

BAB 1 RUNTUNAN

1.1 Tujuan Pembelajaran

1. Mahasiswa diharapkan untuk memahami program Java dengan menggunakan IDE Eclipse.
2. Mahasiswa mampu mengimplementasikan algoritma dasar di pemrograman Java.
3. Mahasiswa dapat memahami atribut dan method dalam sebuah pemrograman.

1.2 Dasar Teori

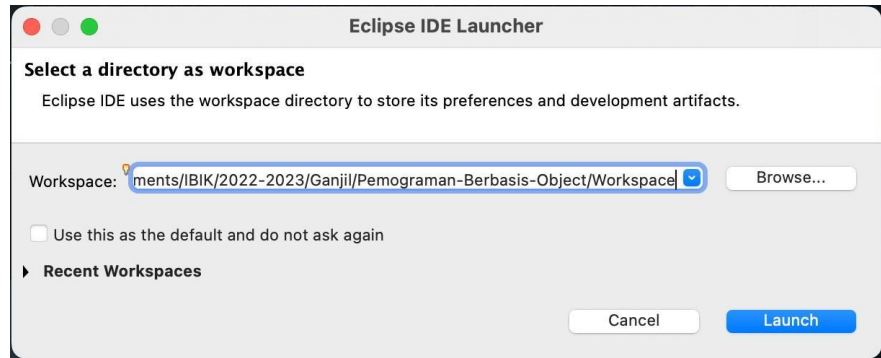
1.2.1 IDE Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

- **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- **Multi-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Langkah-langkah membuat project baru adalah sebagai berikut:

1. Buka Eclipse IDE, pilih directory workspace yang diinginkan klik tombol Launch

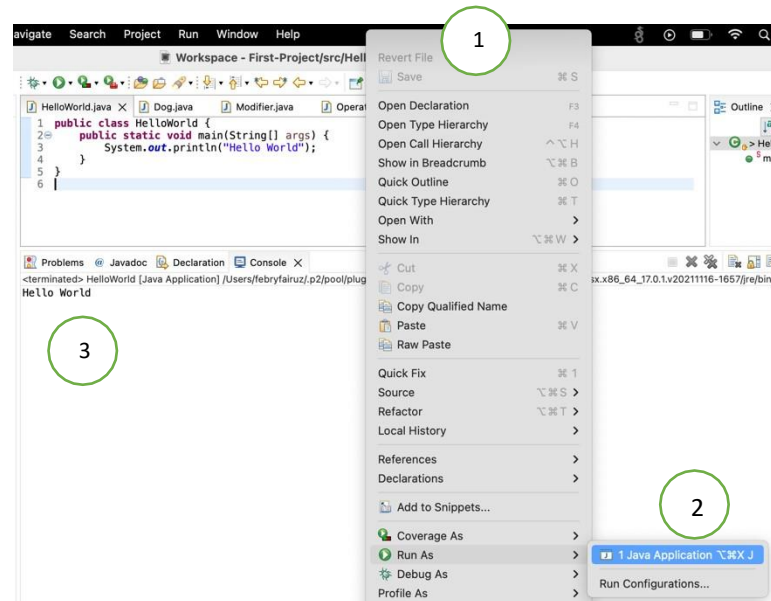


2. Pada IDE Eclipse pilih menu *File* ☐ *New* ☐ *Other...* maka akan muncul *pop up window Select Wizard*.
 3. Pilih *Java* ☐ *Java Project* lalu klik *Next*. Pada *pop up window New Java Project* tuliskan nama proyek pada *field Project Name* lalu tekan tombol *Finish*.
 4. Selanjutnya membuat file class pada Java. Pada *collation view*, *expand project* yang telah dibuat. Klik kanan pada folder *src*, jika memilih *New* ☐ *Class* maka lanjut ke Langkah 6 namun jika memilih *New* ☐ *Other...* lanjut ke langkah 5.
 5. Pada *pop up window Select a Wizard* pilih *Java* ☐ *Class* maka akan menampilkan *pop up window New Java Class*
 6. Pada *pop up New Java Class* isilah isian field *Name* dengan *Class Name* yang akan dibuat lalu klik tombol *Finish*.
- kebawah lgi tulisan bab uppercasse yak
7. Setelah berhasil membuat file kelas masukan *syntax* sederhana untuk

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

menampilkan tulisan “Hello World” seperti dibawah ini:

8. Untuk menjalankannya pada *source code* file *class main*, klik kanan pilih *Run As => Java Application*.



1.2.2 Variabel

Variabel digunakan untuk menampung data yang digunakan pada program. Variabel menampung data tergantung pada tipe data yang dideklarasikan sebelumnya.

