

LAPORAN PRAKTIKUM PROGRAM DASAR

Diajukan untuk memenuhi UAS mata kuliah Pemrograman Dasar Semester Ganjil

Tahun Ajaran 2021-2022

Dosen Pengampu

Rizdania S.T M.Kom



Disusun oleh:

AHMAD RUDIANTO (21157201114)

PROGRAM STUDI ILMU KOMPUTER 2021 B
FAKULTAS TEKNOLOGI DAN SAINS
SEKOLAH TINGGI KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS PGRI WIRANEGARA
2022

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa., Selawat serta salam semoga senantiasa dilimpahkan kepada Nabi muhammad SAW, beserta keluarganya, sahabatnya tabu'innya dan seluruh umatnya yang istiqomah mengikuti tuntunan dan teladan sampai akhir zaman.

Atas berkat Rahmat Tuhan Yang Maha Esa ,Kami dapat menyelesaikan tugas makalah yang berjudul " Laporan Praktikum Program Dasar " dengan tepat waktu.

Makalah disusun untuk memenuhi UAS Mata Kuliah Praktikum Pemrograman Dasar . saya selaku penulis mengucapkan terima kasih kepada Ibu Rizdania S.T M.Kom selaku guru Mata Kuliah Praktikum Pemrograman Dasar. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu diselesaikannya makalah ini.

Bagi saya sebagai penyusun merasa bahwa masih banyak kekurangan dalam penyusunan makalah ini karena keterbatasan pengetahuan dan pengalaman kami. Untuk itu kami sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan.

Pasuruan,11 Januari 2022

DAFTAR ISI

DAFTAR ISI	3
BAB 1	4
1.1 Pengenalan java	4
1.2 Instalasi JDK.....	4
1.3 Editor Java.....	4
BAB 2	6
2.1 Membuat kelas Sederhana	6
2.2 VARIABEL, DATA DAN OPERATOR	7
2.3 Pemilihan 1	12
2.3 Percabangan dalam percabangan (Nested if)	16
2.4 ARRAY 1	20
2.5 PERULANGAN.....	25
2.6 Perulangan 2	30
2.7 Array 2.....	33

BAB 1

PENGENALAN JAVA

1.1 Pengenalan java

Java adalah salah satu bahasa pemrograman yang bersifat open source yang merupakan produk dari Sun Microsystem. Bahasa Java adalah bahasa modern yang telah diterima masyarakat komputasi dunia. Hampir semua perusahaan perangkat lunak dan komputer besar mendukung dan mengembangkan aplikasi sistem berbasis Java. Java adalah salah satu bahasa pemrograman Multiplatform (Bisa berjalan di berbagai macam system operasi) karena pada dasarnya java mempunyai Jre (java runtime environment) atau dapat kita artikan sebagai mesin tersendiri untuk mengeksekusi binary code hasil dari kompilasi program yang telah kita buat, berbeda dengan bahasa pemrograman vb, c++ yang memanfaatkan komponen system dalam Windows untuk mengeksekusi binary code hasil kompilasi program.

Java adalah salah satu bahasa pemrograman Multiplatform (Bisa berjalan di berbagai macam system operasi) karena pada dasarnya java mempunyai Jre (java runtime environment) atau dapat kita artikan sebagai mesin tersendiri untuk mengeksekusi binary code hasil dari kompilasi program yang telah kita buat, berbeda dengan bahasa pemrograman vb, c++ yang memanfaatkan komponen system dalam Windows untuk mengeksekusi binary code hasil kompilasi program. Pada Tahun 1996, Sun Microsystem secara resmi merilis versi awal Java yang kemudian terus berkembang hingga muncul JDK 1.1 (Java Development Kit versi 1.1). Perkembangan terus dilakukan hingga muncul versi baru yang disebut Java 2. Perubahan utama antara versi sebelumnya adalah adanya Swing yang merupakan teknologi Graphical User Interface (GUI) yang mampu menghasilkan aplikasi desktop yang benar-benar baik. Untuk mengembangkan aplikasi berbasis java, kita memerlukan Java Development Kit(JDK), saat ini java telah mencapai versi 2 yang dikenal dengan java 2 platform. Java 2 platform, dibagi menjadi tiga kategori yaitu:

1. Java 2 Standard Edition (J2SE)
2. Java 2 Enterprise Edition(J2EE)
3. Java 2 Micro Edition(J2ME)

1.2 Instalasi JDK

Sebelum anda membuat aplikasi dengan java maka anda harus menginstal software Bahasa pemrograman java. java dapat anda dapatkan dengan cara download di alamat link berikut <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Kemudian setelah anda mendownloadnya maka mulai dengan menginstal, pada dasarnya proses instalasi java sangat mudah yaitu dengan cara mengklik ganda pada filemaster javakemudian ikuti prosedur yang diperintahkan.

1.3 Editor Java

Ada banyak editor yang bisa kita gunakan untuk menuliskan source code bahasa pemrograman java, bahkan semua editor dimungkinkan selama editor tersebut mampu menyimpan file dengan ekstensi .java yang merupakan ekstensi dari file untuk menyimpan source code bahasa pemrograman java.

BAB 2

PEMBAHASAN

2.1 Membuat kelas Sederhana

- Kelas sederhana percoaan ke 1

Membuat kelas sederhana pada pemrograman java, pada neatbeans klik file dan pilih new project, pilih java dan java applications kemudian next, isi project name dengan PemrogramanPertama lalu klik finish.

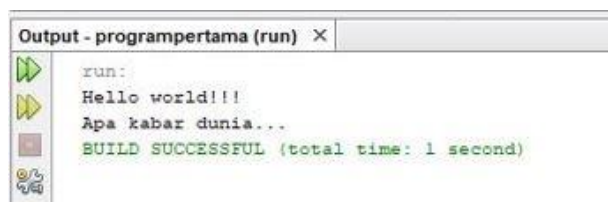
Dibawah ini hasil program saya.

```
package programpertama;

/**
 *
 * @author RUDIYANTO
 */
public class Programpertama {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello world!!!");
        System.out.println("Apa kabar dunia...");
    }
}
```

Untuk menjalankan program klik kanan lalu Run file, maka hasilnya ;



```
Output - programpertama (run) X
run:
Hello world!!!
Apa kabar dunia...
BUILD SUCCESSFUL (total time: 1 second)
```

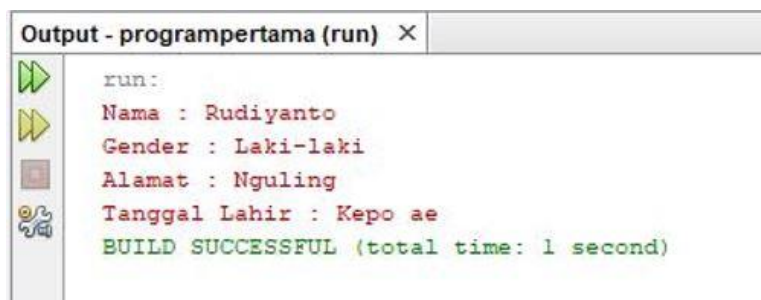
- kelas sederhana percobaan ke 2

Buat proyek baru di netbeans dan sebuah class lalu ketik ulang listing program berikut ini!

```
package programpertama;

/**
 *
 * @author RUDIYANTO
 */
public class HaloDunia {
    public static void main(String[] args) {
        System.err.println("Nama : Rudiyanto");
        System.err.println("Gender : Laki-laki");
        System.err.println("Alamat : Nguling");
        System.err.println("Tanggal Lahir : Kepo ae");
    }
}
```

Untuk menjalankan program klik kanan lalu Run file, maka hasilnya ;



The screenshot shows the 'Output - programpertama (run)' window in NetBeans. It displays the output of the Java program, which prints personal information to the standard error stream. The output is as follows:

```
run:
Nama : Rudiyanto
Gender : Laki-laki
Alamat : Nguling
Tanggal Lahir : Kepo ae
BUILD SUCCESSFUL (total time: 1 second)
```

2.2 VARIABEL, DATA DAN OPERATOR

➤ Keyword dalam Java

Keyword dalam Java Kata kunci adalah identifier yang telah dipesan untuk didefinisikan sebelumnya oleh Java untuk tujuan tertentu. Anda tidak dapat menggunakan keyword sebagai nama variabel, class, method.

➤ Tipe Data Primitif

Tipe Data Primitif Bahasa pemrograman Java mendefinisikan delapan tipe data primitif. Mereka diantaranya adalah boolean (untuk bentuk logika), char (untuk bentuk tekstual), byte, short, int, long (integral), double and float (floating point).

➤ Variabel Variabel

Variabel Variabel adalah item yang digunakan data untuk menyimpan pernyataan objek. Variabel memiliki tipe data dan nama. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri. Nama variabel harus mengikuti aturan untuk identifier. Berikut Aturan penamaan variable :

- Diawali dengan – huruf/abjad – karakter mata uang – underscore (_)
- Terdiri dari huruf/abjad, angka dan underscore
- Tidak boleh mengandung karakter khusus atau spasi
- Tidak boleh diawali dengan angka

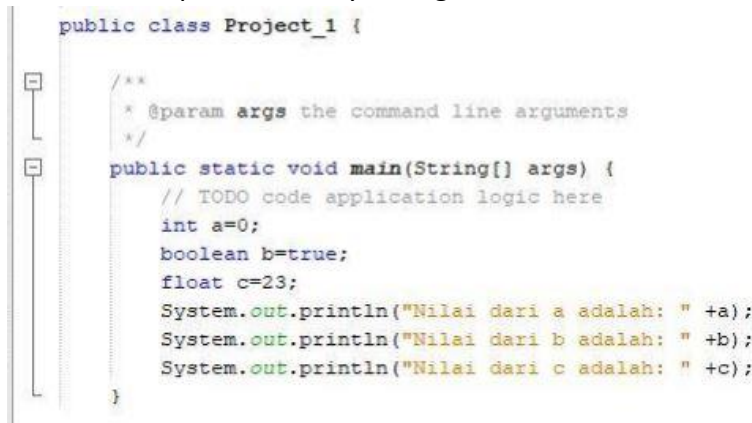
➤ Deklarasi dan Inisialisasi Variabel

Untuk deklarasi variabel adalah sebagai berikut, [=initial value]; Nilainya berada diantara <> adalah nilai yang disyaratkan, sementara nilai dalam tanda [] bersifat optional sesuai dengan tipe data yang dipakai untuk deklarasi variabel.

➤ Menampilkan Data Variabel

Untuk mengeluarkan nilai dari variabel yang diinginkan, kita dapat menggunakan perintah sebagai berikut: System.out.println() atau System.out.print().

- Contoh hasil praktikum saya dengan nama Contohvariabel1 :



```
public class Project_1 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int a=0;
        boolean b=true;
        float c=23;
        System.out.println("Nilai dari a adalah: " +a);
        System.out.println("Nilai dari b adalah: " +b);
        System.out.println("Nilai dari c adalah: " +c);
    }
}
```

Jika di Run maka hasilnya :



```
run:
Nilai dari a adalah: 0
Nilai dari b adalah: true
Nilai dari c adalah: 23.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

➤ Teknik Konversi Data

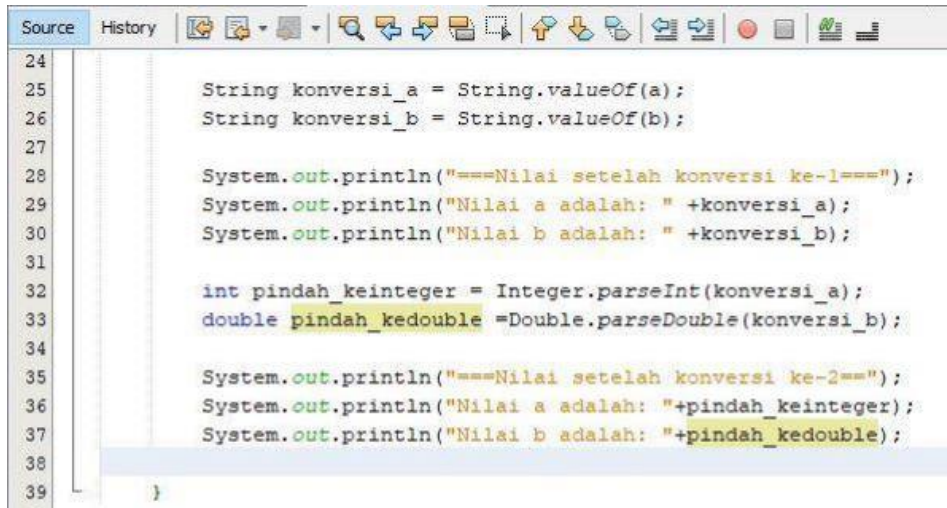
Konversi adalah cara untuk memberikan nilai suatu data dari satu tipe data ke tipe data yang lain. Yang perlu diketahui, bahwa konversi data dilakukan jika kita membutuhkan suatu tipe data yang sama untuk menjalankan sebuah proses, jika data tersebut sudah sama tipe datanya maka tidak perlu dilakukan proses konversi data. Untuk konversi data harus memenuhi dua criteria berikut ini :

1. Dua tipe data yang kompatibel
2. Tipe data awal lebih kecil dibandingkan dengan tipe data tujuan.

Jika dua syarat tersebut tidak terpenuhi maka bisa dilakukan teknik casting, yaitu melakukan perubahan tipe data dengan cara dipaksa. Casting ini bisa langsung dilakukan

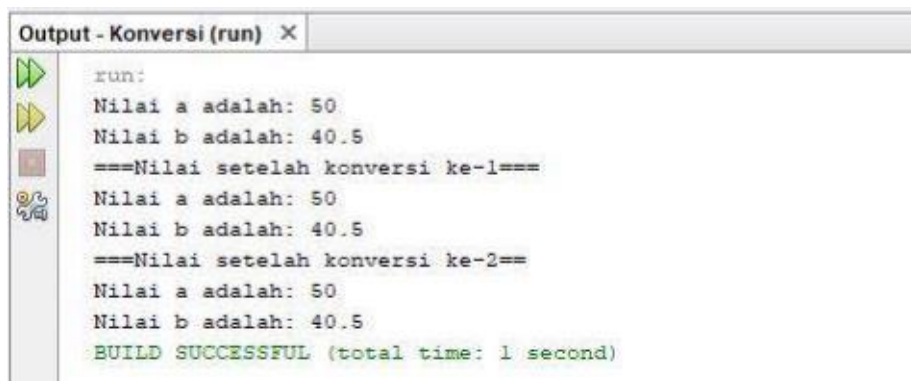
otomatis oleh java dengan cara menambahkan tipe data cating dikurung kemudian dilanjutkan dengan proses yang mau dijalankan. contoh : int a=4; int b=5; double c=(double)a/b;

- Contoh hasil praktikum saya dengan nama konversi



