variabel ini akan digunakan untuk menyimpan nilai boolean (true atau false).
4. **bool = true;:** Memberikan nilai true kepada variabel bool.
5. **if(bool) { System.out.print("true\n"); } else System.out.print("false\n");:** Ini adalah sebuah struktur percabangan IF. Jika nilai bool adalah true, maka program akan mencetak "true" ke konsol. Jika tidak, program akan mencetak "false".
6. **if(!bool) { System.out.print("salah\n"); } else System.out.print("benar\n");:** Ini juga merupakan sebuah struktur percabangan IF. Namun, dalam kondisi ini, kita menggunakan operator negasi (!) untuk mengecek apakah bool adalah false. Jika bool adalah false, maka program akan mencetak "salah". Jika tidak, program akan mencetak "benar".

Dengan demikian, program ini menunjukkan cara menggunakan tipe data boolean dalam ekspresi kondisional. Itu menunjukkan bahwa kita dapat menggunakan kondisi boolean langsung dalam percabangan IF atau menggunakan operator negasi (!) untuk mengecek kondisi yang berlawanan.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac KasusSwitch.java
KasusSwitch.java:15: error: unclosed string literal
case 'a': { System.out.print (" Yang anda ketik adalah a
^
KasusSwitch.java:16: error: illegal character: '\'
\n"); break;}
^
KasusSwitch.java:16: error: unclosed string literal
\n"); break;}
^
KasusSwitch.java:16: error: not a statement
\n"); break;}
^
KasusSwitch.java:17: error: orphaned case
case 'u': { System.out.print (" Yang anda ketik adalah u
^
KasusSwitch.java:17: error: unclosed string literal
case 'u': { System.out.print (" Yang anda ketik adalah u
^
KasusSwitch.java:18: error: illegal character: '\'
\n"); break;}
^
KasusSwitch.java:18: error: unclosed string literal
\n"); break;}
^
KasusSwitch.java:18: error: not a statement
\n"); break;}
^
KasusSwitch.java:19: error: orphaned case
case 'e': { System.out.print (" Yang anda ketik adalah e
^
KasusSwitch.java:19: error: unclosed string literal
case 'e': { System.out.print (" Yang anda ketik adalah e
^
KasusSwitch.java:20: error: illegal character: '\'
```

7. KasusSwitch.java

Kode di atas adalah contoh penggunaan struktur switch-case dalam bahasa pemrograman Java. Program ini meminta pengguna untuk memasukkan sebuah huruf, kemudian menampilkan pesan yang sesuai berdasarkan huruf yang dimasukkan.

Mari kita jelaskan bagian-bagian utama dari kode tersebut:

1. **import java.util.Scanner;;** Mendeklarasikan impor kelas Scanner dari paket java.util. Ini memungkinkan program untuk menggunakan Scanner untuk membaca input dari pengguna.
2. **public class KasusSwitch { ... };** Mendefinisikan sebuah kelas Java bernama KasusSwitch.
3. **public static void main(String[] args) { ... };** Mendefinisikan metode utama (main) dari program. Metode ini adalah titik masuk utama ketika program dijalankan.
4. **char cc;;** Mendeklarasikan sebuah variabel cc dengan tipe data char. Variabel ini akan digunakan untuk menyimpan huruf yang dimasukkan oleh pengguna.
5. **Scanner masukan = new Scanner(System.in);;** Membuat sebuah objek Scanner yang digunakan untuk membaca input dari pengguna melalui aliran input standar (System.in).
6. **System.out.print ("Ketikkan sebuah huruf, akhiri dengan RETURN \n");** Mencetak pesan ke konsol untuk memberikan petunjuk kepada pengguna tentang apa yang harus dimasukkan.
7. **cc = masukan.next().charAt(0);** Membaca sebuah huruf yang dimasukkan oleh pengguna dan menyimpannya dalam variabel cc.
8. **switch (cc) { ... };** Ini adalah struktur switch-case. Program akan mengevaluasi nilai variabel cc dan akan memilih salah satu dari beberapa kasus berdasarkan nilai tersebut.
9. **case 'a': { System.out.print (" Yang anda ketik adalah a \n"); break; };** Ini adalah salah satu kasus dalam struktur switch-case. Jika nilai cc adalah 'a', maka program akan mencetak pesan "Yang anda ketik adalah a".
10. **default: System.out.print (" Yang anda ketik adalah huruf mati \n");** Ini adalah bagian default dari struktur switch-case. Jika nilai cc tidak cocok dengan salah satu kasus yang ditentukan sebelumnya, maka program akan mencetak pesan "Yang anda ketik adalah huruf mati".

Dengan demikian, program ini memungkinkan pengguna untuk memasukkan sebuah huruf dan menampilkan pesan yang sesuai berdasarkan huruf yang

dimasukkan, baik itu 'a', 'u', 'e', 'i', 'o', atau huruf lainnya. Jika huruf yang dimasukkan tidak termasuk dalam kasus yang ditentukan, maka program akan memberikan pesan "Yang anda ketik adalah huruf mati".

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac Konstant.java
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java Konstant
Jari-jari lingkaran =60
Luas lingkaran = 11309.4
Akhir program
```

