

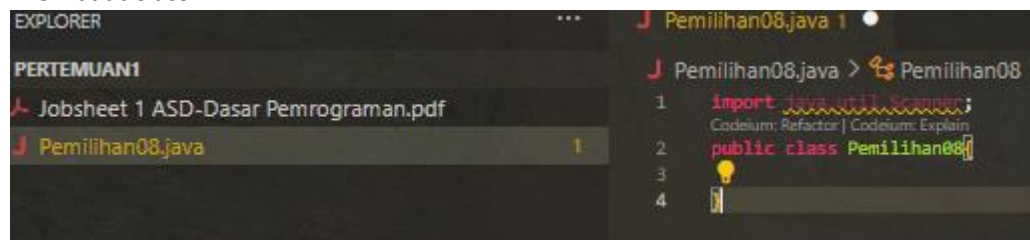
**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 1**



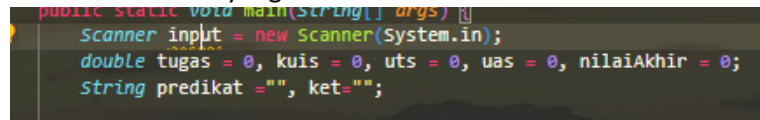
**Oleh:**  
**DZULFIKAR MUHAMMAD AL GHIFARI**  
**NIM. 2341760071**  
**SIB-1F / 08**  
**D-IV SISTEM INFORMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

## a. PEMILIHAN

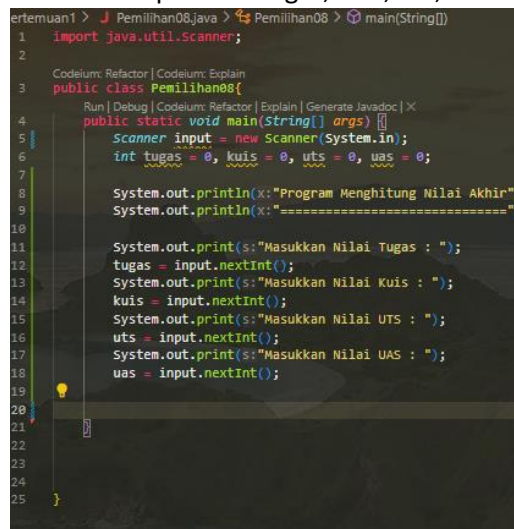
### 1. Membuat class



### 2. Deklarasi variable yang dibutuhkan



### 3. Membuat input nilai tugas, kuis, uts, dan uas



4. Membuat validasi dan kalkulasi nilai yang telah di inputkan

```
if(tugas > 100 || kuis > 100 || uts > 100 || uas > 100 ){
    System.out.println(x:"Nilai tidak valid");
}else{
    nilaiAkhir = (tugas*20/100) + (kuis*20/100) + (uts*30/100) + (uas*30/100);
    if(nilaiAkhir > 80){
        predikat = "A";
        ket = "SELAMAT ANDA LULUS";
    }else if(nilaiAkhir > 73){
        predikat = "B+";
        ket = "SELAMAT ANDA LULUS";
    }else if(nilaiAkhir > 65){
        predikat = "B";
        ket = "SELAMAT ANDA LULUS";
    }else if(nilaiAkhir > 60){
        predikat = "C+";
        ket = "SELAMAT ANDA LULUS";
    }else if(nilaiAkhir > 50){
        predikat = "C";
        ket = "SELAMAT ANDA LULUS";
    }else if(nilaiAkhir > 39){
        predikat = "D";
        ket = "MAAF, ANDA TIDAK LULUS";
    }else {
        predikat = "E";
        ket = "MAAF, ANDA TIDAK LULUS";
    }

    System.out.println("Nilai Akhir : " + nilaiAkhir);
    System.out.println("Nilai Huruf : " + predikat);
}
System.out.println("\n");
```

5. Hasil Akhir

```
Program Menghitung Nilai Akhir
=====
Masukkan Nilai Tugas : 50
Masukkan Nilai Kuis : 80
Masukkan Nilai UTS : 80
Masukkan Nilai UAS : 80
=====
Nilai Akhir : 74.0
Nilai Huruf : B+
=====
SELAMAT ANDA LULUS
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma> 
```

## b. PERULANGAN

### 1. Membuat class

```
pertemuan1 > J Perulangan08.java > Perulangan08 > main(String[])
1  import java.util.Scanner;
   Codeium: Refactor | Codeium: Explain
2  public class Perulangan08 {
   Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
3  public static void main(String[] args) {
4      Scanner input = new Scanner(System.in);
5
6
7  }
8
```

