

LAPORAN PRAKTIKUM

ALGORITMA DAN PEMROGRAMAN LANJUT

PERTEMUAN KE-2



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya
NIM : 195410257
JURUSAN : Informatika
JENJANG : S1

SEMESTER ANTARA
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-2

(PERULANGAN BERTINGKAT DUA DAN TIGA)

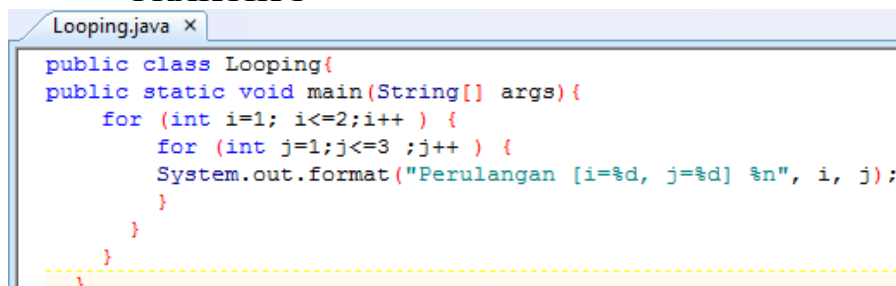
TUJUAN

Mahasiswa dapat membuat program untuk menyelesaikan kasus menggunakan perulangan bertingkat 2 maupun 3

DASAR TEORI

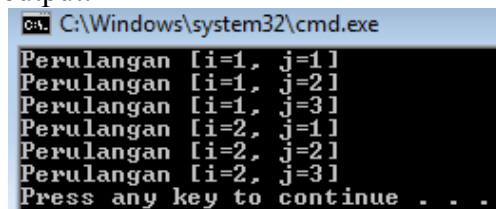
Dalam Java, ada tiga struktur kontrol perulangan yaitu: for, while, dan do-while. Untuk yang belum tahu: Perulangan (atau yang disebut Looping) adalah suatu proses yang dilakukan secara berulang-ulang hingga mencapai kondisi tertentu. Sebagai contoh ketika anda ingin mencetak deretan angka hingga batas tertentu (contoh: 1-100), maka anda bisa menggunakan fungsi looping dalam program. Biasanya fungsi looping digunakan dan berperan penting dalam algoritma sorting, karena kita akan menukar nilai variabel hingga menghasilkan nilai berurutan.

----- PRAKTIK 1 -----



```
public class Looping{
    public static void main(String[] args){
        for (int i=1; i<=2;i++ ) {
            for (int j=1;j<=3 ;j++ ) {
                System.out.format("Perulangan [i=%d, j=%d] %n", i, j);
            }
        }
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
Perulangan [i=1, j=1]
Perulangan [i=1, j=2]
Perulangan [i=1, j=3]
Perulangan [i=2, j=1]
Perulangan [i=2, j=2]
Perulangan [i=2, j=3]
Press any key to continue . . .
```

penjelasan:

program looping ini merupakan contoh program looping dengan For. Perulangan for adalah Perulangan yang digunakan ketika ingin mengeksekusi perintah program yang sama dengan jumlah proses perulangan yang sudah diketahui dengan mengacu pada kondisi yang ditetapkan. Untuk membuat program di atas, kita awali dengan membuat class bermodifier public bernama Looping, hal itu dituliskan dengan **public class Looping**. Setelah itu, kita tulis **public static void main (String args[])**. Dalam baris kode ini, public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah.

Selanjutnya kita masuk ke looping (perulangan). Kita tulis baris kode **for (int i=1; i<=2;i++)**. **int i = 1** berarti hitungan perulangan dimulai dari 1, **i<=2** berarti selama nilai hitungannya lebih kecil atau sama dengan 2, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 2 kali. **i++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap pengulangan.

Kita tulis baris kode **for (int j=1;j<=3 ;j++)**. **int j = 1** berarti hitungan perulangan dimulai dari 1, **j<=3** berarti selama nilai hitungannya lebih kecil atau sama dengan 3, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 3 kali. **j++** fungsinya adalah

untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Kita tulis `System.out.format("Perulangan [i=%d, j=%d] %n", i, j);`

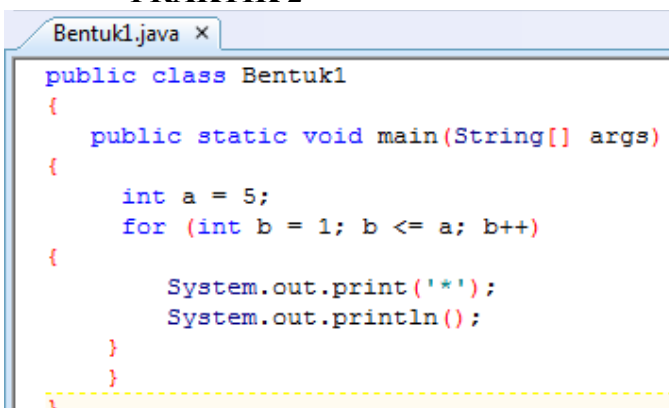
%d berfungsi untuk menentukan bahwa variabel yang dimaksudkan adalah bilangan desimal atau angka, sehingga kita harus menulis `[i=%d, j=%d]`. Lalu kita gunakan %n untuk membuat baris baru di setiap perulangan supaya program terlihat rapi. Terakhir, kita masukkan nilai variabel di akhir kalimat yaitu dengan menuliskan `i, j`.

Lalu kenapa outputnya bisa berbentuk begini?

```
Perulangan [i=1, j=1]
Perulangan [i=1, j=2]
Perulangan [i=1, j=3]
Perulangan [i=2, j=1]
Perulangan [i=2, j=2]
Perulangan [i=2, j=3]
Press any key to continue . . .
```

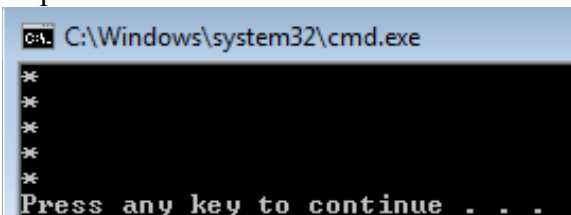
Karena hal itu disebabkan oleh looping bertingkat. Perulangan for pertama milik variabel `i`, bertugas sebagai penentu berapa kali perulangan `j` (1,2,3) akan dilakukan. Kita tadi menuliskan `i<=2` pada perulangan `i`, sehingga hal itu menyebabkan perulangan `j` (1,2,3) dilakukan hingga 2 kali, sehingga ada total 6 baris. Pada perulangan `j` pertama, muncul angka 1 2 3 lalu diikuti oleh perulangan `i` 1 1 1 yang berarti baris tersebut merupakan perulangan pertama. Pada perulangan `j` kedua, muncul angka 1 2 3 lalu diikuti oleh perulangan `i` 2 2 2 yang berarti baris tersebut merupakan perulangan kedua.