```
24
25     String konversi_a = String.valueOf(a);
26     String konversi_b = String.valueOf(b);
27
28     System.out.println("===Nilai setelah konversi ke-1===");
29     System.out.println("Nilai a adalah: " +konversi_a);
30     System.out.println("Nilai b adalah: " +konversi_b);
31
32     int pindah_keinteger = Integer.parseInt(konversi_a);
33     double pindah_kedouble =Double.parseDouble(konversi_b);
34
35     System.out.println("===Nilai setelah konversi ke-2===");
36     System.out.println("Nilai a adalah: "+pindah_keinteger);
37     System.out.println("Nilai b adalah: "+pindah_kedouble);
38
39 }
```

Jika di Run maka hasilnya :



```
Output - Konversi (run) X
run:
Nilai a adalah: 50
Nilai b adalah: 40.5
===Nilai setelah konversi ke-1===
Nilai a adalah: 50
Nilai b adalah: 40.5
===Nilai setelah konversi ke-2===
Nilai a adalah: 50
Nilai b adalah: 40.5
BUILD SUCCESSFUL (total time: 1 second)
```

➤ Operator

Operator Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti bermacam-macam prioritas yang pasti sehingga compilernya akan tahu yang mana operator untuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan.

- Contoh hasil praktikum saya dengan nama OperasiAritmatika

```

Source History
20     int y = 4;
21     float a = 12.5f;
22     float b = 7f;
23
24     System.out.println("x= " + x + ", y =" + y);
25     System.out.println("x + y =" + (x + y));
26     System.out.println("x - y =" + (x - y));
27     System.out.println("x / y =" + (x / y));
28     System.out.println("x % y =" + (x % y));
29     System.out.println("a = " + a + ", b = " + b);
30     System.out.println("a / b =" + (a / b));
31     System.out.println("x / b =" + (x / b));
32     System.out.println("b / x =" + (b / x));
33
34 }

```

Jika di Run maka hasilnya :

```

Output - OperasiAritmatika (run) X
run:
x= 6, y =4
x + y =10
x - y =2
x / y =1
x % y =2
a = 12.5, b = 7.0
a / b =1.7857143
x / b =0.85714287
b / x =1.1666666
BUILD SUCCESSFUL (total time: 0 seconds)

```

➤ Operator Relasi

Operator Relasi membandingkan dua nilai dan menentukan keterhubungan diantara nilai-nilai tersebut. Hasil keluarannya berupa nilai boolean yaitu true atau false.

- Contoh hasil praktikum saya dengan nama Operator :

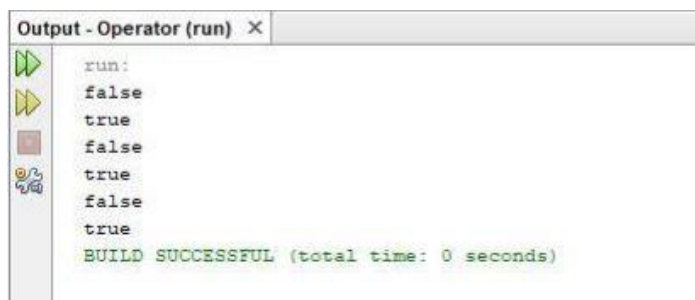
```

// Java code application logic here
int a = 4;
int b = 3;

System.out.println(a==b);
System.out.println(a!=b);
System.out.println(a<b);
System.out.println(a>b);
System.out.println(a<=b);
System.out.println(a>=b);
}

```

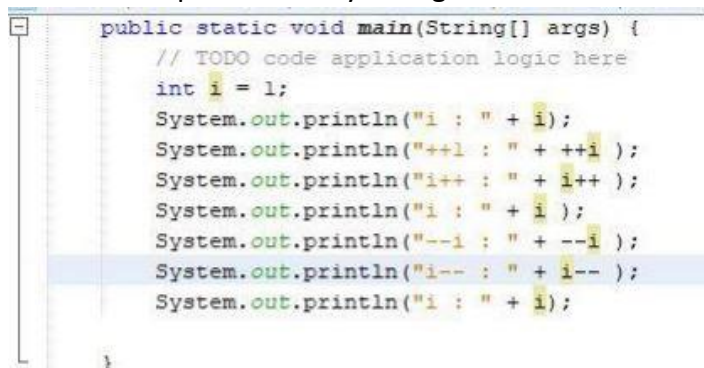
Jika di Run maka hasilnya :



```
Output - Operator (run) X
run:
false
true
false
true
false
true
BUILD SUCCESSFUL (total time: 0 seconds)
```


➤ Operator Penjumlahan dan Pengurangan

- Contoh hasil praktikum saya dengan nama IncremenDecrement.



```
public static void main(String[] args) {
    // TODO code application logic here
    int i = 1;
    System.out.println("i : " + i);
    System.out.println("++1 : " + ++i );
    System.out.println("i++ : " + i++ );
    System.out.println("i : " + i );
    System.out.println("--i : " + --i );
    System.out.println("i-- : " + i-- );
    System.out.println("i : " + i);
}
```

Jika di Run maka hasilnya :



```
Output - IncremenDecrement (run) X
run:
i : 1
++1 : 2
i++ : 2
i : 3
--i : 2
i-- : 2
i : 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

➤ Operator Logika

Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean. Terdapat enam operator logika yaitu: && (logika AND), & (boolean logika AND), || (logika OR), | (boolean logika inclusive OR), ^ (boolean logika exclusive OR), dan ! (logika NOT).

- Contoh hasil praktikum saya dengan nama Operatorlogika

```
// TODO code application logic here
int a = 4;
int b = 3;

System.out.println(a==b);
System.out.println(a!=b);
System.out.println(a<b);
System.out.println(a>b);
System.out.println(a<=b);
System.out.println(a>=b);
}
```

Jika di Run maka hasilnya :

```
Output - Operator (run) X
run:
false
true
false
true
false
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.3 Pemilihan 1

- **Fungsi kondisi “ if ”**
“if statement” adalah statement conditional yang paling sederhana dan sering digunakan. Jika kondisi pada ekspresi if menghasilkan nilai true, maka kode dalam blok akan dieksekusi, bila tidak program akan dilanjutkan ke baris kode setelah blok if.
- **Fungsi kondisi “ if else ”**
Variasi lain adalah “if else statement” yang menyajikan blok kode untuk dieksekusi bila kondisi bernilai false.
- **Kondisional Else if**
Statement “else if” dapat digunakan untuk memeriksa beberapa kondisi sekaligus.
- **Switch Case**
Kondisi SWITCH CASE adalah percabangan kode program dimana kita membandingkan isi sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan true, maka block kode program akan di proses. Kondisi SWITCH CASE terdiri dari 2 bagian, yakni

perintah SWITCH dimana terdapat nama variabel yang akan diperiksa, serta 1 atau lebih perintah CASE untuk setiap nilai yang akan diperiksa.

○ Ternary Operator

Operator ternary adalah salah satu operator yang digunakan di dalam bahasa pemrograman java. Mempunyai fungsi untuk menggantikan kondisi if else ,yang dimana jika suatu kondisi dapat bernilai true atau false, akan menampilkan output atau statement. Selain itu juga dengan menggunakan operator ternary , code akan terlihat lebih ringkas dan mudah di baca. Ciri-ciri jika baris code tersebut , menggunakan operator ternary. Adanya pernyataan atau kondisi yang di apit oleh simbol (?) dan juga (:).

• Contoh hasil praktikum saya dengan nama Pemilihan01

```
package pemilihan01.pkg1;

import java.util.Scanner;

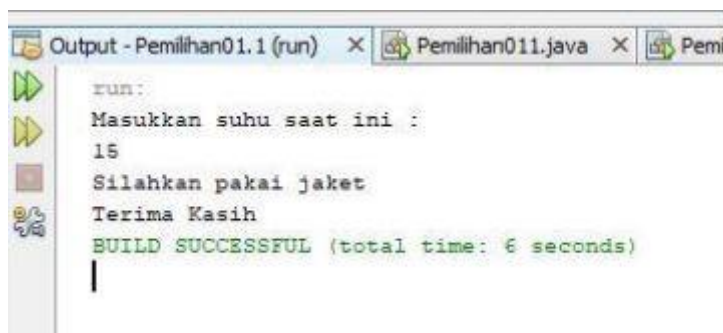
public class Pemilihan011 {
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner input = new Scanner(System.in);
        int suhu;

        System.out.println("Masukkan suhu saat ini : ");
        suhu = input.nextInt();

        if(suhu < 16 ){
            System.out.println("Silahkan pakai jaket");
            System.out.println("Terima Kasih");
        }

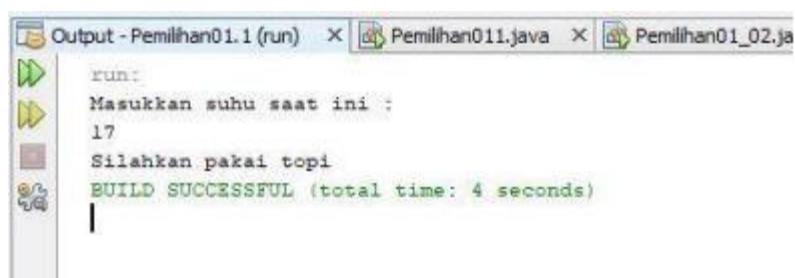
        else {
            System.out.println("Silahkan pakai topi ");
        }
    }
}
```

Jika di Run maka hasilnya : Jika di inputkan angka < 16 misal,15 maka hasilnya “Silahkan pakai jaket”.



The screenshot shows the 'Output - Pemilihan01.1 (run)' window. The output text is: 'run: Masukkan suhu saat ini : 15 Silahkan pakai jaket Terima Kasih'. Below the output, it says 'BUILD SUCCESSFUL (total time: 6 seconds)'.

Jika di inputkan angka 16 atau >16 misal,17 maka hasilnya “Silahkan pakai topi”.

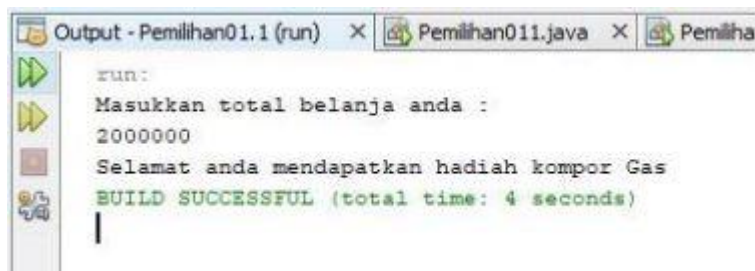


The screenshot shows the 'Output - Pemilihan01.1 (run)' window. The output text is: 'run: Masukkan suhu saat ini : 17 Silahkan pakai topi'. Below the output, it says 'BUILD SUCCESSFUL (total time: 4 seconds)'.

- fungsi kondisional “ else if ”,
 - Contoh hasil praktikum saya dengan nama pemilihan02

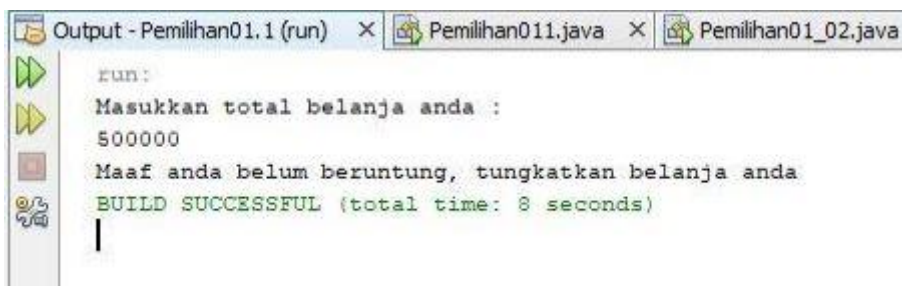
```
package pemilihan01.pkg1;  
import java.util.Scanner;  
public class Pemilihan01_02 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        int bayar;  
  
        System.out.println("Masukkan total belanja anda : ");  
        bayar = input.nextInt();  
  
        if (bayar >= 2000000) {  
            System.out.println("Selamat anda mendapatkan hadiah kompor Gas ");  
        } else if (bayar >= 1000000) {  
            System.out.println("Selamat anda mendapatkan hadiah piring");  
        } else {  
            System.out.println("Maaf anda belum beruntung, tungkatkan belanja anda");  
        }  
    }  
}
```

Hasil running program; Jika di inputkan jumlah Rp 2000000 maka, hasilnya “Selamat anda mendapatkan hadiah kompor gas”



The screenshot shows the 'Output - Pemilihan01.1 (run)' window. The output text is: 'run: Masukkan total belanja anda : 2000000 Selamat anda mendapatkan hadiah kompor Gas BUILD SUCCESSFUL (total time: 4 seconds)'. The input '2000000' is entered on the line following the prompt.

Jika di inputkan jumlah Rp 1000000 maka, hasilnya “Selamat anda mendapatkan hadiah piring”



The screenshot shows the 'Output - Pemilihan01.1 (run)' window. The output text is: 'run: Masukkan total belanja anda : 500000 Maaf anda belum beruntung, tungkatkan belanja anda BUILD SUCCESSFUL (total time: 8 seconds)'. The input '500000' is entered on the line following the prompt.

- fungsi “ Switch Case ”, buatlah project baru
 - Contoh hasil praktikum saya dengan nama pemilihan01_03


```

package pemilihan01.pkg1;
import java.util.Scanner;

public class Pemilihan01_03 {
    public static void main(String [] args){
        Scanner input = new Scanner (System.in);
        int pilihan;

        System.out.println("Silahkan masukkan pilihan anda =");
        pilihan = input.nextInt();

        switch (pilihan){
            case 1: System.out.println("Hari Senin");
                    break;
            case 2: System.out.println("Hari Selasa");
                    break;
            case 3: System.out.println("Hari Rabu");
                    break;
            case 4: System.out.println("Hari Kamis");
                    break;
            case 5: System.out.println("Hari Jumat");
                    break;
            case 6: System.out.println("Hari Sabtu");
                    break;
            case 7: System.out.println("Hari Minggu");
                    break;

            default:
                System.out.println("Maaf angka yang anda masukkan salah");
        }
    }
}

```

Jika di inputkan angka 1 maka, hasilnya “Hari senin

```

run:
Silahkan masukkan pilihan anda =
1
Hari Senin
BUILD SUCCESSFUL (total time: 4 seconds)

```

Dan jika di inputkan angka 2 maka, hasilnya “Hari selasa” Begitupun seterusnya sampai ke 7

Jika yang di inputkan angka selain 1 sampe 7 misal, angka 10 maka, hasilnya “Maaf angka yang anda masukkan salah”

➤ .fungsi “ Ternary Operator “

Contoh hasil praktikum saya dengan nama pemilihan01_04

```

package pemilihan01.pkg1;

public class pemilihan01_04 {
    public static void main (String [] arg){
        double angka = 5.5;

        String keterangan;

        keterangan = (angka > 0.0) ? "Bilangan positif" : "Bilangan negatif";
        System.out.println(angka + " adalah " + keterangan);
    }
}

```

Hasil running

