LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK



Laporan Untuk Melengkapi Tugas Terstruktur Mata Kuliah Pemrograman Berorientasi Object

Dosen Pengajar :

M. Imron Rosyadi, MM

Disusun oleh :

TEKNIK INFORMATIKA-3A

Dhyan Putri Fidanti (201869040022)

**UNIVERSITAS YUDHARTA** **PASURUAN**

Jl.Yudharta No.7 (Pondok Pesantren Ngalah)

Sengonagung Purwosari Pasuruan Telp.(0343)611186

**2019**

**KATA PENGANTAR**

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, atas segala limpahan rahmat dan karunia-Nya kepada kita semua, sehingga saya dapat menyelesaikan laporan praktikum ini.

Saya menyadari bahwa di dalam pembuatan laporan ini berkat bantuan dan tuntunan Tuhan Yang Maha Esa dan tidak lepas dari bantuan berbagai pihak, untuk itu dalam kesempatan ini saya menghaturkan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang membantu dalam pembuatan laporan ini.

Saya menyadari bahwa dalam proses penulisan laporan ini masih dari jauh dari kesempurnaan baik materi maupun cara penulisannya. Namun demikian, saya telah berupaya dengan segala kemampuan dan pengetahuan yang dimiliki sehingga dapat selesai dengan baik dan oleh karenanya, saya dengan rendah hati menerima masukan, saran dan usul guna penyempurnaan laporan ini.

Akhirnya saya berharap semoga makalah ini dapat bermanfaat bagi seluruh pembaca.

Pasuruan, 5 Desember 2020  
Penyusun,

Dhyan Putri Fidanti

**DAFTAR ISI**

**HALAMAN SAMPUL i**

**KATA PENGANTAR ii**

**DAFTAR ISI iii**

**BAB I. PENDAHULUAN 1**

1.1 Latar Belakang 1

1.2 Tujuan 2

1.3 Manfaat 2

**BAB II. PEMBAHAAN 3**

2.1 Landasan Teori 3

2.1.1 Java OOP 3

2.1.2 Java Class&Object 3

2.1.3 Java Class Attribute 4

2.1.4 Java Class Method 4

2.1.5 Java Constructor 5

2.1.6 Java Modifire 5

2.1.7 Java Encapsulation 6

2.1.8 Java API 6

2.1.9 Java Inherintance 6

2.1.10 Java Polymorphism 7

2.1.11 Java Iner Class 7

2.1.12 Java Interface 7

2.1.13 Java Abstraction 7

2.1.14 Java Enums 8

2.1.15 Java User Input 8

2.1.16 Java Date 8

2.1.17 Java Array List 8

2.1.18 Java Hash Map 9

2.1.19 Java Wraper Class 9

2.1.20 Java Exception 10

**BAB III. PRAKTIKUM 11**

3.1 Praktikum 11

3.1.1 Java Constructor 11

3.1.2 Java Modifire 14

3.1.3 Java Encapsulation 16

3.1.4 Java API 19

3.1.5 Java Inherintance 22

3.1.6 Java Polymorphism 24

3.1.7 Java Iner Class 27

3.1.8 Java Interface 31

3.1.9 Java Abstraction 34

3.1.10 Java Enums 36

3.1.11 Java User Input 41

3.1.12 Java Date 45

3.1.13 Java Array List 47

3.1.14 Java Hash Map 49

3.1.15 Java Wraper Class 51

3.1.16 Java Exception 52

**BAB IV. PENUTUP 55**

4.1. Kesimpulan 55

4.2. Saran 55

**DAFTAR PUSTAKA 56**

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Pemrograman Berorientasi Objek (*Object Oriented Programming* atau OOP) merupakan paradigm pemrograman yang berorientasikan kepada objek. Objek adalah struktur data yang terdiri dari bidang data dan metode bersama dengan interaksi mereka untu merancang aplikasi dan program computer. Semua data dan fungsi di dalam paradigm ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Pada jaman sekarang, banyak bahasa pemrograman yang mendukung OOP.

OOP adalah paradigm pemrograman yang cukup mendominasi saat ini, karena mampu memberikan solusi kaidah pemrograman modern. Meskipun demikian, bukan berarti bahwa pemrogaman procedural sudah tidak layak lagi. OOP diciptakan karena dirasakan masih adanya keterbatasan pada bahasa pemrograman tradisional. Konsep OOP sendiri adalah semua pemecahan masalah dibagi kedalam objek. Dalam OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai objek. Pembahasan mengenai orientasi objek tidak akan terlepas dari konsep objek seperti *inheritance* atau pewarisan, *encapsulation* atau pembungkusan, dan *polymorphism* atau kebanyakrupaan. Konsep-konsep ini merupakan fundamental dalam orientasi objek yang perlu dipahami serta digunakan dengan baik, dan menghindari penggunaanya yang tidak tepat.

OOP lebih mudah dipelajari bagi pemula dibandingkan dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

* 1. **Tujuan**

1. Untuk melengkapi nilai tugas mata kuliah Pemrograman Berorientasi Objek.
2. Mampu membuat program java pada Netbeans berdasarkan aturan OOP.
3. Untuk mengkaji kembali pengetahuan yang telah didapatkan pada kuliah Pemrograman Berorientasi Objek.
   1. **Manfaat**
4. Dapat melengkapi nilai tugas mata kuliah Pemrograman Berorientasi Objek.
5. Dapat membuat program Java pada Netbeans berdasarkan ilmu aturan OOP.
6. Dapat mengkaji kembali pengetahuan yang telah didapatkan pada kuliah Pemrograman Berorientasi Objek.

**BAB II**

**PEMBAHASAN**

* 1. **Landasan Teori**
     1. Java OOP

OOP adalah singkatan dari **Object-Oriented Programming**. pemrograman prosedural adalah tentang menulis prosedur atau metode yang melakukan operasi pada data, sedangkan pemrograman berorientasi objek adalah tentang menciptakan benda-benda yang mengandung data dan metode. pemrograman berorientasi objek memiliki beberapa keunggulan dibandingkan pemrograman prosedural:

* OOP lebih cepat dan lebih mudah untuk mengeksekusi
* OOP menyediakan struktur yang jelas untuk program
* OOP membantu untuk menjaga kode Java DRY "Don’t Try Yourself", dan membuat kode lebih mudah untuk mempertahankan, memodifikasi dan debug
* OOP memungkinkan untuk membuat aplikasi yang dapat digunakan kembali penuh dengan kode kurang dan waktu pengembangan yang lebih pendek
  + 1. Java Class dan Object

Class merupakan cetak biru (blue print) dari objek atau dengan kata lain sebuah class menggambarkan ciri – ciri objek secara umum.Sebagai contoh Honda Beat, Yamaha R6, dan Suzuki Satria merupakan objek dari class sepeda motor.

Kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu system / konsep yang memiliki sifat, karakter dan fungsi. Contoh :

|  |
| --- |
| public class MyClass {  int x = 5;  public static void main(String[] args) {  MyClass **myObj** = new MyClass();  System.out.println(myObj.x);  } } |

* + 1. Java Class Attributes

Segala sesuatu di Jawa dikaitkan dengan **Class** dan **Object**, bersama dengan **Attributes** dan **Method**. Sebagai contoh: dalam kehidupan nyata, mobil adalah obyek. Mobil memiliki **atribut** , seperti berat badan dan warna, dan **metode** , seperti drive dan rem. Contoh : Class MyClass dengan Attributes x dan y.

|  |
| --- |
| public class MyClass {  int x = 5;  int y = 3;  } |

* + 1. Java Class Method

Method adalah kumpulan program yang mempunyai nama. Method merupakan sarana bagi programmer untuk memecah program menjadi bagian-bagian yang kecil agar jadi lebih kompleks sehingga dapat di gunakan berulang-ulang. Method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. Contoh, pada object Mobil : terdapat method drive dan rem. Contoh :

|  |
| --- |
| public class MyClass {  static void myMethod() {  System.out.println("Hello World!");  }  public static void main(String[] args) {  myMethod();  }  }  // Outputs "Hello World!" |

* + 1. Java Constructors

Construktor adalah method yang secara otomatis dipanggil/dijalankan pada saat sebuah class diinstransi. Jika dalam sebuah class tidak terdapat construktor maka secara otomatis java akan membuatkan sebuah default construktor. Nama construktor harus sama dengan nama class dan tidak boleh memiliki tipe return value. Sama halnya dengan method, construktor dapat memiliki satu atau banyak parameter maupun tanpa parameter.

* + 1. Java Modifiers

Access Modifier adalah sebuah “hak akses” yang diberikan kepada sebuah variabel/method/class dengan tujuan untuk menjaga integritas dari data tersebut ketika ingin diakses object lain. Hak akses sendiri diberikan oleh pembuat program sebagai salah satu bentuk implementasi dari OOP itu sendiri. Dengan adanya Access Modifier, kita dapat membatasi resource-resource mana saja yang dapat diakses oleh object tertentu, turunannya, ataupun oleh method tertentu. Di bahasa JAVA sendiri ada 4 jenis access modifier yang dapat digunakan :

1. Public

Access modifier public mempunyai hak akses paling luas dibanding yang lainnya. Karena aksesnya sangat luas, maka access modifier ini biasanya digunakan untuk method setter getter sesuai konsep OOP.

1. Prottected

Access modifier protected biasanya digunakan untuk mewariskan variabel yang ada di super class terhadap child class.

1. Non Access Modifire

Sesuai namanya, hak akses yang satu ini tidak perlu dituliskan di method/variabelnya. Dengan hak akses ini, variabel/method dapat diakses dari class lain asalkan masih dalam satu package yang sama.

1. Private

Access modifier private bersifat tertutup. Sesuai dengan konsep OOP Encapsulation, maka setiap variabel wajib untuk dilindungi hak aksesnya secara langsung dari luar. Oleh karena itu, variabel diberikan hak akses private dan untuk melakukan pengaksesan/perubahan data digunakan setter getter.

* + 1. Java Encapsulation

Enkapsulasi (encapsulation) adalah untuk memastikan bahwa data “sensitive” disembunyikan dari pengguna. Artinya encapsulation pada java digunakan untuk membungkus class dan menjaga apa saja yang ada dalam class baik itu method maupun attribute , agar tidak dapat di akses oleh kelas lain. Untuk mencapai ini maka perlu diperhatikan hal-hal berikut ini :

* Mendeklarasikan variabel / attribute kelas sebagai Private
* Memberikan metode public **Get** dan **Set**  untuk mengakses dan memperbarui nilai suatu Private variabel

**Get** dan **Set**

Variabel Private hanya dapat diakses didalam kelas yang sama (kelas luar tidak memiliki akses). Namun, memungkinkan mengaksesnya jika diberi tambahan metode Public **Get** dan **Set.**

Get adalah metode mendapatkan nilai variabel , dan Set metode mengatur nilai Sintaks untuk keduanya adalah bawa dimulai dengan salah satu get atau set , diikuti oleh nama variabel , dengan huruf pertama menggunakan huruf kapital.

* + 1. Java API

API Java adalah perpustakaan kelas prewritten, yang bebas untuk digunakan, termasuk dalam Java Development Environment. Dalam library API berisi komponen untuk mengelola input, pemrograman database, dan lebih banyak lagi.

Library ini dibagi menjadi **paket** dan **kelas** . Berarti Anda baik dapat mengimpor satu kelas (bersama dengan metode dan atribut), atau seluruh paket yang berisi semua kelas yang milik paket tertentu.

* + 1. Java Inheritance

Konsep pewarisan (inheritance) : Proses pembentukan kelas baru dari kelas yang sudah ada menjadi kelas Superclass dan kelas Subclasses. Proses pembentukan kelas menghasilkan hierarki kelas yaitu puncak hierarki yang disebut **kelas abstrak**.

Dengan inheritance, class yang baru (subclass) akan mirip dengan class yang lama (superclass) namun memiliki karakteristik yang baru. Dalam java, subclass hanya bisa memiliki satu superclass (single inheritance) sedangkan superclass bisa memiliki satu subclass atau lebih.

* + 1. Java Polymorphism

Polymorphism merupakan konsep sederhana dalam bahasa pemrograman berorientasi obyek yang berarti kemampuan sebuah obyek untuk menghasilkan aksi yang berbeda, bila method yang sama dipanggil, dimana aksi method yang dikerjakan tergantung dari tipe obyeknya.

Polymorphism merupakan suatu konsep yang menyatakan sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku berbeda. Jadi dapat disimpulkan bahwa polymorphism adalah satu objek yang memiliki banyak bentuk.

* + 1. Java Inner Class

Inner class adalah kelas yang disisipkan didalam kelas lain. Fungsi kelas sisipan ini adalah mendukung suatu proses yang dijalankan  oleh kelas utamanya. Inner class bersifat tersarang terhadap kelas-kelas utamnya.

* + 1. Java Interface

Interface adalah sekumpulan konstanta dana tau deklarasi mothod tanpa munyertakan / menuliskan body methodnya. Interface bisa digunakan untuk mendeklarasikan koleksi method dan konstanta yang dapat digunakan oleh satu atau lebih class.

* + 1. Java Abstaction

A**bstraction** adalah proses menyembunyikan rincian tertentu dan hanya menampilkan informasi penting kepada pengguna.

