TUGAS WORKSHOP SISTEM INFORMASI BERBASIS DEKSTOP

TUGAS INDIVIDU

OLEH:

NAMA : FIRMAN WAHYUDI

NIM : E41190305

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2020

Pertemuan 1

Practice 4.2

Packages	A group of related Java classes.
Code Block	Sections of code that are enclosed inside a set of curly braces. {}
Upper Camel Case	First letter uppercase and the first letter of each internal word capitalized. Example: savingsaccount
Constant	A named value that does not change.
Lower Camel Case	First letter lowercase and the first letter of each internal word capitalized. Example: studentfirstname
Driver Class	A class that contains a main method.
Import Statement	A code statement in a Java class file that includes java code from another package or class.
Programmer-created Object Class	A class that defines instances of objects to be used in another class.
Java Comments	Code that is preceded by //. Comments are used to clarify programming logic. Comments are ignored by the compiler.
Java Keywords	A word that has a special function in the Java language, and cannot be used as names for classes, methods, or variables.
Java API	The library of Java classes available to import into a programmercreated class.
Object Class	The outline of an object, including class variables, constructors, and methods.
Constructor	Values that are sent into a method or constructor to be used in a calculation or substituted with values from the class.
Parameters	Values, such as numbers, characters, or booleans. References to objects, such as a bankaccount object.
Variables	Keywords used to specify the accessibility of a class (or type) and its members. Ex: public, private, protected, default

Access Modifiers	A block of code inside a class that is used to change or access information about the class.
Methods	A block of code inside a class that is used to change or access Information about the class.

Practice 4.3

Variables	Named primitive or object storage mechanisms defined in a program. The assigned value may or may not (constants) change.
Arithmetic Operators	Symbols are used to do addition, subtraction, multiplication, division, and modular arithmetic in math expressions and formulas.
Primitive Data Types	The group of Java data types that do not use the keyword new when declared or initialized. They store the value in the same place in memory as the variable name.
Byte	The smallest java primitive type that can hold an integer value.
Long	This data type (8 bytes) is the largest integer type.
Conventions	The formatting and naming standards that most programmers follow.
Int	This Java primitive data type (4 bytes) can hold integer values.
Double	This Java primitive data type (8 bytes) is the largest primitive that can hold a decimal value.
Intialization	When a variable is assigned a value for the first time.
Float	This Java primitive data type (4 bytes) can be initialized with a decimal number preceding letter f.
Literal	Can be any number, text, or other information that represent a value; used to initialize a primitive type.
Declaration	A Java statement when a variable is defined but not necessarily assigned a value.
Order of Operations	This word describes the mathematical precedence that a variable has in a Java program.

Char	A java primitive data type (2 bytes) that can hold single character values.
Scope	Used to describe the block of code where a variable exists in a program. A block of code is denoted by ().
Type Casting	The process of explicitly modifying one data type to become a different data type.
Truncation	A concept where a number is always rounded down to the nearest integer.
Assignment Operator	The equal sign '=' used in a Java statement to assign a value 'to a variable'.
Type Conversion	The process of modifying one data type to become a different data type. This may be implicit or explicit.
Short	A Java primitive data type (2 bytes) that holds integer numbers within a shorter range than an int.
Boolean	A one-bit java primitive type that can hold the value true or false.

Practice 4.4

Concatenation	Joining multiple String objects together.
Escape Sequences	Specific characters that are preceded by a \ character. When evaluated, the special character is evaluated as a special function, such as tabs, newlines, etc.
Instantiate	Assigning a value to a String object reference.
Object Reference	A data type that references the location in memory where an object is stored rather than a single, specific value.
String Methods	Code available in the Java API to manipulate or return strings.
String Object	An Object type that stores sentences, words, or multiple characters.

