

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN
PERCABANGAN IF DAN SWITCH DALAM PEMOGRAMAN JAVA



disusun Oleh:
NIDA TSABITA Ariba
2511532025
Dosen Pengampu:
Dr. WAHYUDI, S.T, M.T
Asisten Praktikum:
AUFAN TAUFIQURRAHMAN

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

DAFTAR ISI

DAFTAR ISI.....	i
BAB I	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat.....	2
BAB II.....	3
2.1 Program Ifelse1	3
2.1.1 Uraian Kode Program Ifelse1	3
2.1.2 Langkah Kerja Program Ifelse1	3
2.1.3 Analisis Hasil Program Ifelse1	5
2.2 Program LatIf1	5
2.2.1 Uraian Kode Program LatIf1	5
2.2.2 Langkah Kerja Program LatIf1	6
2.2.3 Analisis Hasil Program LatIf1	8
2.3 Program MultiIf.....	8
2.3.1 Uraian Kode Program MultiIf.....	8
2.3.2 Langkah Kerja Program MultiIf	9
2.3.3 Analisis Hasil Program MultiIf.....	11
2.4 Program Nilai	12
2.4.1 Uraian Kode Program Nilai	12
2.4.2 Langkah Kerja Program Nilai.....	13
2.4.3 Analisis Hasil Program MultiIf.....	15
2.5 Program NamaBulan	15
2.5.1 Uraian Kode Program NamaBulan	15
2.5.2 Langkah Kerja Program Ifelse1	17
2.5.3 Analisis Hasil Program NamaBulan	19
BAB III	20

BAB I

PENDAHULUAN

1.1 Latar Belakang

Java merupakan salah satu bahasa pemrograman yang populer dan banyak digunakan di dunia teknologi. Dalam proses belajar Java, memahami struktur percabangan adalah hal dasar yang sangat penting. Percabangan digunakan untuk membuat keputusan dalam program berdasarkan kondisi tertentu. Dengan percabangan, program bisa menampilkan output yang berbeda sesuai dengan input pengguna.

Beberapa bentuk percabangan yang digunakan dalam praktikum ini antara lain:

1. If digunakan untuk mengeksekusi suatu perintah hanya jika kondisi bernilai benar.
2. If-Else digunakan untuk memilih dua kemungkinan kondisi, benar atau salah.
3. If-Else-If digunakan untuk memeriksa beberapa kondisi secara berurutan.
4. Switch-Case digunakan sebagai alternatif if-else-if ketika terdapat banyak pilihan kondisi.

Dengan mempelajari percabangan ini, dapat dipahami bagaimana sebuah program bisa mengambil keputusan secara logis sesuai kebutuhan. Praktikum ini dilakukan agar bisa mencoba langsung penerapan percabangan dalam program Java sederhana.

1.2 Tujuan

Tujuan dari praktikum ini adalah sebagai berikut:

1. Memahami konsep dasar percabangan dalam Java.
2. Mempelajari penggunaan if, if-else, dan if-else-if untuk menentukan hasil berdasarkan kondisi.
3. Mengetahui cara kerja switch-case sebagai alternatif untuk banyak kondisi.

4. Melatih kemampuan dalam membuat program sederhana yang melibatkan keputusan logika.

1.3 Manfaat

Manfaat yang diperoleh dari praktikum ini antara lain:

1. Menambah pemahaman tentang percabangan dalam bahasa Java.
2. Membiasakan diri menggunakan percabangan untuk menyelesaikan masalah pemrograman sederhana.
3. Melatih kemampuan berpikir logis dalam menentukan alur program.
4. Memberikan dasar untuk memahami struktur program yang lebih kompleks.