8. Konstant.java

1. Deklarasi Konstanta:

- `final float PHI = 3.1415f;` Menetapkan nilai konstanta PHI dengan nilai 3.1415f (tipe data float).

2. Deklarasi variabel:

- `float r;` Mendeklarasikan variabel r dengan tipe data float untuk menyimpan jari-jari lingkaran.
- `Scanner masukan=new Scanner(System.in);` Membuat objek Scanner baru dengan nama masukan untuk membaca input dari pengguna.

3. Membaca data:

- `System.out.print ("Jari-jari lingkaran =");` Menampilkan teks "Jari-jari lingkaran =" ke layar.
- `r = masukan.nextFloat();` Membaca input dari pengguna dan menyimpannya dalam variabel r.

4. Menghitung dan Menampilkan Hasil:

- `System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");`:: Menghitung luas lingkaran dengan rumus $PHI * r * r$ dan menampilkan hasilnya ke layar.
- `System.out.print ("Akhir program \n");`:: Menampilkan teks "Akhir program" ke layar.

Ringkasan:

Kode ini mendemonstrasikan cara mendeklarasikan konstanta, variabel, membaca input dari pengguna, menghitung luas lingkaran, dan menampilkan hasil dalam bahasa pemrograman Java.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac Max2.java
Max2.java:14: error: unclosed string literal
System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN :
                ^
Max2.java:15: error: illegal character: '\'
\n");
^
Max2.java:15: error: unclosed string literal
\n");
^
Max2.java:15: error: not a statement
\n");
^
```

9. Max2.java

Berikut penjelasan singkat kode Java untuk mencari nilai maksimum dari dua bilangan:

1. Deklarasi variabel:

- `int a, b;` Mendeklarasikan dua variabel a dan b dengan tipe data int untuk menyimpan dua bilangan yang akan dibandingkan.
- `Scanner masukan=new Scanner(System.in);` Membuat objek Scanner baru dengan nama masukan untuk membaca input dari pengguna.

2. Membaca data:

- `System.out.print ("Maksimum dua bilangan : \n");` Menampilkan teks "Maksimum dua bilangan :" ke layar.

- `System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN : \n");` Menampilkan instruksi untuk memasukkan dua bilangan ke layar.
-
- `a=masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel a.
- `b=masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel b.

3. Menentukan Nilai Maksimum:

- `if (a >= b){` : Memeriksa apakah nilai a lebih besar atau sama dengan b.
 - `System.out.println ("Nilai a yang maksimum "+ a);` Jika a lebih besar atau sama dengan b, maka tampilkan pesan "Nilai a yang maksimum" dan nilai a.
- `else /* a > b */{` : Jika a tidak lebih besar atau sama dengan b, maka:
 - `System.out.println ("Nilai b yang maksimum: "+ b);` Tampilkan pesan "Nilai b yang maksimum" dan nilai b.

4. Menampilkan Hasil:

- Kode ini menampilkan dua pesan:
 - `"Ke dua bilangan : a = " + a + " b = " + b;` Menampilkan nilai a dan b yang telah diinputkan.
 - Pesan "Nilai a yang maksimum" atau "Nilai b yang maksimum" beserta nilainya, tergantung pada hasil perbandingan a dan b.

Ringkasan:

Kode ini mendemonstrasikan cara membaca dua bilangan dari pengguna, membandingkannya, dan menentukan nilai maksimumnya.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PriFor.java
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java PriFor
Baca N, print 1 s/d N N = 8
1
2
3
4
5
6
7
8
Akhir program
```