### 2. Deklarasi variable yang dibutuhkan

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

Long nim;
int angka;
```

### 3. Mencari 2 angka terakhir dan kalkulasi apabila angka > 10

```
angka = (int) (nim % 100);

if (angka < 10) {
    angka += 10;
}
```

### 4. Menampilkan angka dengan perulangan dengan catatan angka ganjil diganti dengan \* dan angka 6 & 10 di hilangkan

```
for (int i = 1; i <= angka; i++) {
    if (i != 6 && i != 10) {
        if (i % 2 == 0) {
            System.out.print(i + " ");
        } else {
            System.out.print(i + "* ");
        }
    }
}
```

### 5. Hasil akhir

```
Masukkan NIM Anda: 2341760071
* 2 * 4 * * 8 * * 12 * 14 * 16 * 18 * 20 * 22 * 24 * 26 * 28 * 30 * 32 * 34 * 36 * 38 * 40 * 42 * 44 * 46 * 48 * 50 * 52 * 54 * 56 * 58 * 60 * 62 * 64 * 66 * 68 * 70 *
```

### c. ARRAY

#### 1. Membuat class

```
temuan1 > Array08.java > Array08 > main(String[])
import java.util.Scanner;

Codeium: Refactor | Codeium: Explain
public class Array08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
    }
}
```

#### 2. Deklarasi variable yang dibutuhkan

```
public class Array08 {
    static String[][] dataMatkul = new String[8][5];
    static Scanner input = new Scanner(System.in);
    static double ips = 0, countNilai = 0, countsks = 0;
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        dataMatkul[0][0] = "Pancasila";
        dataMatkul[1][0] = "Konsep Teknologi Informasi";
        dataMatkul[2][0] = "Critical Thingking dan Problem Solving";
        dataMatkul[3][0] = "Matematika Dasar";
        dataMatkul[4][0] = "Bahasa Inggris";
        dataMatkul[5][0] = "Dasar Pemrograman";
        dataMatkul[6][0] = "Praktikum Dasar Pemrograman";
        dataMatkul[7][0] = "Keselamatan dan Kesehatan Kerja";
    }
}
```

#### 3. Input data nilai dari tiap tiap matkul beserta sksnya

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void inputData(){
    System.out.println(x:"=====");
    System.out.println(x:"Program Menghitung IP Semester");
    System.out.println(x:"=====");

    for(int i=0; i < dataMatkul.length; i++){
        System.out.print("Masukkan Nilai Angka untuk Mk "+dataMatkul[i][0]+ " : ");
        dataMatkul[i][1] = input.next();
        System.out.print("Masukkan SKS untuk Mk "+dataMatkul[i][0]+ " : ");
        dataMatkul[i][2] = input.next();
        System.out.println();
    }
}
```

#### 4. Mencari predikat dan bobot setiap nilai yang telah di inputkan

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void getPredicate(){
    for(int i=0; i<dataMatkul.length; i++){
        if(Double.valueOf(dataMatkul[i][1]) > 80){
            dataMatkul[i][3] = "A";
            dataMatkul[i][4] = "4.00";
        } else if(Double.valueOf(dataMatkul[i][1]) > 73){
            dataMatkul[i][3] = "B+";
            dataMatkul[i][4] = "3.50";
        } else if(Double.valueOf(dataMatkul[i][1]) > 65){
            dataMatkul[i][3] = "B";
            dataMatkul[i][4] = "3.00";
        } else if(Double.valueOf(dataMatkul[i][1]) > 60){
            dataMatkul[i][3] = "C+";
            dataMatkul[i][4] = "2.50";
        } else if(Double.valueOf(dataMatkul[i][1]) > 50){
            dataMatkul[i][3] = "C";
            dataMatkul[i][4] = "2.00";
        } else if(Double.valueOf(dataMatkul[i][1]) > 39){
            dataMatkul[i][3] = "D";
            dataMatkul[i][4] = "1.50";
        } else {
            dataMatkul[i][3] = "E";
            dataMatkul[i][4] = "1.00";
        }
    }
}
```



5. Kalkulasi ips dengan mengalikan bobot dengan sks dan dibagi dengan jumlah sks

```
for(int i=0; i<dataMatkul.length; i++){
    countNilai += (Double.valueOf(dataMatkul[i][4]) * Double.valueOf(dataMatkul[i][2]));
    countsks += Double.valueOf(dataMatkul[i][2]);
}
ips = countNilai/countsks;
```

6. Menampilkan hasil dan ips