BAB II

PEMBAHASAN

2.1 Program Ifelse1

2.1.1 Uraian Kode Program Ifelse1

Kode program yang di praktikkan adalah sebagai berikut:

```
package Pekan4;
import java.util.Scanner;

public class Ifelse1 {
    public static void main(String[] args) {
        double IPK;
        Scanner input = new Scanner(System.in);
        System.out.print("Input IPK Anda: ");
        IPK = input.nextDouble();
        input.close();
        if (IPK > 2.75) {
            System.out.println("Anda lulus Sangat Memuaskan
dengan IPK " + IPK);
        }
        else {
            System.out.println("Anda tidak lulus");
        }
    }
}
```

Program ini digunakan untuk menentukan kelulusan mahasiswa berdasarkan nilai IPK yang dimasukkan pengguna. Jika nilai IPK lebih dari 2.75, maka mahasiswa dinyatakan lulus sangat memuaskan. Jika tidak, maka mahasiswa dinyatakan tidak lulus.

2.1.2 Langkah Kerja Program Ifelse1

1. Membuat *package* dan *import library*

```
package Pekan4;
import java.util.Scanner;
```

- package Pekan4; menandakan bahwa program berada di dalam *package* bernama Pekan4. *Package* digunakan untuk mengelompokkan kelas agar lebih terorganisir.
- import java.util.Scanner; Digunakan untuk mengimpor kelas *Scanner* dari *library* Java. *Scanner* berfungsi untuk membaca input dari pengguna.

2. Membuat kelas utama

```
public class Ifelse1 {
```

Mendefinisikan kelas dengan nama Ifelse1. Semua kode program ditulis di dalam kelas ini.

3. Membuat *method main*

```
public static void main(String[]args) {
```

Method utama yang menjadi titik awal eksekusi program Java.

4. Deklarasi variabel

```
double IPK;
```

Variabel IPK bertipe double karena IPK biasanya berupa bilangan desimal (misalnya 3.25).

5. Membuat objek *Scanner*

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

Membuat objek keyboard dari kelas *Scanner* untuk membaca input pengguna.

6. Input nilai dari pengguna

```
System.out.print("Input IPK Anda: ");  
IPK = input.nextDouble();
```

Program meminta pengguna memasukkan nilai IPK, lalu menyimpannya ke variabel IPK.

7. Menutup *Scanner*

```
input.close();
```

Menutup objek *Scanner* agar tidak terjadi kebocoran *resource*.

8. Membuat percabangan If-Else

```
if (IPK>2.75) {  
    System.out.println("Anda lulus Sangat  
Memuaskan dengan IPK " + IPK);  
}  
else {  
    System.out.println("Anda tidak lulus");  
}
```

```
}
```

Jika IPK lebih dari 2.75 program menampilkan pesan lulus sangat memuaskan.

Jika tidak program menampilkan pesan tidak lulus.

9. Kurung kurawal penutup

```
}  
}
```

Kurung kurawal (}) pertama : menutup method main.

Kurung kurawal (}) kedua : menutup kelas Ifelse1.

2.1.3 Analisis Hasil Program Ifelse1

Contoh Output 1 (IPK > 2.75):

```
Input IPK Anda: 4.0  
Anda lulus Sangat Memuaskan dengan IPK 4.0
```

Contoh Output 2 (IPK <= 2.75):

```
Input IPK Anda: 2.7  
Anda tidak lulus
```

Analisis Hasil:

Hasil program menunjukkan bahwa percabangan if-else dapat digunakan untuk menentukan status kelulusan mahasiswa berdasarkan IPK. Jika IPK lebih dari 2.75, maka program menampilkan pesan “lulus sangat memuaskan”, sedangkan jika kurang atau sama dengan 2.75, maka ditampilkan pesan “tidak lulus”. Misalnya, jika IPK 4.0 output menampilkan lulus, dan jika IPK 2.7 output menampilkan tidak lulus. Dengan demikian, logika percabangan if-else dapat mengklasifikasi kondisi secara sederhana dan tepat.