Cara mendeklarasikan variabel yaitu :

Sintaks dasar :

[tipe data] [nama variabel]

Untuk mendefinisikan variabel, suatu identifier dapat digunakan untuk menamai variabel tersebut. Identifier adalah kumpulan karakter yang dapat digunakan untuk menamai variabel, method, class, interface, dan package. Dalam pemrograman Java identifier bisa disebut sah apabila diawali dengan :

- Huruf/ abjad
- Karakter Mata Uang
- Underscore(_)

Identifier tidak boleh mengandung @, spasi atau diawali dengan angka serta tidak boleh menggunakan *keyword* yang telah digunakan di pemrograman java. Selain karakter, Unicode juga dapat digunakan sebagai identifier.

Java memiliki dua jenis tipe data yang dikategorikan menjadi dua yaitu tipe data primitif dan tipe data referensi.

Macam tipe data primitif diantaranya :

- **Integer (Bilangan bulat)**

Integer merupakan tipe data numerik yang digunakan untuk mendefinisikan bilangan bulat. Tipe data numeric yang termasuk integer diantaranya :

Tipe	Deskripsi
Byte	-128 s/d +127 menempati 8 bits di memori
Short	-32768 s/d +32767 menempati 16 bits di memori
Int	-2147483648 s/d +2147483647 menempati 32 bits di memori
Long	-9223372036854775808 s/d +9223372036854775807 menempati 64 bits di memori

- **Floating Point (Bilangan pecahan)**

Floating point digunakan untuk menangani bilangan desimal atau perhitungan yang lebih detail dibanding integer.

Tipe	Deskripsi
Float	-3.4×10^8 s/d $+3.4 \times 10^8$
Double	-1.7×10^{308} s/d $+1.7 \times 10^{308}$

- **Char**

Char adalah karakter tunggal yang pendefinisian di awal dan akhir menggunakan tanda petik tunggal ('). Tipe char mengikuti aturan Unicode, sehingga bisa dapat menggunakan kode untuk kemudian diikuti bilangan dari 0 sampai 65535, tetapi yang biasa digunakan adalah bilangan heksadesimal dari 0000 sampai FFFF.

- **Boolean**

Tipe data Boolean terdiri dari dua nilai saja, yaitu *true* dan *false*. Boolean sangat penting untuk mengevaluasi suatu kondisi.

- **Tipe Data Referensi**

Kelebihan pemrograman dengan orientasi objek adalah dapat mendefinisikan tipe data baru yang merupakan objek dari *class* tertentu. Tipe data ini digunakan untuk mereferensikan objek atau class tertentu, seperti String.

1.2.3 Operator

Operator adalah simbol khusus yang menyajikan operasi khusus pada satu, dua, atau tiga operand dan kemudian mengembalikan hasilnya. Jenis operator antara lain :

- **Operator Aritmatika**

Operator ini digunakan pada operasi-operasi aritmatika seperti penjumlahan, pengurangan, pembagian dan lain-lain.

Operat or	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus (Sisa bagi)
++	Increment
Operat or	Keterangan
--	Decrement

- **Operator Relasional**

Untuk membandingkan 2 nilai (variabel) atau lebih digunakan operator

relasional, dimana operator ini akan mengembalikan atau menghasilkan nilai *True* atau *False*.

Operat or	Keterangan
<	Lebih Kecil
<=	Lebih Kecil Sama Dengan
>	Lebih Besar
>=	Lebih Besar Sama Dengan
==	Sama Dengan
!=	Tidak Sama Dengan

▪ Operator Kondisional

Operator ini menghasilkan nilai yang sama dengan operator relasional, hanya saja penggunaannya lebih pada operasi – operasi Boolean.

Operat or	Keterangan
&&	And
	Or
!	Not

▪ Operator Shift dan Bitwise

Kedua operator ini digunakan untuk memanipulasi nilai dari bitnya, sehingga diperoleh nilai yang lain.

Operat or	Keterangan
--------------	------------

&	And
Operat or	Keterangan
	Or

^	Xor
~	Not
>>	Shift Right
<<	Shift Left

▪ Operator Assignment

Operator assignment digunakan untuk memberikan suatu nilai ke sebuah variabel. Operator assignment tidak hanya berupa '=', namun selain itu dalam C# terdapat beberapa shortcut assignment yang lain.