Abstraksi dapat dicapai dengan baik **kelas abstrak atau inerface.** kata kunci Abtraksi adalah pengubah non-akses, digunakan untuk kelas dan metode :

* **Kelas abstrak:** adalah kelas terbatas yang tidak dapat digunakan untuk membuat objek (untuk mengaksesnya, harus diwariskan dari kelas lain).
* **Metode abstrak:** hanya dapat digunakan di kelas abstrak, dan tidak memiliki tubuh. Tubuh disediakan oleh subclass (diwarisi dari).
  + 1. Java Enums

Enum (enumeration)adalah salah satu kelas library di J2SE (versi 5), yang dapat kita gunakan untuk membuat kumpulan variable yang telah didefinisikan menjadi tipe data **konstanta** (variabel tidak bisa diubah, seperti final variabel).setiap objek baru yang didefinisikan berdasarkan tipe data Enum, objek tersebut dapat mengakses atribut atau method (Enum), lalu mengaplikasikan nilai atau value yang ada di dalamnya. Enum bisa juga disebut array, namun dalam bentuk konstanta, dan juga bertipe class.

* + 1. Java User Input

Kelas **Scanner** digunakan untuk mendapatkan input pengguna, dan ditemukan dalam paket java.util . Untuk menggunakan kelas **Scanner ,** buat objek kelas dan gunakan salah satu metode yang tersedia yang ditemukan dalam dokumentasi kelas **Scanner** .

* + 1. Java Date

Java tidak memiliki kelas tanggal bawaan , tetapi dapat mengimpor paket java.time agar berfungsi dengan API tanggal dan waktu. Paket terdiri atas beberapa kelas tanggal dan waktu .Sebagai contoh :

|  |  |
| --- | --- |
| Class | Deskripsi |
| Local Date | Menjelaskan tentang tanggal (tahun,bulan,hari  (yyyy-MM-dd)) |
| Local Time | Menjelaskan tentang waktu (jam,menit,detik dan seperinu detik (HH-mm-ss-zzz)) |
| Local Date Time | Menjelaskan tentang tanggal dan waktu (yyyy-MM-dd-HH-mm-ss.zzz) |
| Date Time Formatter | Menampilkan dan menguraikan objek tanggal dan waktu |

* + 1. Java Arraylist

Kelas ArrayList adalah [array](https://translate.googleusercontent.com/translate_c?depth=1&hl=id&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://www.w3schools.com/java/java_arrays.asp&xid=25657,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271,15700283&usg=ALkJrhiTH6rPlA-QuvlB_z7aByHZNOCRkQ) resizable, yang dapat ditemukan dalam paket java.util. Perbedaan antara array bawaan dan ArrayList di Java, adalah bahwa ukuran array tidak dapat dimodifikasi (jika Anda ingin menambah atau menghapus elemen ke / dari array, Anda harus membuat yang baru). Sementara elemen dapat ditambahkan dan dihapus dari ArrayList kapan pun Anda mau.

* + 1. Java HashMap

Anda pelajari dari bab sebelumnya, bahwa Array menyimpan barang-barang sebagai koleksi memerintahkan, dan Anda memiliki akses mereka dengan nomor indeks ( int tipe). Sebuah HashMap Namun, toko item dalam " **key** / **value** " pasangan, dan Anda dapat mengaksesnya dengan indeks jenis lain (misalnya String). HashMap mempunai banyak metode :

1. Add Item : Untuk menambah item, menggunakan metode put().
2. Access an Item : Untuk mengakses nilai item, menggunakan metode get().
3. Remove an Item : Untuk menghapus item, mengunakan metode remove(). Untuk menghapus semua item menggunakan metode clear().
4. HashMap Size : Unuk mengetahui berapa banyak item yang ada, menggunakan metode size.
5. Loop Through a HashMap : Gunakan keyset() Jika hanya ingin key, dan gunakan values() jika hanya inin value.
6. Other Type : Kunci dan nilai-nilai dalam HashMap sebenarnya objek. Dalam contoh di atas, kami menggunakan objek tipe "String". Ingat bahwa String di Jawa adalah obyek (bukan tipe primitif). Untuk menggunakan jenis lain, seperti int, Anda harus menentukan setara [kelas wrapper](https://www.w3schools.com/java/java_wrapper_classes.asp) : Integer. Untuk jenis primitif, gunakan: Boolean untuk boolean, Character untuk char,  Double untuk ganda, dll.
   * 1. JavaWrapper **Class**

Di dalam bahasa pemrograman Java, *Wrapper Class* (Kelas Pembungkus) adalah suatu mekanisme yang digunakan untuk mengconvert atau mengubah suatu nilai yang di definisikan, dari tipe data **primitif**menjadi sebuah nilai dengan tipe data **referensi**(Objek). Selain itu, Wrapper Class mendukung method dari tipe data primitif, contohnya seperti Boolean, Character, Integer, Long, Float, dsb.

Pada bahasa pemrograman Java, terdapat 2 jenis tipe data, yaitu primitif dan referensi (objek), Tipe Data *Primitif*adalah tipe data yang tidak memiliki method, hanya memiliki data saja. Tipe data ini bukanlah sebuah object, sedangkan Tipe Data *Referensi*adalah tipe data yang digunakan untuk memegang referensi dari sebuah object ( instance dari class). Dan berikut ini merupakan table yang menjelaskan macam-macam tipe data primitif dan referensi pada Java:

|  |  |
| --- | --- |
| **Tipe Data Primitif** | **Tipe Data Referensi** |
| char | Character |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| boolean | Boolean |
| float | Float |
| double | Double |

metode berikut ini digunakan untuk mendapatkan nilai yang terkait dengan objek pembungkus yang sesuai:

intValue() , byteValue() , shortValue() , longValue() , floatValue() , doubleValue() , charValue() , booleanValue() .

* + 1. Java Exeption

Ketika mengeksekusi kode Java, kesalahan yang berbeda dapat terjadi: coding kesalahan yang dibuat oleh programmer, kesalahan karena masukan yang salah, atau hal-hal tidak terduga lainnya.

Ketika terjadi kesalahan, Jawa biasanya akan berhenti dan menghasilkan pesan kesalahan. Istilah teknis untuk ini adalah: Java akan melemparkan sebuah **pengecualian** (melempar kesalahan).

**BAB III**

**PRAKTIKUM**

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Constructor

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

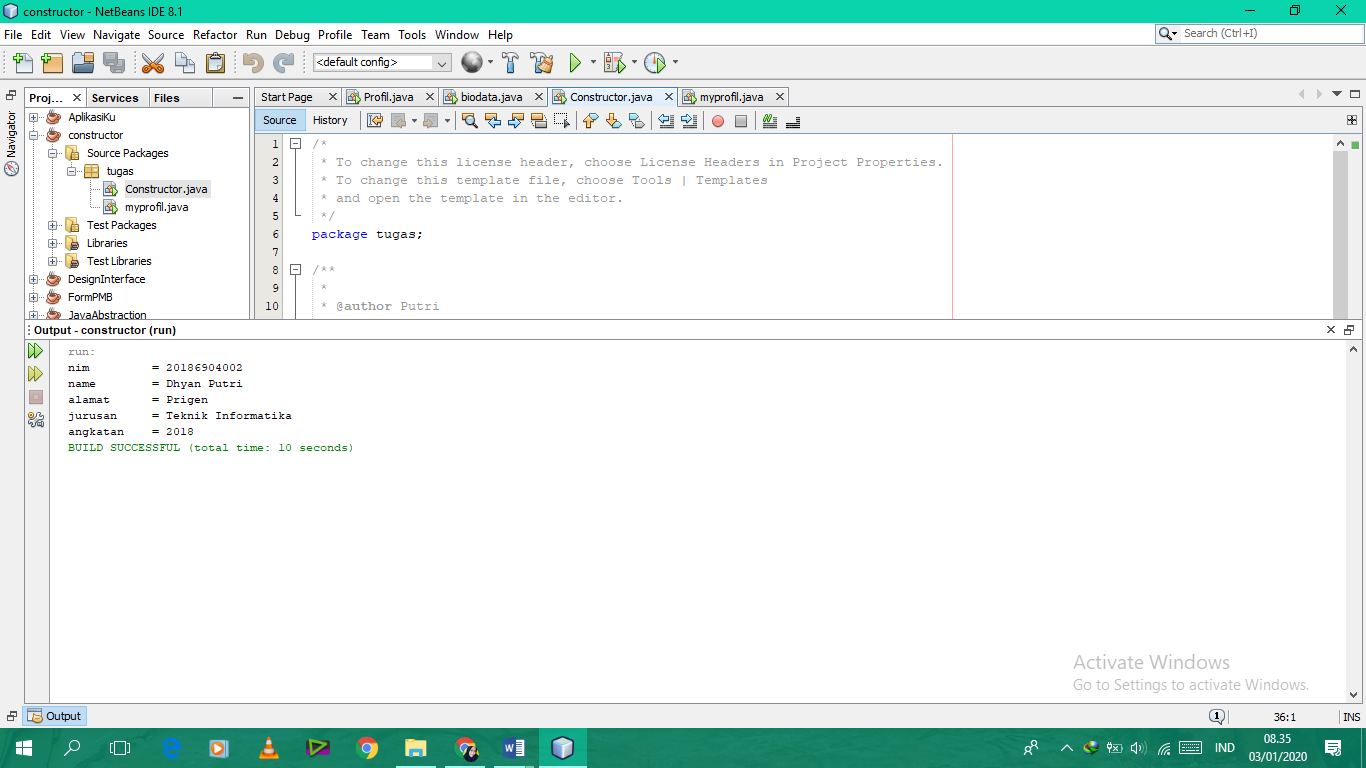
**Script Program :**

|  |
| --- |
| package tugas;  /\*\*  \*  \* @author Putri  \*/  public class Constructor {  long modelnim;  String modelname;  String modelalamat;  String modeljurusan;  int modelangkatan;    public Constructor() {  modelnim = 20186904002L;  modelname = "Dhyan Putri";  modelalamat = "Prigen";  modeljurusan = "Teknik Informatika";  modelangkatan = 2018;  }  public static void main(String[] args) {  Constructor profil = new Constructor();  System.out.println("nim = "+profil.modelnim);  System.out.println("name = "+profil.modelname);  System.out.println("alamat = "+profil.modelalamat);  System.out.println("jurusan = "+profil.modeljurusan);  System.out.println("angkatan = "+profil.modelangkatan);  }  } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama tugas.
2. public class Constructor Merupakan pendeklarasian sebuah class dengan nama Constructor. Selanjutnya nama class ini digunakan untuk menamai file-nya dengan extension .java. { Tanda { digunakan untuk mengawali class Tes yang di dalamnya ada method main dan beberapa statement.
3. modelname = "Dhyan Putri" untuk memasukan data berupa nama.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. { Tanda { digunakan untuk mengawali method main yang di dalamnya ada beberapa statement.
6. System.out.println("name = "+profil.modelname) Berfungsi untuk mencetak NAMA ke layar dengan pindah baris.
7. } Tanda } digunakan untuk mengakhiri Method main yang di dalamnya ada beberapa statement.
8. } Tanda } digunakan untuk mengakhiri class Tes yang di dalamnya ada method main dan beberapa statement.

**Ouput Program :**



**Penjelasan Program :** Program ini digunakan untuk menampilkan data berupa biodata mahasiswa. Program ini memiliki class bernama constructor dan method bernama main.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Modifire (public)