2.2 Program LatIf1

2.2.1 Uraian Kode Program LatIf1

Kode program yang di praktikkan adalah sebagai berikut:

```
package Pekan4;  
import java.util.Scanner;
```

```
public class LatIf1 {

    public static void main(String[] args) {
        double IPK;
        Scanner Input = new Scanner(System.in);
        System.out.print("Input IPK anda: ");
        IPK = Input.nextDouble();
        Input.close();
        if (IPK>2.75) {
            System.out.println("Anda Lulus Sangat Memuaskan
dengan IPK " + IPK);
        }
    }
}
```

Program ini digunakan untuk menentukan kelulusan mahasiswa berdasarkan nilai IPK yang dimasukkan pengguna. Jika IPK lebih dari 2.75, maka mahasiswa dinyatakan lulus sangat memuaskan. Berbeda dengan program sebelumnya (Ifelse1), pada program ini tidak ada blok else, sehingga jika $IPK \leq 2.75$ maka tidak ada output yang ditampilkan.

2.2.2 Langkah Kerja Program LatIf1

1. Membuat *package* dan *import library*

```
package Pekan4;
import java.util.Scanner;
```

- package Pekan4; menandakan bahwa program berada di dalam *package* bernama Pekan4. *Package* digunakan untuk mengelompokkan kelas agar lebih terorganisir.
- import java.util.Scanner; Digunakan untuk mengimpor kelas *Scanner* dari *library* Java. *Scanner* berfungsi untuk membaca input dari pengguna.

2. Membuat kelas utama

```
public class LatIf1 {
```

Mendefinisikan kelas dengan nama LatIf1. Semua kode program ditulis di dalam kelas ini.

3. Membuat *method main*

```
public static void main(String[]args) {
```


Method utama yang menjadi titik awal eksekusi program Java.

4. Deklarasi variabel

```
double IPK;
```

Variabel IPK bertipe double karena IPK biasanya berupa bilangan desimal (misalnya 3.25).

5. Membuat objek *Scanner*

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

Membuat objek keyboard dari kelas *Scanner* untuk membaca input pengguna.

6. Input nilai dari pengguna

```
System.out.print("Input IPK Anda: ");  
IPK = input.nextDouble();
```

Program meminta pengguna memasukkan nilai IPK, lalu menyimpannya ke variabel IPK.

7. Menutup *Scanner*

```
input.close();
```

Menutup objek *Scanner* agar tidak terjadi kebocoran *resource*.

8. Membuat percabangan If

```
if (IPK > 2.75) {  
    System.out.println("Anda Lulus Sangat  
Memuaskan dengan IPK " + IPK);  
}
```

Jika nilai IPK lebih dari 2.75, program menampilkan pesan bahwa mahasiswa lulus sangat memuaskan.

Jika nilai IPK ≤ 2.75 , program tidak menampilkan output apapun (karena tidak ada blok else).

9. Kurung kurawal penutup

```
}  
}
```

Kurung kurawal (}) pertama : menutup method main.

Kurung kurawal (}) kedua : menutup kelas LatIf1.

2.2.3 Analisis Hasil Program LatIf1

Contoh Output:

```
Input IPK anda: 4.0
Anda Lulus Sangat Memuaskan dengan IPK 4.0
```

Analisis Hasil:

Program LatIf1 hanya menggunakan satu kondisi if tanpa else. Hasil program menampilkan pesan “lulus sangat memuaskan” hanya jika IPK lebih dari 2.75. Namun jika IPK kurang atau sama dengan 2.75, tidak ada output yang muncul karena tidak ada perintah lain setelah if. Hal ini menunjukkan bahwa percabangan if tunggal hanya mengeksekusi kondisi tertentu dan mengabaikan kondisi lain. Program ini memperlihatkan bahwa logika sederhana bisa cukup dalam kasus tertentu.

2.3 Program MultiIf

2.3.1 Uraian Kode Program MultiIf

Kode program yang di praktikkan adalah sebagai berikut:

```
package Pekan4;
import java.util.Scanner;

public class MultiIf {
    public static void main(String[] args) {
        int umur;
        char sim;
        Scanner a = new Scanner(System.in);
        System.out.print("Input umur anda: ");
        umur = a.nextInt();
        System.out.print("Apakah anda sudah punya SIM? ");
        sim = a.next().charAt(0);
        a.close();
        if ((umur >= 17)&&(sim == 'y')) {
            System.out.print("Anda sudah dewasa dan boleh
bawa motor ");
        }
        if ((umur >= 17)&&(sim != 'y')) {
            System.out.print("Anda sudah dewasa tetapi tidak
boleh bawa motor ");
        }
        if ((umur < 17)&&(sim != 'y')) {
```

```

        System.out.print("Anda belum cukup umur bawa
motor ");
    }
    if ((umur < 17)&&(sim == 'y')) {
        System.out.print("Anda belum cukup umur punya SIM
");
    }
}
}
}