Operat or	Contoh	Ekivalen dengan
+=	b += a	b = b + a
-=	b -= a	b = b - a
*=	b *= a	b = b * a
/=	b /= a	b = b / a
%=	b %= a	b = b % a
&=	b &= a	b = b & a

<code> =</code>	<code>b = a</code>	<code>b = b a</code>
<code>^=</code>	<code>b ^= a</code>	<code>b = b ^ a</code>
<code><<=</code>	<code>b <<= a</code>	<code>b = b << a</code>
<code>>>=</code>	<code>b >>= a</code>	<code>b = b >> a</code>

Contoh

```
public class Operator
{ //block start

    //class body
    public static void main(String[] args) {
        int A, C;
        int B = 25;
        A = 10;
        C = A + B;
        System.out.println("C = " + C );
        System.out.println("AxB = " + A*B );
        System.out.println("A increment = " + A++ );
        System.out.println("A = " + A );
    }

} //end block
```

1.2.4 Object dan Class

Object Objek memiliki status dan perilaku. Contoh: Seekor anjing memiliki status - warna, nama, ras dll. Objek adalah turunan dari kelas.

Class Kelas dapat didefinisikan sebagai templat/cetak biru yang menjelaskan perilaku/status yang didukung oleh objek dari tipenya.

Dalam sebuah Class, class memiliki 3 struktur bagian yaitu, Atribute, Method/Function, dan Object.

Contoh Class pada Java:

```
public class Dog {
    String breed;
    int age;
    String color;

    void barking() {
        System.out.println("woof..woof..");
    }

    void hungry() {
        System.out.println("woof.. it's time for lunch");
    }

    void sleeping() {
        System.out.println("snoring.. zzz.. zzz..");
    }

    void puppyName(String name) {
        System.out.println("My puppy name is: "+name);
    }
}
```

Pada source code diatas terdapat sebuah class bernama Dog. Dimana class ini memiliki 3 buah attribute yaitu breed, age dan color. Dan memiliki 4 buah method Bernama barking(), hungry(), sleeping(), dan puppyName().

Contoh class main pada Java:

```
public class AppMain {
    static Dog mydog = new Dog();

    public static void main(String[] args) {
        mydog.hungry();
        String name = "Tomi";
        mydog.puppyName(name);
    }
}
```

Pada source code di atas class Main Bernama AppMain menginisialisasi object dari class Dog sebagai attribute dengan nama mydog. Sedangkan pada method main, di dalam method ini mengeksekusi perintah untuk memanggil method hungry() dan puppyName() yang berada pada class Dog. Dimana method

mydog.puppyName(name) memiliki sebuah parameter yang harus diisi Bernama name. nilai dari parameter ini diambil dari variable instance bernama name dengan tipe data String dan bernilai “Tomi”.



The screenshot shows a Java IDE console window with tabs for Problems, Javadoc, Declaration, and Console. The console output is as follows:

```
<terminated> HelloWorld [Java Application] /Users/febryfairuz/.p2/p
woof.. it's time for lunch
My puppy name is: Tomi
```

1.2.5 Atribute dan Method

- **Atribute**

Atribute merupakan bagian dari variable, dimana variable memiliki 3 buah tipe, yaitu:

- **Local Variable** □ didefinisikan atau diinisialisasi di dalam method, constructor, atau didalam satu blok yang sama. Tipe ini akan dideklarasikan dan diinisialisasi di dalam method dan tipe ini akan memusnahkan nilai inisialisasi jika penggunaan method sudah selesai.
- **Instance Variable** □ type ini didefinisikan dalam class dan berada diluar method manapun. Variable ini disebut dengan **Atribute**, karena akan diinisialisasi ketika class dipakai.
- **Class Variable** □ variable ini biasanya diberikan inisialisasi keyword static dan pendeklarasiannya berada di dalam class dan berada diluar method manapun.
- **Method**

Metode Java merupakan kumpulan pernyataan yang dikelompokkan bersama untuk melakukan sebuah operasi. Saat memanggil metode

System.out.println(), misalnya, sistem sebenarnya mengeksekusi beberapa pernyataan untuk menampilkan pesan pada konsol atau terminal.

Method dapat dibuat sendiri dengan atau tanpa nilai balik, memanggil metode dengan atau tanpa parameter, dan menerapkan abstraksi metode dalam desain program.

Syntax dasar:

[tipe data] name_of_method() or

[tipe data] name_of_method([tipe_data] name_of_variable).