**Bahasa Pemrograman :** JAVA

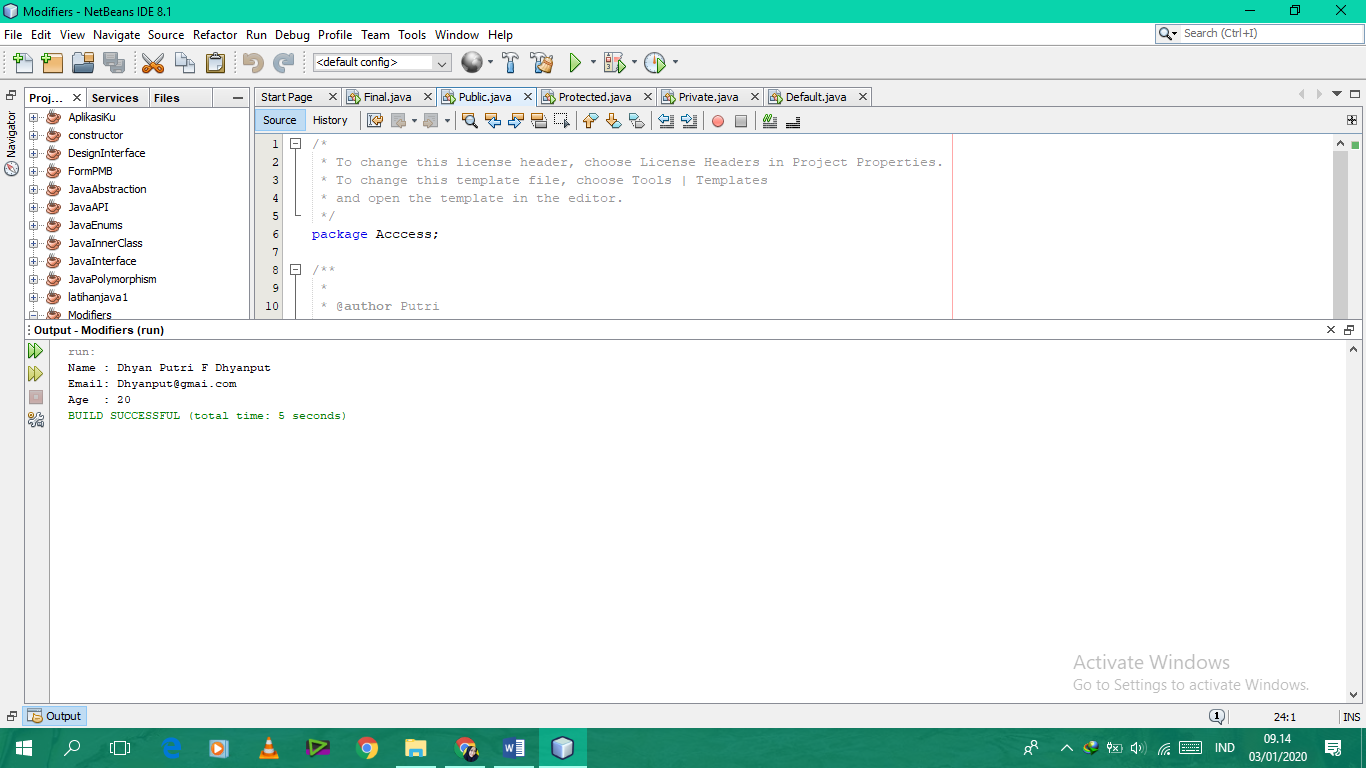
**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package Acccess;  /\*\*  \*  \* @author Putri  \*/  public class Public {  public String fname = "Dhyan Putri F";  public String lname = "Dhyanput";  public String email = "Dhyanput@gmai.com";  public int age = 20;  public static void main(String[] args) {  Public myObj = new Public();  System.out.println("Name : " + myObj.fname + " " + myObj.lname);  System.out.println("Email: " + myObj.email);  System.out.println("Age : " + myObj.age);  }  } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama Access.
2. public class Public Merupakan pendeklarasian sebuah class dengan nama Public. Selanjutnya nama class ini digunakan untuk menamai file-nya dengan extension .java.
3. { Tanda { digunakan untuk mengawali class Tes yang di dalamnya ada method main dan beberapa statement.
4. public String fname = "Dhyan Putri F" Berfungsi untuk memasukan data berupa huruf (Nama).
5. public int age = 20 integer Berfungsi untuk memasukan data berupa angka (Umur).
6. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
7. Public myObj = new Public() Digunakan untuk menambahkan objek ke dalam kelas.
8. { Tanda { digunakan untuk mengawali method main yang di dalamnya ada beberapa statement.
9. System.out.println("Name : " + myObj.fname + " " + myObj.lname) Berfungsi mencetak method Nama.
10. } Tanda } digunakan untuk mengakhiri Method main yang di dalamnya ada beberapa statement.
11. } Tanda } digunakan untuk mengakhiri class Tes yang di dalamnya ada method main dan beberapa statement.

**Ouput Program :**

**Penjelasan Program :** Modifier public akan membuat member dan class bisa di akses dari mana saja. Program ini digunakan untuk menampilkan data seseorang.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Encapsulation

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

(1)

|  |
| --- |
| package getandset;  /\*\*  \*  \* @author Putri  \*/  public class Person {  public String name;    //getter  public String getName(){  return name;  }    //setter  public void setName(String newName) {  this.name = newName;  }    } |

(2)

|  |
| --- |
| package explained;  import getandset.Person;  /\*\*  \*  \* @author Putri  \*/  public class MyClass {  public static void main(String[]args) {  Person myObj = new Person ();  myObj.setName("Dhyanput");  System.out.println(myObj.getName());  }    } |

**Keterangan Script :** (1)

1. Mendeklarasikan Package bernama getandset.
2. public class Person

public String name

membuat kelas Person , menggunakan tipe variabel String karena mengambil contoh ‘nama orang’.

1. //getter

public String getName(){

return name;

}

Metode get artinya mengembalikan nilai variabel nama.

1. //setter

public void setName(String newName) {

this.name = newName;

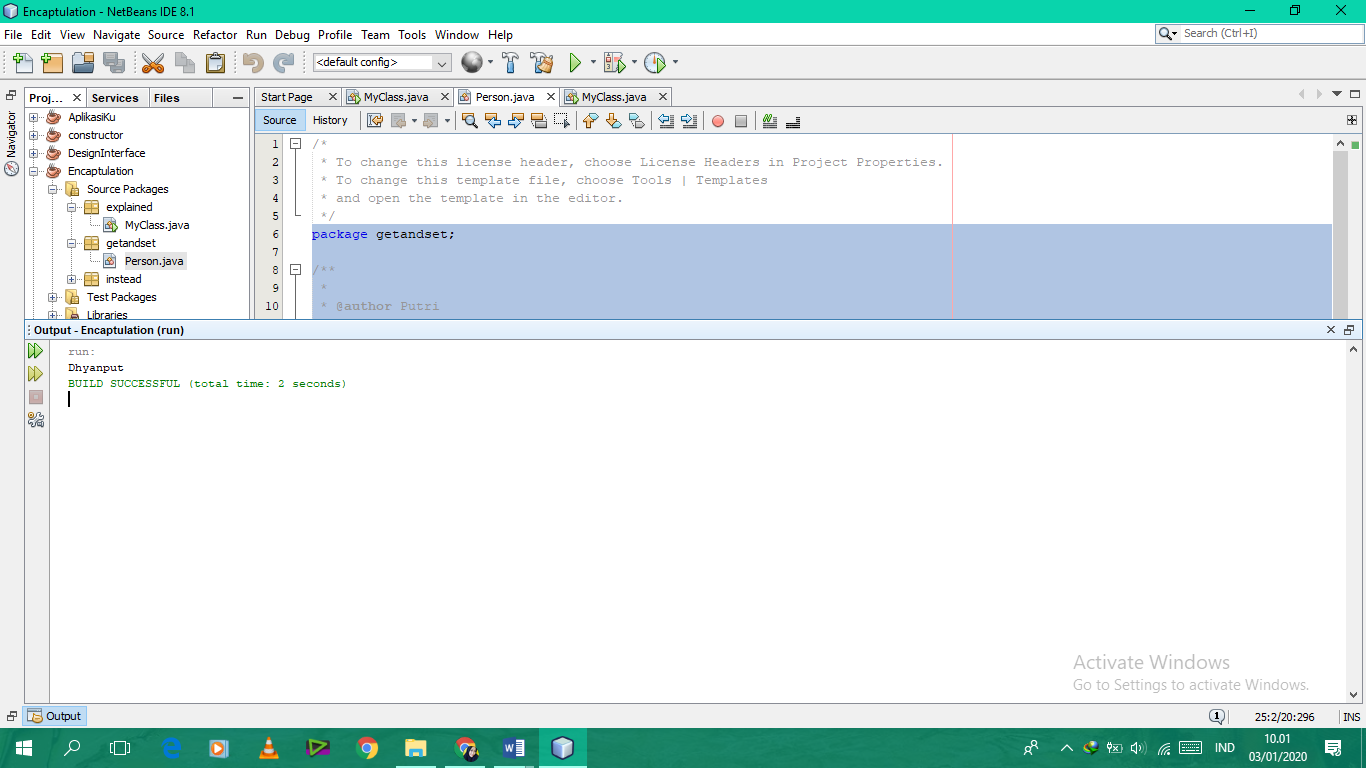
}

Metode Set artinya mengambil parameter (newName) dan mengisi data ke dalam objek ,kata this merupakan kunci yang digunakan untuk merujuk ke objek saat ini.

Variabel ‘nama’ dideklarasikan sebagai ‘public’ agar dapat mengaksesnya di luar kelas sedangkan jika kita mendeklarasikan dengan ‘private’ maka tidak dapat diakses diluar kelas.

**Keterangan Script :** (2)

1. Mendeklarasikan Package bernama explained.
2. import getandset.Person Kode ini maksudnya adalah mengimpor class dan interface yang kita butuhkan. Mengimport fungsi get dan set pada kelas ‘main’.
3. public class MyClass Menggunakan public karena akan di akses dari luar kelas.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. Person myObj = new Person () digunakan untuk Menambahkan objek ke dalam kelas.
6. myObj.setName("Dhyanput") digunakan untuk mengisi data kedalam objek.
7. System.out.println(myObj.getName())Berfungsi mencetak method Nama.

**Ouput Program :**

**Penjelasan Program :** Program di atas menggunakan metode getName() dan setName() untuk mengakses dan memperbarui variabel.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java API

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package Scanner;  import java.util.Scanner;  /\*\*  \*  \* @author Putri  \*/  public class biodata {  public static void main(String[] args){  Scanner input = new Scanner (System.in);  long nim ;  String nama;  String alamat;  String jurusan;  int tahunAngkatan;  String jenisKelamin;  System.out.print ("nama :");  nama = input.nextLine ();    System.out.print ("alamat :");  alamat = input.nextLine ();    System.out.print ("jurusan :");  jurusan = input.nextLine ();    System.out.print ("jenisKelamin :");  jenisKelamin = input.nextLine ();    System.out.print ("nim :");  nim = input.nextLong ();    System.out.print ("tahunAngkatan :");  tahunAngkatan = input.nextInt ();    System.out.println("nim "+ nim);  System.out.println("nama "+ nama);  System.out.println("alamat "+ alamat);  System.out.println("jurusan "+ jurusan);  System.out.println("tahunAngkatan "+ tahunAngkatan);  System.out.println("jenisKelamin "+ jenisKelamin);  }  } |

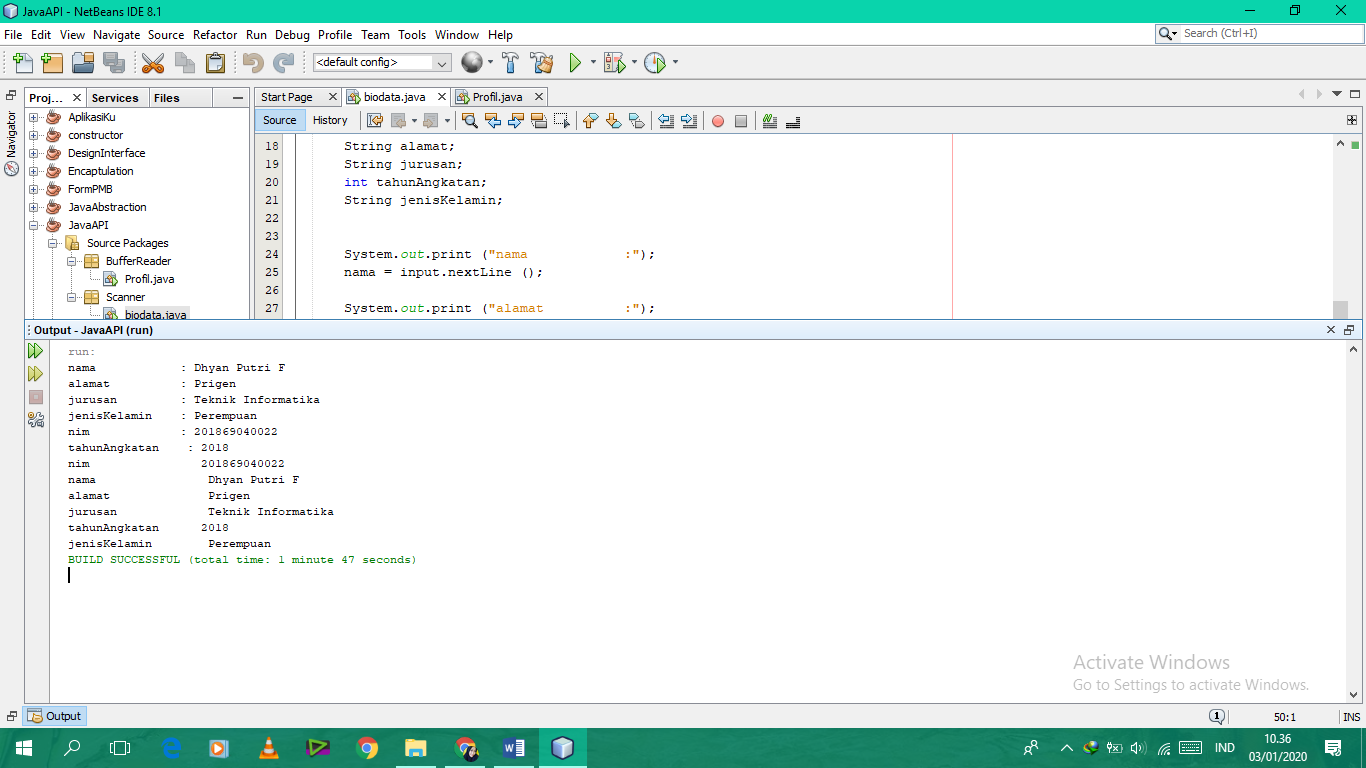
**Keterangan Script :**

1. Mendeklarasikan Package bernama Scanner.
2. import java.util.Scanner Digunakan untuk mengimpor utilitas java.
3. public class biodata Merupakan pendeklarasian sebuah class dengan nama biodata. Selanjutnya nama class ini digunakan untuk menamai file-nya dengan extension .java.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. Scanner input = new Scanner (System.in) Digunakan untuk menginput data scanner.
6. long nim Digunakan untuk memasukkan kombinasi angka yang agak panjang.
7. String nama Digunakan untuk memasukkan data berupa huruf (nama).
8. System.out.print ("nama :")

nama = input.nextLine ()

Berfungsi untuk mencetak method nama.