```
System.out.println(x:"=====");
System.out.println(x:"Hasil Konversi Nilai");
System.out.println(x:"=====");

System.out.println(x:"Mk \t\t\t\t\t Nilai Angka \t Nilai Huruf \t Bobot Nilai \t SKS");
for(int i=0; i<dataMatkul.length; i++){
    System.out.println(dataMatkul[i][0] + "\t\t\t" + dataMatkul[i][1] + "\t\t" + dataMatkul[i][3] + "\t\t" + dataMatkul[i][4] + "\t\t\t" + dataMatkul[i][2]);
}

System.out.println(x:"=====");
System.out.println("IP : " + ips);
```

7. Hasil akhir

Masukkan SKS untuk Mk Keselamatan dan Kesehatan Kerja : 2

=====

Hasil Konversi Nilai

=====

MK	Nilai Angka	Nilai Huruf	Bobot Nilai	SKS
Pancasila	90	A	4.00	2
Konsep Teknologi Informasi	80	B+	3.50	2
Critical Thingking dan Problem Solving	80	B+	3.50	2
Matematika Dasar	80	B+	3.50	2
Bahasa Inggris	80	B+	3.50	2
Dasar Pemrograman	80	B+	3.50	2
Praktikum Dasar Pemrograman	80	B+	3.50	2
Keselamatan dan Kesehatan Kerja	80	B+	3.50	2

=====

IP : 3.5625

#### d. FUNGSI

##### 1. Membuat Class

```
Codeium: Explain  
public class Fungsi {
```

##### 2. Deklarasi variable yang dibutuhkan

```
static Scanner input = new Scanner(System.in);  
static String[][] tokoBunga = new String[4][5];  
static String[][] bungaRusak = new String[4][5];  
static int aglomema=75000, keladi=50000, alocasia=60000, mawar=10000;  
static String[] namaBunga = {"Aglonema", "Keladi \t", "Alocasia", "Mawar \t"};
```

```
int pilihan = 0;  
tokoBunga[0][0] = "RoyalGarden 1";  
tokoBunga[0][1] = "10";  
tokoBunga[0][2] = "5";  
tokoBunga[0][3] = "15";  
tokoBunga[0][4] = "7";  
  
tokoBunga[1][0] = "RoyalGarden 2";  
tokoBunga[1][1] = "6";  
tokoBunga[1][2] = "11";  
tokoBunga[1][3] = "9";  
tokoBunga[1][4] = "12";  
  
tokoBunga[2][0] = "RoyalGarden 3";  
tokoBunga[2][1] = "2";  
tokoBunga[2][2] = "10";  
tokoBunga[2][3] = "10";  
tokoBunga[2][4] = "5";  
  
tokoBunga[3][0] = "RoyalGarden 4";  
tokoBunga[3][1] = "5";  
tokoBunga[3][2] = "7";  
tokoBunga[3][3] = "12";  
tokoBunga[3][4] = "9";
```

##### 3. Menu untuk memilih aksi selanjutnya

```
while (true) {  
    System.out.println(x:"1. Cek Pendapatan Sold Out Semua Cabang");  
    System.out.println(x:"2. Cek Stok Cabang RoyalGarden 4");  
    System.out.println(x:"3. Exit");  
  
    System.out.print(s:"Pilih (1/2/3) :");  
    pilihan = 0;  
    pilihan = input.nextInt();  
    switch (pilihan) {  
        case 1:  
            int[][] income = getIncome();  
            showData(income);  
            break;  
        case 2:  
            updateStok();  
            showStok();  
            break;  
        case 3:  
            return;  
        default:  
            System.out.println(x:"Pilihan tidak tersedia");  
            break;  
    }  
}
```

- Fungsi `getIncome` untuk mendapatkan pendapatan tiap tiap bunga yang terjual pada tiap tiap cabang

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static int[][] getIncome(){
    int[][] income = new int[4][5];

    for (int i=0; i<income.length; i++){
        income[i][0] = Integer.valueOf(tokoBunga[i][1]) * aglonema;
        income[i][1] = Integer.valueOf(tokoBunga[i][2]) * keladi;
        income[i][2] = Integer.valueOf(tokoBunga[i][3]) * alocasia;
        income[i][3] = Integer.valueOf(tokoBunga[i][4]) * mawar;
        income[i][4] = 0;
    }

    return income;
}
```