----- PRAKTIK 2 -----



```
Bentuk1.java x
public class Bentuk1
{
    public static void main(String[] args)
    {
        int a = 5;
        for (int b = 1; b <= a; b++)
        {
            System.out.print('*');
            System.out.println();
        }
    }
}
```

output:



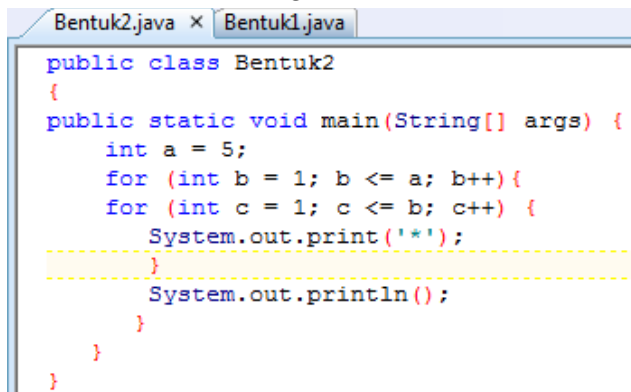
```
C:\Windows\system32\cmd.exe
*
*
*
*
*
Press any key to continue . . .
```

penjelasan:

pertama, kita membuat pendeklarasian kelas menggunakan baris kode **public class Bentuk1**. Class Bentuk1 bermodifier public yang artinya dapat diakses oleh seluruh class didalam package yang sama atau diluar package yang berbeda, modifier jenis ini memiliki tingkat akses yang sangat luas, hingga seluruh sumber daya dapat diakses oleh class manapun tanpa batasan. Setelah itu kita awali eksekusi program dengan **public static void main (String args[])**. Dalam baris kode ini, public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah. Langkah selanjutnya adalah kita harus mendeklarasikan variabel bertipe integer bernama `a` yang kita beri nilai 5. hal tersebut kita tulis dengan baris kode

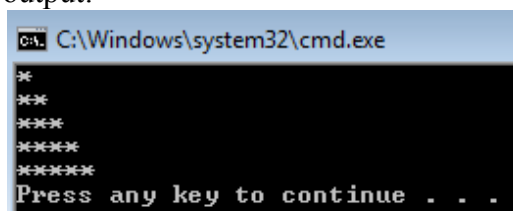
int a = 5; kemudian kita buat looping nya yaitu: **for (int b = 1; b <= a; b++)**. int b = 1 berarti hitungan perulangan dimulai dari 1, b<=a berarti selama nilai hitungannya lebih kecil atau sama dengan a, maka perulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 5 kali karena tadi kita memberi nilai 5 ke variabel a. b++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Terakhir kita tentukan data yang akan diulang. Pada program ini, yang diulang adalah '*' atau bisa kita tulis dengan baris kode **System.out.print('*');** Sehingga, hasilnya adalah perulangan tanda '*' sebanyak 5 baris lurus secara vertikal.

----- PRAKTIK 3 -----



```
public class Bentuk2
{
    public static void main(String[] args) {
        int a = 5;
        for (int b = 1; b <= a; b++){
            for (int c = 1; c <= b; c++) {
                System.out.print('*');
            }
            System.out.println();
        }
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
**
***
****
*****
Press any key to continue . . .
```

penjelasan:

pada program ini, kita menggunakan perulangan bertingkat dua. Seperti biasa, pertama-tama kita buat sebuah class bernama Bentuk2 bermodifikasi public dengan baris kode **public class Bentuk2**. Lalu sebelum memulai eksekusi looping, kita deklarasikan dahulu **public static void main(String[] args)**. Public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah.

Selanjutnya kita buat variabel bertipe integer bernama a yang kita beri nilai 5 yaitu dengan menuliskan baris kode **int a = 5;**. Tujuan dari pemberian nilai 5 kepada variabel a adalah supaya program yang terlooping akan menghasilkan 5 baris saja. Setelah mendeklarasikan variabel a, kita langsung mulai penulisan baris-baris kode looping for bertingkat. Pertama, kita buat looping for milik variabel b yaitu **for (int b = 1; b <= a; b++)**. int b = 1 berarti hitungan perulangan dimulai dari 1, b<=a berarti selama nilai hitungannya lebih kecil atau sama dengan a, maka perulangan akan terus dilakukan. Jadi, perulangan ini akan mengulang sebanyak 5 kali karena tadi kita memberi nilai 5 ke variabel a. b++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

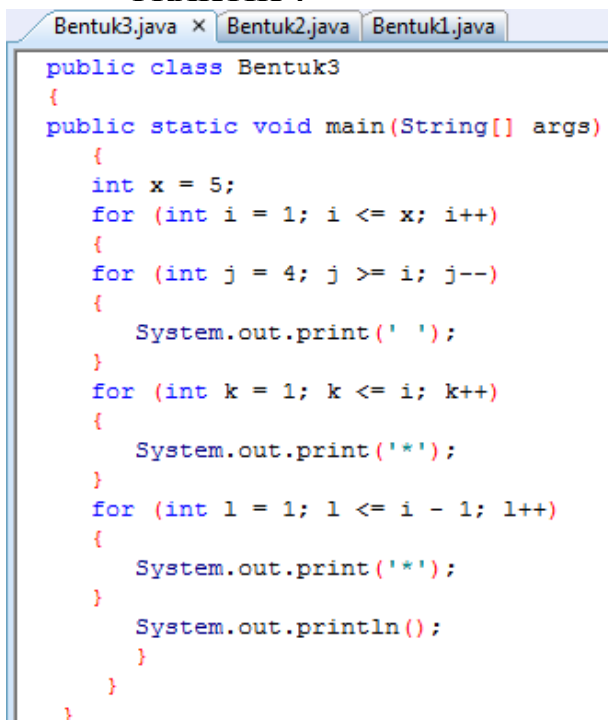
Kedua, kita buat perulangan for variabel c di dalam perulangan for variabel b yaitu dengan menuliskan baris kode **for (int c = 1; c <= b; c++)**. int c = 1 berarti hitungan perulangan dimulai dari 1, c<=b berarti selama nilai hitungannya lebih kecil atau sama dengan b, maka perulangan akan terus dilakukan. c++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Lalu kita tentukan objek atau data yang akan digunakan sebagai perulangan, yaitu tanda '*', yang kita eksekusi dengan baris kode **System.out.print('*');**

Lalu, mengapa hasil outputnya bisa begini?

```
*
**
***
****
*****
Press any key to continue
```

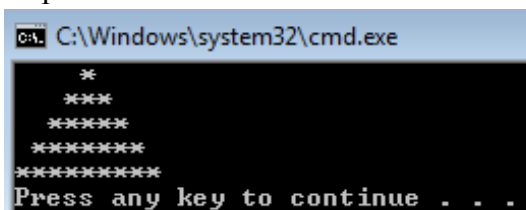
Program mengeluarkan output seperti itu karena kita mempunyai 3 variabel utama. Yang pertama adalah *a* yang menentukan banyaknya baris yaitu 5 baris. Kedua yaitu perulangan for variabel *b*, yang berfungsi untuk memberikan enter atau garis baru, yang berfungsi untuk membuat pola segitiga. Ketiga yaitu perulangan for variabel *c* yang merupakan perulangan yang mencetak tanda '*' mulai dari jumlah 1 yang bertambah 1 setiap barisnya hingga baris ke 5.