```

run:
5.5 adalah Bilangan positif
BUILD SUCCESSFUL (total time: 3 seconds)

```

2.3 Percabangan dalam percabangan (Nested if)

Kita telah mempelajari penggunaan pernyataan IF untuk memilih sebuah tidak, pernyataan IF-ELSE untuk memilih antara dua tindakan, serta pernyataan IF-ELSE IF-ELSE dan SWITCH-CASE untuk memilih beberapa tindakan (3 atau lebih). Terkadang kita membutuhkan pengambilan keputusan dalam bentuk level (bertingkat) sehingga di dalam suatu pernyataan IF (atau IF-ELSE) bisa saja terdapat pernyataan IF (atau IF-ELSE) yang lain. Jenis percabangan seperti ini disebut NESTED IF (percabangan bersarang).

Pada bentuk penulisan pernyataan NESTED-IF tersebut, kondisi yang akan diseleksi pertama kali adalah kondisi IF yang berada di posisi terluar (kondisi 1).

- Jika kondisi 1 bernilai salah, maka pernyataan ELSE terluar (pasangan dari IF yang bersangkutan) yang akan diproses. Namun, jika pernyataan ELSE (pasangan dari IF) tidak ditulis, maka penyeleksian kondisi akan dihentikan.
- Jika ternyata kondisi 1 bernilai benar, maka kondisi berikutnya yang lebih dalam (kondisi 2) akan diseleksi. Jika kondisi 2 bernilai salah, maka pernyataan ELSE (pasangan dari IF yang bersangkutan) yang akan diproses. Namun, jika pernyataan ELSE (pasangan dari IF) tidak ditulis, maka penyeleksian kondisi akan dihentikan.

Dengan cara yang sama, penyeleksian kondisi akan dilakukan sampai dengan kondisi n, jika kondisikondisi sebelumnya bernilai benar.

Berikut ini adalah contoh kode program untuk penggunaan NESTED IF pada contoh kasus pembayaran di kasir :

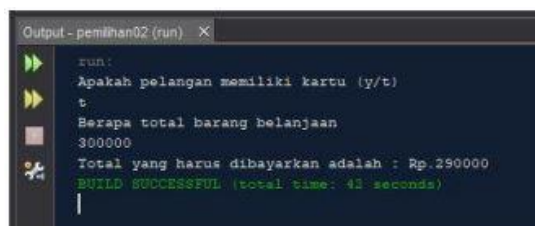
➤ Percabangan dalam percabangan (Nested if)

```
package pemilihan02;
import java.util.Scanner;

public class Pemilihan02 {
    public static void main(String[] args) {
        int total, diskon, bayar;
        String kartu;
        Scanner sc = new Scanner(System.in);
        System.out.println("Apakah pelanggan memiliki kartu (y/t)");
        kartu = sc.nextLine();
        System.out.println("Berapa total barang belanjaan");
        total = sc.nextInt();

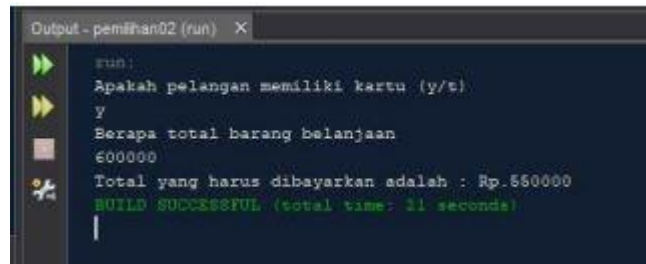
        if (kartu.equals("y")) {
            if (total > 500000) {
                diskon = 50000;
            } else {
                diskon = 25000;
            }
        } else {
            if (total > 200000) {
                diskon = 10000;
            } else {
                diskon = 0;
            }
        }
        bayar = total - diskon;
        System.out.println("Total yang harus dibayarkan adalah : Rp." + bayar );
    }
}
```

Jika di inputkan (t) sebagai pelanggan tidak memiliki kartu dan di inputkan total barang belanjaan 300000 maka akan di diskon 10000 menjadi =290000



```
Output - pemilihan02 (run) X
run:
Apakah pelanggan memiliki kartu (y/t)
t
Berapa total barang belanjaan
300000
Total yang harus dibayarkan adalah : Rp.290000
BUILD SUCCESSFUL (total time: 43 seconds)
```


Jika di inputkan (y) sebagai pelanggan yang memiliki kartu dan di inputkan total barang belanjaan 600000 maka akan di diskon 50000 menjadi =550000



```
Output - pemilihan02 (run) X
run:
Apakah pelanggan memiliki kartu (y/t)
y
Berapa total barang belanjaan
600000
Total yang harus dibayarkan adalah : Rp.550000
BUILD SUCCESSFUL (total time: 11 seconds)
```

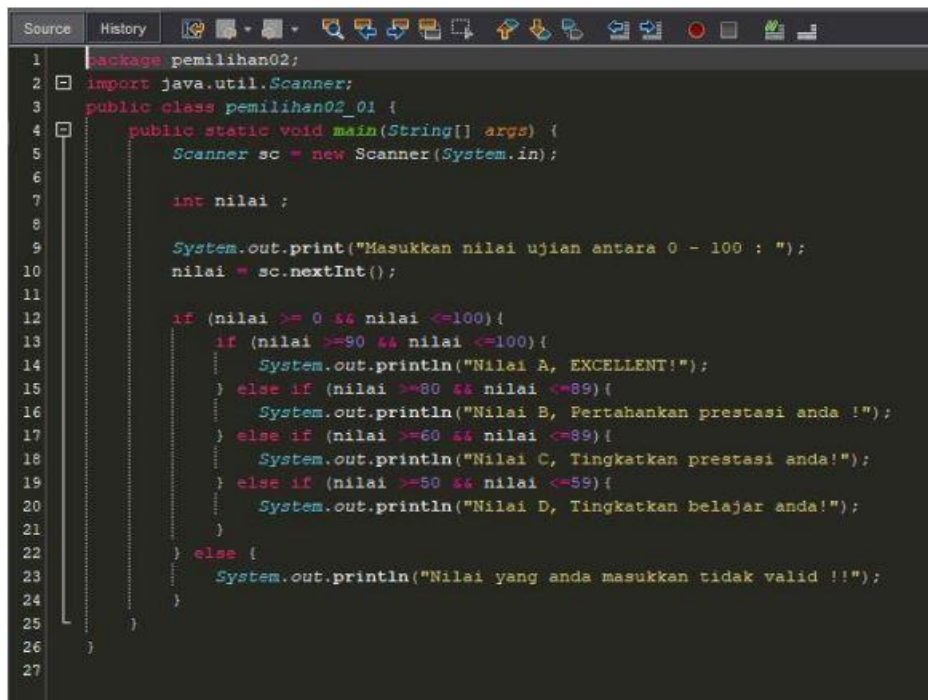
➤ Penggunaan operator logika dalam percabangan Operator

logika merupakan jenis operator yang akan membandingkan logika hasil dari Relational operator.

- && (AND) Jika semua operand bernilai benar (TRUE), maka kondisi tersebut dinyatakan bernilai benar.
- || (OR) Jika salah satu operand bernilai benar (TRUE), maka kondisi tersebut dinyatakan bernilai benar. !
- (NOT) Operator ini juga disebut sebagai invers atau negasi. Operator ini akan membalikan nilai boolean true menjadi false dan begitu juga sebaliknya.

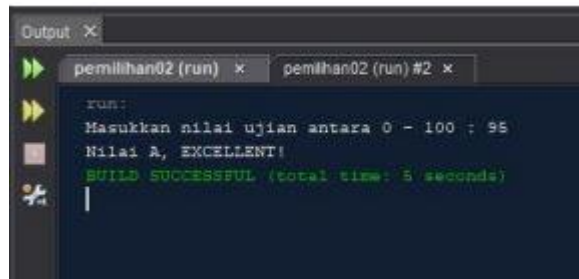
Contoh implementasi operator logika dalam percabangan: Operator logika digunakan untuk menggabungkan beberapa kondisi. Jika Anda ingin program yang menghasilkan output "Welcome!" hanya ketika usia variabel lebih besar dari 18 dan uang variabel lebih besar dari 500. Dalam kasus tersebut, Anda bisa menggunakan logika AND (&&) (deretan kode bagian atas)

Berikut ini adalah contoh kode program untuk penggunaan operator logika dalam percabangan Operator :



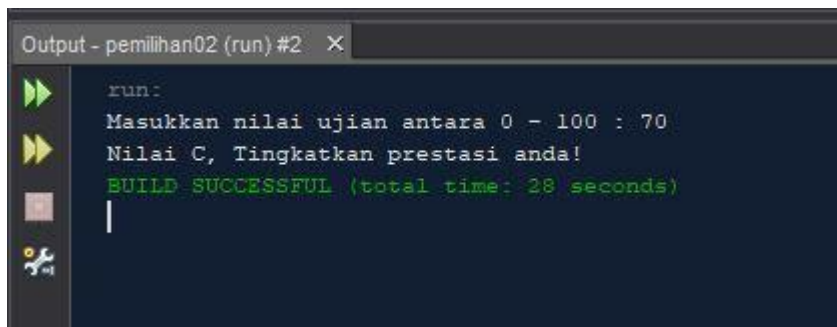
```
Source History
1 package pemilihan02;
2 import java.util.Scanner;
3 public class pemilihan02_01 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int nilai ;
8
9         System.out.print("Masukkan nilai ujian antara 0 - 100 : ");
10        nilai = sc.nextInt();
11
12        if (nilai >= 0 && nilai <=100){
13            if (nilai >=90 && nilai <=100){
14                System.out.println("Nilai A, EXCELLENT!");
15            } else if (nilai >=80 && nilai <=89){
16                System.out.println("Nilai B, Pertahankan prestasi anda !");
17            } else if (nilai >=60 && nilai <=59){
18                System.out.println("Nilai C, Tingkatkan prestasi anda!");
19            } else if (nilai >=50 && nilai <=49){
20                System.out.println("Nilai D, Tingkatkan belajar anda!");
21            }
22        } else {
23            System.out.println("Nilai yang anda masukkan tidak valid !!");
24        }
25    }
26 }
27 }
```

jika di inputkan nilai antara 90-100 misal 95 maka hasil nilainya (nilai A, EXCELLENT!)



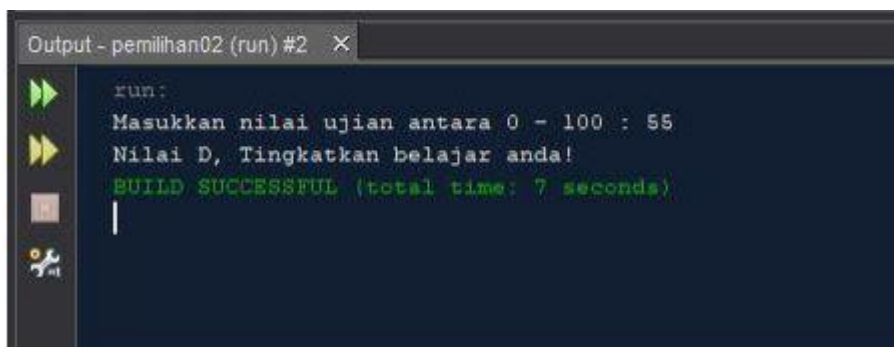
```
Output X
pemilihan02 (run) x pemilihan02 (run) #2 x
run:
Masukkan nilai ujian antara 0 - 100 : 95
Nilai A, EXCELLENT!
BUILD SUCCESSFUL (total time: 5 seconds)
```

jika yang di inputkan nilai antara 60-79 misal 70 maka hasil nilainya (nilai C, Tingkatkan prestasi anda!



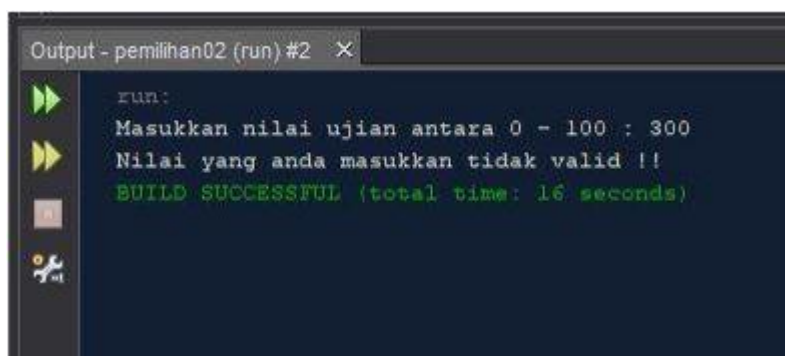
```
Output - pemilihan02 (run) #2 X
run:
Masukkan nilai ujian antara 0 - 100 : 70
Nilai C, Tingkatkan prestasi anda!
BUILD SUCCESSFUL (total time: 28 seconds)
```

jika yang di inputkan nilai antara 50-59 misal 55 maka hasil nilainya (nilai D, Tingkatkan belajar anda!)



```
Output - pemilihan02 (run) #2 X
run:
Masukkan nilai ujian antara 0 - 100 : 55
Nilai D, Tingkatkan belajar anda!
BUILD SUCCESSFUL (total time: 7 seconds)
```

jika yang di inputkan di luar progra atau ngasal misal 300 maka hasil nilainya (nilai yang anda masukkan tidak valid!!)