- Fungsi `updateStok` untuk mengupdate stok bunga yang rusak/mati pada cabang 4

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void updateStok(){
    bungaRusak[3][0] = tokoBunga[3][0];
    bungaRusak[3][1] = "1";
    bungaRusak[3][2] = "2";
    bungaRusak[3][3] = "0";
    bungaRusak[3][4] = "5";
}
```

- Fungsi `showData` untuk menampilkan data pendapatan tiap tiap cabang, sedangkan fungsi `showStok` untuk menampilkan data stok pada cabang 4

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void showData(int[][] income){
    System.out.println("x: \t\t\t Aglonema \t Keladi \t Alocasia \t Mawar \t Total");
    for (int i=0; i<income.length; i++){
        System.out.println(tokoBunga[i][0] + "\t\t\t " + income[i][0] + "\t " + income[i][1] + "\t " + income[i][2] + "\t " + income[i][3] + "\t " + (income[i][0]+income[i][1]+income[i][2]+income[i][3]));
    }
}

Codeium: Refactor | Explain | Generate Javadoc | X
public static void showStok(){
    System.out.println("Toko Bunga RoyalGarden 4");
    System.out.println("Jenis \t\t Stok Awal \t Rusak/Mati \t Stok Akhir");
    for (int i=0; i<namaBunga.length; i++){
        System.out.println(namaBunga[i] + "\t\t\t " + tokoBunga[3][i+1] + "\t\t\t " + bungaRusak[3][i+1] + "\t\t\t " + (Integer.valueOf(tokoBunga[3][i+1]) - Integer.valueOf(bungaRusak[3][i+1])));
    }
}
```

- Hasil Akhir

```
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma> & 'C:\Program Files\Java\jdk-16.0.2\bin\java
g\Code\User\workspaceStorage\d65d9a1783773716e9a1da7df3f1a6f8\redhat.java\jdt_ws\PraktikumAlgoritma
1. Cek Pendapatan Sold Out Semua Cabang
2. Cek Stok Cabang RoyalGarden 4
3. Exit
Pilih (1/2/3) :1

      Aglonema      Keladi      Alocasia      Mawar      Total
RoyalGarden 1      750000      250000      900000      0      1900000
RoyalGarden 2      450000      550000      540000      0      1540000
RoyalGarden 3      150000      500000      600000      0      1250000
RoyalGarden 4      375000      350000      720000      0      1445000

1. Cek Pendapatan Sold Out Semua Cabang
2. Cek Stok Cabang RoyalGarden 4
3. Exit
Pilih (1/2/3) :2
Toko Bunga RoyalGarden 4
Jenis      Stok Awal      Rusak/Mati      Stok Akhir
Aglonema      5      1      4
Keladi      7      2      5
Alocasia      12      0      12
Mawar      9      5      4

1. Cek Pendapatan Sold Out Semua Cabang
2. Cek Stok Cabang RoyalGarden 4
3. Exit
Pilih (1/2/3) :3
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma>
```



## e. TUGAS 1

1. Membuat class dan deklarasi variable sesuai dengan petunjuk jobsheet

```

import java.util.Scanner;

public class Tugas108 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String plat = "";
        int index = -1;

        char[] kode = {'A', 'B', 'D', 'E', 'F', 'G', 'H', 'L', 'N', 'T'};
        char[][] kota = {
            {'B', 'A', 'N', 'T', 'E', 'N', ' ', ' ', ' ', ' ', ' ', ' '},
            {'J', 'A', 'K', 'A', 'R', 'T', 'A', ' ', ' ', ' ', ' ', ' '},
            {'B', 'A', 'N', 'D', 'U', 'N', 'G', ' ', ' ', ' ', ' ', ' '},
            {'C', 'I', 'R', 'E', 'B', 'O', 'N', ' ', ' ', ' ', ' ', ' '},
            {'B', 'O', 'G', 'O', 'R', ' ', ' ', ' ', ' ', ' ', ' '},
            {'P', 'E', 'K', 'A', 'L', 'O', 'N', 'G', 'A', 'N', ' '},
            {'S', 'E', 'M', 'A', 'R', 'A', 'N', 'G', ' ', ' ', ' ', ' '},
            {'S', 'U', 'R', 'A', 'B', 'A', 'Y', 'A', ' ', ' ', ' ', ' '},
            {'M', 'A', 'L', 'A', 'N', 'G', ' ', ' ', ' ', ' ', ' '},
            {'T', 'E', 'G', 'A', 'L', ' ', ' ', ' ', ' ', ' ', ' '}
        };
    }
}

```

2. Input plat nomor yang akan di cari

```

System.out.print(s:"Masukkan kode plat kendaraan : ");
plat = input.nextLine().toUpperCase();