----- PRAKTIK 4 -----



```
public class Bentuk3
{
    public static void main(String[] args)
    {
        int x = 5;
        for (int i = 1; i <= x; i++)
        {
            for (int j = 4; j >= i; j--)
            {
                System.out.print(' ');
            }
            for (int k = 1; k <= i; k++)
            {
                System.out.print('*');
            }
            for (int l = 1; l <= i - 1; l++)
            {
                System.out.print('*');
            }
            System.out.println();
        }
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
*
**
***
****
*****
Press any key to continue . . .
```

penjelasan:

pada program di atas, kita membuat segitiga sama kaki atau segitiga piramida. Kita membutuhkan 4 variabel yaitu *x*, *a*, *b*, *c*, dan *d*. Pada program ini, kita membutuhkan 3 inner loop yang diantaranya masing-masing berfungsi untuk memberikan spasi ke samping, menampilkan bintang dari kanan ke kiri, dan terakhir untuk menampilkan bintang dari kiri ke kanan. Pertama, mari kita bahas dan analisis outer loop yaitu **for (int i = 1; i <= x; i++)**. Inti dari outer loop ini adalah variabel *i* dimulai dari 1, *i* akan terus di looping dari 1 hingga bertemu dengan *x* (5 baris) jika kondisi *i* adalah kurang dari sama dengan *x*, lalu *i* akan terus bertambah hingga menyentuh garis 5.

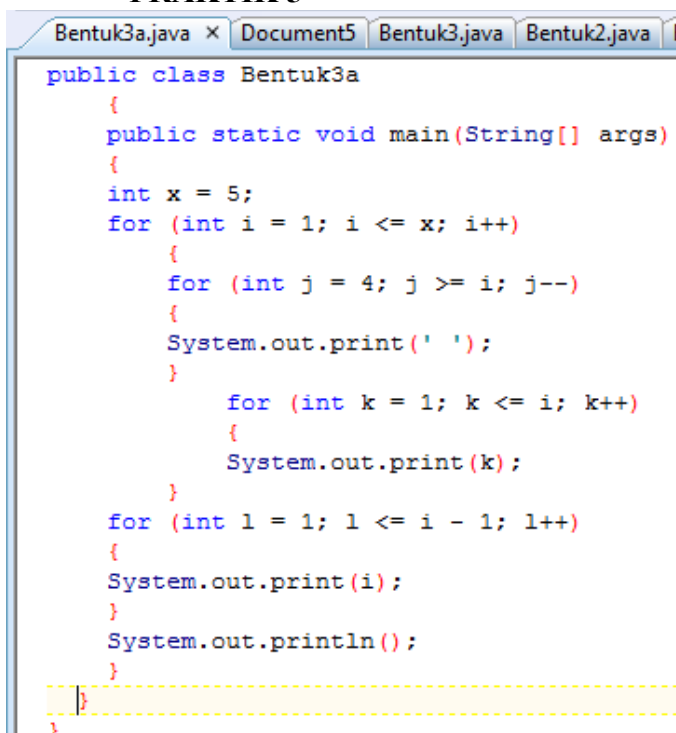
inner loop pertama adalah **(int j = 4; j >= i; j--)** yang fungsinya untuk memberikan spasi ke samping, maka dari itu variabel *j* terus berkurang hingga 4 langkah.

Inner loop kedua adalah **for (int k = 1; k <= i; k++)** yang fungsinya untuk menampilkan bintang dari kanan ke kiri. **int k = 1** berarti hitungan perulangan dimulai dari 1, **k <= i** berarti selama nilai hitungannya lebih kecil atau sama dengan i, maka perulangan akan terus dilakukan. **k++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Inner loop ketiga adalah **for (int l = 1; l <= i - 1; l++)** yang fungsinya untuk menampilkan bintang dari kiri ke kanan. **int l = 1** berarti hitungan perulangan dimulai dari 1, **l <= i** berarti selama nilai hitungannya lebih kecil atau sama dengan i, maka perulangan akan terus dilakukan. **l++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Terakhir, kita print atau cetak objek/data yang akan diloop yaitu '*'. Hal tersebut kita lakukan dengan menuliskan baris kode **System.out.print('*');**. Akhirnya, kita bisa menghasilkan program segitiga sama kaki atau piramida bintang.

----- PRAKTIK 5 -----

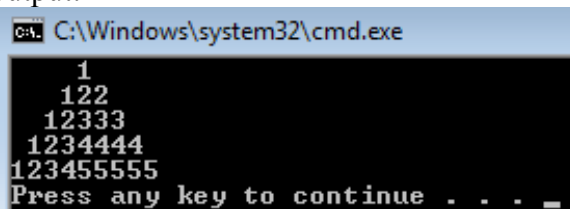


```
public class Bentuk3a
{
    public static void main(String[] args)
    {
        int x = 5;
        for (int i = 1; i <= x; i++)
        {
            for (int j = 4; j >= i; j--)
            {
                System.out.print(' ');
            }

            for (int k = 1; k <= i; k++)
            {
                System.out.print(k);
            }

            for (int l = 1; l <= i - 1; l++)
            {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
1
122
12333
1234444
123455555
Press any key to continue . . . _
```

penjelasan:

sebenarnya konsepnya disini hampir sama dengan konsep di praktik 4. Hanya saja, di program ini kita menggunakan angka sebagai pola segitiganya. Pertama, kita buat dulu class bermodifer public bernama Bentuk3a dengan menuliskan baris kode **public class Bentuk3a**. Lalu dilanjutkan dengan pendeklarasian **public static void main (String args[])**. **public** berarti metode ini dapat dipanggil dari luar class, **static** menunjukkan metode ini bersifat sama untuk semua class, **void** berarti metode ini tidak mengembalikan nilai, **args[]** adalah array objek string argument baris-baris perintah. Kemudian kita buat integer bernama x yang kita beri nilai 5. di dalam program ini, kita membutuhkan 4 perulangan for. 1 berupa outer loop, dan sisanya merupakan inner loop. masing-

masing berfungsi untuk memberikan spasi ke samping, menampilkan bintang dari kanan ke kiri, dan terakhir untuk menampilkan bintang dari kiri ke kanan. Outer loop pada program di atas adalah **for (int i = 1; i <= x; i++)** yang berfungsi untuk memberikan enter atau garis baru, yang berfungsi untuk membuat pola segitiga sebanyak 5 baris (karena kita tadi memberi nilai 5 ke variabel x). pada loop **for (int i = 1; i <= x; i++)**, variabel i dimulai dari 1, i akan terus dilooping dari 1 hingga bertemu dengan x (5 baris) jika kondisi i adalah kurang dari sama dengan x, lalu i akan terus bertambah hingga menyentuh garis 5.

inner loop pertama adalah **for (int j = 4; j >= i; j--)** memberikan spasi ke samping, maka dari itu variabel j terus berkurang hingga 4 langkah. int j = 1 berarti hitungan perulangan dimulai dari 4, j<=i berarti selama nilai hitungannya lebih besar atau sama dengan i, maka perulangan akan terus dilakukan. j-- fungsinya adalah untuk mengurangi satu (-1) nilai hitungan pada setiap perulangan.