```

Program ini digunakan untuk menentukan status umur dan kepemilikan SIM seseorang. Program mengecek kombinasi dua kondisi: umur (≥ 17 atau < 17) dan kepemilikan SIM ('y' = sudah punya SIM, selain itu dianggap tidak). Dari kombinasi kondisi ini, ada 4 kemungkinan output.

2.3.2 Langkah Kerja Program MultiIf

1. Membuat *package* dan *import library*

```

package Pekan4;
import java.util.Scanner;

```

- package Pekan4; menandakan bahwa program berada di dalam *package* bernama Pekan4. *Package* digunakan untuk mengelompokkan kelas agar lebih terorganisir.
- import java.util.Scanner; Digunakan untuk mengimpor kelas *Scanner* dari *library* Java. *Scanner* berfungsi untuk membaca input dari pengguna.

2. Membuat kelas utama

```

public class MultiIf {

```

Mendefinisikan kelas dengan nama MultiIf. Semua kode program ditulis di dalam kelas ini.

3. Membuat *method main*

```

    public static void main(String[]args) {

```

Method utama yang menjadi titik awal eksekusi program Java.

4. Deklarasi variabel

```

        int umur;
        char sim;

```

umur digunakan menyimpan umur dalam angka bulat (int).

sim digunakan menyimpan input berupa karakter ('y' atau lainnya).

5. Membuat objek *Scanner*

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

Membuat objek keyboard dari kelas *Scanner* untuk membaca input pengguna.

6. Input data dari pengguna

```
System.out.print("Input umur anda: ");  
umur = a.nextInt();  
System.out.print("Apakah anda sudah punya SIM? ");  
sim = a.next().charAt(0);
```

Program meminta pengguna memasukkan umur, lalu menanyakan apakah sudah punya SIM ('y' untuk ya, karakter lain untuk tidak).

7. Menutup *Scanner*

```
a.close();
```

Menutup objek *Scanner* agar tidak terjadi kebocoran *resource*.

8. Membuat percabangan multi-if

```
System.out.print("Anda sudah dewasa dan  
boleh bawa motor ");  
}  
if ((umur >= 17)&&(sim != 'y')) {  
    System.out.print("Anda sudah dewasa tetapi  
tidak boleh bawa motor ");  
}  
if ((umur < 17)&&(sim != 'y')) {  
    System.out.print("Anda belum cukup umur  
bawa motor ");  
}  
if ((umur < 17)&&(sim == 'y')) {  
    System.out.print("Anda belum cukup umur  
punya SIM ");  
}
```

Ada 4 kondisi berbeda:

Jika `((umur >= 17)&&(sim == 'y'))` maka program akan mengeluarkan output 'Anda sudah dewasa dan boleh membawa motor'.

Jika `((umur >= 17)&&(sim != 'y'))` maka program akan mengeluarkan output 'Anda sudah dewasa tetapi tidak boleh bawa motor'.

Jika `((umur < 17)&&(sim != 'y'))` maka program akan mengeluarkan ouput 'Anda belum cukup umur dan belum cukup umur bawa motor'.

Jika `((umur < 17)&&(sim == 'y'))` maka program akan mengeluarkan ouput 'Anda belum cukup umur punya SIM'.

9. Kurung kurawal penutup

```
}  
}
```

Kurung kurawal `()` pertama : menutup method main.

Kurung kurawal `()` kedua : menutup kelas MultiIf.