```
Output - pemilihan02 (run) #2 X
run:
Masukkan nilai ujian antara 0 - 100 : 300
Nilai yang anda masukkan tidak valid !!
BUILD SUCCESSFUL (total time: 16 seconds)
```

Pengecekan pertama digunakan untuk mengecek kategori (pekerja atau pebisnis). Selanjutnya dilakukan pengecekan kedua untuk menentukan besarnya pajak berdasarkan penghasilan yang telah dimasukkan. Kemudian tambahkan kode program untuk menghitung gaji bersih yang diterima setelah dipotong pajak.

- Deklarasi variabel katagori, penghasilan gaji bersih dan pajak

```
package pemilihan02;
import java.util.Scanner;

public class pemilihan02_03 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        String katagori;
        int penghasilan, gajiBersih;
        double pajak = 0;

        System.out.println("Masukkan kkatagori : ");
        katagori = scan.nextLine();
        System.out.println("Masukkan penghasilan : ");
        penghasilan = scan.nextInt();

        if(katagori.equalsIgnoreCase("pekerja")){
            if (penghasilan <= 2000000) {
                pajak = 0.1;
            } else if (penghasilan <= 3000000) {
                pajak = 0.15;
            } else {
                pajak = 0.2;
            }
            gajiBersih = (int)(penghasilan - (penghasilan * pajak));
            System.out.println("Gaji bersih yang anda terima : " + gajiBersih);
        } else if (katagori.equalsIgnoreCase("pebisnis")){
            if (penghasilan <= 2500000) {
                pajak = 0.15;
            } else if (penghasilan <= 3500000) {
                pajak = 0.2;
            } else {
                pajak = 0.25;
            }
            gajiBersih = (int)(penghasilan - (penghasilan * pajak));
            System.out.println("Gaji bersih yang anda terima : " + gajiBersih);
        } else {
            System.out.println("Katagori yang anda masukkan salah");
        }
    }
}
```

Running pebisnis

```
pemilihan02 (run) #2 x  pemilihan02 (run) #4 x
run:
Masukkan kkatagori :
pekerja
Masukkan penghasilan :
2000000
Gaji bersih yang anda terima : 1800000
BUILD SUCCESSFUL (total time: 10 seconds)
```

Running pekerja

```
Output x
pemilihan02 (run) #2 x  pemilihan02 (run) #4 x
run:
Masukkan kkatagori :
pebisnis
Masukkan penghasilan :
2500000
Gaji bersih yang anda terima : 2125000
BUILD SUCCESSFUL (total time: 27 seconds)
```

2.4 ARRAY 1

Array adalah object yang digunakan untuk menyimpan banyak data dengan tipe yang sama. Tipe dari array bisa : tipe data primitif atau class. Terdapat 3 langkah untuk membuat array:

- Mendeklarasikan variabel array
- Memcreate array beserta ukurannya.
- Memberikan sebuah nilai pada setiap element array.

Membuat Array (Mendeklarasikan Variabel Array)

- `int[] ints`
- `double[] dubs`
- `Dimension[] dims ;`
- `float[][] twoDee`
- `int [5] scores;`

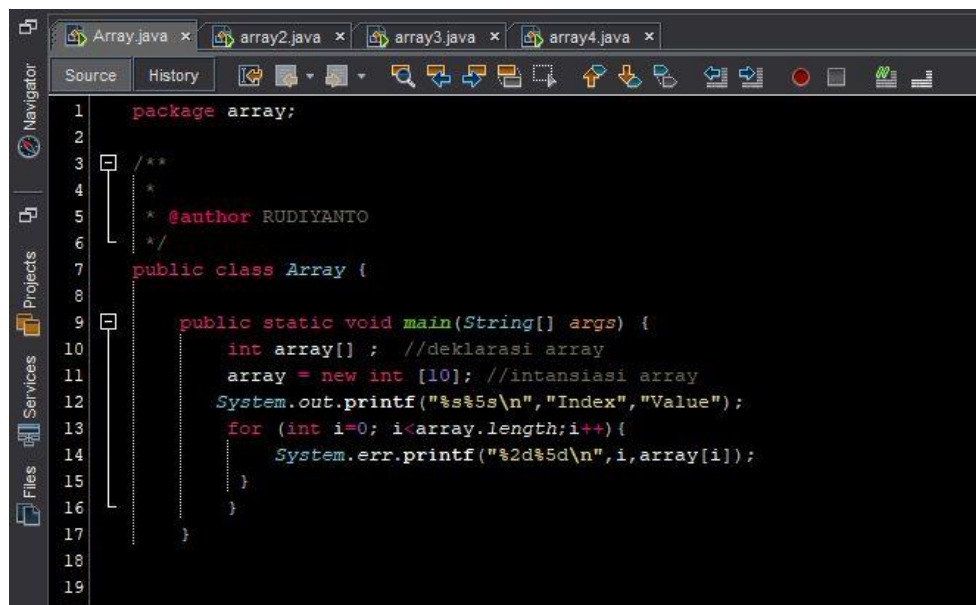
Membuat Array (Men-create Array Beserta Ukurannya) Karena array adalah sebuah object, maka create array dengan operator `new`. Besar array ditentukan pada saat runtime

- `int[] ints ; –`
- `ints = new int[25] ;`

Pada saat array di buat, isi array diinisialisasi dengan default value.

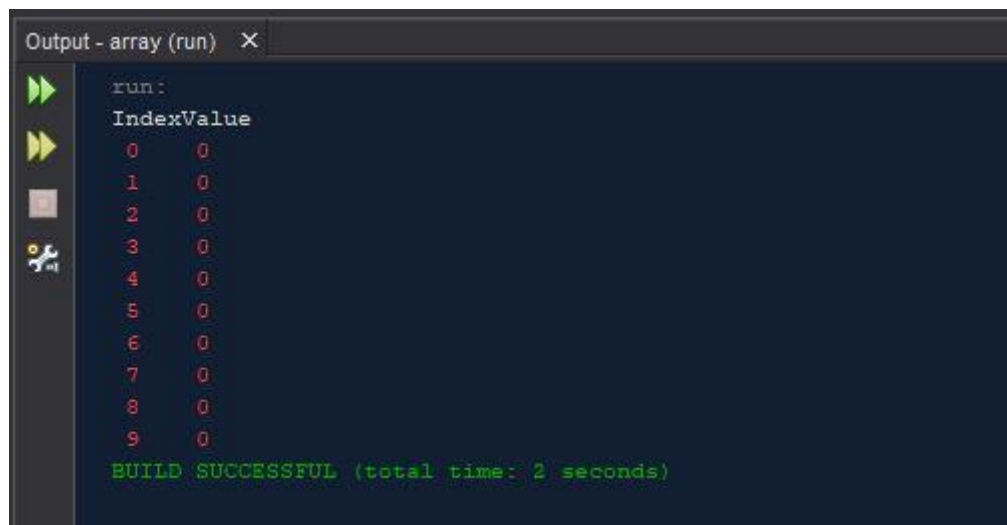
➤ Berikut ini adalah contoh kode program untuk penggunaan array

1. - Percobaan array 1

A screenshot of an IDE window with multiple tabs labeled 'Array.java', 'array2.java', 'array3.java', and 'array4.java'. The 'Source' tab is active, showing Java code. The code defines a package 'array', a class 'Array', and a 'main' method. In the 'main' method, an integer array 'array' of size 10 is declared and initialized. A loop prints the index and value of each element. The code is as follows:

```
1 package array;
2
3 /**
4  *
5  * @author RUDIYANTO
6  */
7 public class Array {
8
9     public static void main(String[] args) {
10         int array[] ; //deklarasi array
11         array = new int [10]; //intansiiasi array
12         System.out.printf("%s%5s\n","Index","Value");
13         for (int i=0; i<array.length;i++){
14             System.err.printf("%2d%5d\n",i,array[i]);
15         }
16     }
17 }
18
19
20
```

-hasil Running

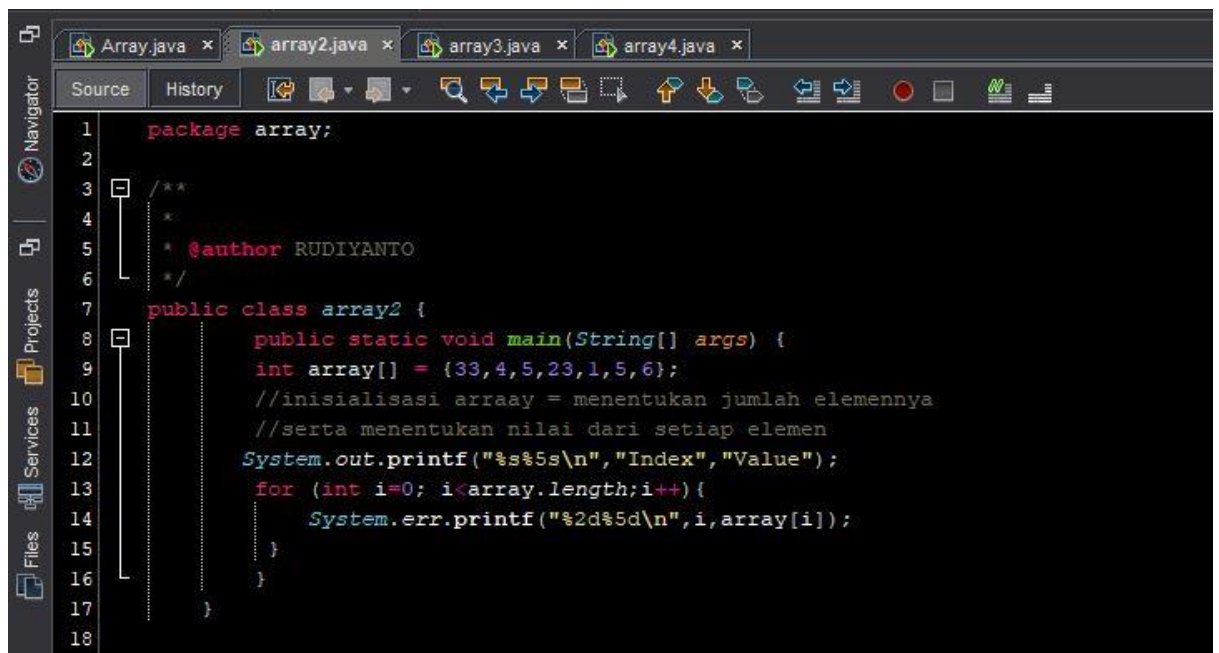


The screenshot shows an IDE's output window titled "Output - array (run)". On the left, there is a vertical toolbar with icons for running (green play button), stepping through (yellow play button), stopping (red square), and debugging (bug icon). The output text is as follows:

```
run:
IndexValue
0      0
1      0
2      0
3      0
4      0
5      0
6      0
7      0
8      0
9      0

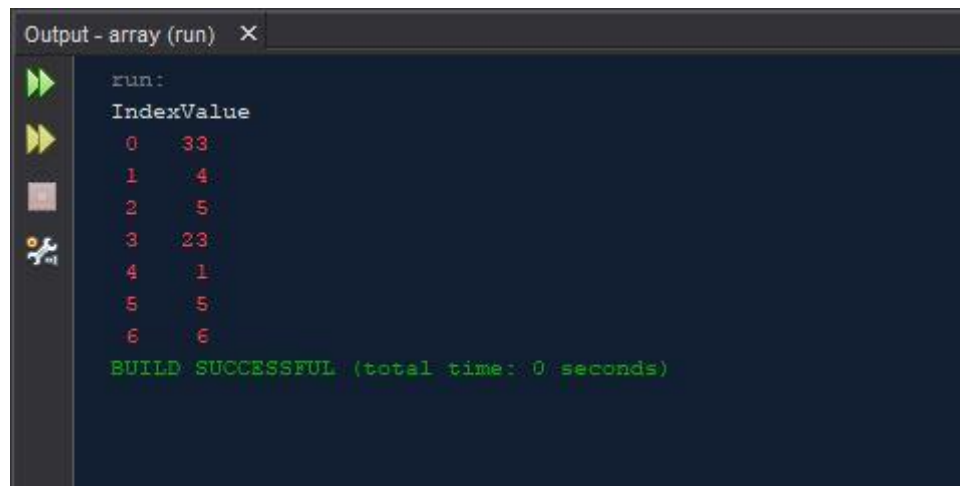
BUILD SUCCESSFUL (total time: 2 seconds)
```

2 - Percobaan array2



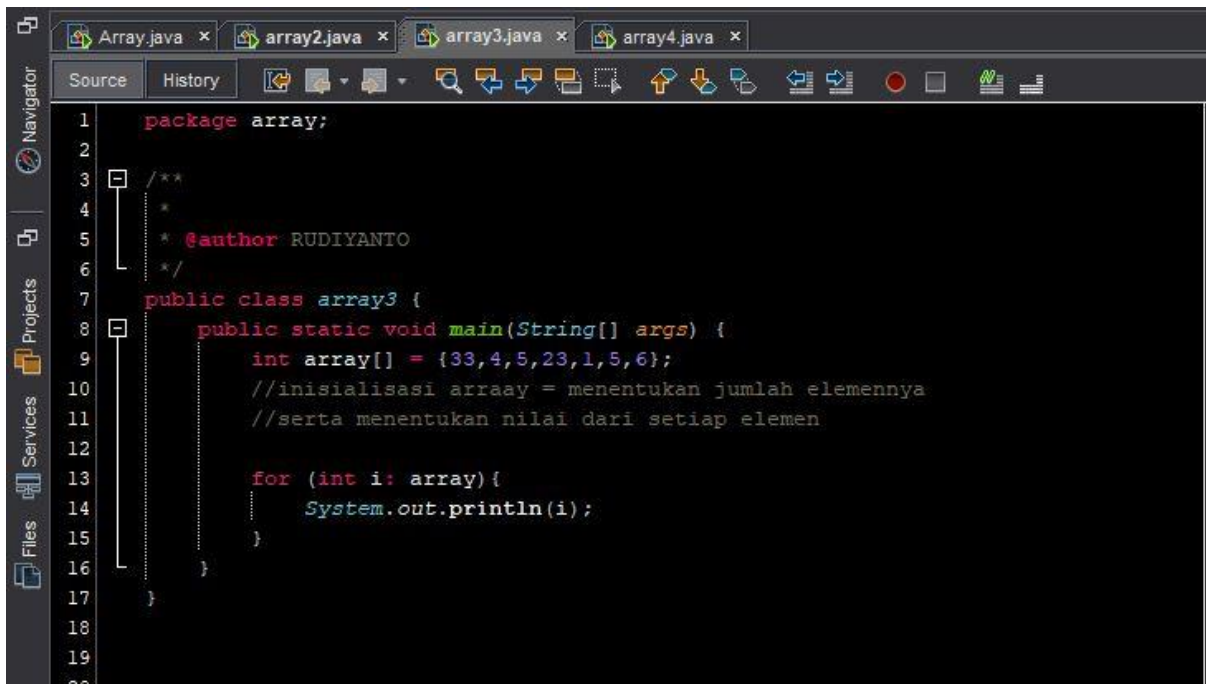
```
1 package array;
2
3 /**
4  *
5  * @author RUDIYANTO
6  */
7 public class array2 {
8     public static void main(String[] args) {
9         int array[] = {33,4,5,23,1,5,6};
10        //inisialisasi array = menentukan jumlah elemennya
11        //serta menentukan nilai dari setiap elemen
12        System.out.printf("%s%5s\n","Index","Value");
13        for (int i=0; i<array.length;i++){
14            System.err.printf("%2d%5d\n",i,array[i]);
15        }
16    }
17 }
18
```

- Hasil Runnung



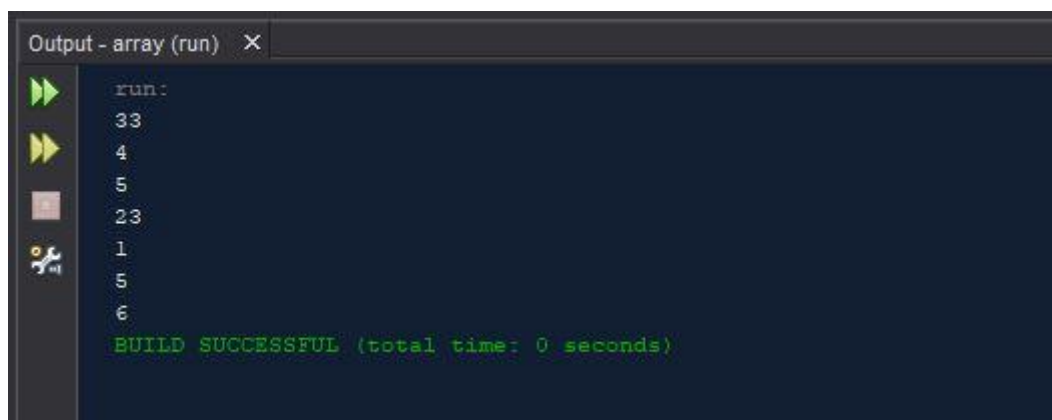
```
Output - array (run) X
run:
IndexValue
0 33
1 4
2 5
3 23
4 1
5 5
6 6
BUILD SUCCESSFUL (total time: 0 seconds)
```


3 – Percobaan array3



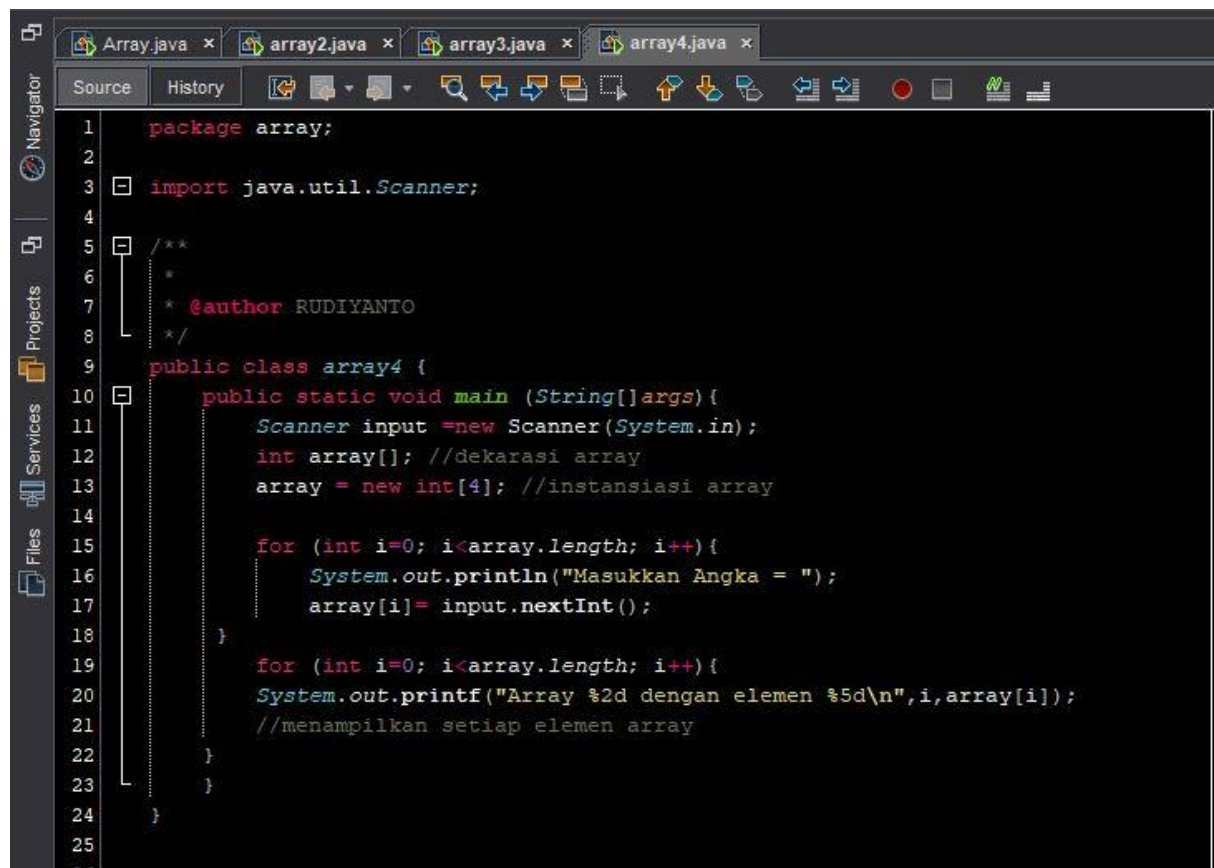
```
1 package array;
2
3 /**
4  *
5  * @author RUDIYANTO
6  */
7 public class array3 {
8     public static void main(String[] args) {
9         int array[] = {33,4,5,23,1,5,6};
10        //inisialisasi arraay = menentukan jumlah elemennya
11        //serta menentukan nilai dari setiap elemen
12
13        for (int i: array){
14            System.out.println(i);
15        }
16    }
17 }
18
19
20
```

- Hasil Running



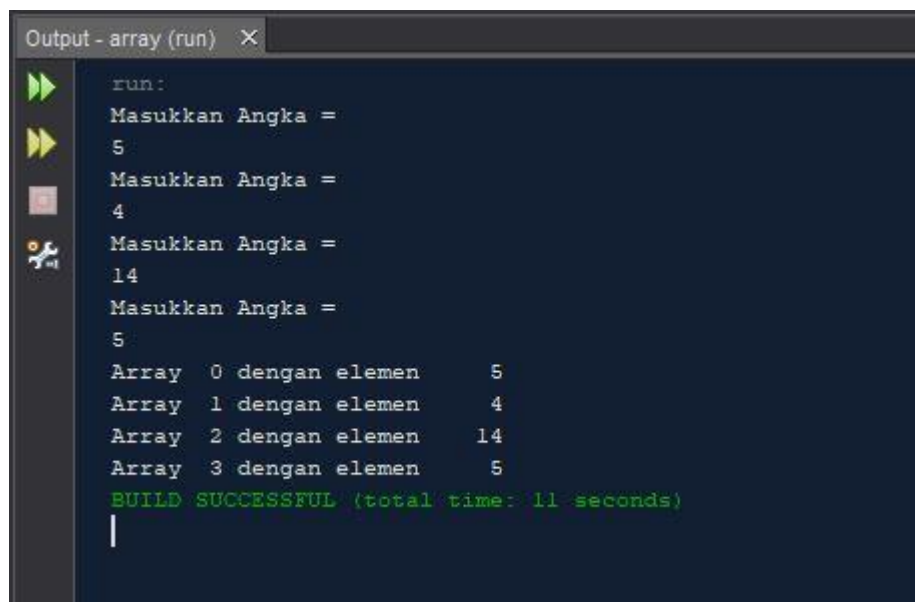
```
Output - array (run) X
run:
33
4
5
23
1
5
6
BUILD SUCCESSFUL (total time: 0 seconds)
```

4 – Percobaan array4