Tugas

A. Membuat object class berikut dalam file person.java:

Student Name: Lisa Palombo

Student ID: 123456789

Student Status: Active

Nama variable yang digunakan: fName, lName, stuId, stuStatus

Tampilkan dengan System.out.println

```
1   package studenttester;
2
3   public class Person {
4       public String fName,lName,stuStatus;
5       public int stuID;
6       public Person(String fName, String lName, int stuID, String stuStatus){
7           this.fName=fName;
8           this.lName = lName;
9           this.stuID = stuID;
10          this.stuStatus = stuStatus;
            toString();
12      }
13      public String getFName(){
14          return fName;
15      }
16      public void setFName(String fName){
17          this.fName = fName;
18      }
19
20      public String getLName(){
21          return lName;
22      }
23
24      public void setLName(String lName){
25          this.lName = lName;
26      }
27      public int getStuID(){
28          return stuID;
29      }
30      public void setStuID(int stuID){
31          this.stuID = stuID;
32      }
```

```
30      public void setStuID(int stuID){
31          this.stuID = stuID;
32      }
33      public String getStuStatus(){
34          return stuStatus;
35      }
36      public void setStuStatus(String stuStatus){
37          this.stuStatus = stuStatus;
38      }
        public String toString(){
            String Output = "";
41          Output = "Name       : "+getFName()+" "+getLName()+
42                   "\nStudent ID : "+getStuID()+
43                   "\nStatus     : "+getStuStatus();
44          return Output;
45      }
46      public static void main(String[] args) {
47          Person s1 = new Person("Lisa","Palombo",123456789,"Active");
48          System.out.println(s1);
49      }
50  }
```

Output - StudentTester (run)

```
run:
Name       : Lisa Palombo
Student ID : 123456789
Status     : Active
BUILD SUCCESSFUL (total time: 3 seconds)
```

B. Membuat object class berikut dalam file managingpeople.java

```
1  package studenttester;
2  public class ManagingPeople {
3
4      public static void main(String[] args) {
5          // TODO code application logic here
6          Person p1 = new Person("Arial",37);
7          Person p2 = new Person("Joseph", 15);
8
9          if(p1.getAge()==p2.getAge()){
10
11             System.out.println(p1.getName()+"is same age as"+p2.getName());
12         }else{
13
14             System.out.println(p1.getName()+"is NOT same age as"+p2.getName());
15         }
16     }
17 }
18
```

Managing People tidak bisa di run karena di class person tidak ada method getAge() dan getName(),dan dibagian new Person(“Arial”,37) Seharusnya ada 4 parameter yang dimasukkan contohnya new Peson(“Lisa”,”Palombo”,123456789,”Active”)

C. Terdapat kesalahan dalam pendeklarasian variable. Jelaskan

```
boolean gameOver = false;
int students=50,classes=3;
double sales_tax;
short number1;
```

```
int 2beOrNot2be;
float price index;
double lastYear'sPrice;
long class;
```

int 2beOrNot2be : Salah Karena Variabel tidak boleh diawali oleh angka

float price index : Salah karena tidak boleh ada spasi jika ingin diberi jarak menggunakan garis bawah

double lastYear’sPrice : Salah karena symbol yang diizinkan adalah garis bawah dan tanda dollar

long class : salah karena variabel tidak boleh mengandung keyword java

D. Tuliskan output dari operasi string berikut:

```java
package stringobject;
public class StringObject {

    public static void main(String[] args) {
        // TODO code application logic here
        String s1 = "ABC";
        String s2 = new String("DEF");
        String s3 = "AB" + "C";

        System.out.println("s1.compareTo(s2) : "+s1.compareTo(s2));
        System.out.println("s2.equals(s3) :"+s2.equals(s3));
        System.out.print("s3 == s1 :");
        System.out.println(s3 == s1);
        System.out.println("s2.compareTo(s3) : "+s2.compareTo(s3));
        System.out.println("s3.equals(s1) : "+s3.equals(s1));
    }
}