10. PriFor.java

Berikut penjelasan singkat kode Java tersebut:

1. Deklarasi Variabel:

- `int i, N;` Mendeklarasikan dua variabel integer:
 - `i`: Akan digunakan sebagai counter dalam perulangan.
 - `N`: Akan digunakan untuk menyimpan nilai batas atas untuk mencetak angka.
- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Baca N, print 1 s/d N ");` Menampilkan teks instruksi.
- `System.out.print("N = ");` Menampilkan teks untuk input nilai N.
- `N = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel N.

3. Perulangan dengan for:

- `for (i = 1; i <= N; i++) {` Memulai perulangan dengan:
 - Inisialisasi counter `i = 1`
 - Kondisi perulangan: `i` harus kurang dari atau sama dengan N
 - Peningkatan nilai `i` setelah setiap iterasi
- `System.out.println(i);` Mencetak nilai `i` pada setiap iterasi perulangan.

4. Pesan Akhir:

- `System.out.println("Akhir program \n");`: Menampilkan teks "Akhir program" sebagai penanda akhir program.

Ringkasan:

Kode ini meminta pengguna memasukkan nilai N, kemudian mencetak angka-angka dari 1 sampai N dengan menggunakan perulangan for.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintIterasi.java
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java PrintIterasi
Nilai N >0 = 17
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
```

11. PrintIterasi.java

1. Deklarasi Variabel:

- `int N`; Mendeklarasikan variabel integer N untuk menyimpan nilai batas atas pencetakan angka.

- `int i`:: Mendeklarasikan variabel integer `i` untuk digunakan sebagai counter dalam perulangan.
- `Scanner masukan = new Scanner(System.in)`:: Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Nilai N >0 = ")`:: Menampilkan teks untuk meminta input nilai `N`.
- `N = masukan.nextInt()`:: Membaca input dari pengguna dan menyimpannya dalam variabel `N`.

3. Inisialisasi Counter:

- `i = 1`:: Menginisialisasi counter `i` dengan nilai 1 untuk memulai pencetakan angka dari 1.

4. Perulangan dengan for (Iterate):

- `for (;) {`: Memulai perulangan tanpa kondisi awal dan akhir yang eksplisit, sehingga akan terus berulang hingga ada perintah `break`.
 - `System.out.println(i)`:: Mencetak nilai `i` pada setiap iterasi perulangan.
 - `if (i == N) break`:: Jika nilai `i` sudah mencapai `N`, maka perulangan dihentikan dengan perintah `break`.
 - `else { i++; }`: Jika belum mencapai `N`, nilai `i` diincrement (ditambahkan 1) untuk mencetak angka berikutnya.

Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai `N`, kemudian mencetak angka-angka dari 1 sampai `N` dengan menggunakan perulangan `for` yang dimodifikasi menjadi bentuk `iterate`. Bentuk ini berbeda dengan perulangan `for` biasa karena tidak memiliki kondisi awal dan akhir yang eksplisit, sehingga mengandalkan perintah `break` untuk menghentikan perulangan.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java PrintRepeat.java
Nilai N >0 = 19
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

12. PrintRepeat.java

1. Deklarasi Variabel:

- `int N`; Mendeklarasikan variabel integer N untuk menyimpan nilai batas atas pencetakan angka.
- `int i`; Mendeklarasikan variabel integer i untuk digunakan sebagai counter dalam perulangan.