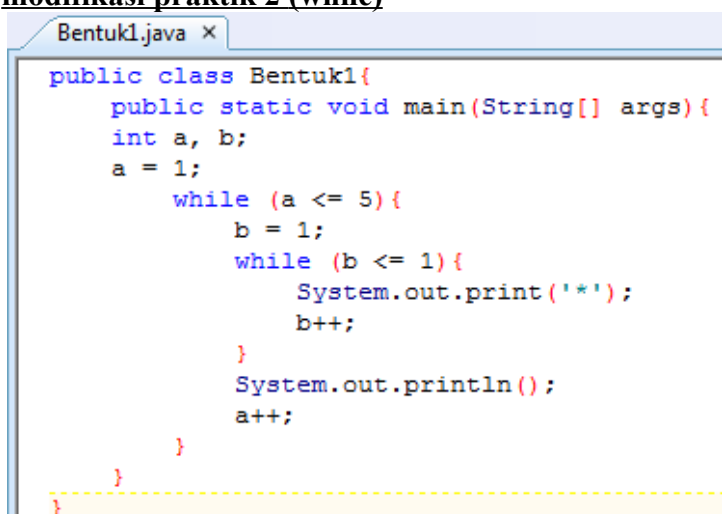
Inner loop kedua adalah **for (int k = 1; k <= i; k++)** yang fungsinya untuk menampilkan bintang dari kanan ke kiri. int k = 1 berarti hitungan perulangan dimulai dari 1, k<=i berarti selama nilai hitungannya lebih kecil atau sama dengan i, maka perulangan akan terus dilakukan. k++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Inner loop ketiga adalah **for (int l = 1; l <= i - 1; l++)** yang fungsinya untuk menampilkan bintang dari kiri ke kanan. int l = 1 berarti hitungan perulangan dimulai dari 1, l<=i berarti selama nilai hitungannya lebih kecil atau sama dengan i, maka perulangan akan terus dilakukan. l++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Lalu, apa bedanya dengan program praktik 4? Perbedaan terletak pada objek/data yang digunakan untuk membuat pola segitiga. Kita print atau cetak objek/data yang akan diloop yaitu '*'. Hal tersebut kita lakukan dengan menuliskan baris kode **System.out.print('i');**. Hal itu membuat perulangan dimulai dari angka 1, lalu di baris kedua ada angka 1 diikuti 2 berjumlah 2, selanjutnya ada angka 1 2 diikuti 3 berjumlah 3, selanjutnya ada angka 1 2 3 diikuti 4 berjumlah 4, terakhir ada angka 1 2 3 4 diikuti angka 5 berjumlah 5. Jadi terbentuklah segitiga sama kaki berpola angka.

----- PRAKTIK 6 -----

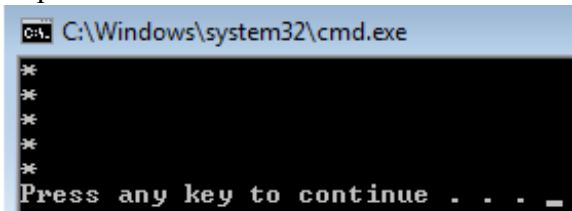
modifikasi praktik 2 (while)



```
public class Bentuk1{
    public static void main(String[] args){
        int a, b;
        a = 1;
        while (a <= 5){
            b = 1;
            while (b <= 1){
                System.out.print('*');
                b++;
            }
            System.out.println();
            a++;
        }
    }
}
```

output di halaman selanjutnya

output:



```
C:\Windows\system32\cmd.exe
*
*
*
*
*
Press any key to continue . . .
```

penjelasan:

seperti yang bisa kita lihat, program di modifikasi praktik 2 ini menampilkan output yang sama persis dengan program di praktik 2. Maka dari itu, perbedaannya adalah di bagian script program pada praktik ini sendiri. Jika sebelumnya kita menggunakan for, maka di script program ini kita menggunakan while sebagai perulangan untuk menggantikan perulangan for. Apa itu perulangan atau looping while? Perulangan while adalah jenis perulangan yang digunakan ketika kita belum mengetahui jumlah proses perulangan. Banyaknya perulangan ditentukan dari proses inisialisasi dan kondisi yang didefinisikan. Hal itu tentu berbeda dengan perulangan for yang sudah dapat kita ketahui jumlah proses perulangan dari kondisi yang ditetapkan.

Struktur perulangan while adalah seperti berikut:

```
while (kondisi){  
    pernyataan;  
}
```

Kondisi – Sebagai sebuah syarat untuk menghentikan proses perulangan

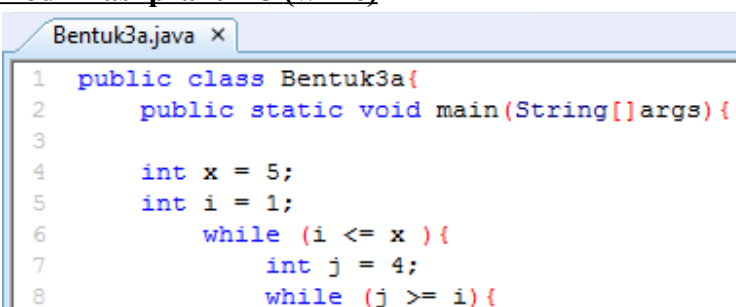
Pernyataan – Bagian instruksi-instruksi program yang akan diulang saat kondisi bernilai benar (true)

Untuk menghentikan perulangan while, kita harus menambahkan suatu perintah yang digunakan untuk mengubah nilai kondisi menjadi false agar proses perulangan dapat berhenti.

Untuk pembuatan program di atas, kita gunakan baris kode **public class Bentuk1** untuk membuat class bermodifier public bernama Bentuk1. Class Bentuk1 bermodifier public yang artinya dapat diakses oleh seluruh class didalam package yang sama atau diluar package yang berbeda, modifier jenis ini memiliki tingkat akses yang sangat luas, hingga seluruh sumber daya dapat diakses oleh class manapun tanpa batasan tertentu. Kemudian kita lanjutkan dengan deklarasi **public static void main(String[] args)**. Selanjutnya kita buat dua buah variabel bertipe integer yaitu variabel a dan variabel b dengan menuliskan baris kode **int a, b;** dan sekaligus memberi nilai 1 pada variabel a dengan menuliskan baris kode **a = 1;**

Sekarang kita masuk ke proses perulangan atau looping while. Yang pertama adalah perulangan while outer yaitu **while (a <= 5)** artinya jika nilai variabel a kurang atau sama dengan 5, maka b dimulai hitungannya dari 1. Lalu di dalamnya kita buat inner looping while yaitu **while (b <= 1)** yang artinya jika perulangan b kurang dari atau sama dengan 1, maka print '*'. Lalu modifier b++ kita letakkan di akhir inner loop, dan modifier a++ kita letakkan di akhir outer loop.

modifikasi praktik 5 (while)



```
Bentuk3a.java x
1 public class Bentuk3a{
2     public static void main(String[] args){
3
4         int x = 5;
5         int i = 1;
6         while (i <= x ){
7             int j = 4;
8             while (j >= i){
```