```

Output - StringObject (run)

```
run:
s1.compareTo(s2) : -3
s2.equals(s3) :false
s3 == s1 :true
s2.compareTo(s3) : 3
s3.equals(s1) : true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pertemuan 2

Practice 5.1

Ternary operator	A shorthand form of an if/else statement.
Scanner	A Java class used for reading keyboard or file input during program execution.
Switch statements/if statements	A type of program control that allows different segments of code to execute when the input value matches a given condition.
Switch statements/if statements	A type of program control that allows different segments of code to execute when the input value matches a given condition.

Practice 5.2

Do-while loop	A post-test loop that executes an unknown number of times until a condition is met, but always executes the first time through the loop.
For loop	A pre-test loop that uses an iterator to keep track of how many times a loop will execute.
Continue	A keyword used to skip over the remaining code in a loop and return program control to the beginning of the loop to execute again.
While loop	A pre-test loop that executes an unknown number of times until a condition is met.
Break	A keyword used to terminate a loop from executing before the loop condition is met.

Practice 6.1

Iterate	The act of progressing through an array.

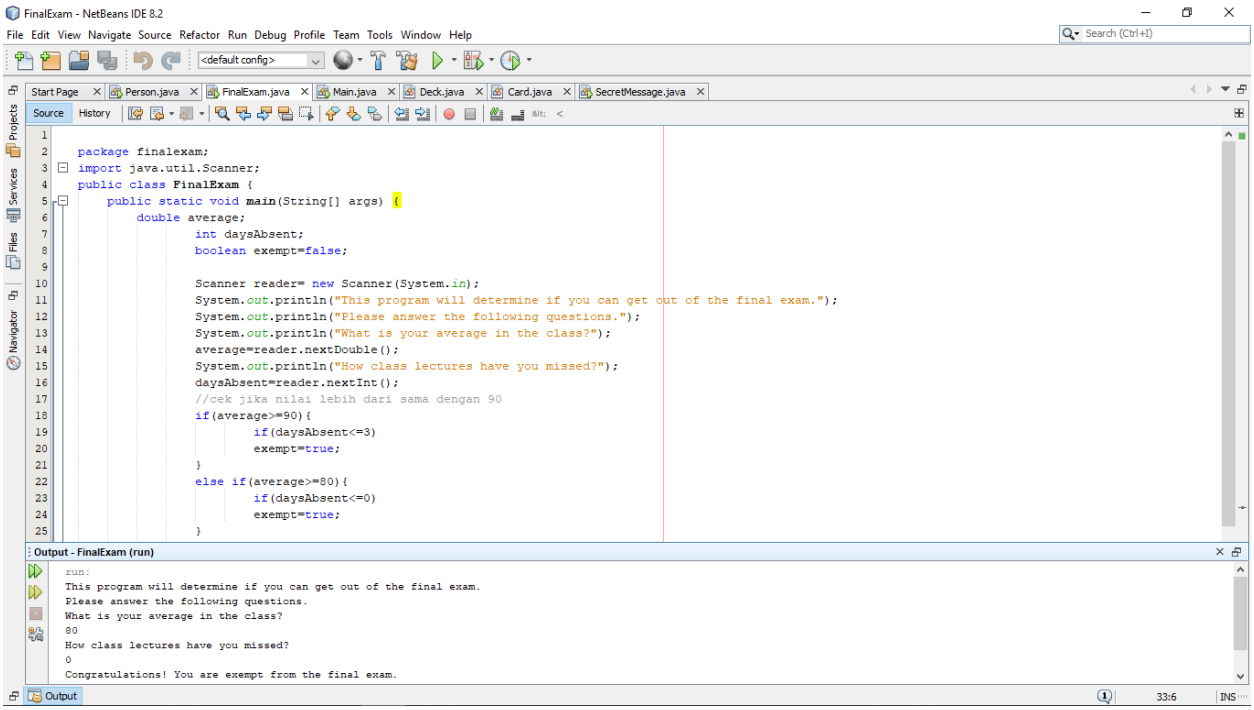
Array	A structure that stores multiple values of the same data type.
Array of Arrays	A two-dimensional array.
Index	An integer that identifies the location of a value in an array.