```
1 package array;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author RUDIYANTO
8  */
9 public class array4 {
10     public static void main (String[] args) {
11         Scanner input = new Scanner(System.in);
12         int array[]; //dekarasi array
13         array = new int[4]; //instansiasi array
14
15         for (int i=0; i<array.length; i++){
16             System.out.println("Masukkan Angka = ");
17             array[i]= input.nextInt();
18         }
19
20         for (int i=0; i<array.length; i++){
21             System.out.printf("Array %2d dengan elemen %5d\n",i,array[i]);
22             //menampilkan setiap elemen array
23         }
24     }
25 }
```

- Hasil Running



```
Output - array (run) X
run:
Masukkan Angka =
5
Masukkan Angka =
4
Masukkan Angka =
14
Masukkan Angka =
5
Array 0 dengan elemen    5
Array 1 dengan elemen    4
Array 2 dengan elemen   14
Array 3 dengan elemen    5
BUILD SUCCESSFUL (total time: 11 seconds)
```

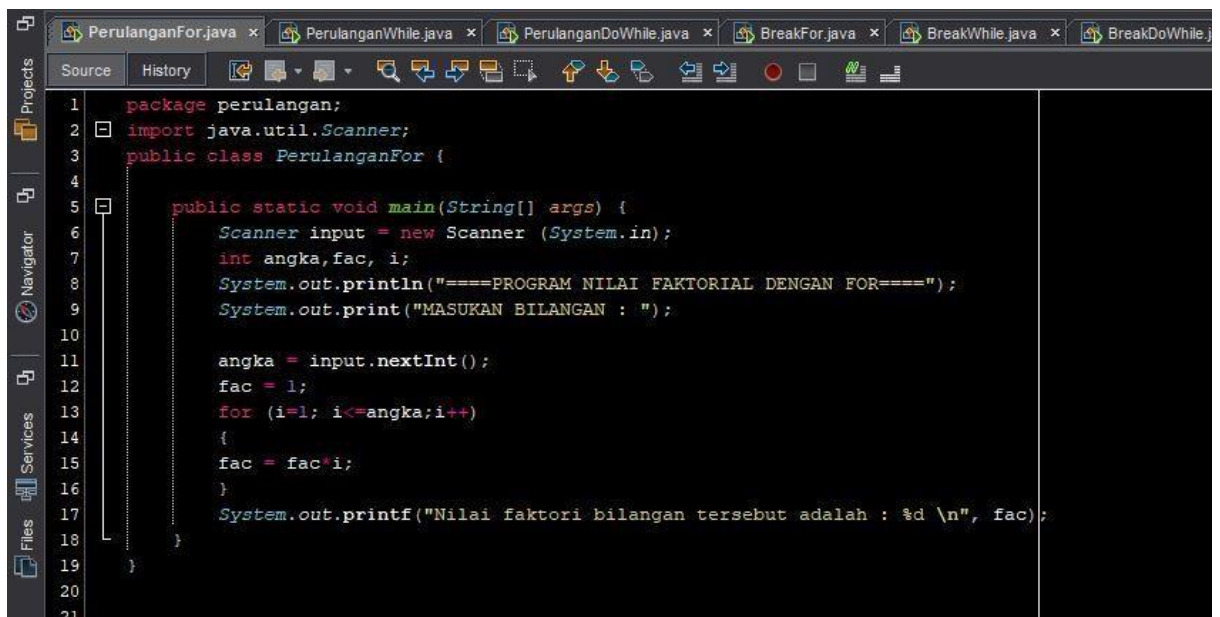

2.5 PERULANGAN

Perulangan adalah suatu blok atau kelompok instruksi yang dilaksanakan secara berulang-ulang. Perulangan akan membuat efisiensi proses dibandingkan jika dioperasikan secara manual. Kebanyakan aplikasi perangkat lunak melakukan pekerjaan berulang-ulang sampai sebuah kondisi yang diinginkan, oleh karena itu pengulangan merupakan bagian yang penting dalam pemrograman karena dengan adanya pengulangan pembuat program tidak perlu menulis kode program sebanyak pengulangan yang diinginkan. Perulangan yang dijelaskan pada jobsheet ini adalah :

- Perulangan dengan for
- Perulangan dengan while
- Perulangan dengan do-while For For adalah kode yang digunakan untuk menjalankan serangkaian kode secara berulang-ulang.
-

➤ For

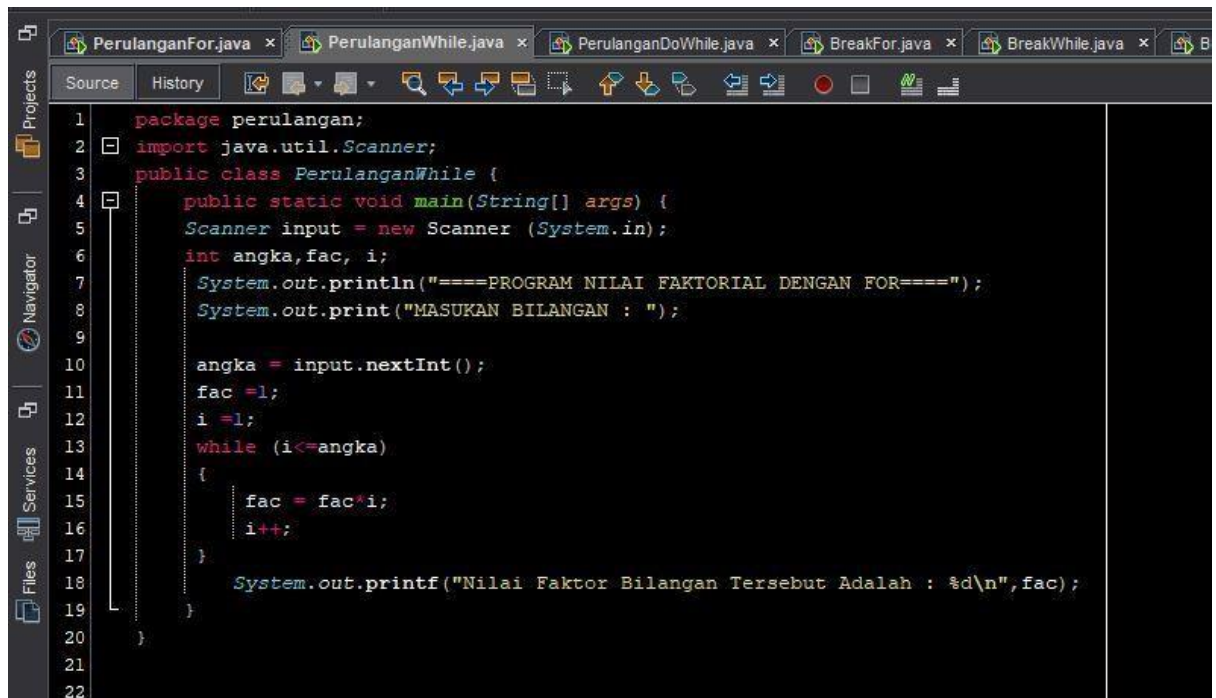
For adalah kode yang digunakan untuk menjalankan serangkaian kode secara berulang-ulang. Pada kode for ini terdapat beberapa komponen yang dicantumkan, antara lain: (1) inisialisasi, (2) kondisi, (3) perubahan nilai, (4) statement yang diulang