2.3.3 Analisis Hasil Program MultiIf

Contoh Output 1:

```
Input umur anda: 17  
Apakah anda sudah puhnya SIM? y  
Anda sudah dewasa dan boleh bawa motor
```

Contoh Output 2:

```
Input umur anda: 17  
Apakah anda sudah puhnya SIM? belum  
Anda sudah dewasa tetapi tidak boleh bawa motor
```

Contoh Output 3:

```
Input umur anda: 11  
Apakah anda sudah puhnya SIM? belum  
Anda belum cukup umur bawa motor
```

Contoh Output 4:

```
Input umur anda: 11
Apakah anda sudah punya SIM? y
Anda belum cukup umur punya SIM
```

Analisis Hasil:

Hasil program menunjukkan bahwa percabangan multi-if dapat digunakan untuk menguji lebih dari satu kondisi. Program mengecek kombinasi antara umur (≤ 17 atau < 17) dan kepemilikan SIM ('y' atau tidak). Dari kombinasi ini, program menghasilkan empat kemungkinan output berbeda. Misalnya, umur 17 dengan SIM "boleh bawa motor", umur 17 tanpa SIM "dewasa tapi tidak boleh bawa motor", umur 11 tanpa SIM "belum cukup umur bawa motor", dan umur 11 dengan SIM "belum cukup umur punya SIM". Program ini membuktikan bahwa multi-if bisa digunakan untuk menyelesaikan permasalahan dengan banyak kondisi.

2.4 Program Nilai

2.4.1 Uraian Kode Program Nilai

Kode program yang di praktikkan adalah sebagai berikut:

```
package Pekan4;
import java.util.Scanner;

public class Nilai {
    public static void main(String[] args) {
        int nilai;
        Scanner input = new Scanner(System.in);
        System.out.print("Input nilai angka: ");
        nilai = input.nextInt();
        input.close();

        if (nilai >= 81) {
            System.out.print("A");
        } else if (nilai >= 70) {
            System.out.print("B");
        } else if (nilai >= 60) {
            System.out.print("C");
        } else if (nilai >= 50) {
            System.out.print("D");
        } else {
            System.out.print("E");
        }
    }
}
```

Program ini digunakan untuk menentukan nilai huruf berdasarkan input nilai angka yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan sebuah nilai angka melalui keyboard, lalu nilai tersebut disimpan ke dalam variabel bertipe data *int* bernama *nilai*. Setelah itu, program menggunakan struktur percabangan *if-else-if* untuk mengklasifikasikan nilai ke dalam huruf A, B, C, D, atau E. Jika nilai ≥ 81 maka hasilnya A. Jika nilai ≥ 70 maka hasilnya B. Jika nilai ≥ 60 maka hasilnya C. Jika nilai ≥ 50 maka hasilnya D. Jika nilainya di bawah 50 maka akan ditampilkan E.

2.4.2 Langkah Kerja Program Nilai

1. Membuat *package* dan *import library*

```
package Pekan4;  
import java.util.Scanner;
```

- `package Pekan4;` menandakan bahwa program berada di dalam *package* bernama *Pekan4*. *Package* digunakan untuk mengelompokkan kelas agar lebih terorganisir.
- `import java.util.Scanner;` Digunakan untuk mengimpor kelas *Scanner* dari *library* Java. *Scanner* berfungsi untuk membaca input dari pengguna.

2. Membuat kelas utama

```
public class Nilai {
```

Mendefinisikan kelas dengan nama *Nilai*. Semua kode program ditulis di dalam kelas ini.

3. Membuat *method main*

```
public static void main(String[]args) {
```

Method utama yang menjadi titik awal eksekusi program Java.

4. Deklarasi variabel

```
int nilai;
```

Variabel *nilai* bertipe *int* karena nilai ujian berupa bilangan bulat.

5. Membuat objek *Scanner*

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

Membuat objek keyboard dari kelas *Scanner* untuk membaca input pengguna.

6. Input nilai dari pengguna

```
System.out.print("Input nilai angka: ");  
nilai = input.nextInt();
```

Program meminta pengguna memasukkan nilai angka, kemudian menyimpannya ke variabel nilai.

7. Menutup *Scanner*

```
input.close();
```

Menutup objek *Scanner* agar tidak terjadi kebocoran *resource*.