Contoh method **tanpa** nilai baik:

```
public class Dog {  
    void barking() {  
        System.out.println("woof..woof..");  
    }  
}
```

Method dengan tipe data void merupakan method yang tidak akan mengembalikan nilai apapun, seluruh operasi atau syntax akan langsung dieksekusi di dalam method tersebut.

Contoh method **dengan** nilai baik:

```
public class Dog {  
    int Age = 13;
```

```

String Gender;

void barking() {
    System.out.println("woof..woof..");
}

Integer showAge() {
    return Age;
}

Float showWeight(int height) {
    Float weight = (float) ((height-100)-((height-100)*100));
    return weight;
}

String showGender(char value) {
    Gender = "Female";
    if(value == 'M') {
        Gender = "Male";
    }
    return Gender;
}
}

```

Berikut ini adalah cara pemanggilan method dengan atau tanpa nilai balik yang telah dibuat pada class Dog.

```

public class AppMain {
    static Dog mydog = new Dog();

    public static void main(String[] args) {
        Integer age = mydog.showAge();
        System.out.println("Age of my dog is "+age);
        int height = 40;
        Float weight = mydog.showWeight(height);
        System.out.println("Weight of my dog is "+weight);

        System.out.print("Enter the Gender of dog:");
        char gender = 'M';
        String genderName = mydog.showGender(gender);
    }
}

```

a) Acces Modifier

Java menyediakan sejumlah pengubah akses (*access modifier*) untuk

mengatur tingkat akses untuk class, variabel, metode, dan construct. Empat tingkat akses adalah:

Access Modifier	Class	Package	Outside class	Outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Contoh

```
public class ClassName {  
    // ...  
}  
  
private boolean myFlag;  
static final double weeks = 9.5;  
protected static final int BOXWIDTH = 42;  
  
public static void main(String[] arguments) {  
    // body of method  
}
```

1.2.6 Input dan Output

Untuk menginput sebuah nilai pada konsol dapat menggunakan library Scanner milik java. Scanner digunakan untuk mendapatkan masukan nilai, dan dapat ditemukan dalam paket java.util. Untuk menggunakan kelas Scanner, buat objek kelas dan gunakan salah satu metode yang tersedia dalam dokumentasi kelas Scanner. Berikut ini adalah tipe-tipe dari model Scanner:

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

Contoh Scanner:

```
import java.util.Scanner;

public class HelloWorld {
    static Dog mydog = new Dog();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Integer age = mydog.showAge();
        System.out.println("Age of my dog is "+age);

        System.out.print("Enter the Height of dog:");
        int height = scanner.nextInt();
        Float weight = mydog.showWeight(height);
        System.out.println("Weight of my dog is "+weight);

        System.out.print("Enter the Gender of dog:");
        char gender = scanner.next().charAt(0);
        String genderName = mydog.showGender(gender);
        System.out.println("Gender of my dog is "+genderName);

        scanner.close();
    }
}
```

Sedangkan untuk Output pada konsol memiliki dua tipe yaitu:

- System.out.print("...") = menampilkan dalam 1 baris tanpa break line.
- System.out.println("...") = menampilkan dalam 1 baris dengan breakline.

1.3 Latihan Pembelajaran

1. Buatlah project Java dengan nama Pembelajaran-01-Runtunan.
2. Buatlah program main JAVA untuk melakukan konversi SUHU dari Celcius ke Fahrenheit, Reamur, dan Kelvin. Dan gunakan Local Variable pada program yang dibuat.
Simpanlah dengan nama Latihan01.java pada project anda.
3. Buatlah program main JAVA untuk menghitung luas dan keliling atau volume luas bangunan:
 - Segitiga Kerucut (untuk npm digit terakhir Ganjil)
 - Layang-layang (untuk npm digit terakhir Genap)
 - Tabung (untuk npm digit terakhir Ganjil)
 - Prisma Segitiga (untuk npm digit terakhir Genap)Dimana untuk menghitung luas dan keliling luas bangunan tersebut menggunakan METHOD dengan atau tanpa nilai balik. Dan memanfaatkan Local Variable dan Class Variable dalam membangun program tersebut.
Simpanlah file dengan nama Latihan03.java pada project anda.
4. Buatlah program main JAVA dengan studi kasus seperti berikut ini:
Terdapat sebuah toko kelontong Bernama “TOKO SERBAGUNA IBIK”. Produk Promo yang sedang dijual adalah “ROTI ENAK“ dengan harga satuannya ialah Rp 6.300,-. Jika consumer membeli dengan kelipatan 3 mendapatkan diskon sebesar 5%.
Buatlah program sederhana untuk mencetak kwitansi pembelian dari toko tersebut. Berdasarkan desain dibawah ini:

TOKO SERBAGUNA IBIK			
Masukan jumlah produk yang dibeli : 7			
10 Sep 2022 (13:00)			
ITEM	QTY	HARGA	TOTAL
ROTI ENAK.	7	Rp 6.300,-	Rp 44.100
Diskon : 10%			
Sub Total : Rp 39.60,-			

Gunakanlah Scanner sebagai library untuk memasukan nilai input produk dan library Date Time untuk menampilkan waktu saat ini.