```
1 package perulangan;
2 import java.util.Scanner;
3 public class PerulanganFor {
4
5     public static void main(String[] args) {
6         Scanner input = new Scanner (System.in);
7         int angka,fac, i;
8         System.out.println("====PROGRAM NILAI FAKTORIAL DENGAN FOR====");
9         System.out.print("MASUKAN BILANGAN : ");
10
11         angka = input.nextInt();
12         fac = 1;
13         for (i=1; i<=angka;i++)
14         {
15             fac = fac*i;
16         }
17         System.out.printf("Nilai faktori bilangan tersebut adalah : %d \n", fac);
18     }
19 }
20
21
```

➤ While

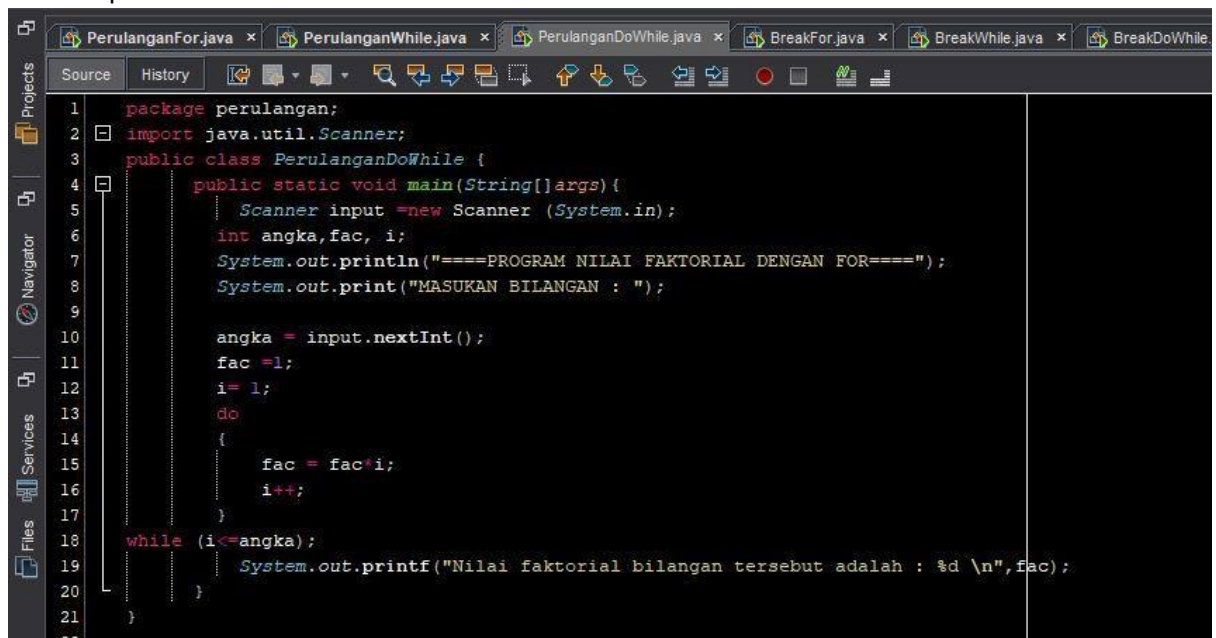
Kode while merupakan kode alternatif untuk melakukan perulangan selain for. Perulangan while biasanya digunakan jika jumlah perulangan tidak diketahui atau memiliki kemungkinan dapat dilakukan kurang dari batas perulangan yang telah ditentukan. Perulangan while hanya akan melakukan perulangan selama kondisi perulangan terpenuhi. Perintah-perintah akan dilaksanakan apabila ekspresi boolean dalam keadaan true. Di dalam loop ada nilai yang mengontrol loop dan nilainya harus berubah, sehingga pada akhir program akan keluar dari loop.



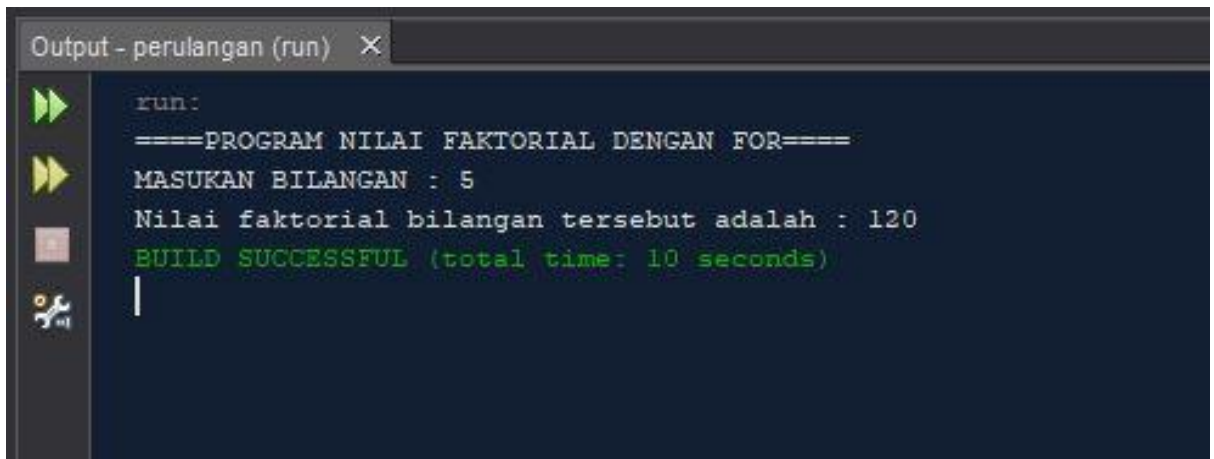
```
1 package perulangan;
2 import java.util.Scanner;
3 public class PerulanganWhile {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         int angka,fac, i;
7         System.out.println("====PROGRAM NILAI FAKTORIAL DENGAN FOR====");
8         System.out.print("MASUKAN BILANGAN : ");
9
10        angka = input.nextInt();
11        fac =1;
12        i =1;
13        while (i<=angka)
14        {
15            fac = fac*i;
16            i++;
17        }
18        System.out.printf("Nilai Faktor Bilangan Tersebut Adalah : %d\n",fac);
19    }
20 }
21
22
```

➤ Do – While

Kode do-while merupakan kode while-do dengan sintaks yang berbeda. Cara kerja do-while relatif sama dengan while. Perintah do-while() akan mengulang statement miliknya selama syarat pengulangannya terpenuhi. Hanya saja, perintah do-while() menjalankan statementnya terlebih dahulu, setelah itu baru memeriksa syaratnya. Sedangkan perintah while() memeriksa syarat terlebih dahulu. Oleh karena itu, perintah do-while() akan menjalankan statementnya paling tidak sebanyak satu kali, meskipun syarat pengulangan tidak terpenuhi.



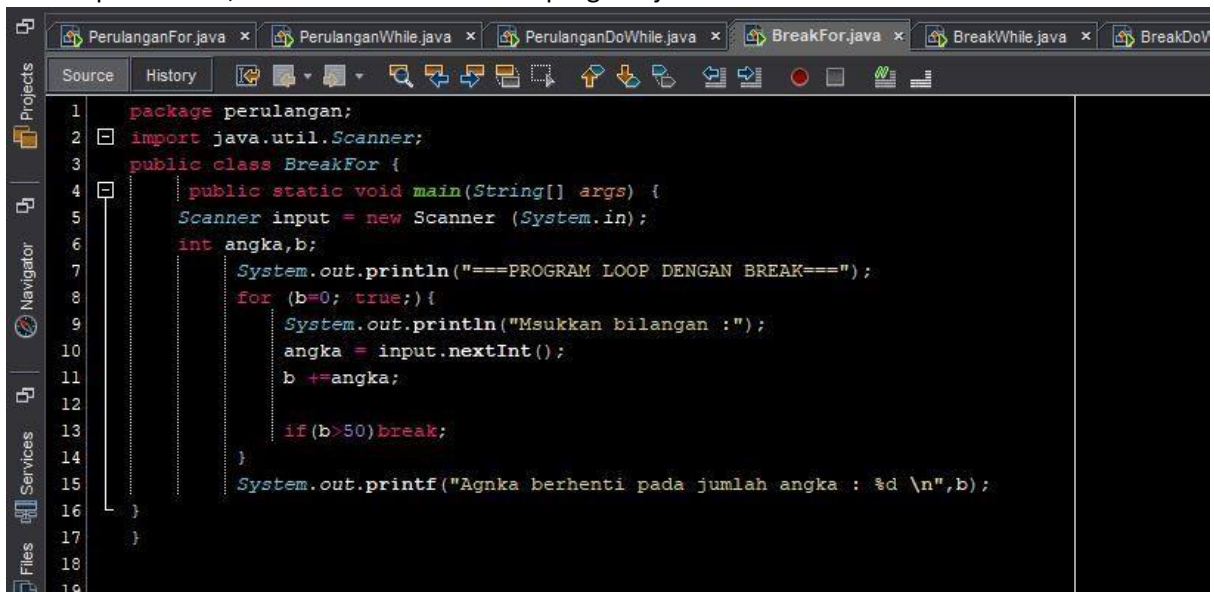
```
1 package perulangan;
2 import java.util.Scanner;
3 public class PerulanganDoWhile {
4     public static void main(String[] args){
5         Scanner input =new Scanner (System.in);
6         int angka,fac, i;
7         System.out.println("====PROGRAM NILAI FAKTORIAL DENGAN FOR====");
8         System.out.print("MASUKAN BILANGAN : ");
9
10        angka = input.nextInt();
11        fac =1;
12        i= 1;
13        do
14        {
15            fac = fac*i;
16            i++;
17        }
18        while (i<=angka);
19        System.out.printf("Nilai faktorial bilangan tersebut adalah : %d \n",fac);
20    }
21 }
22
```



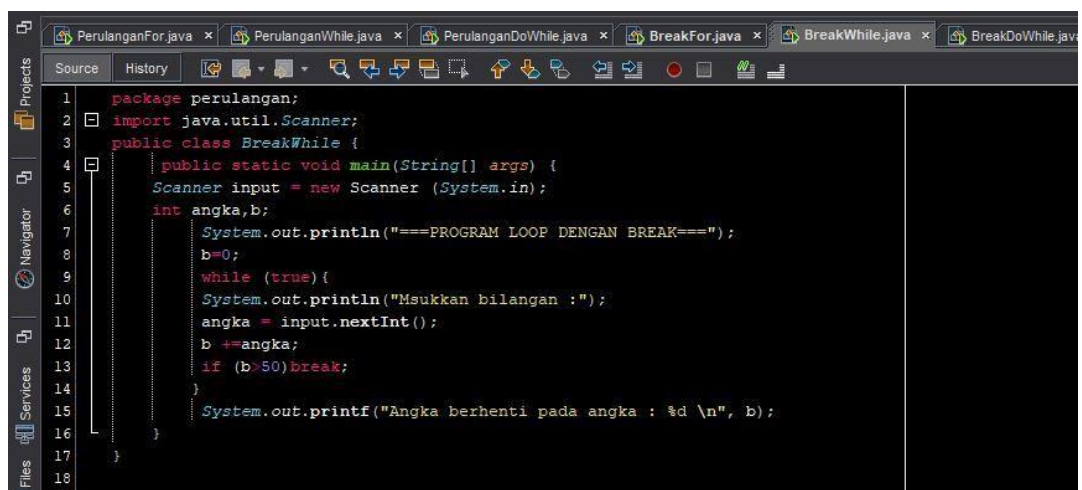
```
run:
====PROGRAM NILAI FAKTORIAL DENGAN FOR====
MASUKAN BILANGAN : 5
Nilai faktorial bilangan tersebut adalah : 120
BUILD SUCCESSFUL (total time: 10 seconds)
```

➤ **Break dan Continue**

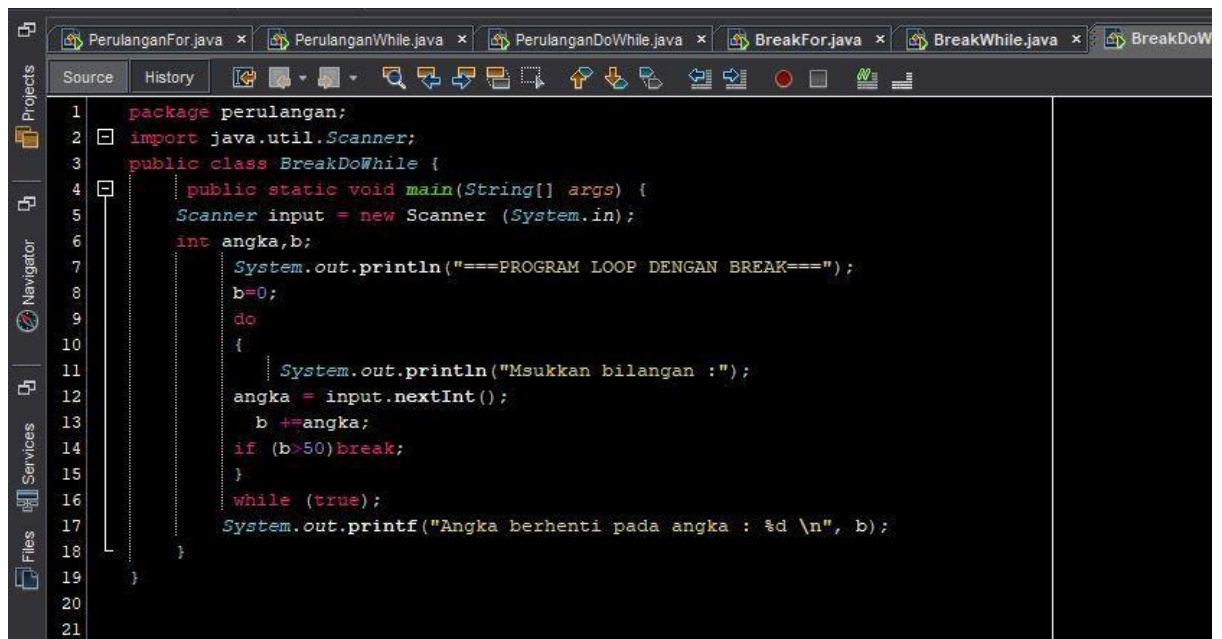
Break dan continue tergolong ke dalam keyword di bahasa pemrograman java, yang keduanya digunakan pada suatu kondisi tertentu , pada perulangan seperti while ,do while dan for. Jika fungsi break digunakan untuk menghentikan suatu pernyataan (statement), dan jika fungsi continue digunakan untuk mengabaikan ,lalu melanjutkan suatu pernyataan pada perulangan. Keyword break dan continue juga biasa digunakan, bersamaan dengan Control Flow seperti if else, dan switch case di dalam program java



```
1 package perulangan;
2 import java.util.Scanner;
3 public class BreakFor {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         int angka,b;
7         System.out.println("===PROGRAM LOOP DENGAN BREAK===");
8         for (b=0; true;){
9             System.out.println("Msukkan bilangan :");
10            angka = input.nextInt();
11            b +=angka;
12
13            if (b>50) break;
14        }
15        System.out.printf("Agnka berhenti pada jumlah angka : %d \n",b);
16    }
17 }
18
19
```

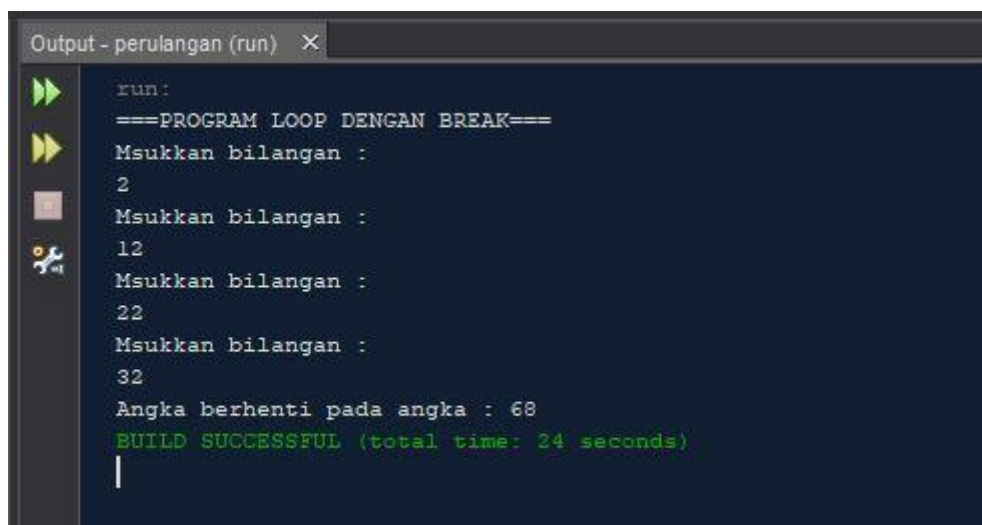