Command-Line Arguments	The ability to pass data into the main function and access it as an element of an array.
Algorithm	A logical computational procedure that if correctly applied ensures the solution of a problem.
Two-dimensional Array	An array of arrays, similar to a table, matrix, or spreadsheet.
Nested For Loop	A for loop inside of a for loop.
Single-dimensional Array	A named object used to store more than one value.

Practice 6.2

Catch	A keyword in Java that signals the following block of code handles a specified exception.
Unchecked Exceptions	An exception that is optional to be handled.
Checked Exception	An exception that MUST be handled.
Error	Indicates that there is a problem with interpreting your program.
Throw	This stops the interpreter from running the rest of the code until it finds a catch.
Syntax Error	An error that indicates an issue with coding format.
Run-Time Error	An error that occurs while the program is running, also known as an exception.
Logic Error	An error that occurs as a result of incorrect programmer logic.
Try/Catch Block	A block of code that handles exceptions by dealing with the exception if it is thrown.
Exceptions	Errors that occur during run-time and can be corrected or *handled* by your code.

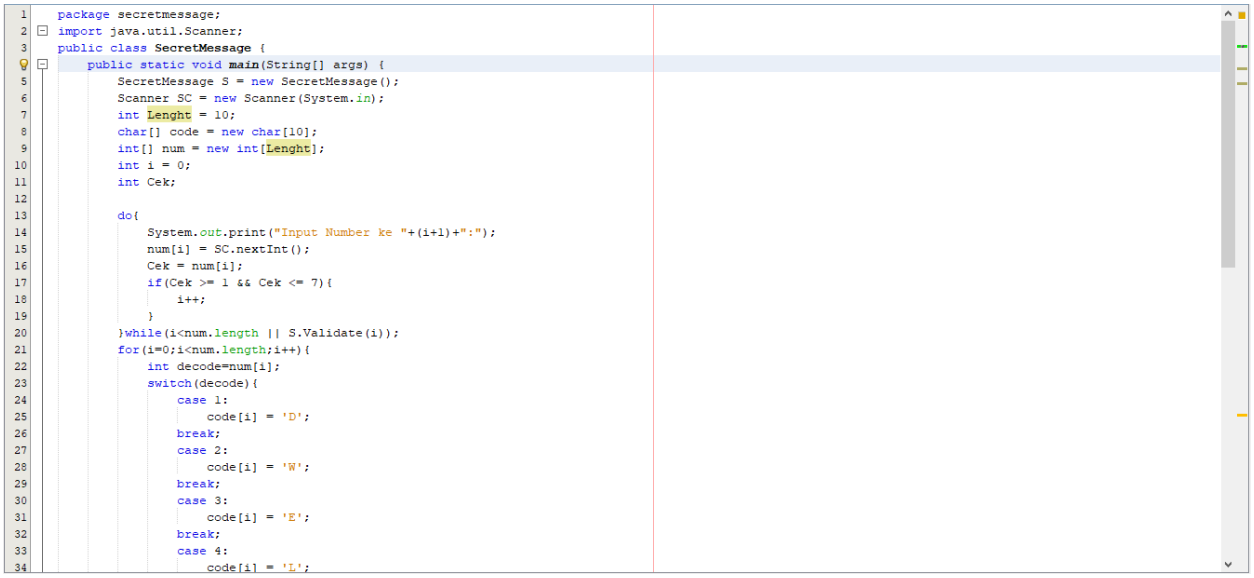
Tugas

A. Ketikkan code program berikut dan jelaskan outputnya



Output dari code programnya adalah menginput data nilai rata-rata dan berapa kali tidak absen. Dengan ketentuan nilai rata-rata diatas 80 dan tidak absen tidak lebih dari 0 maka diperbolehkan ikut ujian

B. Decode a Secret Message