**Output Program :**



**Penjelasan Program :** Program ini digunakan untuk mengisi data berupa biodata mahasiswa. Program ini memiliki class bernama biodata dan method bernama main.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Inheritance

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package Bangun\_datar;  import java.util.\*;  /\*\*  \*  \* @author Putri  \*/  public class Persegi extends Bangundatar{  public void Persegi(){  double sb,lb,kb;  Scanner c = new Scanner(System.in);  System.out.println("Persegi");  System.out.println("=============");  System.out.println("Masukkan sisi : ");  sb = c.nextDouble();  lb = sb\*sb;  kb = 4\*sb;  System.out.println("Luas Bujur Sangkar adalah : "+lb);  System.out.println("Keliling Bujur Sangkar adalah : "+kb);  }  } |

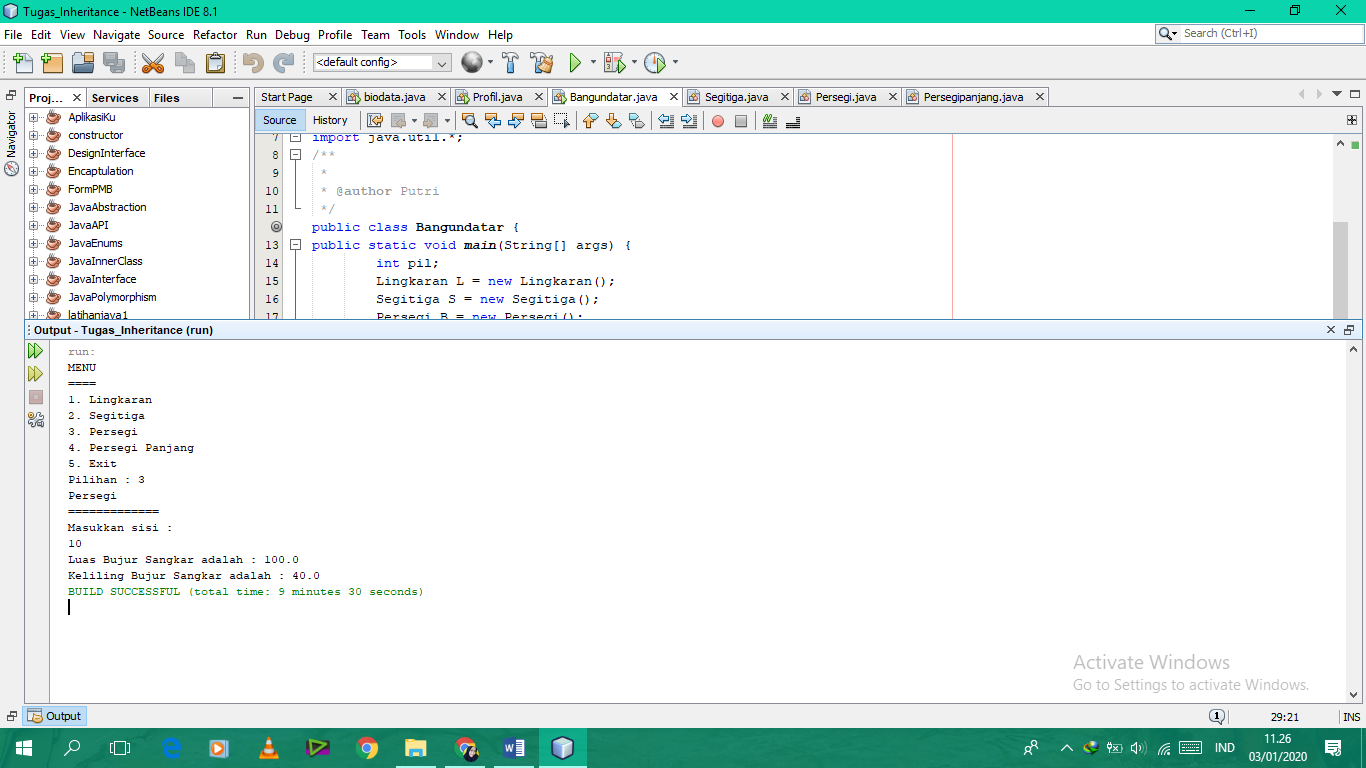
**Keterangan Script :**

1. Mendeklarasikan Package bernama Bangun\_datar.
2. import java.util.\*; Digunakan untuk mengimpor utilitas java.
3. public class Persegi extends Bangundatar Untuk mendeklarasikan kelas Persegi perluasan dari kelas Bangundatar.
4. public void Persegi() Untuk mendeklarasikan metode Persegi bertipe void dan bersifat publik.
5. double sb,lb,kb Untuk mendeklarasikan nilai (sb,lb,kb).
6. Scanner c = new Scanner(System.in) Untuk memindai dan memasukkan kedalam sistem.
7. System.out.println("Persegi") Untuk mencetak nama "Persegi".
8. sb = c.nextDouble() Untuk memanggil deklarasi nilai (sb,lb,kb).
9. lb = sb\*sb

kb = 4\*sb

Untuk memasukkan rumus persegi kedalam program.

1. System.out.println("Luas Bujur Sangkar adalah : "+lb) Untuk mencetak kata "Luas Bujur Sangkar adalah : ". Sekaligus untuk mencetak rumus luas.

**Output Program :**

**Penjelasan Program :** Program ini digunakan untuk mengisi data menghitung luas serta keliling persegi. Program ini memiliki class bernama persegi dan method tipe void.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Polymorphism

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

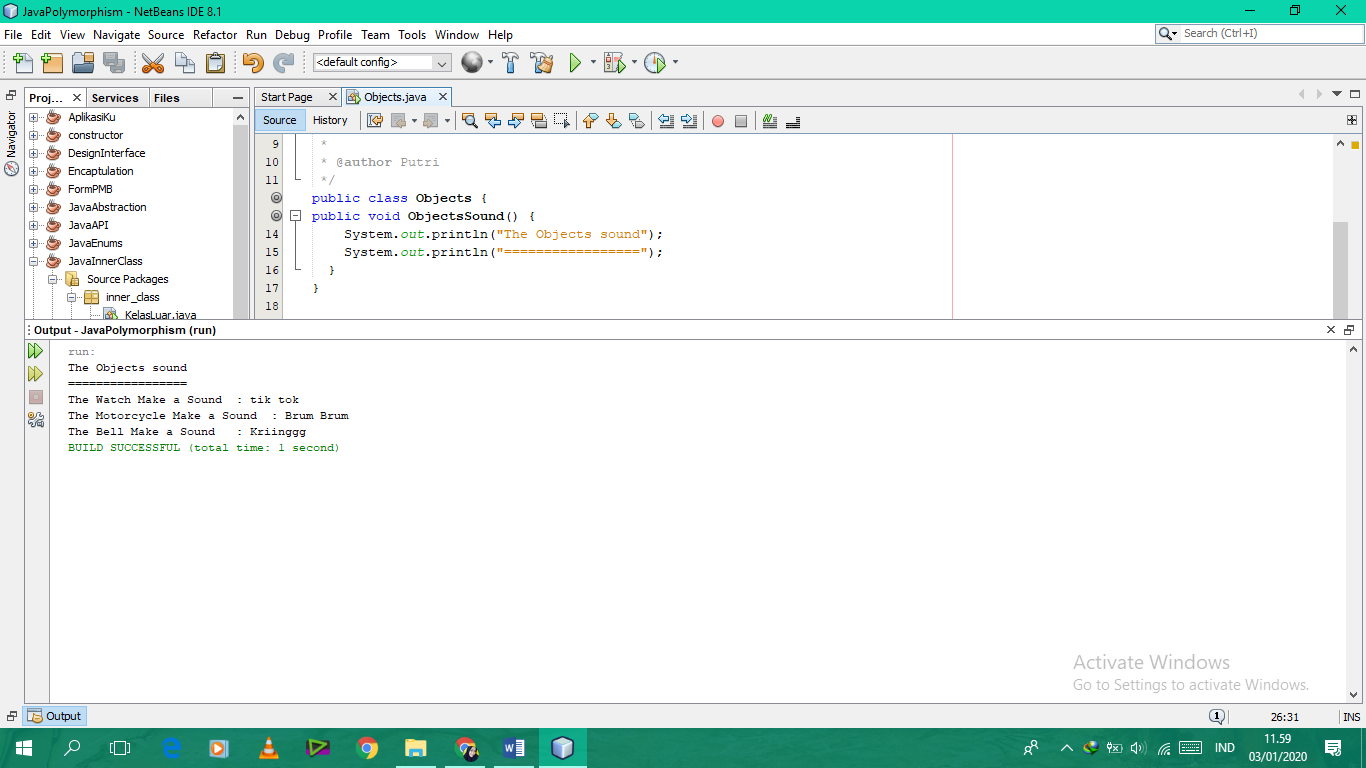
**Script Program :**

|  |
| --- |
| package Polymorphism2;  /\*\*  \*  \* @author Putri  \*/  public class Objects {  public void ObjectsSound() {  System.out.println("The Objects makes a sound");  }  }  class Watch extends Objects {  public void ObjectsSound() {  System.out.println("The Watch Make a Sound : tik tok");  }  }  class Motorcycle extends Objects {  public void ObjectsSound() {  System.out.println("The Watch Make a Sound : Brum Brum");  }  }  class Bell extends Objects {  public void ObjectsSound() {  System.out.println("The Bell Make a Sound : Kriinggg");  }  }  class MyMainClass {  public static void main(String[] args) {  Objects myObjects = new Objects();  Objects myWatch = new Watch();  Objects myMotorcycle = new Motorcycle();  Objects myBell = new Bell();    myObjects.ObjectsSound();  myWatch.ObjectsSound();  myMotorcycle.ObjectsSound();  myBell.ObjectsSound();  }  } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama polymorphism2.
2. public class Objects Merupakan pendeklarasian sebuah class dengan nama Objects. Selanjutnya nama class ini digunakan untuk menamai file-nya dengan extension .java.
3. public void ObjectsSound() Untuk mendeklarasikan suara benda.
4. System.out.println("The Bell Make a Sound : Kriinggg") Untuk mencetak kata "The Bell Make a Sound : Kriinggg".
5. class Watch extends Objects Untuk mendeklarasikan kelas Watch perluasan dari kelas Objects.
6. public void ObjectsSound() Untuk mendeklarasikan suara benda (ObjectsSound).
7. System.out.println("The Watch Make a Sound : tik tok") Untuk mencetak kata "The Watch Make a Sound : tik tok".
8. class MyMainClass Mendeklarasikan kelas MyMainClass.
9. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
10. Objects myObjects = new Objects() Mendeklarasikan variabel Objects.
11. Objects myWatch = new Watch() Mendeklarasikan variable Watch.
12. myObjects.ObjectsSound() Mendeklarasikan Sound dari variable Objects.
13. myWatch.ObjectsSound() Mendeklarasikan Sound dari variable Watch.

**Output Program :**



**Penjelasan Program :** Program ini digunakan untuk menampilkan suara benda-benda. Program ini memiliki class bernama Objects dan method tipe void.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Inner Class

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package study\_kasus;  /\*\*  \*  \* @author Putri  \*/  public class KelasLuar {  private class Order{  private String merk = "CHANNEL";  private int Harga = 500000;  private void jalankan(){  System.out.println("Merk Baju: "+merk);  System.out.println("Harga Baju: "+Harga);  }  }  private class Pemesan{  private String nama = "SUZY";  private int umur = 24;  private void identitas(){  System.out.println("Nama Saya: "+nama);  System.out.println("Usia Saya: "+umur);  }  }  public static void main(String[] args){  study\_kasus.KelasLuar outerclass = new study\_kasus.KelasLuar();  study\_kasus.KelasLuar.Order data1 = outerclass.new Order();  study\_kasus.KelasLuar.Pemesan data2 = outerclass.new Pemesan();  System.out.println("===== DATA DARI CLASS ORDER ========");  data1.jalankan();  System.out.println("===== DATA DARI CLASS PENGGUNA =====");  data2.identitas();  }    } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama study\_kasus.
2. public class KelasLuar Merupakan pendeklarasian sebuah class dengan nama KelasLuar. Selanjutnya nama class ini digunakan untuk menamai file-nya dengan extension .java.
3. private class Order Mendeklarasikan kelas Order yang bersifat private.
4. private String merk = "CHANNEL" Mendeklarasikan variabel merk bertipe string dan bersifat private.
5. private int Harga = 500000 Mendeklarasikan variabel harga bertipe integer dan bersifat private.
6. private void jalankan() Mendeklarasikan objek Jalankan bertipe void dan bersifat private.
7. System.out.println("Merk Baju: "+merk) Untuk mencetak kata "Merk Baju: ".
8. private class Pemesan Mendeklarasikan kelas pemesan yang bersifat private.
9. private String nama = "SUZY" Mendeklarasikan variabel merk bertipe string dan bersifat private.
10. private int umur = 24 Mendeklarasikan variabel harga bertipe integer dan bersifat private.
11. private void identitas()Mendeklarasikan objek Jalankan bertipe void dan bersifat private.
12. System.out.println("Nama Saya: "+nama) Untuk mencetak kata "Nama Saya: ".
13. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
14. study\_kasus.KelasLuar.Order data1 = outerclass.new Order() Mendeklarasikan Order dari variable KelasLuar.
15. System.out.println("===== DATA DARI CLASS ORDER ========")