- `Scanner masukan = new Scanner(System.in)`; Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Nilai N >0 = ")`; Menampilkan teks untuk meminta input nilai N.
- `N = masukan.nextInt()`; Membaca input dari pengguna dan menyimpannya dalam variabel N.

3. Inisialisasi Counter:

- `i = 1`:: Menginisialisasi counter `i` dengan nilai 1 untuk memulai pencetakan angka dari 1.

4. Perulangan dengan do-while:

- `do { ... } while (i <= N)`:: Memulai perulangan do-while:
 - `System.out.print(i + "\n")`:: Mencetak nilai `i` pada setiap iterasi perulangan.
 - `i++`:: Mengincrement (menambahkan 1) nilai `i` untuk mencetak angka berikutnya.
 - `while (i <= N)`:: Kondisi perulangan: Perulangan akan terus berlanjut selama nilai `i` masih kurang dari atau sama dengan `N`.

Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai `N`, kemudian mencetak angka-angka dari 1 sampai `N` dengan menggunakan perulangan do-while. Perulangan do-while menjamin bahwa blok kode di dalamnya akan dijalankan minimal sekali, baru kemudian kondisi perulangan diperiksa. Hal ini berbeda dengan perulangan while biasa yang memeriksa kondisi terlebih dahulu sebelum menjalankan blok kode.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java PrintWhile
Nilai N >0 = 9
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9
```

13. PrintWhile.java

1. Deklarasi Variabel:

- `int N`;; Mendeklarasikan variabel integer N untuk menyimpan nilai batas atas pencetakan angka.
- `int i`;; Mendeklarasikan variabel integer i untuk digunakan sebagai counter dalam perulangan.
- `Scanner masukan = new Scanner(System.in)`;; Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Nilai N >0 = ")`;; Menampilkan teks untuk meminta input nilai N.
- `N = masukan.nextInt()`;; Membaca input dari pengguna dan menyimpannya dalam variabel N.

3. Inisialisasi Counter:

- `i = 1`;; Menginisialisasi counter i dengan nilai 1 untuk memulai pencetakan angka dari 1.

4. Perulangan dengan while:

- `while (i <= N) { ... }`: Memulai perulangan while:
 - `System.out.println(i)`;; Mencetak nilai i pada setiap iterasi perulangan.
 - `i++`;; Mengincrement (menambahkan 1) nilai i untuk mencetak angka berikutnya.
 - `while (i <= N)`;; Kondisi perulangan: Perulangan akan terus berlanjut selama nilai i masih kurang dari atau sama dengan N.

Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai N, kemudian mencetak angka-angka dari 1 sampai N dengan menggunakan perulangan while. Perulangan while akan terus mengeksekusi blok kode di dalamnya selama kondisi perulangan terpenuhi.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java PrintWhile1
Nilai N >0 = 15
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

14. PrintWhile1.java

Berikut penjelasan singkat kode Java tersebut:

1. Deklarasi Variabel:

- `int N`;; Mendeklarasikan variabel integer N untuk menyimpan nilai batas atas pencetakan angka.
- `int i = 1`;; Mendeklarasikan variabel integer i dengan nilai awal 1 untuk digunakan sebagai counter dalam perulangan. Inisialisasi dilakukan langsung saat deklarasi.
- `Scanner masukan = new Scanner(System.in)`;; Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Nilai N >0 = ")`;; Menampilkan teks untuk meminta input nilai N.