```
                case 5:
                    code[i] = 'H';
                break;
                case 6:
                    code[i] = 'O';
                break;
                case 7:
                    code[i] = 'R';
                break;
                default:
                    System.out.println("?");
            }
        }
        for(i=0;i<num.length;i++){
            System.out.print(code[i]);
        }
        System.out.println();
    }
    private boolean Validate(int i){
        if(i > 7 || i  < 1)
            return false;
        else return true;
    }
}
```

```
Input Number ke 7:6
Input Number ke 8:7
Input Number ke 9:4

Input Number ke 10:1
HELLOWORLD
BUILD SUCCESSFUL (total time: 26 seconds)
```

C. Ketikkan code program berikut, jelaskan outputnya. Identifikasi class, method, dan object

Card Class

```
package pointc;
public class Card {
        String suit,name;
        int points;
    Card(int n1,int n2){
        suit = getSuit(n1);
        name = getName(n2);
        points = getPoints(name);
    }
    public String toString(){
        return "The "+name+" of "+suit;
    }
    public String getName(int i){
        if(i == 1) return "Ace";
        if(i == 2) return "Two";
        if(i == 3) return "Three";
        if(i == 4) return "Four";
        if(i == 5) return "Five";
        if(i == 6) return "Six";
        if(i == 7) return "Seven";
        if(i == 8) return "Eight";
        if(i == 9) return "Nine";
        if(i == 10) return "Ten";
        if(i == 11) return "Jack";
        if(i == 12) return "Queen";
        if(i == 13) return "King";
        return "error";
    }
    public int getPoints(String n){
    if(n == "Jack"||n == "Queen"||n == "King"||n == "Ten")
        return 10;
    if(n == "Two")
        return 2;
```

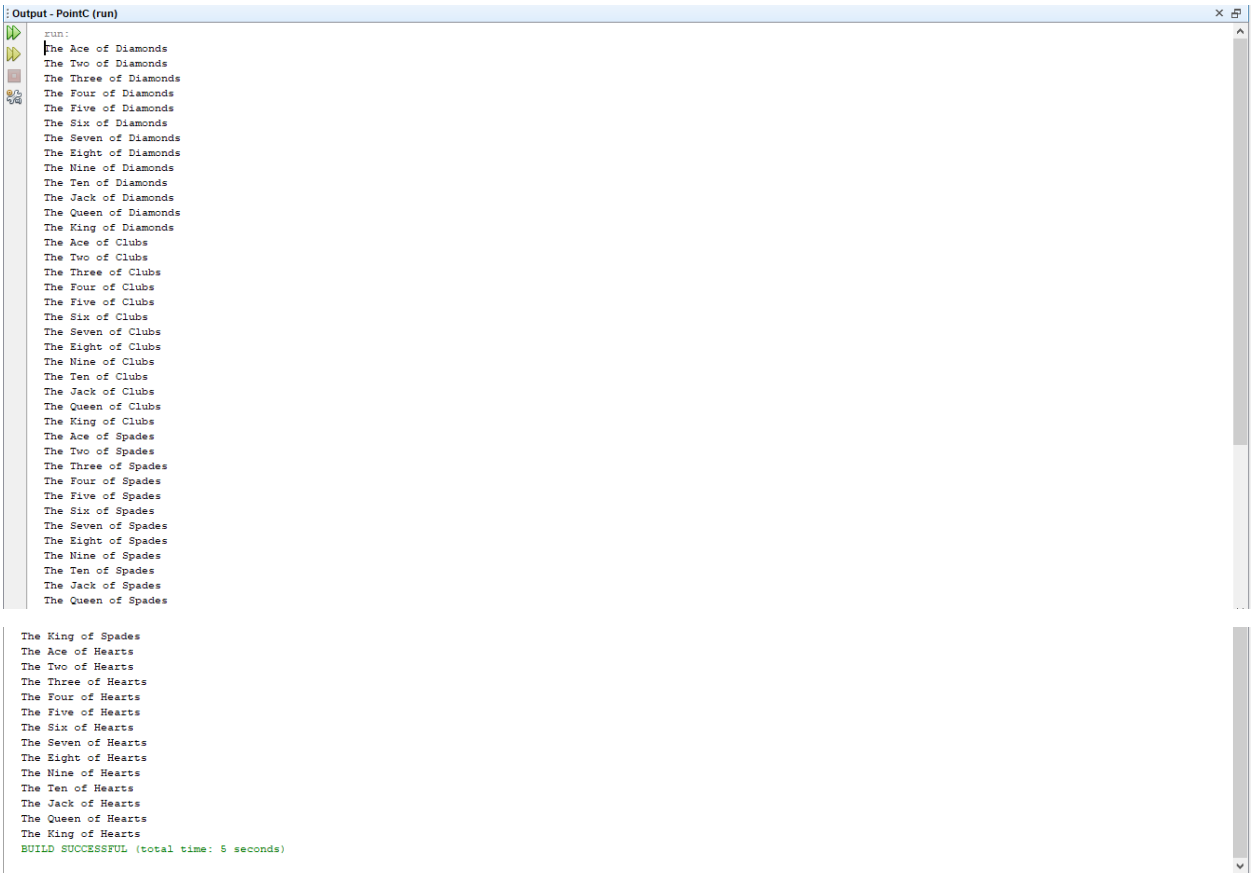
```
        if(n == "Three")
            return 3;
        if(n == "Four")
            return 4;
        if(n == "Five")
            return 5;
        if(n == "Six")
            return 6;
        if(n == "Seven")
            return 7;
        if(n == "Eight")
            return 8;
        if(n == "Nine")
            return 9;
        if(n == "Ace")
            return 1;
        return -1;
        }