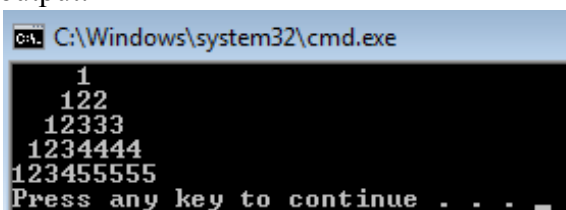


```

9         System.out.print(' ');
10        j--;
11    }
12    int k = 1;
13    while (k <= i){
14        System.out.print(k);
15        k++;
16    }
17    int l = 1;
18    while (l <= i - 1){
19        System.out.print(i);
20        l++;
21    }
22    System.out.println();
23    i++;
24 }
25 }
26

```

output:



```

C:\Windows\system32\cmd.exe
1
12
123
1234
12345
Press any key to continue . . . _

```

penjelasan:

sama seperti modifikasi program praktik 2 tadi, di program kali ini kita juga mengubah perulangan program yang tadinya menggunakan for menjadi menggunakan perulangan while. Seperti yang sudah dijelaskan tadi, perulangan while adalah jenis perulangan yang digunakan ketika kita belum mengetahui jumlah proses perulangan. Banyaknya perulangan ditentukan dari proses inisialisasi dan kondisi yang didefinisikan. Untuk pembuatan program di atas, kita mulai dengan pemberian nama class yaitu Bentuk3a yang ditulis dengan baris kode **public class Bentuk3a**. Kemudian langsung kita lanjutkan dengan deklarasi **public static void main(String[]args)**. Public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah. Lalu kita buat dua variabel yaitu x yang bernilai 5 dan i yang bernilai 1.

pertama, kita buat perulangan while untuk variabel j yang berfungsi untuk memberikan spasi ke samping. **while (i <= x)** yang artinya jika dan selama i bernilai kurang dari atau sama dengan x, maka program akan terus berjalan, dan j dimulai dari 4 yaitu dituliskan dengan **int j = 4;**. setelah itu kita buat inner loop yaitu **while (j >= i)**, maka kita buat spasi supaya kita bisa memberikan spasi ke samping di dalam program yang dituliskan dengan baris kode **System.out.print(' ')**. Lalu modifier j-- kita beri di luar inner class yang berarti nilai j terus berkurang sampai 4 langkah.

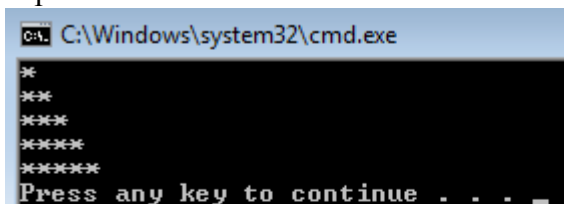
Kedua, kita buat perulangan **while (k <= i)** yang sebelumnya telah kita beri nilai 1 yang berarti perulangan while k dimulai dari 1. Selama k kurang dari atau sama dengan i, maka program akan mencetak k dan akan bertambah terus hingga 5 kali. Fungsi dari perulangan ini adalah untuk menampilkan bintang dari kanan ke kiri.

Ketiga, kita buat perulangan **while (l <= i - 1)** yang sebelumnya telah kita beri nilai 1 yang berarti perulangan while l dimulai dari 1. Selama k kurang dari atau sama dengan i dikurangi 1, maka program akan mencetak i dan akan bertambah terus hingga 5 kali. Fungsi dari perulangan ini adalah untuk menampilkan bintang dari kiri ke kanan. Diperlukan pengurangan 1 pada i untuk menarik pola segitiga dari kiri ke kanan. Terakhir, kita beri modifier l++ pada inner loop dan i++ pada outer loop. Hingga terbentuk segitiga sama kaki angka.

modifikasi praktik 3 (do-while)

```
Bentuk2.java x
public class Bentuk2
{
    public static void main(String[] args) {
        int a = 5;
        int b = 1;
        do
        {
            int c = 1;
            do {
                System.out.print("*");
                c++;
            }
            while ( c <= b);
            b++;
            System.out.println();
        }
        while(b <= a);
    }
}
```

output:



```
CA: C:\Windows\system32\cmd.exe
*
**
***
****
*****
Press any key to continue . . . _
```

penjelasan:

pada modifikasi program praktik 3 ini, kita menggunakan perulangan yang berbeda dari kedua perulangan sebelumnya. Kali ini, kita menggunakan perulangan do-while. Apa itu do while? Perulangan DO WHILE merupakan modifikasi dari perulangan WHILE, yakni dengan memindahkan posisi pemeriksaan kondisi ke akhir perulangan. Artinya, lakukan dahulu sebuah perulangan, baru periksa apakah kondisi variabel counter sudah terpenuhi atau belum di akhir perulangan. Format dasar struktur perulangan do while adalah seperti berikut:

start;

do

{

// kode program

// kode program

increment;

}

while (condition)

Sama seperti perulangan WHILE, di bagian start biasanya terdapat perintah inisialisasi variabel counter, misalnya $i = 0$. Kemudian di dalam block do ditulis kode program yang akan di ulang, kemudian kita buat sebuah perintah untuk menaikkan nilai variabel counter, misalnya dengan perintah $i++$. Di bagian paling bawah, terdapat perintah while (condition). Di sinilah kondisi perulangan akan diperiksa. Selama kondisi ini menghasilkan nilai TRUE, maka perulangan akan lanjut ke iterasi berikutnya.

Pada kasus program modifikasi di atas, kita awali dengan pembuatan class bernama Bentuk 2 dengan menuliskan baris kode **public class Bentuk2**. Lalu dilanjutkan dengan deklarasi **public static void main(String[] args)**. Sebelum melakukan perulangan do while, alangkah baiknya kita membuat dan mendeklarasikan variabel terlebih dahulu. Kita buat dua variabel bernama a yang kita beri nilai 5 dan variabel bernama b yang kita beri nilai 1.

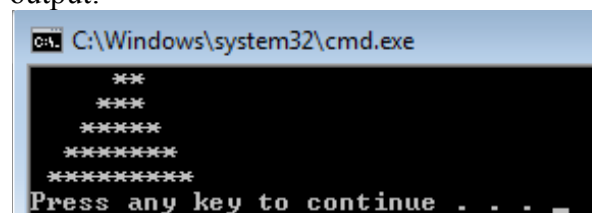
kita mulai proses perulangan do while. Pertama, kita buat outer do terlebih dahulu. Di dalamnya kita deklarasikan variabel baru bernama c yang kita beri nilai 1 yaitu dengan menuliskan baris kode **int c = 1;** Kemudian kita lakukan do untuk langsung memproses program yaitu mencetak bintang "*" yang terus bertambah hingga 5 kali (c++). Untuk menutup do tersebut, kita gunakan **while (c <= b)** yang berarti program akan dijalankan selama nilai c adalah kurang dari atau sama dengan nilai pada b. Kemudian kita tulis **b++;** supaya nilai b terus bertambah hingga batas nilai yang ditentukan tadi. Terakhir, kita deklarasikan **while(b <= a);** untuk menutup outer loop do yang artinya selama b bernilai kurang dari atau sama dengan a (5), maka perulangan b akan dilakukan.