- `N = masukan.nextInt();`: Membaca input dari pengguna dan menyimpannya dalam variabel `N`.

3. Perulangan while (versi ringkas):

- `while (i <= N) { System.out.println(i++); }`:
 - Kondisi perulangan: Perulangan akan terus berlanjut selama nilai `i` masih kurang dari atau sama dengan `N`.
 - `System.out.println(i++);`: Mencetak nilai `i` pada setiap iterasi, kemudian mengincrement nilai `i` dengan 1. Perintah ini menggabungkan proses pencetakan dan increment dalam satu baris.

Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai `N`, kemudian mencetak angka-angka dari 1 sampai `N` dengan menggunakan perulangan `while` versi ringkas. Kode ini lebih singkat karena inisialisasi counter `i` langsung dilakukan saat deklarasi, dan proses pencetakan dan increment dilakukan dalam satu baris.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXinterasi.java
PrintXinterasi.java:15: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg 999
                  ^
PrintXinterasi.java:16: error: unclosed string literal
: ");
  ^
PrintXinterasi.java:24: error: unclosed string literal
System.out.print ("Masukkan nilai x (int),
                  ^
PrintXinterasi.java:25: error: ';' expected
akhiri dg 999 : ");
                ^
PrintXinterasi.java:25: error: unclosed string literal
akhiri dg 999 : ");
                ^
```

15. PrintXinterasi.java

Berikut penjelasan singkat kode Java tersebut:

1. Deklarasi Variabel:

- `int Sum = 0;` Mendeklarasikan variabel integer Sum dengan nilai awal 0 untuk menyimpan hasil penjumlahan.
- `int x;` Mendeklarasikan variabel integer x untuk menyimpan nilai yang diinputkan pengguna.
- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Nilai Pertama:

- `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
- `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.

3. Perulangan for (Iterate):

- `for (;;)` Memulai perulangan for tanpa batasan eksplisit.
 - `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
 - `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.
 - `if (x == 999) break;` Jika nilai x sama dengan 999, maka keluar dari perulangan.
 - `else { Sum = Sum + x; }` Jika nilai x bukan 999, maka tambahkan nilai x ke Sum.

4. Menampilkan Hasil:

- `System.out.println("Hasil penjumlahan = "+ Sum);` Mencetak hasil penjumlahan ke layar.

5. Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai-nilai integer, diakhiri dengan 999. Kode ini kemudian menjumlahkan semua nilai yang diinputkan dan menampilkan hasilnya. Perulangan for digunakan untuk membaca nilai-nilai yang diinputkan pengguna.

Catatan:

- Kode ini menggunakan for tanpa batasan eksplisit. Hal ini dapat diubah dengan menggunakan while loop dengan kondisi `x != 999`.
- Kode ini tidak melakukan validasi input. Pastikan pengguna memasukkan nilai integer yang valid.
- Kode ini dapat dimodifikasi untuk menghitung rata-rata, minimum, dan maksimum dari nilai-nilai yang diinputkan.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXinterasi.java
PrintXinterasi.java:15: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg 999
                  ^
PrintXinterasi.java:16: error: unclosed string literal
: ");
  ^
PrintXinterasi.java:24: error: unclosed string literal
System.out.print ("Masukkan nilai x (int),
                  ^
PrintXinterasi.java:25: error: ';' expected
akhiri dg 999 : ");
                ^
PrintXinterasi.java:25: error: unclosed string literal
akhiri dg 999 : ");
                ^
5 errors

D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXRepeat.java
PrintXRepeat.java:15: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg
                  ^
PrintXRepeat.java:16: error: unclosed string literal
999 : ");
      ^
PrintXRepeat.java:25: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri
                  ^
PrintXRepeat.java:26: error: not a statement
dg 999 : ");
  ^
PrintXRepeat.java:26: error: ';' expected
dg 999 : ");
  ^
PrintXRepeat.java:26: error: unclosed string literal
dg 999 : ");
  ^
```

16. PrintXRepeat.java

Berikut penjelasan singkat kode Java tersebut:

1. Deklarasi Variabel:

- `int Sum`;: Mendeklarasikan variabel integer `Sum` untuk menyimpan hasil penjumlahan.
- `int x`;: Mendeklarasikan variabel integer `x` untuk menyimpan nilai yang diinputkan pengguna.

- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Nilai Pertama:

- `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
- `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.

3. Perulangan do-while:

- `do { ... } while (x != 999);` Memulai perulangan do-while:
 - `Sum = Sum + x;` Menambahkan nilai x ke Sum.
 - `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
 - `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.
 - `while (x != 999);` Perulangan akan terus berlanjut selama nilai x tidak sama dengan 999.

4. Menampilkan Hasil:

- `System.out.println("Hasil penjumlahan = "+Sum);` Mencetak hasil penjumlahan ke layar.

5. Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai-nilai integer, diakhiri dengan 999. Kode ini kemudian menjumlahkan semua nilai yang diinputkan dan menampilkan hasilnya. Perulangan do-while digunakan untuk membaca nilai-nilai yang diinputkan pengguna. Perulangan ini dijamin akan dijalankan minimal sekali, meskipun pengguna langsung memasukkan 999.

Catatan:

- Kode ini tidak melakukan validasi input. Pastikan pengguna memasukkan nilai integer yang valid.
- Kode ini dapat dimodifikasi untuk menghitung rata-rata, minimum, dan maksimum dari nilai-nilai yang diinputkan.

- Kode ini dapat dimodifikasi dengan menambahkan opsi untuk menghentikan perulangan di tengah proses, seperti dengan memasukkan nilai khusus.

Perbedaan dengan Kode Sebelumnya:

Kode ini menggunakan perulangan do-while instead of for loop. Perbedaannya adalah do-while dijamin akan dijalankan minimal sekali, sedangkan for loop tidak.

Contoh:

Misalkan pengguna memasukkan nilai 1, 2, 3, dan 999.

- for loop:
 - Perulangan dijalankan sebanyak 3 kali.
 - Nilai Sum menjadi 6.
- do-while loop:
 - Perulangan dijalankan sebanyak 4 kali.
 - Nilai Sum menjadi 6.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXRepeat.java
PrintXRepeat.java:15: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg
                  ^
PrintXRepeat.java:16: error: unclosed string literal
999 : ");
      ^
PrintXRepeat.java:25: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri
                  ^
PrintXRepeat.java:26: error: not a statement
dg 999 : ");
^
PrintXRepeat.java:26: error: ';' expected
dg 999 : ");
      ^
PrintXRepeat.java:26: error: unclosed string literal
dg 999 : ");
      ^
6 errors