```
1 package perulangan;
2 import java.util.Scanner;
3 public class BreakWhile {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         int angka,b;
7         System.out.println("===PROGRAM LOOP DENGAN BREAK===");
8         b=0;
9         while (true){
10            System.out.println("Msukkan bilangan :");
11            angka = input.nextInt();
12            b +=angka;
13            if (b>50) break;
14        }
15        System.out.printf("Angka berhenti pada angka : %d \n", b);
16    }
17 }
18
19
```



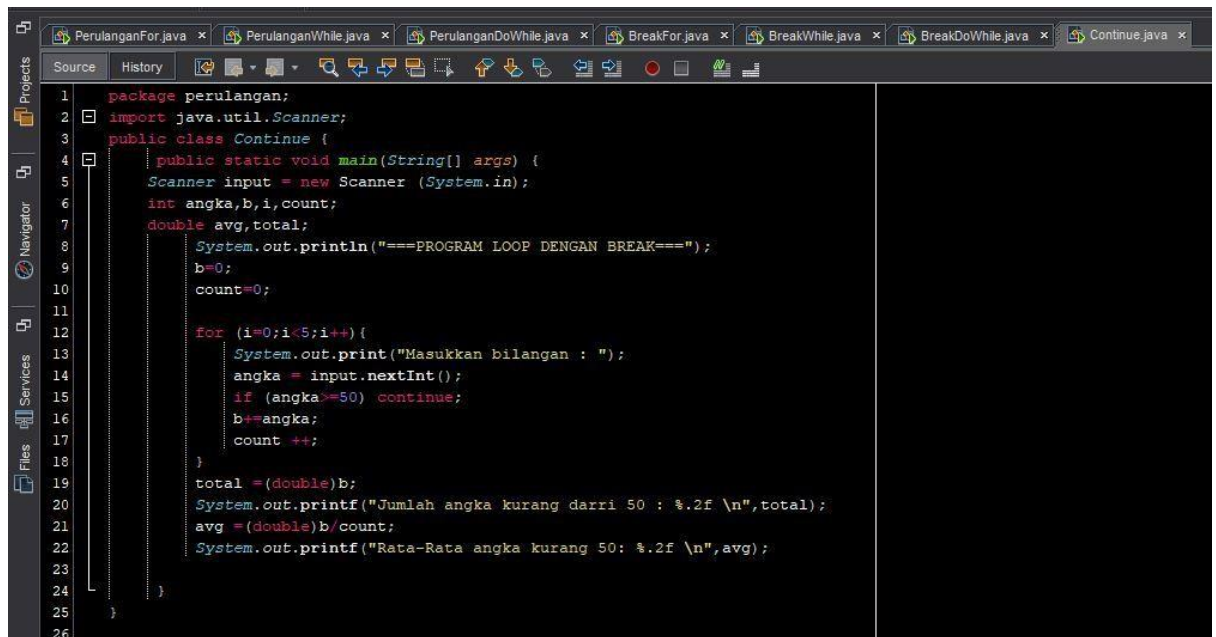
The screenshot shows an IDE with several tabs open: PerulanganFor.java, PerulanganWhile.java, PerulanganDoWhile.java, BreakFor.java, BreakWhile.java, and BreakDoW.java. The active tab is BreakDoW.java, which contains the following Java code:

```
1 package perulangan;
2 import java.util.Scanner;
3 public class BreakDoWhile {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         int angka,b;
7         System.out.println("===PROGRAM LOOP DENGAN BREAK===");
8         b=0;
9         do
10            {
11                System.out.println("Msukkan bilangan :");
12                angka = input.nextInt();
13                b +=angka;
14                if (b>50)break;
15            }
16        while (true);
17        System.out.printf("Angka berhenti pada angka : %d \n", b);
18    }
19 }
20
21
```



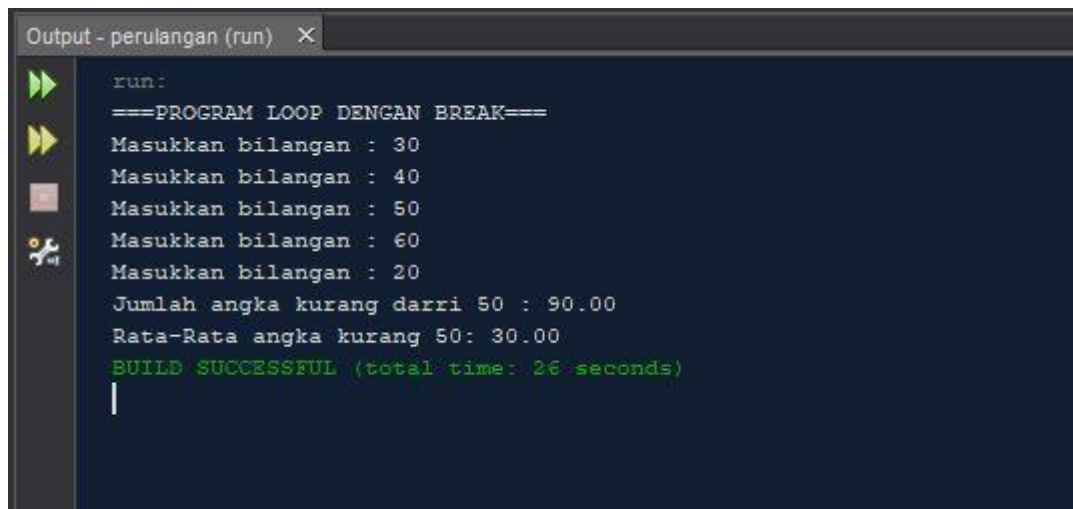
The screenshot shows the output console of the IDE, titled "Output - perulangan (run)". It displays the following text:

```
run:
===PROGRAM LOOP DENGAN BREAK===
Msukkan bilangan :
2
Msukkan bilangan :
12
Msukkan bilangan :
22
Msukkan bilangan :
32
Angka berhenti pada angka : 68
BUILD SUCCESSFUL (total time: 24 seconds)
```

The screenshot shows an IDE with multiple tabs open: 'PerulanganFor.java', 'PerulanganWhile.java', 'PerulanganDoWhile.java', 'BreakFor.java', 'BreakWhile.java', 'BreakDoWhile.java', and 'Continue.java'. The 'Continue.java' tab is active, displaying the following Java code:

```
1 package perulangan;
2 import java.util.Scanner;
3 public class Continue {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         int angka,b,i,count;
7         double avg,total;
8         System.out.println("===PROGRAM LOOP DENGAN BREAK===");
9         b=0;
10        count=0;
11
12        for (i=0;i<5;i++){
13            System.out.print("Masukkan bilangan : ");
14            angka = input.nextInt();
15            if (angka>=50) continue;
16            b+=angka;
17            count ++;
18        }
19        total =(double)b;
20        System.out.printf("Jumlah angka kurang darri 50 : %.2f \n",total);
21        avg =(double)b/count;
22        System.out.printf("Rata-Rata angka kurang 50: %.2f \n",avg);
23    }
24 }
25
26
```



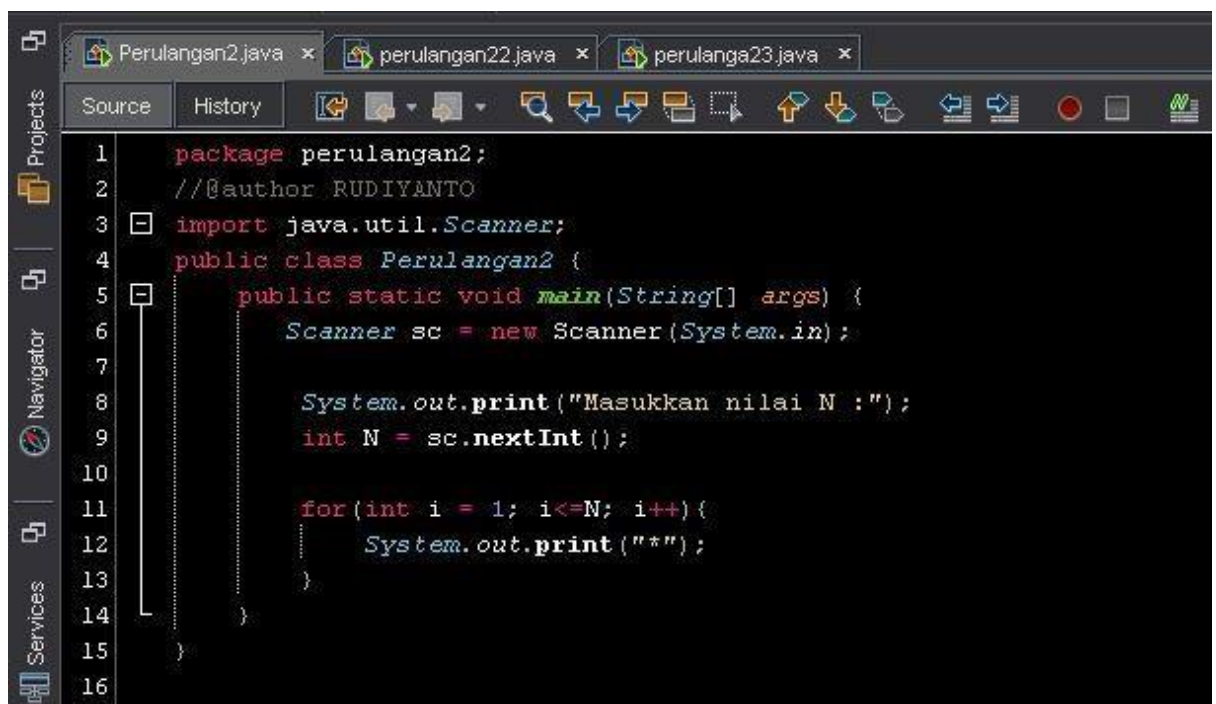
The screenshot shows the 'Output - perulangan (run)' window with the following text:

```
run:
===PROGRAM LOOP DENGAN BREAK===
Masukkan bilangan : 30
Masukkan bilangan : 40
Masukkan bilangan : 50
Masukkan bilangan : 60
Masukkan bilangan : 20
Jumlah angka kurang darri 50 : 90.00
Rata-Rata angka kurang 50: 30.00
BUILD SUCCESSFUL (total time: 26 seconds)
```

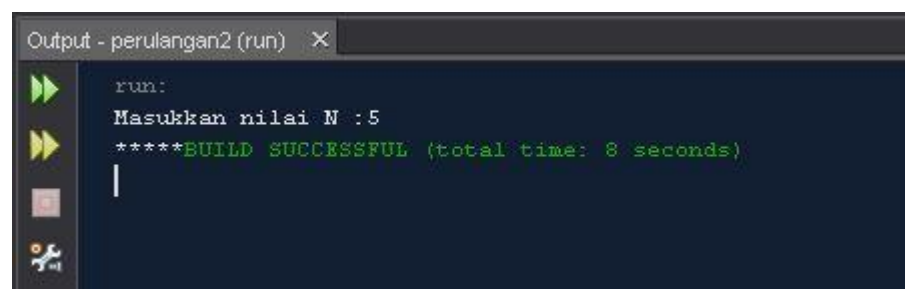
2.6 Perulangan 2

Pengertian Perulangan Bersarang (Nested Loop) Pada bahasan sebelumnya, telah dibahas tentang konsep dasar perulangan. Pada bahasan tersebut disebutkan bahwa logika perulangan digunakan untuk melakukan beberapa proses atau statement program secara berulang-ulang, dengan suatu pola tertentu. Pada perulangan, proses atau statement akan terus dilakukan secara berulang-ulang, selama kondisi perulangan bernilai benar/true. Dan sebaliknya, perulangan akan berhenti dan proses atau statement tidak akan dieksekusi lagi ketika kondisi perulangan bernilai salah/false. Jadi, dalam logika perulangan, suatu kondisi perulangan diperlukan untuk menentukan apakah suatu perulangan masih akan berlangsung lagi atau harus berhenti. Perulangan bersarang (nested loop) adalah struktur perulangan yang berada di dalam perulangan lainnya. Pada umumnya, struktur perulangan yang berada di dalam perulangan lainnya tersebut memiliki hubungan yang saling terkait dalam menyesuaikan sebuah kasus. Pada dasarnya tidak ada batasan dalam jumlah perulangan bersarang. Tetapi sebaiknya tidak terlalu dalam, untuk menghindari kompleksitas yang tinggi serta alur program menjadi lebih sukar untuk dipahami.

Berikut adalah kode program dan Hasil Runningnya :



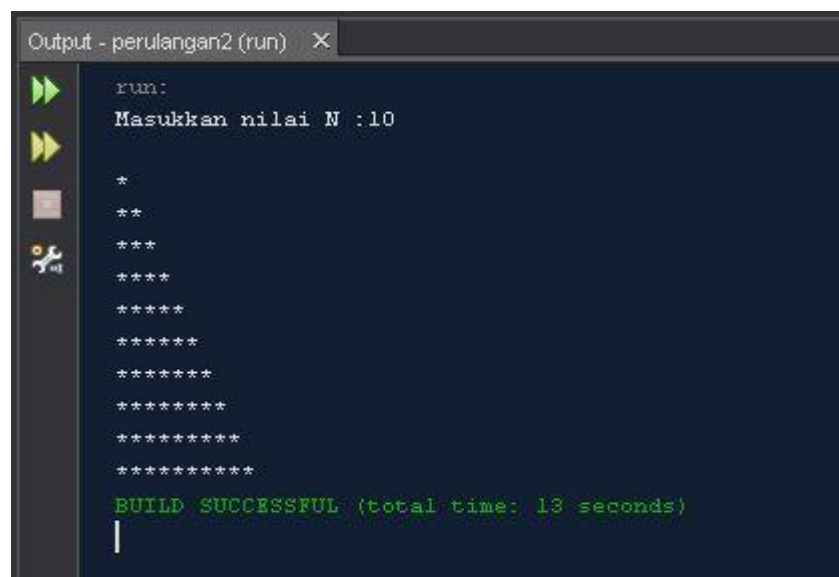
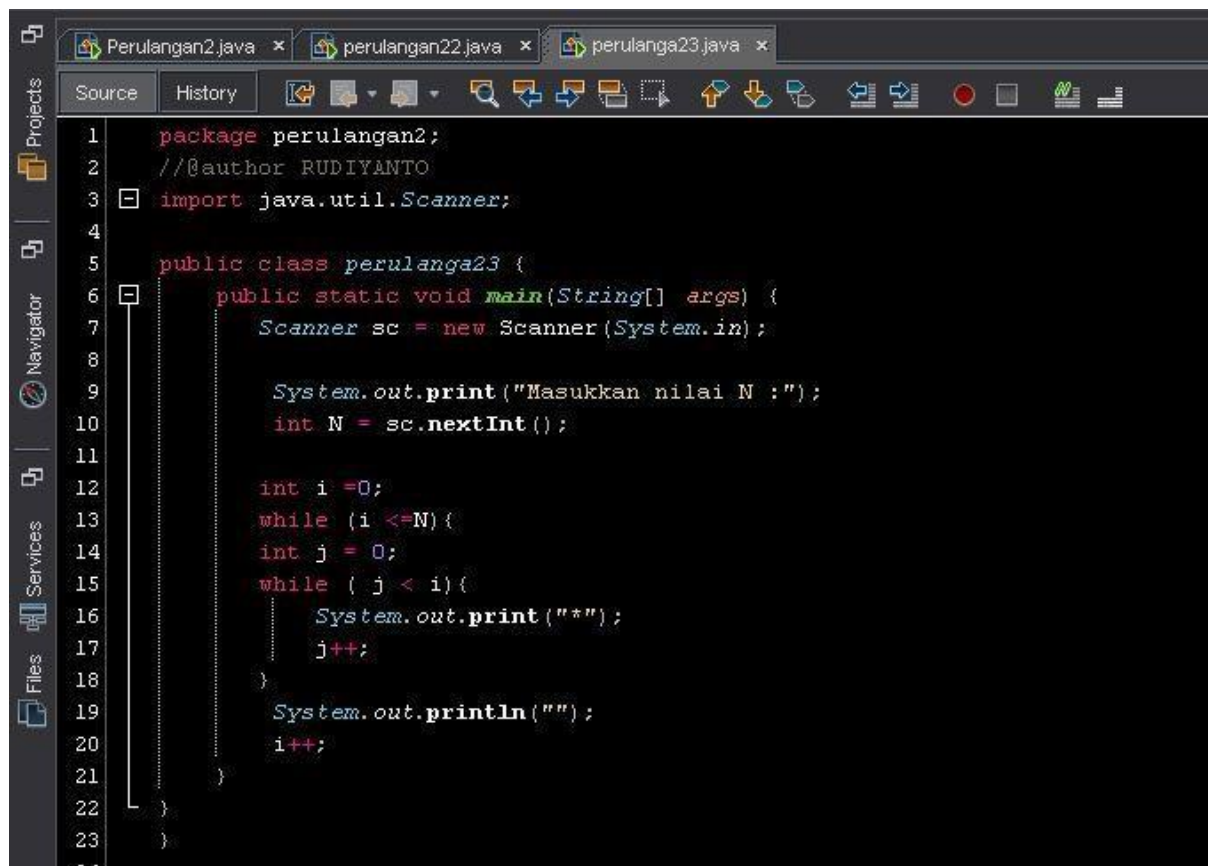
```
1 package perulangan2;
2 // @author RUDIYANTO
3 import java.util.Scanner;
4 public class Perulangan2 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.print("Masukkan nilai N :");
9         int N = sc.nextInt();
10
11         for(int i = 1; i <= N; i++) {
12             System.out.print("*");
13         }
14     }
15 }
16
```