8. Membuat percabangan if-else-if

```
if (nilai >= 81) {  
    System.out.print("A");  
} else if (nilai >= 70) {  
    System.out.print("B");  
} else if (nilai >= 60) {  
    System.out.print("C");  
} else if (nilai >= 50) {  
    System.out.print("D");  
} else {  
    System.out.print("E");  
}
```

Jika nilai ≥ 81 maka ditampilkan huruf A.

Jika nilai ≥ 70 maka ditampilkan huruf B.

Jika nilai ≥ 60 maka ditampilkan huruf C.

Jika nilai ≥ 50 maka ditampilkan huruf D.

Jika tidak memenuhi semua kondisi di atas, maka ditampilkan huruf E.

9. Kurung kurawal penutup

```
}  
}
```


Kurung kurawal (}) pertama : menutup method main.

Kurung kurawal (}) kedua : menutup kelas Nilai.

2.4.3 Analisis Hasil Program MultIf

Contoh Output 1 (nilai ≥ 81) :

```
Input nilai angka: 81  
A
```

Contoh Output 2 (nilai ≥ 70):

```
Input nilai angka: 75  
B
```

Contoh Output 3 (nilai ≥ 60):

```
Input nilai angka: 61  
C
```

Contoh Output 4 (nilai ≥ 50):

```
Input nilai angka: 59  
D
```

Contoh Output 5 (nilai < 50):

```
Input nilai angka: 23  
E
```

Analisis Hasil:

Program Nilai menggunakan percabangan if-else-if untuk menentukan nilai huruf berdasarkan nilai angka yang diinputkan. Misalnya, input 85 menghasilkan huruf A, input 72 menghasilkan B, input 65 menghasilkan C, input 55 menghasilkan D, dan input 40 menghasilkan E. Analisis menunjukkan bahwa program bekerja sesuai kriteria penilaian. Karena hanya satu kondisi yang dieksekusi, percabangan if-else-if efektif untuk mengelompokkan nilai berdasarkan rentang tertentu.

2.5 Program NamaBulan

2.5.1 Uraian Kode Program NamaBulan

Kode program yang di praktikkan adalah sebagai berikut:

```
package Pekan4;  
import java.util.Scanner;
```

```
public class NamaBulan {  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Masukkan angka bulan(1-12): ");  
        int bulan = scanner.nextInt();  
        switch (bulan) {  
            case 1:  
                System.out.println("Januari");  
                break;  
            case 2:  
                System.out.println("Februari");  
                break;  
            case 3:  
                System.out.println("Maret");  
                break;  
            case 4:  
                System.out.println("April");  
                break;  
            case 5:  
                System.out.println("Mei");  
                break;  
            case 6:  
                System.out.println("Juni");  
                break;  
            case 7:  
                System.out.println("Juli");  
                break;  
            case 8:  
                System.out.println("Agustus");  
                break;  
            case 9:  
                System.out.println("September");  
                break;  
            case 10:  
                System.out.println("Oktober");  
                break;  
            case 11:  
                System.out.println("November");  
                break;  
            case 12:  
                System.out.println("Desember");  
                break;  
            default:  
                System.out.println("Angka tidak valid");  
        }  
        scanner.close();  
    }  
}
```

Program ini digunakan untuk menampilkan nama bulan berdasarkan angka yang dimasukkan pengguna. Jika pengguna memasukkan angka 1, program akan menampilkan Januari, jika 2 akan menampilkan Februari, dan seterusnya hingga 12 untuk Desember. Jika angka yang dimasukkan tidak berada pada rentang 1–12, maka program akan menampilkan pesan "Angka tidak valid".

2.5.2 Langkah Kerja Program NamaBulan

10. Membuat *package* dan *import library*

```
package Pekan4;  
import java.util.Scanner;
```

- `package Pekan4;` menandakan bahwa program berada di dalam *package* bernama Pekan4. *Package* digunakan untuk mengelompokkan kelas agar lebih terorganisir.
- `import java.util.Scanner;` Digunakan untuk mengimpor kelas *Scanner* dari *library* Java. *Scanner* berfungsi untuk membaca input dari pengguna.