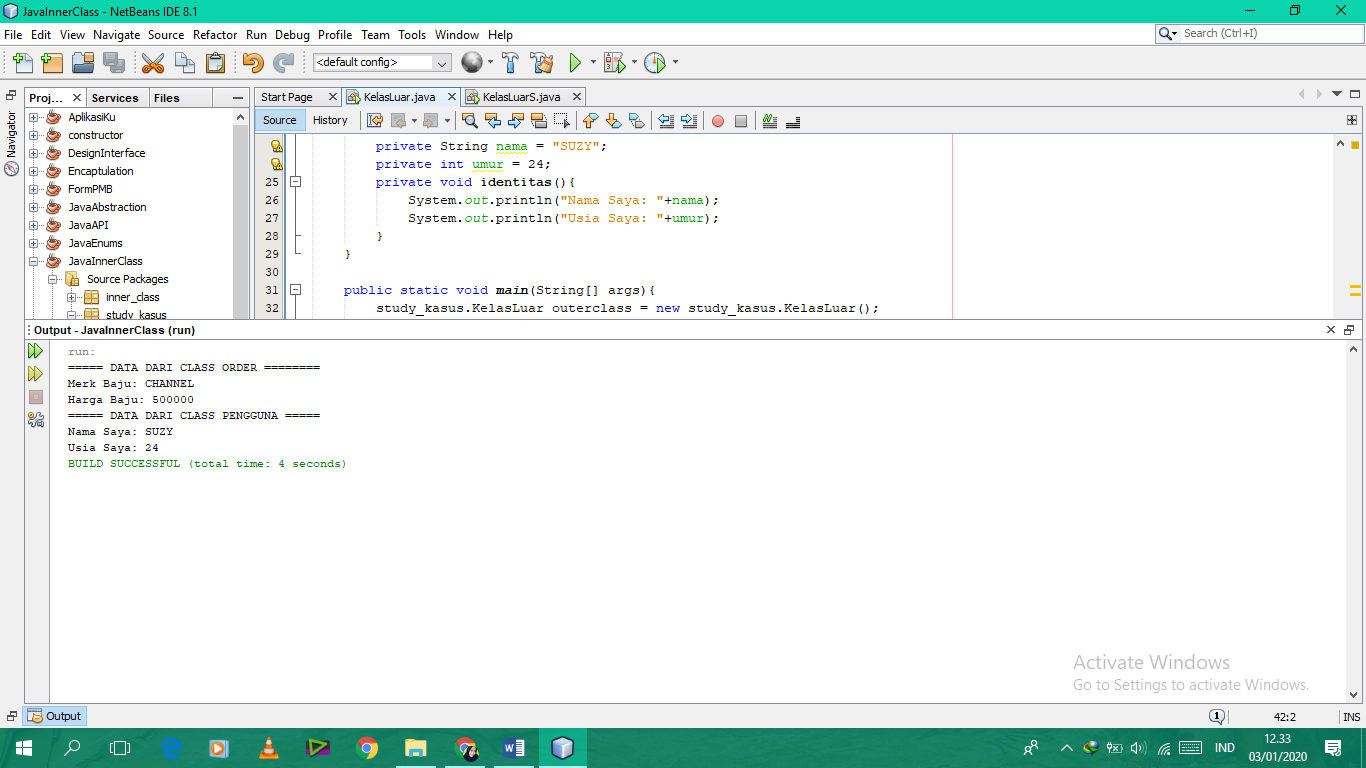
data1.jalankan()

Untuk mencetak method diatas

1. System.out.println("===== DATA DARI CLASS PENGGUNA =====")

data2.identitas()

Untuk mencetak method diatas

**Output Program :**

**Penjelasan Program :** Program ini digunakan untuk menampilkan data dari pemesanan baju. Program ini memiliki class bernama KelasLuar dan memiliki method main.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Interface

**Bahasa Pemrograman :** JAVA

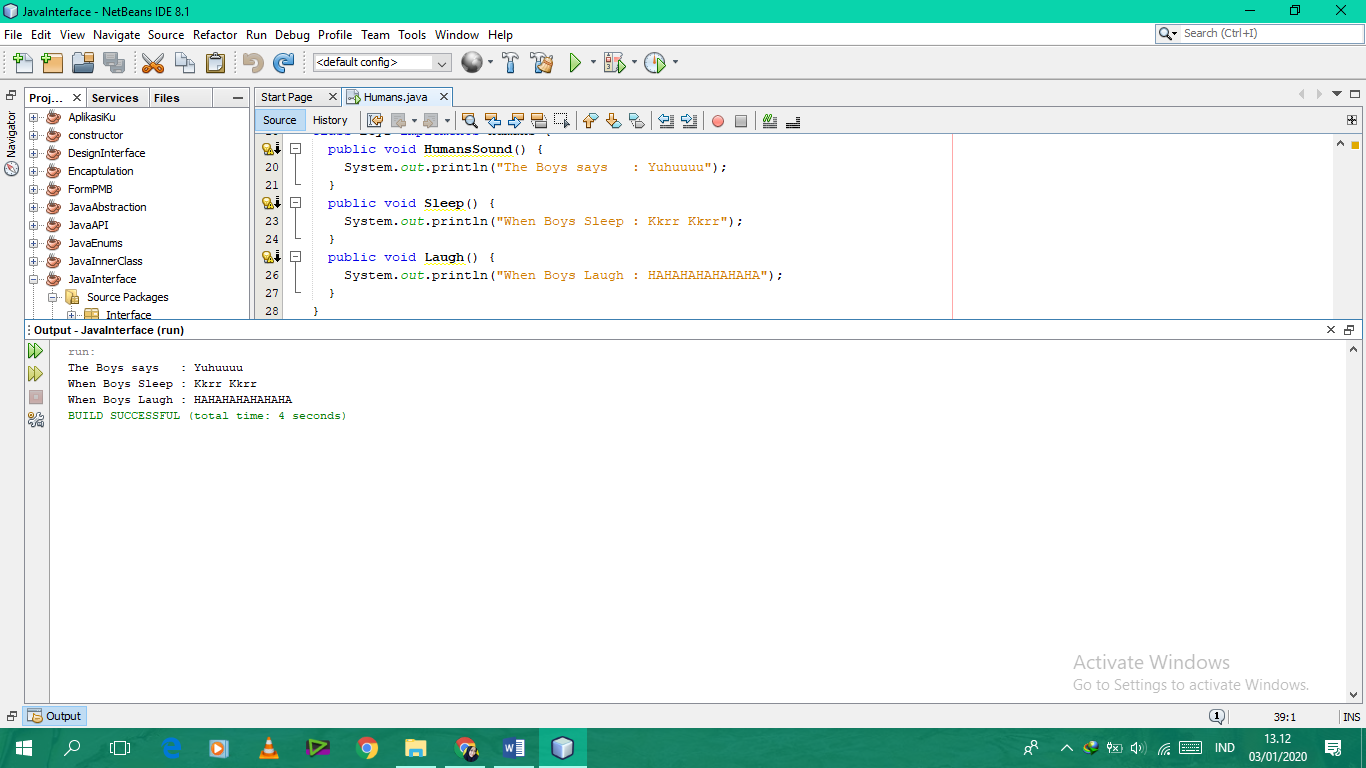
**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package study\_kasus;  /\*\*  \*  \* @author Putri  \*/  interface Humans {  public void HumansSound(); // interface method (does not have a body)  public void Sleep(); // interface method (does not have a body)  public void Laugh();  }  class Boys implements Humans {  public void HumansSound() {  System.out.println("The Boys says : Yuhuuuu");  }  public void Sleep() {  System.out.println("When Boys Sleep : Kkrr Kkrr");  }  public void Laugh() {  System.out.println("When Boys Laugh : HAHAHAHAHAHAHA");  }  }  class Human {  public static void main(String[] args) {  Boys myBoys = new Boys();  myBoys.HumansSound();  myBoys.Sleep();  myBoys.Laugh();  }  } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama study\_kasus.
2. interface Humans Mendeklarasikan interface Humans.
3. public void HumansSound() Mendeklarasikan metode HumansSound bertipe publik dan void.
4. public void Sleep()Mendeklarasikan metode Sleep bertipe publik dan void.
5. class Boys implements Humans Mendeklarasikan kelas Boys yang mengimplementasikan Humans.
6. public void HumansSound() Mendeklarasikan metode HumansSound bertipe publik dan void.
7. System.out.println("The Boys says : Yuhuuuu") Mencetak method kata "The Boys says : Yuhuuuu"
8. class Human Mendeklarasikan kelas Humans
9. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
10. Boys myBoys = new Boys() Untuk mendeklarasikan variabel Boys
11. myBoys.HumansSound() Mendeklarasikan metode HumansSound dari variabel Boys.

**Output Program :**

**Penjelasan Program :** Program ini menampilkan suara laki-laki, yaitu yuhuu saat berkata kemudian kkrr kkrr saat tidur. Program ini memiliki class bernama Humans dan memiliki method main.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Abstraction

**Bahasa Pemrograman :** JAVA

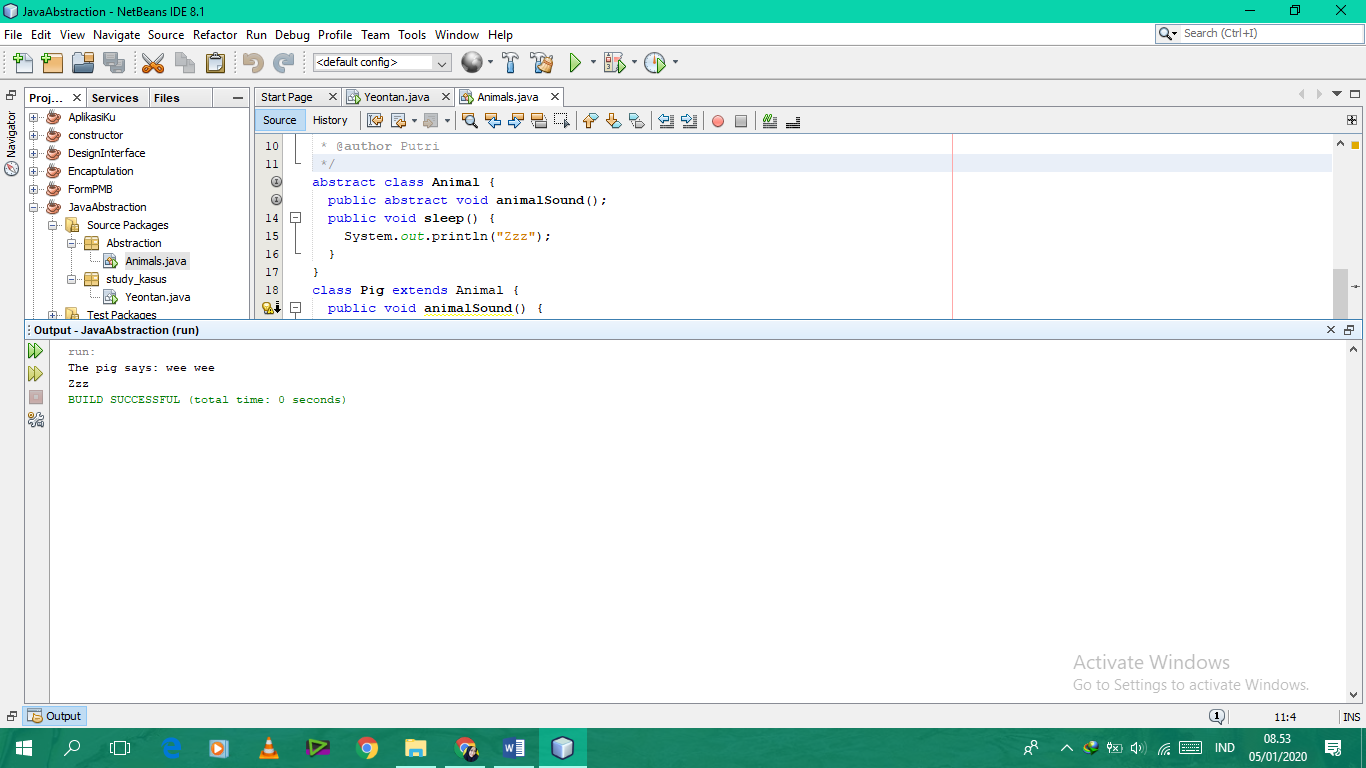
**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package study\_kasus;  /\*\*  \*  \* @author Putri  \*/  abstract class Animal {  public abstract void animalSound();  public void sleep() {  System.out.println("Zzz");  }}  class Pig extends Animal {  public void animalSound() {  System.out.println("The pig says: wee wee");  }}  class Animals {  public static void main(String[] args) {  Pig myPig = new Pig();  myPig.animalSound();  myPig.sleep();  } } |

**Keterangan Script :**

1. Mendeklarasikan Package bernama study\_kasus.
2. abstract class Animal Pendeklarasian kelas dengan nama Animal dan bersifat abstrak.
3. public abstract void animalSoundMendeklarasikan metode animalSound bertipe publik dan void bersifa abstrak.
4. public void sleep() Mendeklarasikan method sleep bertipe publik dan void.
5. System.out.println("Zzzzz") Mencetak kata “Zzzzz".
6. class Pig extends Animal Untuk mendeklarasikan kelas Pig perluasan dari kelas animal.
7. public void animalSound() Mendeklarasikan method animalSound bertipe publik dan void.
8. System.out.println("The pig says: wee wee") Untuk mencetak kata "The pig says: wee wee".
9. class Animals Pendeklarasian kelas Animals.
10. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program.
11. Pig myPig = new Pig() Pendeklarasian variabel Pig.
12. myPig.animalSound() Mendeklarasikan metode animalSound dari variabel Pig.
13. myPig.sleep() Pendeklarasian method Sleep dari variabel Pig.