bisa disimpulkan fungsi perulangan do while pada program di atas adalah: yang pertama adalah a yang menentukan banyaknya baris yaitu 5 baris. Kedua yaitu perulangan variabel b, yang berfungsi untuk memberikan enter atau garis baru, yang berfungsi untuk membuat pola segitiga. Ketiga yaitu perulangan variabel c yang merupakan perulangan yang mencetak tanda '*' mulai dari jumlah 1 yang bertambah 1 setiap barisnya hingga baris ke 5.

modifikasi praktik 4 (do-while)

```
Bentuk3a.java Bentuk3.java x
1 public class Bentuk3{
2     public static void main(String[] args) {
3
4     int x = 5;
5     int i = 1;
6     do{
7         int j = 5;
8         do{
9             System.out.print(' ');
10            j--;
11        }while(j >= i);
12
13        int k = 1;
14        do{
15            System.out.print('*');
16            k++;
17        }while (k <= i);
18
19        int l = 1;
20        do{
21            System.out.print('*');
22            l++;
23        }while (l <= i-1);
24
25        System.out.println();
26        i++;
27    }while (i <= x);
28    }
29 }
```

output:



```
C:\Windows\system32\cmd.exe
**
***
****
*****
*****
Press any key to continue . . . _
```

penjelasan:

pada program kali ini, kita memodifikasi program pada praktik 4. jika sebelumnya kita menggunakan perulangan for, maka pada program ini kita menggunakan perulangan do-while. Seperti yang sudah saya jelaskan sebelumnya, Perulangan DO WHILE merupakan modifikasi dari

perulangan WHILE, yakni dengan memindahkan posisi pemeriksaan kondisi ke akhir perulangan. Artinya, lakukan dahulu sebuah perulangan, baru periksa apakah kondisi variabel counter sudah terpenuhi atau belum di akhir perulangan. Untuk pembuatan script program di atas, bisa kita awali dengan membuat class dengan menuliskan baris kode **public class Bentuk3** dan sekaligus mendeklarasikan **public static void main(String[] args)**.

Setelah itu langsung saja kita mendeklarasikan **int x = 5** dan **int i = 1**. selanjutnya kita mulai perulangan do while pada program ini. Perulangan dimulai dengan membuat outer loop do kemudian di dalamnya kita deklarasikan integer j yang kita beri nilai 5 yaitu dengan menuliskan **int j = 5**; di dalam inner loop do while j, kita langsung eksekusi program dengan do lalu cetak spasi dengan menuliskan **System.out.print(' ');** dan dilanjutkan dengan **j--** yang berarti nilai j terus berkurang hingga 5 langkah. Loop itu diakhiri dengan pengecekan kondisi, jika kondisi **while(j >= i)**, maka loop akan berjalan dan iterasi akan dilanjutkan. Loop j ini berfungsi untuk memberikan spasi ke samping sebanyak 5 langkah.

Lalu kita buat inner loop lagi untuk variabel integer k yang kita beri nilai 1 dengan menuliskan **int k = 1** yang artinya perulangan k dimulai dari perhitungan 1. Lalu langsung kita eksekusi dengan do yang dilanjutkan dengan **System.out.print('*')** dan **k++** yang artinya pola '*' tersebut akan dicetak secara bertambah hingga batas yang telah ditentukan. Kemudian kita cek kondisinya, jika benar berupa **while (k <= i)**, maka looping dijalankan. Loop do while k ini berfungsi untuk menampilkan bintang dari kanan ke kiri.

Kemudian kita buat inner loop lagi untuk variabel integer l yang kita beri nilai 1 dengan menuliskan **int l = 1** yang artinya perulangan k dimulai dari perhitungan 1. Lalu langsung kita eksekusi dengan do yang dilanjutkan dengan **System.out.print('*')** dan **l++** yang artinya pola '*' tersebut akan dicetak secara bertambah hingga batas yang telah ditentukan. Kemudian kita cek kondisinya, jika benar berupa **while (l <= i - 1)**, maka looping dijalankan. Loop do while l ini berfungsi untuk menampilkan bintang dari kiri ke kanan.

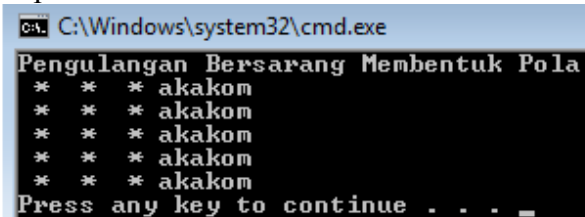
Terakhir, kita beri modifier **i++** untuk outer loop do yang artinya i terus bertambah, dan kita beri kondisi **while (i <= x)** untuk mengecek apakah nilai dari i ada dibawah atau sama dengan x, jika true maka program dijalankan.

----- LATIHAN 1 -----

```
Polal.java x
public class Polal {
    public static void main(String[] args) {
        int i,j;
        System.out.println("Pengulangan Bersarang Membentuk Pola");
        for(i=1;i<=5;i++){
            for(j=1;j<=3;j++){
                System.out.print(" * ");
            }
            System.out.println("akakom ");
        }
    }
}
```

output di halaman selanjutnya

output:



```
C:\Windows\system32\cmd.exe
Pengulangan Bersarang Membentuk Pola
* * * akakom
* * * akakom
* * * akakom
* * * akakom
* * * akakom
Press any key to continue . . . _
```

penjelasan:

Program yang kita buat di latihan ini cukup simple dan sederhana. Pertama-tama, untuk membuat program ini kita buat class bernama Pola1 yang kita tuliskan dengan baris kode **public class Pola1**. Class tersebut memiliki modifier public yang artinya dapat diakses oleh seluruh class didalam package yang sama atau diluar package yang berbeda, modifier jenis ini memiliki tingkat akses yang sangat luas, hingga seluruh sumber daya dapat diakses oleh class manapun tanpa batasan. Kemudian dilanjutkan dengan menuliskan baris kode **public static void main(String[] args)**. Public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah.

Kita buat dua variabel yaitu i dan j, lalu kita cetak sebuah kalimat yaitu dengan menuliskan baris kode **System.out.println("Pengulangan Bersarang Membentuk Pola")** yang akan muncul di awal program sebagai judul. Setelah itu kita mulai perulangan menggunakan for. Perulangan for adalah perulangan yang digunakan ketika ingin mengeksekusi perintah program yang sama dengan jumlah proses perulangan yang sudah diketahui dengan mengacu pada kondisi yang ditetapkan.

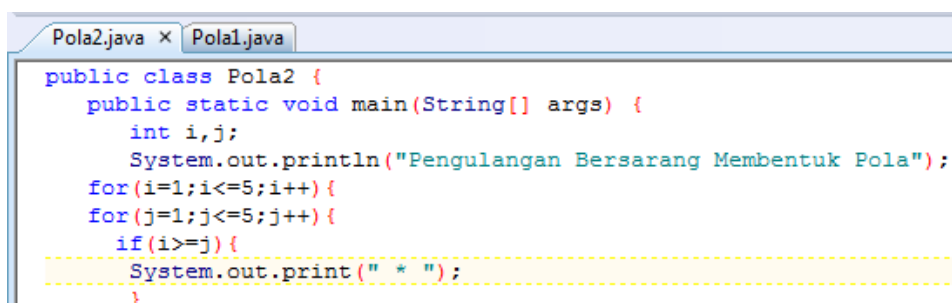
Pada program ini, kita menerapkan perulangan beringkat. Perulangan yang pertama atau bisa disebut dengan perulangan outer adalah **for(i=1;i<=5;i++)**. **int i = 1** berarti hitungan perulangan dimulai dari 1, **i<=5** berarti selama nilai hitungannya lebih kecil atau sama dengan 5, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 5 kali. **i++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Fungsi dari perulangan outer ini adalah untuk mencetak kata "akakom" sebanyak 5 baris.