Simpanlah file dengan nama Latihan04.java pada project anda.

5. Buatlah program main JAVA untuk mengubah nilai inputan String menjadi huruf BESAR semua. Simpanlah file dengan nama Latihan05.java pada project anda.
6. Buatlah program main JAVA untuk dimana terdapat 4 buah inputan dengan Scanner.
 - Memasukan Usia
 - Memasukan Firstname
 - Memasukan Lastname
 - Memasukan NPM

Dimana output pada program ini ialah dengan cara menggunakan CONCAT.

Penggabungan antara nilai variabel. Contoh output: 26FebriFairuz2122223

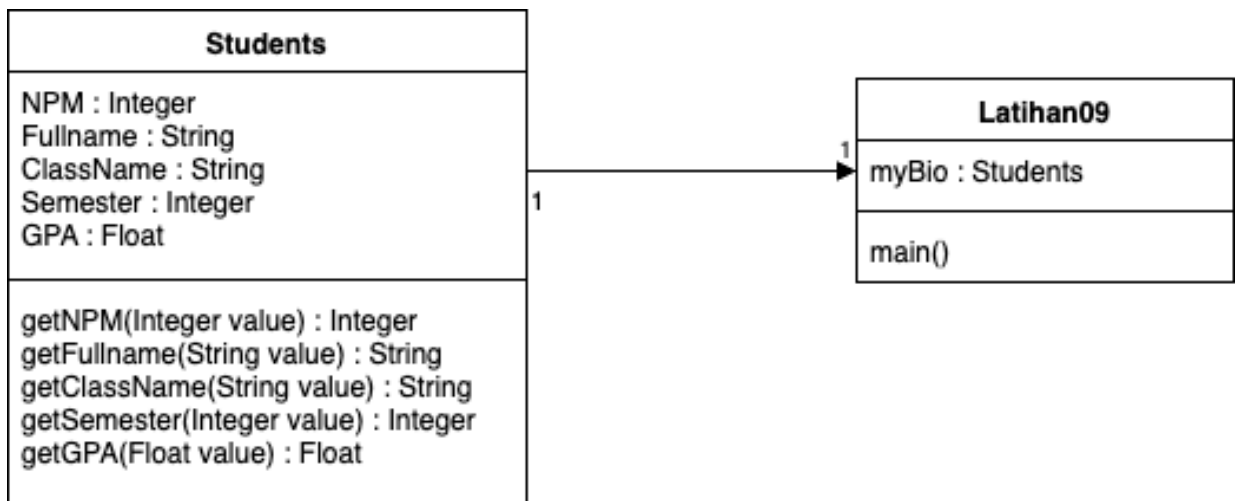
Simpanlah file dengan nama Latihan06.java pada project anda.

7. Buatlah program main JAVA untuk mengubah huruf vocal pada nama lengkap anda menjadi huruf X. Contoh: Febri Fairuz => FXbrX FXXrXz
Simpanlah file dengan nama Latihan07.java pada project anda.
8. Buatlah program main JAVA menampilkan output terdiri dari:

- NPM
- Nama Lengkap
- Nama Kelas
- Semester
- IPK

Gunakanlah local variable untuk menyimpan nilai dari isian output diatas. Simpan file dengan nama Latihan08.java pada project anda.

9. Buatlah class bernama Students. Dimana class tersebut memiliki struktur isian seperti class diagram dibawah ini:



Pada class JAVA main panggilah 5 buah method yang ada di class Students, menggunakan Scanner untuk memasukan nilai value parameter.

Simpanlah dengan nama file Latihan09.java pada project anda.

10. Buatlah program main JAVA berdasarkan soal nomor 4, dengan menambahkan isian berupa nama MEMBER dimana inputan nama member ini diambil dari class Students pada soal nomor 9. Dan tampilkan hasil inputan member di bawah SubTotal dengan label “Member Name : Nama anda”.

Simpanlah dengan nama file Latihan10.java pada project anda.