**Output Program :**

**Penjelasan Program :** Program ini menampilkan suara Pig, yaitu “wee wee” saat berkata kemudian “Zzzz” saat tidur. Program ini memiliki class bernama Animals dan memiliki method main.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Enums

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

(1)

|  |
| --- |
| package study\_kasus;  /\*\*  \*  \* @author Putri  \*/  enum Day {  Mother,  Valentine,  Birth  }  public class SwitchStatement {  public static void main(String[] args) {  Day myVar = Day.Valentine;    switch(myVar) {  case Mother:  System.out.println("Mother Day");  break;  case Valentine:  System.out.println("Valentine Day");  break;  case Birth:  System.out.println("BirthDay");  break;  }  }  } |

(2)

|  |
| --- |
| package study\_kasus;  /\*\*  \*  \* @author Putri  \*/  public class insideAclass {  enum Month {  JANUARI,  FEBRUARI,  MARET,  APRIL,  MEI,  JUNI,  JULI,  AGUSTUS,  SEPTEMBER,  OKTOBER,  NOVEMBER,  DESEMBER  }  public static void main(String[] args) {  Month myVar = Month.AGUSTUS;  System.out.println(myVar);  }  } |

**Keterangan Script :** (1)

1. Mendeklarasikan package bernama study\_kasus.
2. enum Day Mendeklarasikan method Day.
3. Mother,

Valentine,

Birth

Mendeklarasikan masing-masing variabel.

1. public class SwitchStatement Pendeklarasian kelas SwitchStatement dan bersiat publik.
2. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
3. Day myVar = Day.Valentine mendeklarasikan variabe valentine.
4. switch(myVar) Menentukan perintah yang akan di jalankan berdasarkan masukan dari pengguna.
5. case Mother; melakukan perintah dibaris selanjutnya dan masukannya adalah mother.
6. System.out.println("Mother Day"); Mencetak kata “Mother Day”.
7. Break; Mengakhiri perintah case.

**Keterangan Script :** (2)

1. Mendeklarasikan package bernama study\_kasus.
2. public class insideAclass Mendeklarasikan kelas insideAclass bersifat publik.
3. enum Month Mendeklarasikan method Mont.
4. JANUARI,

FEBRUARI,

MARET,

APRIL,

MEI,

JUNI,

JULI,

AGUSTUS,

SEPTEMBER,

OKTOBER,

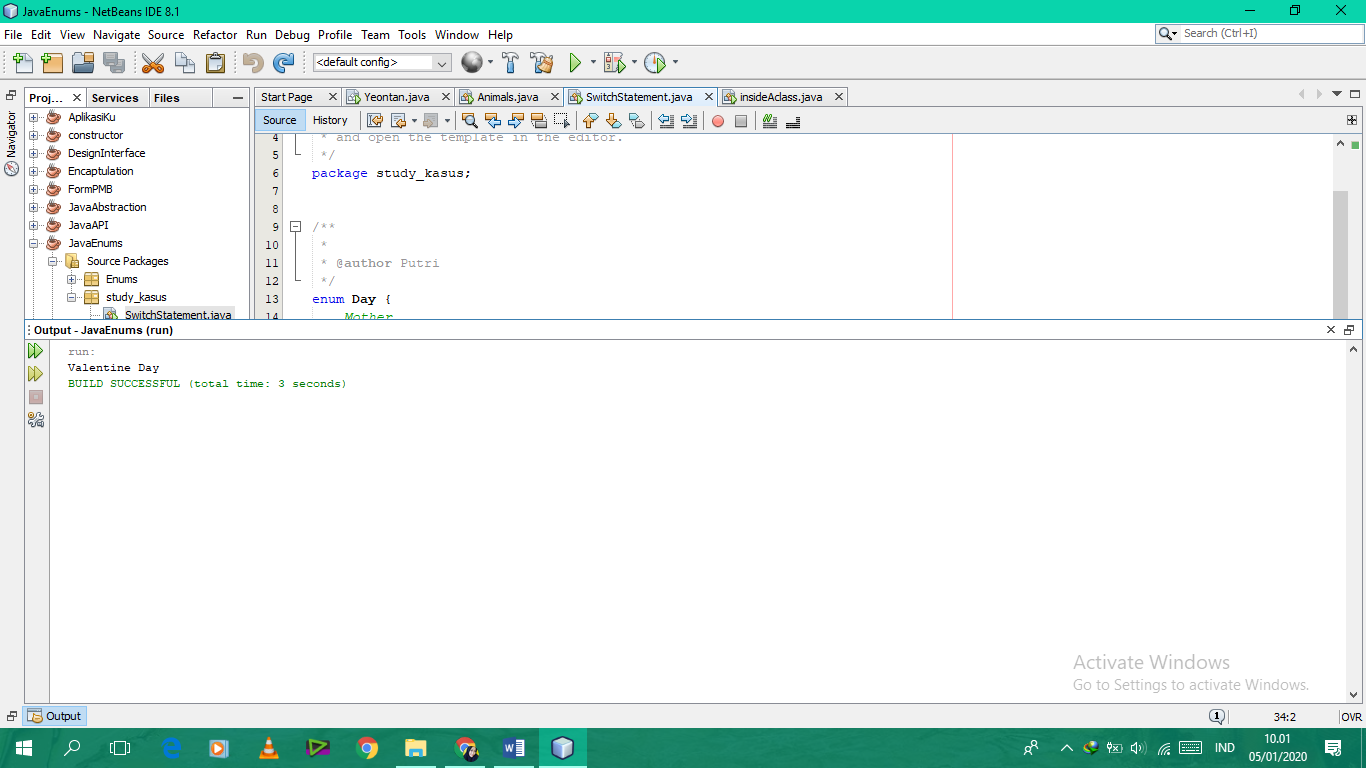
NOVEMBER,

DESEMBER

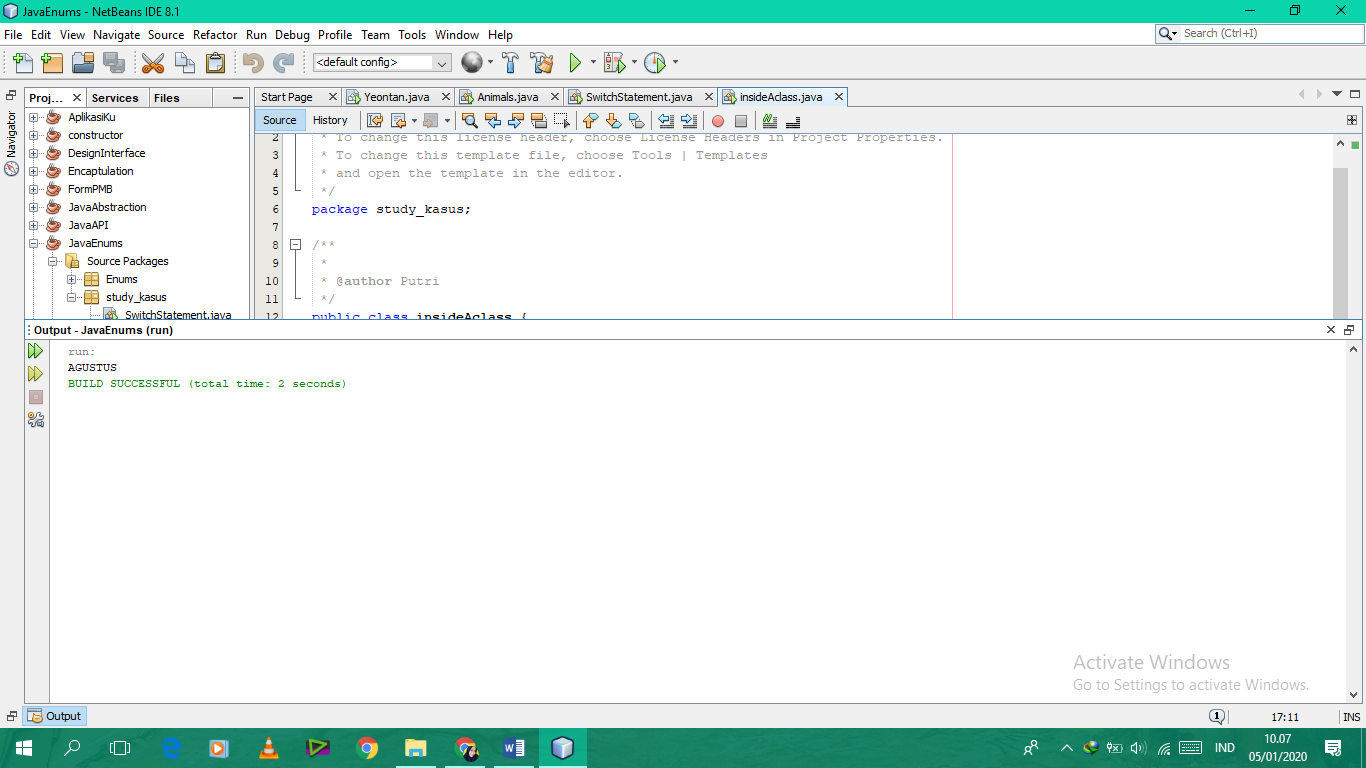
Pendeklarasian masing-masing variabel.

1. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
2. Month myVar = Month.AGUSTUS Pendeklarasian variabel Agustus dalam kelas Month.
3. System.out.println(myVar) Mencetak sekaligus memanggil variabel.

**Output Program :** (1)



**Penjelasan Program :** Program diatas ini melakukan Pencetakan nama hari sesuai perintah dari pengguna seperti contoh output diatas yaitu “Valentine Days”.

**Output Program :** (2)

**Penjelasan Program :** Program ini melakukan pencetakan nama-nama bulan sesuai perintah pengguna.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java User Input

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

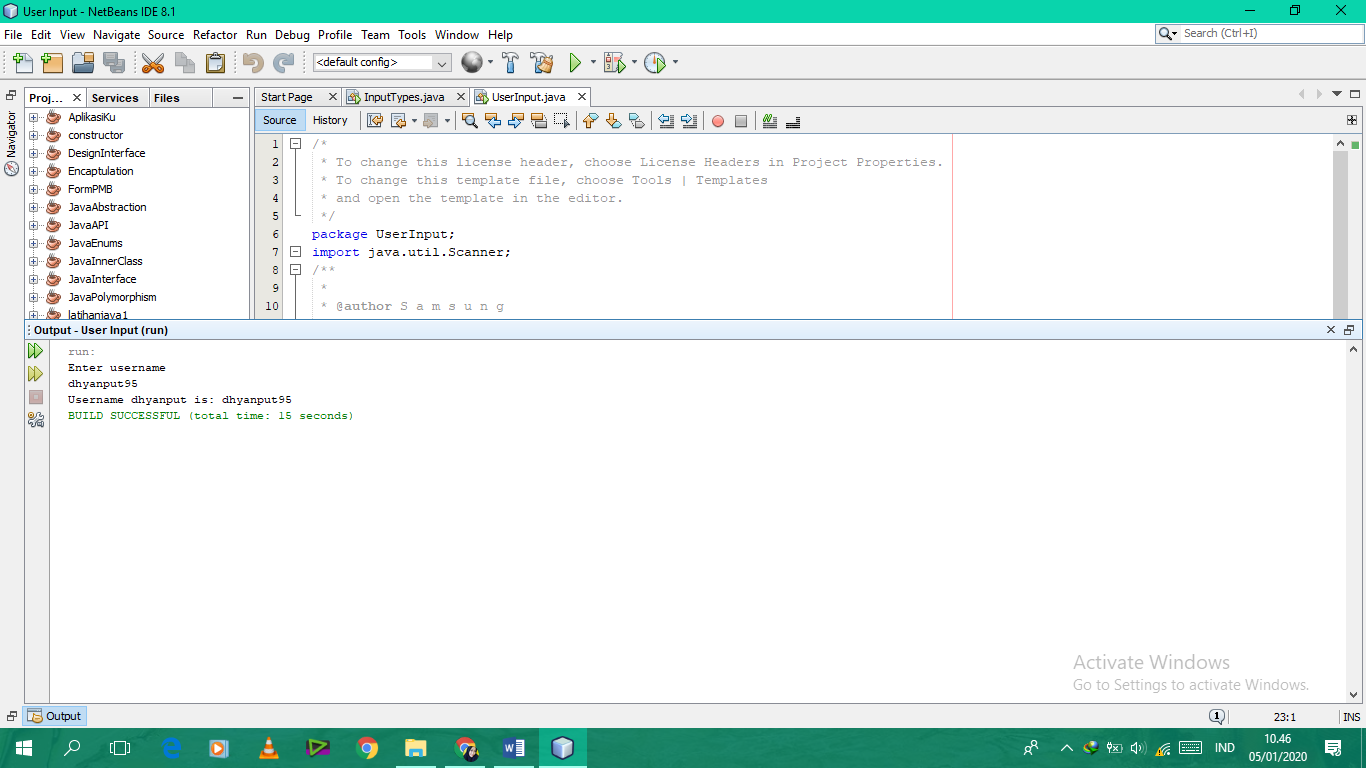
(1)

|  |
| --- |
| package UserInput;  import java.util.Scanner;  /\*\*  \*  \* @author Putri  \*/  public class UserInput {  public static void main(String[] args) {  Scanner myObj = new Scanner(System.in);  System.out.println("Enter username");  String userName = myObj.nextLine();  System.out.println("Username dhyanput is: " + userName);  }  } |

**Keterangan Script :** (1)

1. package UserInput; Mendeklarasikan paket bernama UserInput.
2. import java.util.Scanner; Mengimport utilitas java.
3. public class UserInput Mendeklarasikan kelas UserInput.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. Scanner myObj = new Scanner(System.in); Digunakan untuk memasukkan data scanner object.
6. System.out.println("Enter username"); Untuk mencetak kata “Enter Username”.
7. String userName = myObj.nextLine(); Menyimpan masukan dari pengguna dengan variabel userName bertipe string.
8. System.out.println("Username dhyanput is: " + userName); Untuk Mencetak variabel “Username dhyanput is”.