Perulangan for yang kedua atau perulangan inner adalah **for(j=1;j<=3;j++)**. **int i = 1** berarti hitungan perulangan dimulai dari 1, **i<=3** berarti selama nilai hitungannya lebih kecil atau sama dengan 3, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 3 kali. **i++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Fungsi dari perulangan inner ini adalah untuk mencetak tanda '*' sebanyak 3 pada setiap baris.

Terakhir,

kita tampilkan **System.out.print(" * ")** yang telah dilooping dengan **for(j=1;j<=3;j++)** kita tampilkan **System.out.println("akakom ")**; yang dilooping dengan **for(i=1;i<=5;i++)**

----- LATIHAN 2 -----



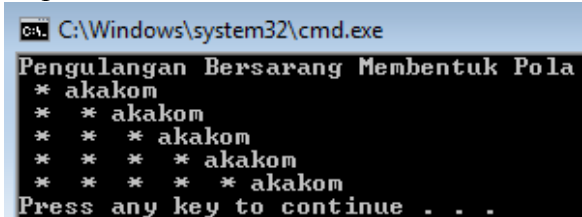
```
Pola2.java x Pola1.java
public class Pola2 {
    public static void main(String[] args) {
        int i,j;
        System.out.println("Pengulangan Bersarang Membentuk Pola");
        for(i=1;i<=5;i++){
            for(j=1;j<=3;j++){
                if(i>=j){
                    System.out.print(" * ");
                }
            }
        }
    }
}
```

```

    }
    }
    System.out.println("akakom ");
    }
}

```

output:



```

C:\Windows\system32\cmd.exe
Pengulangan Bersarang Membentuk Pola
* akakom
* * akakom
* * * akakom
* * * * akakom
* * * * * akakom
Press any key to continue . . .

```

penjelasan:

program yang kita buat pada latihan 2 ini sebenarnya hampir mirip dengan program yang kita buat di latihan 1 tadi. Hanya saja, di program ini jumlah bintang tidak konstan hanya 3 saja, tetapi jumlah bintang (*)urut dari 1-5 mengikuti jumlah baris.

Untuk pembuatan programnya, kita buat class bernama Pola2 yang kita tuliskan dengan baris kode **public class Pola2**. Modifier public pada class Pola 2 artinya class tersebut dapat diakses oleh seluruh class didalam package yang sama atau diluar package yang berbeda, dan modifier jenis ini memiliki tingkat akses yang sangat luas. Lalu kita lanjutkan dengan deklarasi **public static void main(String[] args)**. Public berarti metode ini dapat dipanggil dari luar class, static menunjukkan metode ini bersifat sama untuk semua class, void berarti metode ini tidak mengembalikan nilai, args[] adalah array objek string argument baris-baris perintah.

Langkah selanjutnya kita harus membuat dua variabel yaitu i dan j. Variabel i akan digunakan dalam perulangan outer sementara variabel j akan digunakan dalam perulangan inner. Hal itu kita lakukan dengan menuliskan **int i,j**. Setelah pendeklarasian variabel, kita langsung menuliskan sebuah kalimat pembuka program yaitu **System.out.println("Pengulangan Bersarang Membentuk Pola")**. Kemudian kita mulai perulangannya.

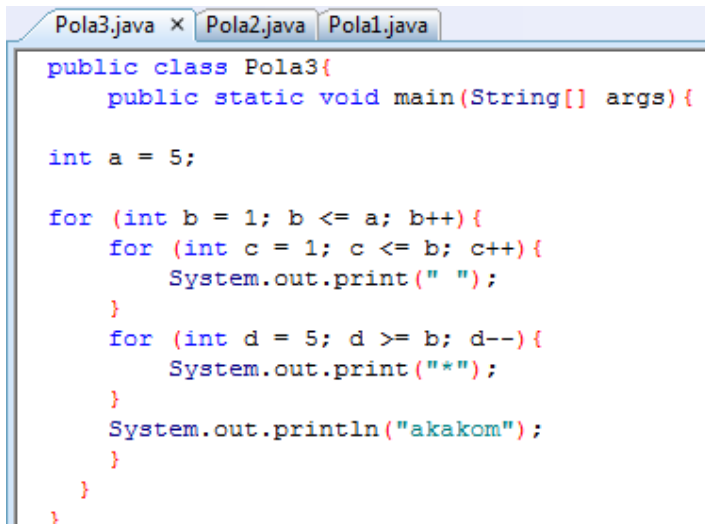
perulangan outer adalah **for(i=1;i<=5;i++)**. **int i = 1** berarti hitungan perulangan dimulai dari 1, **i<=5** berarti selama nilai hitungannya lebih kecil atau sama dengan 5, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 5 kali. **i++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Fungsi dari perulangan outer ini adalah untuk mencetak kata "akakom" sebanyak 5 baris.

for(j=1;j<=5;j++). **int i = 1** berarti hitungan perulangan dimulai dari 1, **i<=5** berarti selama nilai hitungannya lebih kecil atau sama dengan 5, maka pengulangan akan terus dilakukan. Dengan kata lain, perulangan ini akan mengulang sebanyak 5 kali. **i++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Fungsi dari perulangan inner ini adalah untuk mencetak tanda '*' pada setiap baris.

Lalu kita gunakan baris kode **if(i>=j)** supaya tanda '*' muncul pada setiap baris bertahap dimulai dari jumlah 1 hingga 5 di samping tulisan "akakom". Terakhir cetak kalimat **System.out.print(" *")** yaitu data yang diulang oleh perulangan inner dan kalimat **System.out.println("akakom ")** yang dicetak oleh perulangan outer.

Latihan 3 di halaman selanjutnya

----- LATIHAN 3 -----

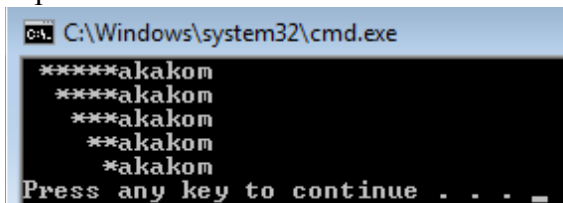


```
public class Pola3{
    public static void main(String[] args){

        int a = 5;

        for (int b = 1; b <= a; b++){
            for (int c = 1; c <= b; c++){
                System.out.print(" ");
            }
            for (int d = 5; d >= b; d--){
                System.out.print("*");
            }
            System.out.println("akakom");
        }
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
*****akakom
****akakom
***akakom
**akakom
*akakom
Press any key to continue . . . _
```

penjelasan:

program ini konsepnya hampir sama dengan program-program latihan tadi. Perbedaannya disini adalah bentuk segitiganya terbalik. Lalu bagaimana cara agar segitiga nya terbalik? Caranya adalah dengan memulai hitungan bintang dari 5 lalu bertahap turun sampai ke 1 bintang. Proses pembuatan program tersebut diawali dengan buat class bernama Pola3 yang kita tuliskan dengan baris kode **public class Pola3** yang diikuti oleh **public static void main(String[] args)** dibawahnya. Kemudian langsung kita buat sebuah variabel a dan kita beri nilai 5 yaitu dengan menuliskan baris kode **int a = 5**. Lalu kita mulai perulangan for nya.