```

3. Perulangan mencari index array kode sesuai dengan kode plat yang telah di input

```

for (int i = 0; i < kode.length; i++) {
    if (kode[i] == plat.charAt(index:0)) {
        index = i;
        break;
    }
}

```

4. Menampilkan nama kota dengan mencari index array kota sesuai index kode plat yang telah di cari pada Langkah sebelumnya

```

if (index != -1) {
    System.out.println("kota: " + new String(kota[index]));
} else {
    System.out.println("Kode plat tidak ditemukan!");
}

```

5. Hasil Akhir

```

PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma> java -cp 'C:\Users\fikar\AppData\Roam
ws\PraktikumAlgoritma_d082531f\bin' 'Tugas108'
Masukkan kode plat kendaraan : N
Kota: MALANG

```

## f. TUGAS 2

1. Membuat class dan deklarasi variable yang dibutuhkan

```
import java.util.Scanner;

Codeium: Explain
public class Tugas208 {
    static Scanner input = new Scanner(System.in);
    static double jarak=0, kecepatan=0, waktu=0;
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
```

2. Fungsi main berfungsi sebagai pengendali menu untuk Langkah selanjutnya

```
public static void main(String[] args) {
    System.out.println(x:"=====");
    System.out.println(x:"Selamat datang !");
    System.out.println(x:"=====");

    while (true) {
        int pilihMenu = menu();

        switch (pilihMenu) {
            case 1:
                hitungKecepatan();
                break;
            case 2:
                hitungJarak();
                break;
            case 3:
                hitungWaktu();
                break;
            case 4:
                return;
            default:
                System.out.println(x:"Pilihan anda tidak tersedia");
                break;
        }
    }
}
```

3. Fungsi menu menampilkan menu yang nantinya akan di pilih

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static int menu(){
    int pilihan = 0;

    System.out.println(x:"1. Hitung Kecepatan");
    System.out.println(x:"2. Hitung Jarak");
    System.out.println(x:"3. Hitung Waktu");
    System.out.println(x:"4. Exit");
    System.out.print(s:"Pilih Salah Satu (1/2/3/4) :");
    pilihan = input.nextInt();

    return pilihan;
}
```

4. Fungsi hitung kecepatan dimulai dari input jarak dan waktu, kemudian di kalkulasi dan hasilnya di tampilkan

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void hitungKecepatan(){
    System.out.print(s:"Masukkan Jarak (km): ");
    jarak = input.nextDouble();

    System.out.print(s:"Masukkan Waktu (jam): ");
    waktu = input.nextDouble();

    kecepatan = jarak / waktu;

    System.out.println("Kecepatan: " + kecepatan + " km/jam");
}
```

5. Fungsi hitung jarak dimulai dengan input kecepatan dan waktu, kemudian dikalkulasi dan ditampilkan hasilnya

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void hitungJarak(){
    System.out.print(s:"Masukkan Kecepatan (km/jam): ");
    kecepatan = input.nextDouble();

    System.out.print(s:"Masukkan Waktu (jam): ");
    waktu = input.nextDouble();

    jarak = kecepatan * waktu;

    System.out.println("Jarak: " + jarak + " km");
}
```

6. Fungsi hitung waktu dimulai dengan input jarak dan kecepatan kemudian dikalkulasikan dan ditampilkan hasilnya

```
Codeium: Refactor | Explain | Generate Javadoc | X
public static void hitungWaktu(){
    System.out.print(s:"Masukkan Jarak (km): ");
    jarak = input.nextDouble();

    System.out.print(s:"Masukkan Kecepatan (km/jam): ");
    kecepatan = input.nextDouble();

    waktu = jarak * kecepatan ;

    System.out.println("Waktu: " + waktu + " jam");
}
```

7. Hasil akhir

```
g:\code\user\workspace\storage\0030
=====
Selamat datang !
=====
1. Hitung Kecepatan
2. Hitung Jarak
3. Hitung Waktu
4. Exit
Pilih Salah Satu (1/2/3/4) :1
Masukkan Jarak (km): 10
Masukkan Waktu (jam): 20
Kecepatan: 0.5 km/jam

1. Hitung Kecepatan
2. Hitung Jarak
3. Hitung Waktu
4. Exit
Pilih Salah Satu (1/2/3/4) :2
Masukkan Kecepatan (km/jam): 5
Masukkan Waktu (jam): 3
Jarak: 15.0 km

1. Hitung Kecepatan
2. Hitung Jarak
3. Hitung Waktu
4. Exit
Pilih Salah Satu (1/2/3/4) :3
Masukkan Jarak (km): 5
Masukkan Kecepatan (km/jam): 10
Waktu: 50.0 jam
```