**Output Program :**



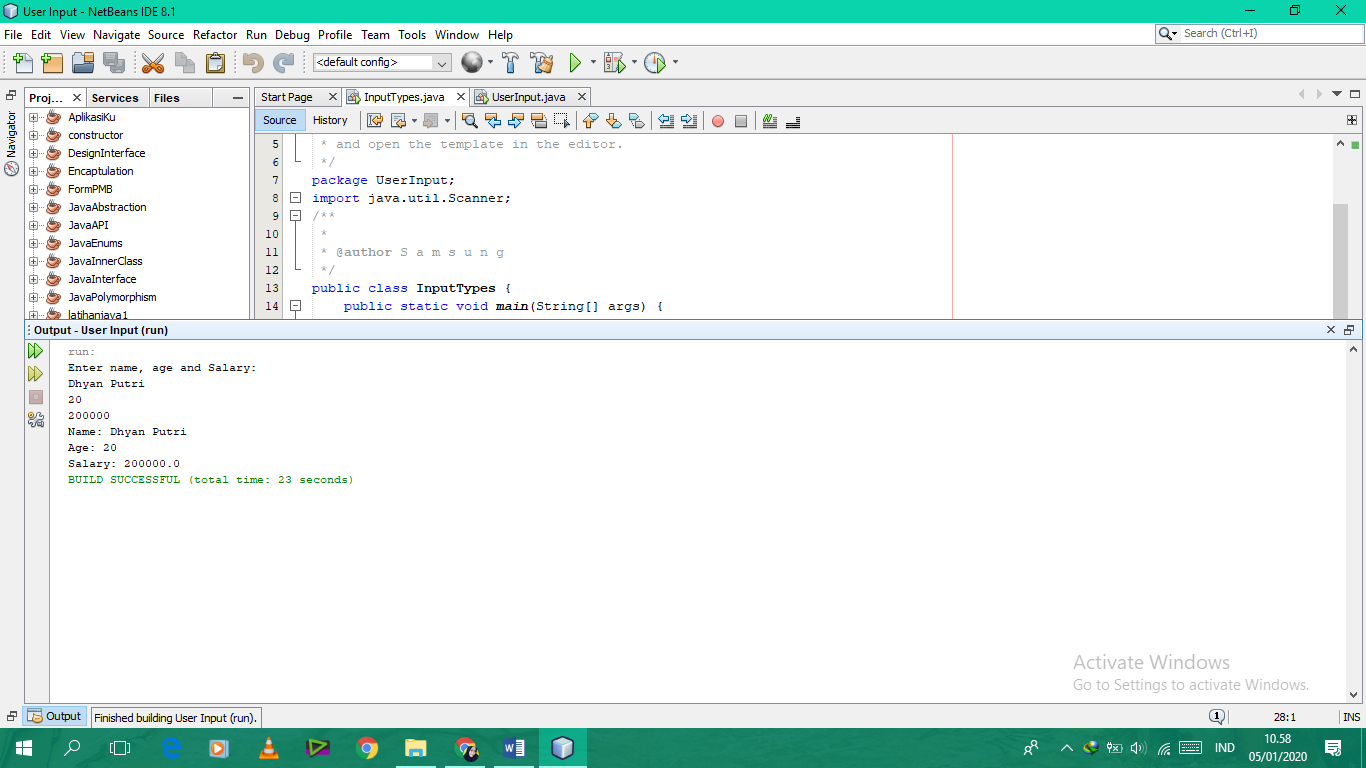
**Penjelasan Program :** Program diatas digunakan untuk menampilkan Username pengguna.

(2)

|  |
| --- |
| package UserInput;  import java.util.Scanner;  /\*\*  \*  \* @author Putri  \*/  public class InputTypes {  public static void main(String[] args) {  Scanner myObj = new Scanner(System.in);  System.out.println("Enter name, age and Salary:");  String name = myObj.nextLine();  int age = myObj.nextInt();  double salary= myObj.nextDouble();  System.out.println("Name: " + name);  System.out.println("Age: " + age);  System.out.println("Salary: " + salary);  }  } |

**Keterangan Script :** (1)

1. package UserInput; Mendeklarasikan paket bernama UserInput.
2. import java.util.Scanner; Mengimport utilitas java.
3. public class InputTypes Mendeklarasikan kelas InputTypes.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. Scanner myObj = new Scanner(System.in); Digunakan untuk memasukkan data scanner object.
6. System.out.println("Enter name, age and Salary:"); Mencetak method kata "Enter name, age and Salary:".
7. int age = myObj.nextInt(); Menyimpan masukan dari pengguna dengan variabel age bertipe integer.
8. System.out.println("Age: " + age); Untuk mencetak variabel “Age”.

**Output Program :**

**Penjelasan Program :** Progam diatas berisi cetakan data pengguna yang berupa nama, umur, dan gaji.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Date

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package Studi\_Kasus;  import java.time.LocalDateTime;  import java.time.format.DateTimeFormatter;  /\*\*  \*  \* @author Putri  \*/  public class MyClass1 {  public static void main(String[] args) {  LocalDateTime myDateObj = LocalDateTime.now();  System.out.println("Before formatting: " + myDateObj);  DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");    String formattedDate = myDateObj.format(myFormatObj);  System.out.println("After formatting: " + formattedDate);  }  } |

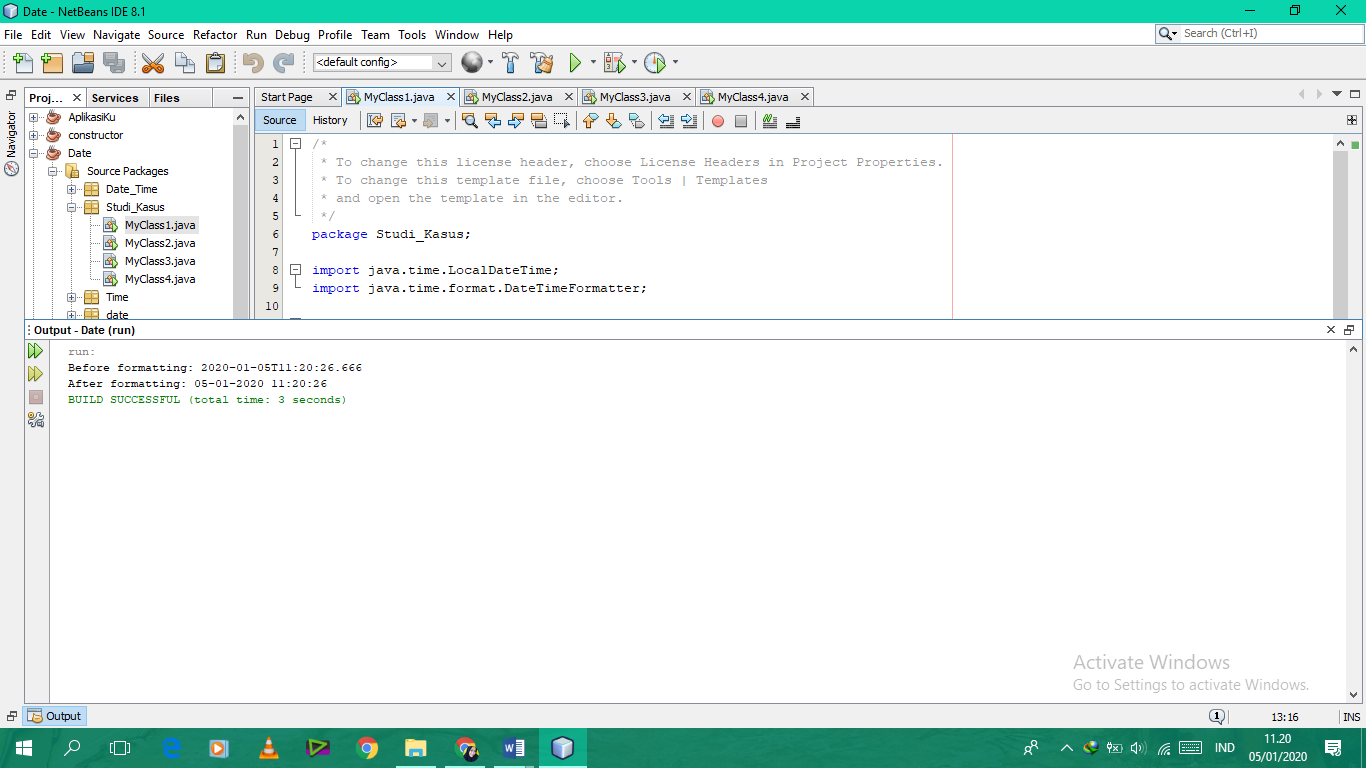
**Keterangan Script :**

1. Mendeklarasikan Package bernama study\_kasus.
2. import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

Mengimport utilitas java.

1. public class MyClass1 Mendeklarasikan kelas MyClass1
2. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
3. LocalDateTime myDateObj = LocalDateTime.now(); LocalTime myObj digunakan Untuk menambahkan objek waktu dengan menggunakan metode Now yang artinya waktu saat ini.
4. System.out.println("Before formatting: " + myDateObj); Ini adalah fungsi untuk menampilkan teks ke layar output-annya dengan menambahkan objek nya.
5. DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss"); Mengubah format tanggal dan waktu , dengan format *ofPattern* yang menerima segala macam nilai .
6. String formattedDate = myDateObj.format(myFormatObj); Menambah objek yang akan kita buat dalam memebrikan format yang berbeda pada tanggal dan waktu.
7. System.out.println("After formatting: " + formattedDate); Memanggil hasil outputan yang dudah di edit pada format yang di ubah.

**Output Program :**

**Penjelasan Program :** Program diatas digunakan untuk menampilkan waktu dan tanggal saat ini.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java ArrayList

**Bahasa Pemrograman :** JAVA

**Compiler :** IDE NetBeans 8.0.1

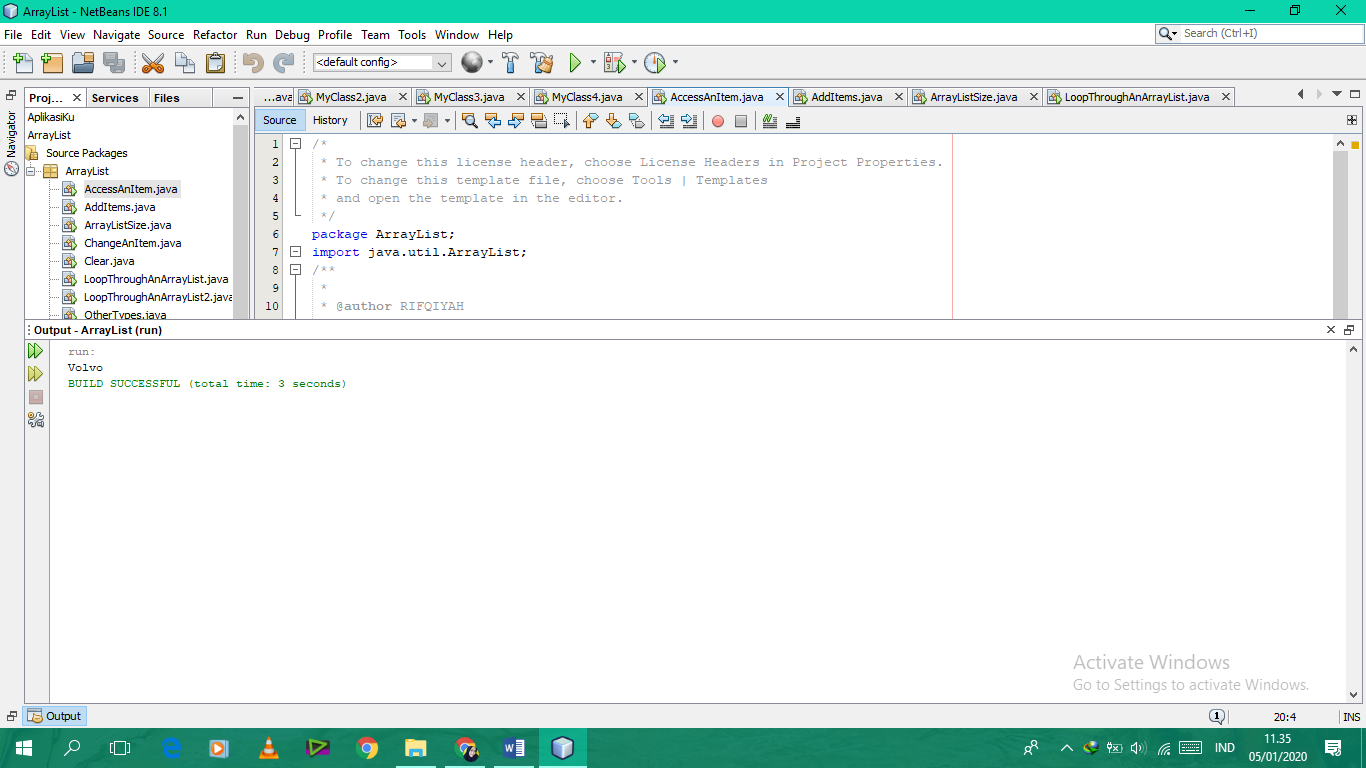
**Script Program :**

|  |
| --- |
| package ArrayList;  import java.util.ArrayList;  /\*\*  \*  \* @author Putri  \*/  public class AccessAnItem {  public static void main(String[] args) {  ArrayList<String> cars = new ArrayList<String>();  cars.add("Volvo");  cars.add("BMW");  cars.add("Ford");  cars.add("Mazda");  System.out.println(cars.get(0));  }  } |