Langkah selanjutnya adalah memulai perulangan for. Pertama, kita buat looping outer for milik variabel b yaitu **for (int b = 1; b <= a; b++)**. **int b = 1** berarti hitungan perulangan dimulai dari 1, **b<=a** berarti selama nilai hitungannya lebih kecil atau sama dengan a, maka perulangan akan terus dilakukan. Jadi, perulangan ini akan mengulang sebanyak 5 kali karena tadi kita memberi nilai 5 ke variabel a. **b++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Jadi inti dari perulangan tersebut adalah berfungsi untuk memberikan enter atau garis baru, yang berfungsi untuk membuat pola segitiga sebanyak 5 baris.

Kedua, kita buat looping inner loop yaitu **for (int c = 1; c <= b; c++)** yang berfungsi memberikan spasi ke samping, maka dari itu nilai c terus menambah hingga 5 langkah. **int c = 1** berarti hitungan perulangan dimulai dari 1, **c<=b** berarti selama nilai hitungannya lebih kecil atau sama dengan b, maka perulangan akan terus dilakukan. **c++** fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

Ketiga, kita buat looping inner **for (int d = 5; d >= b; d--)** yang fungsinya untuk mencetak tanda bintang '*'. **int d = 5** berarti perulangan d dimulai dari 5. Kemudian **d >= b** berarti selama nilai d lebih besar daripada b, maka perulangan akan terus dilakukan. **d--** fungsinya adalah untuk mengurangi satu (-1) nilai hitungan pada setiap perulangan.

kita tampilkan **System.out.print(" * ")** yang telah dilooping dengan **for (int d = 5; d >= b; d--)**
kita tampilkan **System.out.println("akakom ")**; yang dilooping dengan **(int b = 1; b <= a; b++)**

-----TUGAS 1-----

```
Pola4.java x Pola3.java Pola2.java Pola1.java
1 public class Pola4 {
2     public static void main(String[] args) {
3
4         int i,j;
5         System.out.println("Pengulangan bersarang membentuk pola");
6         for(i=1; i<=6; i++){
7             for(j=1; j<=6; j++){
8                 if(i>j){
9                     System.out.print( j+" ");
10                }
11            }
12            System.out.println(" ");
13        }
14        for (i=4; i>=1; i--){
15            for (j=1; j<=6; j++){
16                if(i>=j){
17                    System.out.print(j+ " ");
18                }
19            }
20            System.out.println(" ");
21        }
22    }
23 }
```

output:

```
C:\Windows\system32\cmd.exe
Pengulangan bersarang membentuk pola
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
Press any key to continue . . .
```

penjelasan:

program yang kita buat pada Tugas ini sebenarnya memiliki konsep yang hampir sama dengan pola-pola sebelumnya. Di program ini, pada dasarnya kita membuat segitiga sama kaki yang menghadap terbalik ke kanan. Program kita awali dengan memberi nama kelas dengan menuliskan baris kode **public class Pola4** kemudian kita ikuti dengan **public static void main(String[] args)**.

Lalu kita buat dua buah variabel bernama *i* dan *j*. Kemudian buat sebuah kalimat pembuka program yaitu **System.out.println("Pengulangan bersarang membentuk pola")**. Perulangan pertama yang kita buat adalah **for(i=1; i<=6; i++)**. int *i* = 1 berarti hitungan perulangan dimulai dari 1, *i*<=6 berarti selama nilai hitungannya lebih kecil atau sama dengan 6, maka perulangan akan terus dilakukan. Jadi, perulangan ini akan mengulang sebanyak 6 kali. *i*++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan.

for(j=1; j<=6; j++) yaitu int *j* = 1 berarti hitungan perulangan dimulai dari 1, *i*<=6 berarti selama nilai hitungannya lebih kecil atau sama dengan 6, maka perulangan akan terus dilakukan. Jadi, perulangan ini akan mengulang sebanyak 6 kali. *j*++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Kemudian di dalam perulangan *j* itu ada seleksi **if(i>j)**, jika kondisi sesuai seleksi tersebut, maka kita cetak *j* sekaligus beri spasi yaitu dengan baris kode ini:

System.out.print(j+" ");

kemudian kita tulis **System.out.println(" ")** yang berfungsi untuk memberikan enter atau garis baru, yang berfungsi memberi ruang untuk membuat pola segitiga. Baris kode tersebut diproses dengan looping outer **for(i=1; i<=6; i++)**. kemudian kita buat perulangan outer yaitu dengan menuliskan baris kode **for (i=4; i>=1; i--)** yang berfungsi untuk menurunkan dari angka 5 hingga angka 1. di dalamnya ada perulangan inner yaitu **(j=1; j<=6; j++)**. Int j = 1 berarti hitungan perulangan dimulai dari 1, i<=6 berarti selama nilai hitungannya lebih kecil atau sama dengan 6, maka perulangan akan terus dilakukan. Jadi, perulangan ini akan mengulang sebanyak 6 kali. j++ fungsinya adalah untuk menambah satu (+1) nilai hitungan pada setiap perulangan. Lalu kita beri suatu seleksi (i>=j). Terakhir, kita cetak **System.out.print(j+ " ")** dengan perulangan inner **(j=1; j<=6; j++)** dan beri spasi untuk memberi ruang pembuatan pola segitiga dengan penulisan baris kode **for (i=4; i>=1; i--)**. sehingga pada akhirnya kita seperti membuat dua segitiga siku siku yang jika disatukan seperti satu segitiga yang miring menghadap ke kanan.

KESIMPULAN

Pada laporan ini, kita mempelajari banyak hal tentang perulangan bertingkat. Ada tiga truktur kontrol nilai yang kita praktekan pada pertemuan kali ini yaitu for, while, dan do while. Bisa disimpulkan bahwa:

Perulangan for adalah Perulangan yang digunakan ketika ingin mengekseskusi perintah program yang sama dengan jumlah proses perulangan yang sudah diketahui dengan mengacu pada kondisi yang ditetapkan.

Perulangan while adalah jenis perulangan yang digunakan ketika kita belum mengetahui jumlah proses perulangan. Banyaknya perulangan ditentukan dari proses inisialisasi dan kondisi yang didefinisikan.

Perulangan do while merupakan modifikasi dari perulangan while, yakni dengan memindahkan posisi pemeriksaan kondisi ke akhir perulangan. Artinya, lakukan dahulu sebuah perulangan, baru periksa apakah kondisi variabel counter sudah terpenuhi atau belum di akhir perulangan.

Terima Kasih.