11. Membuat kelas utama

```
public class NamaBulan {
```

Mendefinisikan kelas dengan nama NamaBulan. Semua kode program ditulis di dalam kelas ini.

12. Membuat *method main*

```
public static void main(String[]args) {
```

Method utama yang menjadi titik awal eksekusi program Java.

13. Membuat objek *Scanner*

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

Membuat objek keyboard dari kelas *Scanner* untuk membaca input pengguna.

14. Input angka bulan dari pengguna

```
System.out.print("Masukkan angka bulan(1-12): ");  
int bulan = scanner.nextInt();
```

Program meminta pengguna memasukkan angka bulan, kemudian menyimpannya ke variabel bulan.

15. Menggunakan struktur switch-case

```
System.out.print("Masukkan angka bulan(1-12): ");
int bulan = scanner.nextInt();
switch (bulan) {
    case 1:
        System.out.println("Januari");
        break;
    case 2:
        System.out.println("Februari");
        break;
    case 3:
        System.out.println("Maret");
        break;
    case 4:
        System.out.println("April");
        break;
    case 5:
        System.out.println("Mei");
        break;
    case 6:
        System.out.println("Juni");
        break;
    case 7:
        System.out.println("Juli");
        break;
    case 8:
        System.out.println("Agustus");
        break;
    case 9:
        System.out.println("September");
        break;
    case 10:
        System.out.println("Oktober");
        break;
    case 11:
        System.out.println("November");
        break;
    case 12:
        System.out.println("Desember");
        break;
    default:
        System.out.println("Angka tidak valid");
}
```

switch digunakan untuk memilih salah satu dari banyak kemungkinan berdasarkan nilai bulan. Jika bulan = 1, maka ditampilkan Januari, jika bulan = 2, maka ditampilkan Februari,

dan seterusnya. Jika input tidak sesuai (misalnya 13 atau 0), program masuk ke default dan menampilkan "Angka tidak valid".

16. Menutup *Scanner*

```
scanner.close();
```

Menutup objek *Scanner* agar tidak terjadi kebocoran *resource*.

17. Kurung kurawal penutup

```
}  
}
```

Kurung kurawal (}) pertama : menutup method main.

Kurung kurawal (}) kedua : menutup kelas NamaBulan.

2.5.3 Analisis Hasil Program NamaBulan

Contoh Output 1 (valid):

```
Masukkan angka bulan(1-12): 12  
Desember
```

Contoh Output 2 (tidak valid):

```
Masukkan angka bulan(1-12): 23  
Angka tidak valid
```

Analisis Hasil:

Program NamaBulan bekerja dengan baik dalam menampilkan nama bulan sesuai angka yang dimasukkan pengguna. Jika pengguna memberikan input angka antara 1 sampai 12, program akan menampilkan nama bulan yang tepat, misalnya input 1 menghasilkan Januari dan input 9 menghasilkan September. Jika pengguna memasukkan angka di luar rentang, seperti 0 atau 15, program menampilkan pesan s, sehingga kesalahan input dapat ditangani dengan baik.

BAB III

KESIMPULAN

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa program-program Java yang dipraktikkan berhasil dijalankan sesuai dengan teori. Program OperatorAritmatika mampu melakukan perhitungan dasar matematika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Program Ifelse1 dapat menentukan kelulusan mahasiswa berdasarkan nilai IPK menggunakan struktur if-else, sedangkan program LatIf1 menunjukkan penerapan percabangan sederhana dengan kondisi tunggal. Program MultiIf mampu mengecek lebih dari satu kondisi sekaligus, yaitu umur dan kepemilikan SIM, sehingga menghasilkan empat kemungkinan keluaran. Program Nilai berhasil mengonversi input angka menjadi nilai huruf (A-E) dengan memanfaatkan struktur if-else-if. Selain itu, program NamaBulan dapat menampilkan nama bulan berdasarkan angka input dengan memanfaatkan switch-case, sekaligus memberikan informasi ketika input tidak sesuai. Secara keseluruhan, praktikum ini menunjukkan bahwa penggunaan percabangan dalam Java sangat penting untuk pengambilan keputusan dalam sebuah program.