```
run:
Masukkan nilai N :5
*****BUILD SUCCESSFUL (total time: 8 seconds)
```

```
1 package perulangan2;
2 // @author RUDIYANTO
3 import java.util.Scanner;
4 public class perulangan22 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.print("Masukkan nilai N :");
9         int N = sc.nextInt();
10
11         for(int iOuter = 1; iOuter <=N; iOuter++){
12             for (int i = 1; i<=N; i++){
13                 System.out.print("*");
14             }
15             System.out.println("");
16         }
17     }
18 }
19
```

```
run:
Masukkan nilai N :5
*****
*****
*****
*****
*****
BUILD SUCCESSFUL (total time: 2 seconds)
```



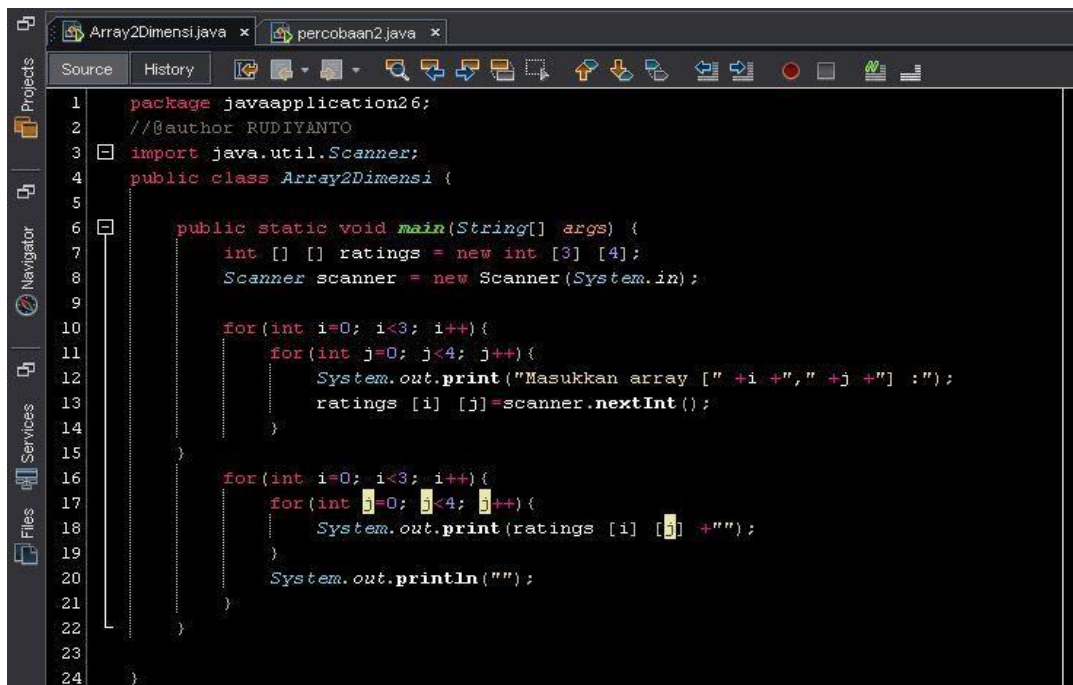
2.7 Array 2

Array yang telah kita pelajari sebelumnya adalah satu dimensi, hanya terdiri dari 1 baris elemen. Terkadang data perlu disajikan dalam bentuk tabel, misalkan spreadsheet yang tentunya membutuhkan array 2 dimensi.

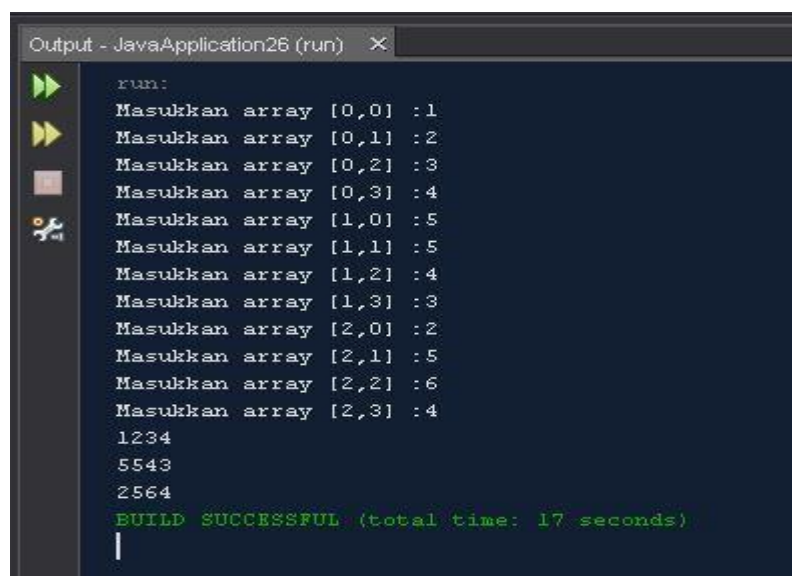
Ukuran Array 2

Dimensi Setiap array memiliki ukuran baik array 1D atau 2D, yaitu sudah tersedia attribute length. Contohnya adalah di bawah ini: – a.length menghasilkan 3, atau barisnya (dimensi yang pertama) – a[0].length menghasilkan 4, atau kolomnya (dimensi yang kedua) Ketika menggunakan attribute/variable length, keuntungannya adalah ketika ukuran array berubah kita tidak perlu mengubah kode untuk input/menampilkan array.

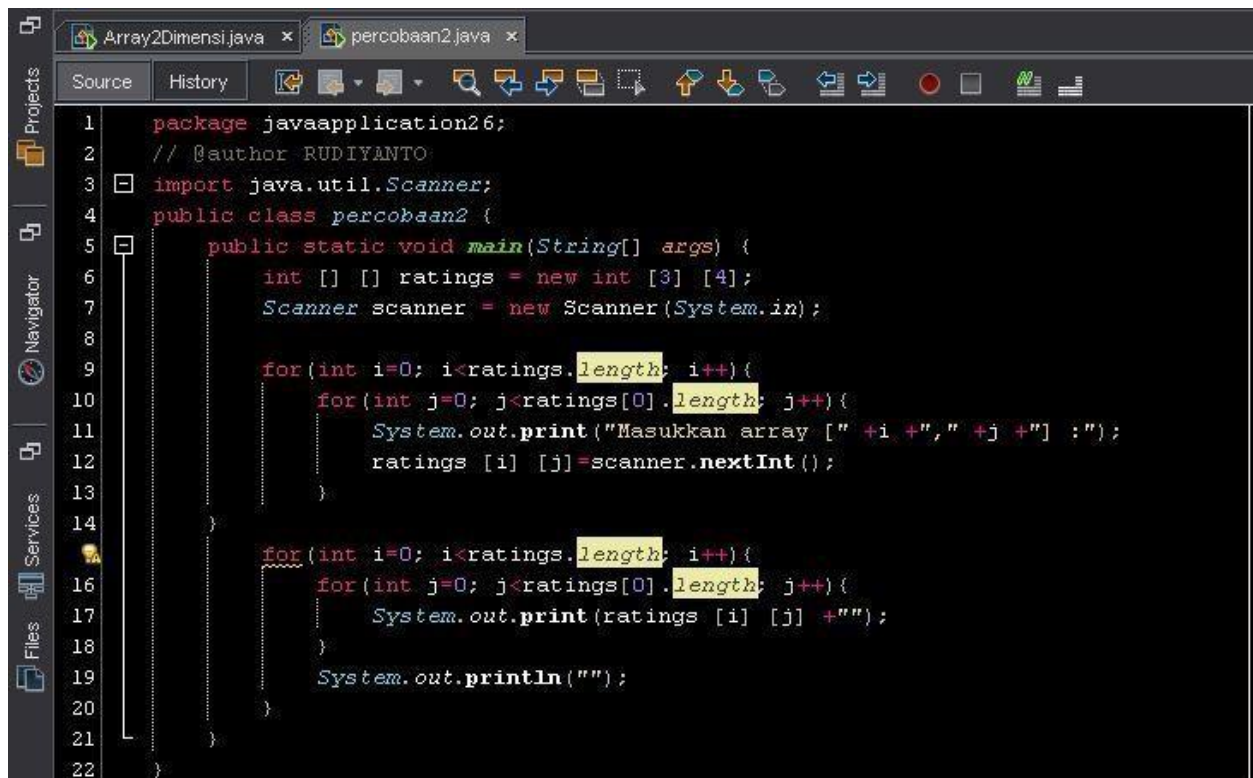
➤ Berikut ini adalah kode program dan hasil Runningnya :



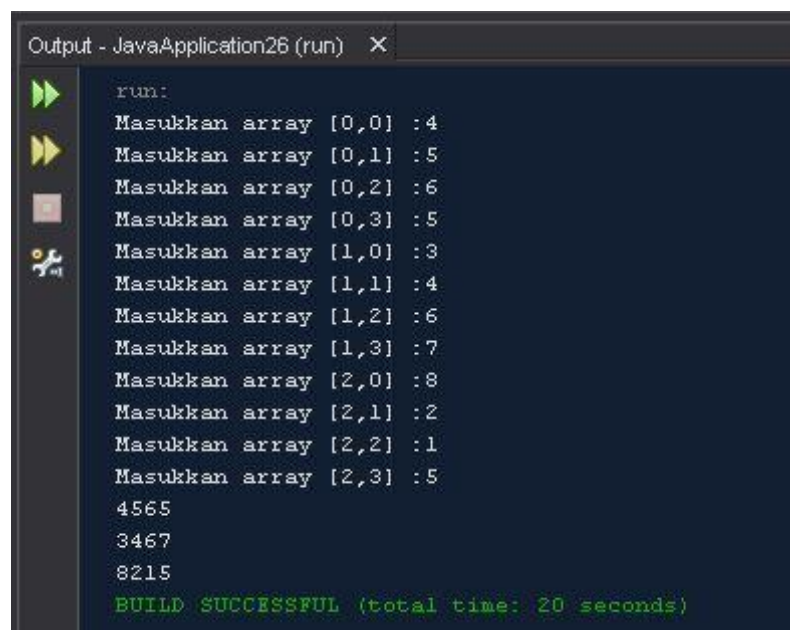
```
1 package javaapplication26;
2 // @author RUDIYANTO
3 import java.util.Scanner;
4 public class Array2Dimensi {
5
6     public static void main(String[] args) {
7         int [] [] ratings = new int [3] [4];
8         Scanner scanner = new Scanner(System.in);
9
10        for(int i=0; i<3; i++){
11            for(int j=0; j<4; j++){
12                System.out.print("Masukkan array [" + i + ", " + j + " ] :");
13                ratings [i] [j]=scanner.nextInt();
14            }
15        }
16
17        for(int i=0; i<3; i++){
18            for(int j=0; j<4; j++){
19                System.out.print(ratings [i] [j] + " ");
20            }
21            System.out.println("");
22        }
23    }
24 }
```



```
Output - JavaApplication26 (run) X
run:
Masukkan array [0,0] :1
Masukkan array [0,1] :2
Masukkan array [0,2] :3
Masukkan array [0,3] :4
Masukkan array [1,0] :5
Masukkan array [1,1] :5
Masukkan array [1,2] :4
Masukkan array [1,3] :3
Masukkan array [2,0] :2
Masukkan array [2,1] :5
Masukkan array [2,2] :6
Masukkan array [2,3] :4
1234
5543
2564
BUILD SUCCESSFUL (total time: 17 seconds)
```



```
1 package javaapplication26;
2 // @author RUDIYANTO
3 import java.util.Scanner;
4 public class percobaan2 {
5     public static void main(String[] args) {
6         int [] [] ratings = new int [3] [4];
7         Scanner scanner = new Scanner(System.in);
8
9         for(int i=0; i<ratings.length; i++){
10             for(int j=0; j<ratings[0].length; j++){
11                 System.out.print("Masukkan array [" +i +"," +j +"] :");
12                 ratings [i] [j]=scanner.nextInt();
13             }
14         }
15
16         for(int i=0; i<ratings.length; i++){
17             for(int j=0; j<ratings[0].length; j++){
18                 System.out.print(ratings [i] [j] +"" );
19             }
20             System.out.println("");
21         }
22     }
23 }
```



```
Output - JavaApplication26 (run) X
run:
Masukkan array [0,0] :4
Masukkan array [0,1] :5
Masukkan array [0,2] :6
Masukkan array [0,3] :5
Masukkan array [1,0] :3
Masukkan array [1,1] :4
Masukkan array [1,2] :6
Masukkan array [1,3] :7
Masukkan array [2,0] :8
Masukkan array [2,1] :2
Masukkan array [2,2] :1
Masukkan array [2,3] :5
4565
3467
8215
BUILD SUCCESSFUL (total time: 20 seconds)
```