**Keterangan Script :**

1. Mendeklarasikan package dengan nama ArrayList.
2. import java.util.ArrayList; Mengimport utilitas java arraylist.
3. public class AccessAnItem Mendeklarasikan kelas dengan nama AccesAnItem.
4. public static void main(String[] args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
5. ArrayList<String> cars = new ArrayList<String>(); Digunakan untuk memasukkan data arraylist dengan variabel cars dan bertipe string.
6. cars.add("Volvo"); Memasukkan data Volvo ke dalam variabel cars.
7. System.out.println(cars.get(0)); Mencetak method dari variabel cars.

**Output Program :**



**Penjelasan program :** Program diatas digumakan untuk menampilkan jenis merek2 mobil.

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Wrapper Class

**Bahasa Pemrograman :** JAVA

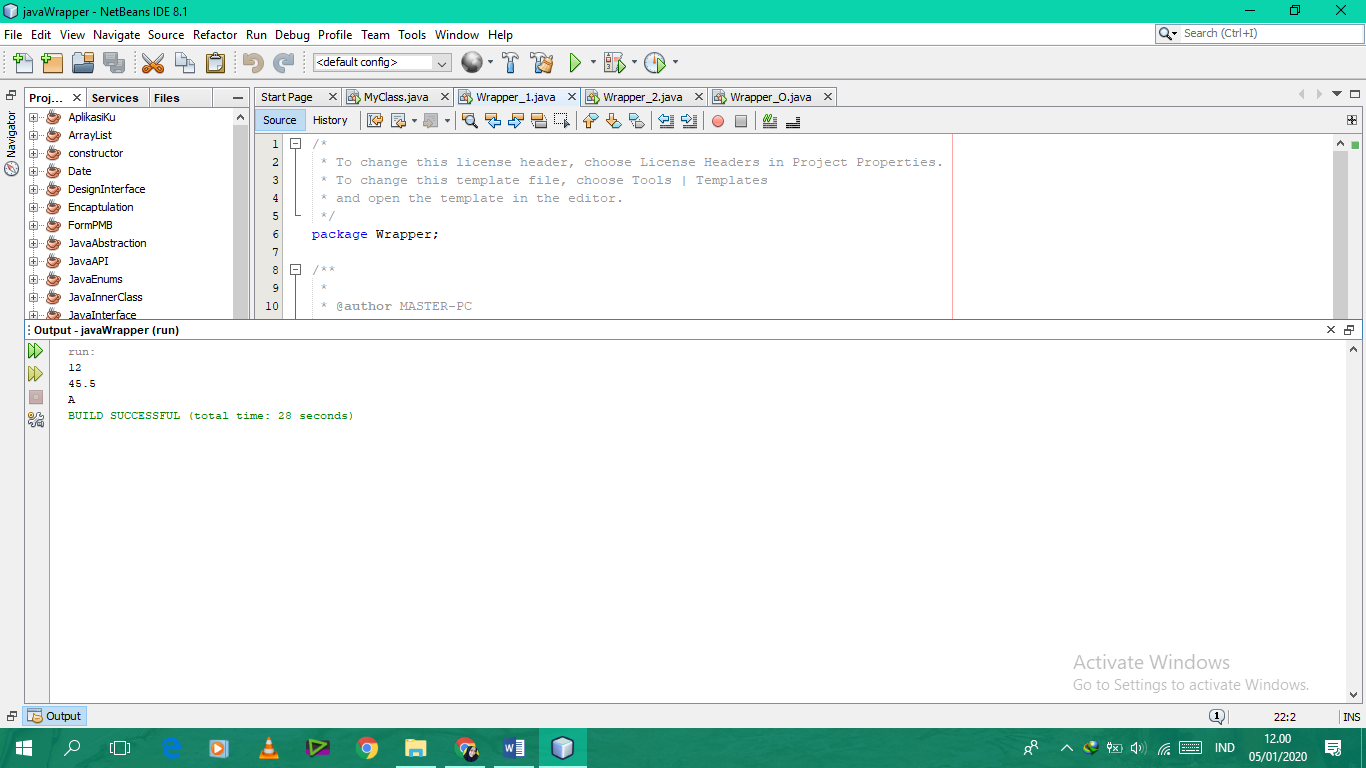
**Compiler :** IDE NetBeans 8.0.1

**Script Program :**

|  |
| --- |
| package Wrapper;  /\*\*  \*  \* @author Putri  \*/  public class Wrapper\_1 {  public static void main(String[]args){  Integer myInt= 12;  Double myDouble = 45.5;  Character myChar = 'A';  System.out.println(myInt.intValue());  System.out.println(myDouble.doubleValue());  System.out.println(myChar.charValue());  }  } |

**Keterangan Script :**

1. Mendeklarasikan package dengan nama Wrepper.
2. public class Wrapper\_1 Mendeklarasikan kelas dengan nama Wrapper\_1 bersifat publik.
3. public static void main(String[]args) Mendeklarasikan suatu method dengan nama main. Nama main di sini merupakan keharusan dalam java karena java akan mencari method yang bernama main ini sebagai awal eksekusi program. Keyword public merupakan access specifier yang menentukan visibility level dari method ini. Public berarti method ini dapat diakses dari manapun dari luar class.
4. Integer myInt= 12; Untuk Memasukkan data yang bersifat integer.
5. System.out.println(myInt.intValue()); Untuk mencetak hasil output integer.

**Output Program :**

**Latihan Praktikum**

**Pemrograman JAVA**

**Nama Program :** Java Wrapper Class

**Bahasa Pemrograman :** JAVA

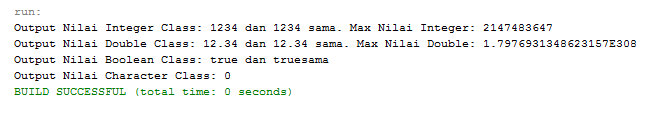
**Compiler :** IDE NetBeans 8.0.1

Source code untuk program java wrapper class yaitu:

Pada studi kasus diatas digunakan wrapper class seperti integer, double, Boolean, dan character. Kemudian menginstalasi objek tersebut dan menggunakan constructor untuk mendefinisikan nilainya pada pernyataan.

Kecuali pada character class, di dalam wrapper class dapat juga digunakan dua tipe pernyataan dengan menggunakan tipe data primitive pada umumnya dan juga bertipe data string. Untuk mengakses method dapat juga dilakukan secara langsung pada wrapper class dengan method keyword static yang telah didefinisikan di dalam wrapper class. Seperti contohnya **Integer.MAX\_VALUE,** dan juga **Double.MAX\_VALUE** yang digunakan untuk menampilkan output hasil, nilai maximal dari tipe data integer dan juga double.

Hasil output:



**Latihan Praktikum**

**Pemrograman JAVA**

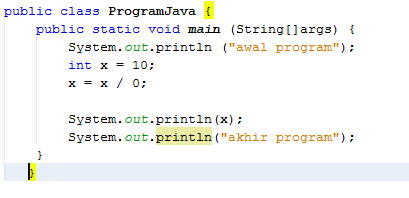
**Nama Program :** Java Exception

**Bahasa Pemrograman :** JAVA

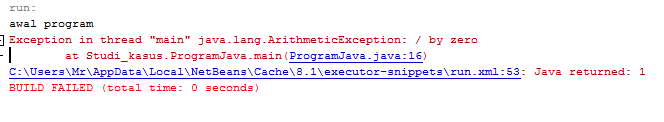
**Compiler :** IDE NetBeans 8.0.1

Untuk menangani error pada java,digunakan sebuah statement yang bernama try..catch. statement tersebut digunakan untuk mengurung eksekusi yang menampilkan error dan dapat membuat program tetap berjalan tanpa dihentikan secara langsung. Error yang ditangani oleh try..catch biasa disebut dengan exception.

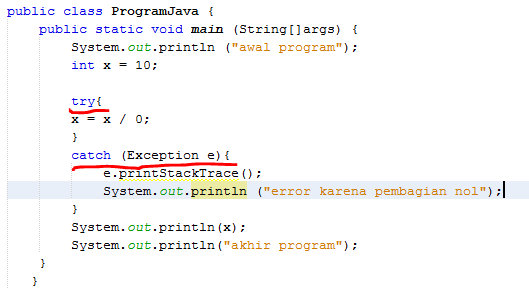
Untuk java exception, dibuat sebuah studi kasus yang mengandung error ketika dijalankan. Berikut adalah source code nya:



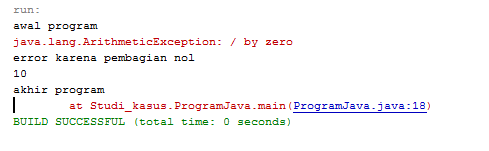
Ketika dijalankan maka akan mendapat hasil output sebagai berikut:



Berbeda ketika operasi pembagian nol diatas dikurung dengan try..catch, maka hasil eksekusi program akan sedikit berbeda.

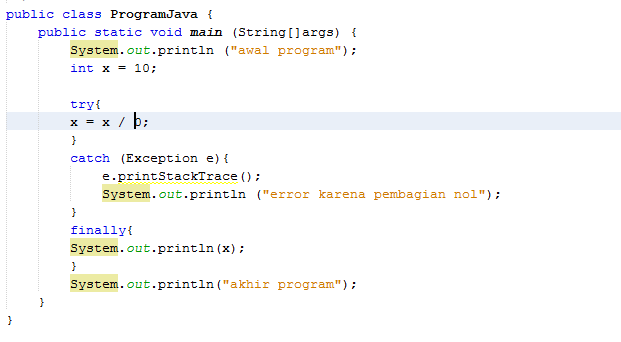


Hasil output:

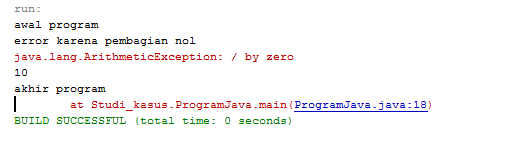


Ketika program di eksekusi kembali maka program akan dijalankan bagian akhir program walaupun terjadi error di tengah-tengah eksekusi.

Untuk menghindari error, akan digunakan statemen finally. Perhatikan source code berikut:



Ketika menjalankan program tersebut maka akan didapat:



Tetap diperoleh hasil error karena pembagi tidak boleh nol.

**BAB IV**

**PENUTUP**

* 1. **KESIMPULAN**

Pemrograman Berorientasi Objek (*Object Oriented Programming* atau OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Objek adalah struktur data yang terdiri dari bidang data dan metode bersama dengan interaksi mereka untu merancang aplikasi dan program computer. Dalam OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai objek. Pembahasan mengenai orientasi objek tidak akan terlepas dari konsep objek seperti *inheritance* atau pewarisan, *encapsulation* atau pembungkusan, dan *polymorphism* atau lainnya. Konsep-konsep ini merupakan fundamental dalam orientasi objek yang perlu dipahami serta digunakan dengan baik, dan menghindari penggunaanya yang tidak tepat. Dalam laporan praktikum ini dilengkapi dengan studi kasus serta hasil percobaan pada setiap babnya.

* 1. **SARAN**

Laporan praktikum ini dapat dikembangkan lebih lanjut sehingga memajukan pendidikan teknologi informasi dan komunikasi Indonesia, terutama pada bidang studi pemrograman brorientasi objek menggunakan bahasa pemrograman java.

**DAFTAR PUSTAKA**

W3Schools. 2019. *Java*.

<https://www.w3schools.com/java/default.asp>

(diakses tanggal 24 Desember 2019 pukul 10.09 WIB)