        public String getSuit(int i){
        if(i == 1) return "Diamonds";
        if(i == 2) return "Clubs";
        if(i == 3) return "Spades";
        if(i == 4) return "Hearts";
        return "error";
        }
    }

```

Deck Class

```
package pointc;
public class Deck {
    Card[] cardArray = new Card[52];
    Deck(){
        int suits= 4;
        int cardType = 13;
        int cardCount = 0;
        for(int i=1;i<=suits;i++){
            for(int j=1;j<=cardType;j++){
                cardArray[cardCount] = new Card(i,j);
                cardCount++;
            }
        }
    }
    public void print(){
        for(int i=0;i<cardArray.length;i++){
            System.out.println(cardArray[i]);
        }
    }
}

```

Main Class

```
package pointc;

public class Main {
    public static void main(String[] args) {
        Deck d = new Deck();
        d.print();
    }
}

```



Output dari code programnya adalah nilai dari 13 adalah Nama Kartu dan 4 Suit yang dimana masukkan kedalam array dan dilooping terlebih dahulu agar susunan nama kartu urut dari 1 sampai 13 dan suit 1 sampai 4 setelah urut array tadi di print

Class :

- Card Class
- Deck Class
- Main Class

Method :

- Card Class : 1. toString()
  2. getName(int i)
  3. getPoints(String n)
  4. getSuit(int i)
- Deck Class : print()

Object :

- Deck Class : Card[] cardArray = new Card[52]
- Main Class : Deck d = new Deck()

D. Tuliskan perbedaan syntax error, logic error, dan exception.

- Syntax Error

  Syntax Error adalah kesalahan dalam coding karena aturan penulisan yang tidak sesuai atau kesalahan pada kontruksi kode
- Logic Error

  Logic Error merupakan kesalahan dalam coding yang terjadi bila tidak memberikan hasil seperti yang di ingkan
- Exception

  Exception artinya pengecualian, yang dimaksud dengan exception adalah kondisi yang akan muncul, jika suatu program tidak sukses dijalankan,atau dengan kata lain,user tidak mengisi input sesuai syarat berlaku atau dengan definisi lain exception adalah suatu kontruksi sebuah Bahasa khusus untuk menangani keadaan yang tidak terduga (biasanya adalah error)