D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXRepeat.java
PrintXRepeat.java:15: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg
                  ^
PrintXRepeat.java:16: error: unclosed string literal
999 : ");
      ^
PrintXRepeat.java:25: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri
                  ^
PrintXRepeat.java:26: error: not a statement
dg 999 : ");
^
PrintXRepeat.java:26: error: ';' expected
dg 999 : ");
      ^
PrintXRepeat.java:26: error: unclosed string literal
dg 999 : ");
```

17. PrintXWhile.java

1. Deklarasi Variabel:

- `int Sum;` Mendeclare variabel integer Sum untuk menyimpan hasil penjumlahan.
- `int x;` Mendeclare variabel integer x untuk menyimpan nilai yang diinputkan pengguna.
- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna.

2. Inisialisasi:

- `Sum = 0;` Menginisialisasi variabel Sum dengan nilai 0.

3. Membaca Nilai Pertama:

- `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
- `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.

4. Perulangan while:

- `while (x != 999) { ... }` Memulai perulangan while:
 - `Sum = Sum + x;` Menambahkan nilai x ke Sum.
 - `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");` Menampilkan teks untuk meminta input nilai x.
 - `x = masukan.nextInt();` Membaca input dari pengguna dan menyimpannya dalam variabel x.
 - `while (x != 999);` Perulangan akan terus berlanjut selama nilai x tidak sama dengan 999.

5. Menampilkan Hasil:

- `System.out.println("Hasil penjumlahan = "+ Sum);` Mencetak hasil penjumlahan ke layar.

6. Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai-nilai integer, diakhiri dengan 999. Kode ini kemudian menjumlahkan semua nilai yang diinputkan dan menampilkan hasilnya. Perulangan while digunakan untuk membaca nilai-nilai yang diinputkan pengguna. Perulangan ini akan terus berlanjut sampai pengguna memasukkan 999.

Catatan:

- Kode ini tidak melakukan validasi input. Pastikan pengguna memasukkan nilai integer yang valid.
- Kode ini dapat dimodifikasi untuk menghitung rata-rata, minimum, dan maksimum dari nilai-nilai yang diinputkan.
- Kode ini dapat dimodifikasi dengan menambahkan opsi untuk menghentikan perulangan di tengah proses, seperti dengan memasukkan nilai khusus.

Perbedaan dengan Kode Sebelumnya:

Kode ini menggunakan perulangan while instead of do-while loop.

Perbedaannya adalah while loop tidak dijamin akan dijalankan, sedangkan do-while loop dijamin akan dijalankan minimal sekali.

Contoh:

Misalkan pengguna memasukkan nilai 1, 2, 3, dan 999.

- while loop:
 - Perulangan dijalankan sebanyak 3 kali.
 - Nilai Sum menjadi 6.
- do-while loop:
 - Perulangan dijalankan sebanyak 4 kali.
 - Nilai Sum menjadi 6.

Ou

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac PrintXWhile.java
PrintXWhile.java:16: error: unclosed string literal
System.out.print ("Masukkan nilai x (int), akhiri dg
                  ^
PrintXWhile.java:17: error: unclosed string literal
999 : ");
      ^
PrintXWhile.java:21: error: unclosed string literal
System.out.print("Masukkan nilai x (int), akhiri
                  ^
PrintXWhile.java:22: error: not a statement
dg 999 : ");
^
PrintXWhile.java:22: error: ';' expected
dg 999 : ");
      ^
PrintXWhile.java:22: error: unclosed string literal
dg 999 : ");
      ^
```

18. SubProgram.java

Penjelasan Kode Java SubProgram.java

1. Definisi Prosedur dan Fungsi:

- maxab(int a, int b): Prosedur untuk mencari maksimum dua bilangan bulat. Prosedur ini mengembalikan nilai maksimum dari a dan b.
- tukar(int a, int b): Prosedur untuk menukar dua bilangan bulat. Prosedur ini tidak mengembalikan nilai, tetapi mengubah nilai a dan b di dalam program.

2. Program Utama:

- Membaca dua bilangan integer a dan b dari pengguna.
- Menuliskan maksimum dua bilangan yang dibaca dengan menggunakan fungsi maxab(a, b).
- Menukar kedua bilangan dengan menggunakan prosedur tukar(a, b).
- Mencetak nilai a dan b setelah ditukar.

3. Penjelasan Lebih Detail:

- Baris 13-15: Deklarasi variabel a dan b untuk menyimpan dua bilangan integer yang dibaca dari pengguna.
- Baris 17-20: Membaca dua bilangan integer dari pengguna dan disimpan dalam variabel a dan b.
- Baris 22: Menuliskan nilai a dan b sebelum ditukar.
- Baris 24: Memanggil fungsi maxab(a, b) untuk mencari maksimum dua bilangan a dan b. Hasilnya disimpan dalam variabel maks.
- Baris 25: Mencetak nilai maksimum maks.
- Baris 27: Menuliskan pesan "Tukar kedua bilangan...".
- Baris 28: Memanggil prosedur tukar(a, b) untuk menukar nilai a dan b.
- Baris 30-31: Mencetak nilai a dan b setelah ditukar.

4. Catatan:

- Kode ini hanya contoh sederhana untuk menunjukkan penggunaan prosedur dan fungsi dalam bahasa Java.
- Anda dapat memodifikasi kode ini untuk menyelesaikan masalah yang lebih kompleks.
- Pastikan Anda memahami cara kerja prosedur dan fungsi sebelum menggunakannya dalam program Anda.
-

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>javac SubProgram.java
SubProgram.java:18: error: unclosed string literal
System.out.println ("Ke dua bilangan setelah tukar: a =
^
SubProgram.java:19: error: unclosed string literal
"+ a +" b = "+ b);
^
SubProgram.java:30: error: unclosed string literal
System.out.print ("Maksimum dua bilangan
^
SubProgram.java:31: error: illegal character: '\'
\n");
^
SubProgram.java:31: error: unclosed string literal
\n");
^
SubProgram.java:31: error: not a statement
\n");
^
SubProgram.java:32: error: unclosed string literal
System.out.print ("Ketikkan dua bilangan,
^
SubProgram.java:33: error: ';' expected
pisahkan dg RETURN : \n");
^
SubProgram.java:33: error: illegal character: '\'
pisahkan dg RETURN : \n");
^
SubProgram.java:33: error: unclosed string literal
pisahkan dg RETURN : \n");
^
SubProgram.java:33: error: not a statement
pisahkan dg RETURN : \n");
^
SubProgram.java:36: error: unclosed string literal
System.out.println ("Ke dua bilangan : a =
^
SubProgram.java:37: error: unclosed string literal
```

19. Tempair.java

Berikut penjelasan singkat kode Java Tempair.java tersebut:

1. Deklarasi Variabel:

- `int T;` Mendeklarasikan variabel integer T untuk menyimpan nilai temperatur yang akan diinputkan pengguna.
- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna.

2. Membaca Input:

- `System.out.print("Contoh IF tiga kasus \n");` Menampilkan teks "Contoh IF tiga kasus".
- `System.out.print("Temperatur (der. C) = ");` Menampilkan teks untuk meminta input temperatur.
- `T = masukan.nextInt();` Membaca input temperatur dari pengguna dan menyimpannya dalam variabel T.

3. Percabangan if-else if:

- Kode menggunakan struktur percabangan if-else if untuk menentukan wujud air berdasarkan nilai temperatur T.

- if ($T < 0$) { ... }: Jika nilai T kurang dari 0, maka:
 - `System.out.print("Wujud air beku \n"+ T);`: Mencetak pesan "Wujud air beku" dan nilai T.
- else if ($(0 \leq T) \ \&\& \ (T \leq 100)$) { ... }: Jika nilai T antara 0 (inklusif) dan 100 (inklusif), maka:
 - `System.out.print("Wujud air cair \n"+ T);`: Mencetak pesan "Wujud air cair" dan nilai T.
- else if ($T > 100$) { ... }: Jika nilai T lebih dari 100, maka:
 - `System.out.print("Wujud air uap/gas \n"+ T);`: Mencetak pesan "Wujud air uap/gas" dan nilai T.

4. Ringkasan:

Kode ini meminta pengguna untuk memasukkan nilai temperatur (derajat Celcius). Kemudian, kode ini menggunakan percabangan if-else if untuk menentukan wujud air (beku, cair, atau uap/gas) berdasarkan nilai temperatur tersebut.

5. Catatan:

- Kode ini dapat dimodifikasi untuk menampilkan output yang lebih informatif, misalnya dengan menyertakan rentang temperatur untuk setiap wujud air.

Output :

```
D:\UMI\SEM 4\PBO\Tugas 3\tugas3_sc>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 18
Wujud air cair
18
```


