***Modul 4***

**Perulangan**

**Tujuan :**

* Mahasiswa mampu menggunakan perintah perulangan dalam pemrograman java
* Mahasiswa mampu menggunakan berbagai bentuk perulangan dalam berbagai kebutuhan.

**Dasar Teori:**

**A. Perulangan**

Adakalanya suatu blok pernyataan harus dieksekusi berulang kali tergantung pada kondisi tertentu. Untuk keperluan tersebut, seperti halnya bahasa pemrograman yang lainnya, JAVA menyediakan beberapa statement perulangan. Dengan menggunakan statement tersebut, suatu blok pernyataan tidak perlu ditulis berulang kali, namun cukup dengan memberikan ekspresi boolean terhadap suatu kondisi.

1. **Perintah: for**

Bentuk for digunakan untuk melakukan perulangan, dimana banyaknya perulangan telah diketahui sebelumnya. Pernyataan dengan for akan memiliki counter yang akan bergerak (naik atau turun) secara otomatis.

Sintaks:

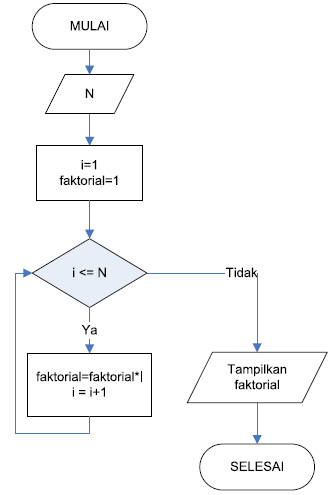
|  |
| --- |
| **for ( <inisialisasi> ; <ekspresi\_boolean> ; <increment> )**  **{**  **<blok pernyataan>**  **}** |

Misal, **for (int i=1; i<=10; i++)**, berarti proses perulangan akan berjalan mulai dari 1 sampai 10, dimana counter i akan naik secara otomatis (i++) hingga i mencapai angka 10.

Contoh 4.1:

Akan dihitung nilai N! (N faktorial). Besarnya N ditentukan melalui input data.

**Gambar 4.1** menunjukkan *flowchart* untuk permasalahan tersebut.



**Gambar 4.1** Diagram blok untuk contoh 4.1.

Kode program untuk permasalahan pada contoh 4.1 tersebut dapat dilihat pada **Gambar 4.2**.



**Gambar 4.2** Kode program untuk contoh 4.1.

Apabila program tersebut dieksekusi, maka hasilnya seperti pada **Gambar 4.3**.



(a)



(b)

**Gambar 4.3** Output yang dihasilkan oleh program pada contoh 4.1.

1. **Perintah: while**

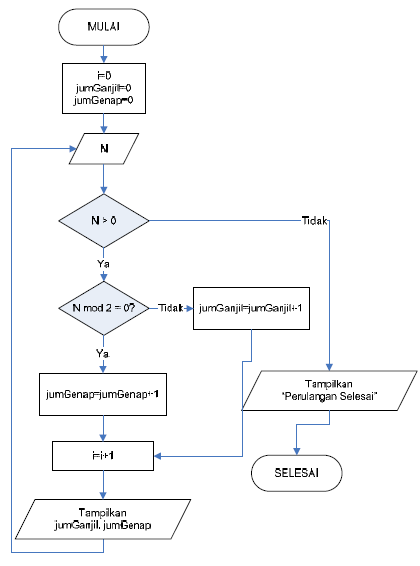
Pernyataan while digunakan untuk melakukan perulangan terhadap sejumlah pernyataan dalam blok selama kondisi (dinyatakan dengan ekspresi boolean) bernilai benar.

Sintaks:

|  |
| --- |
| **while (<ekspresi\_boolean>)**  **{**  **<blok pernyataan>**  **}** |

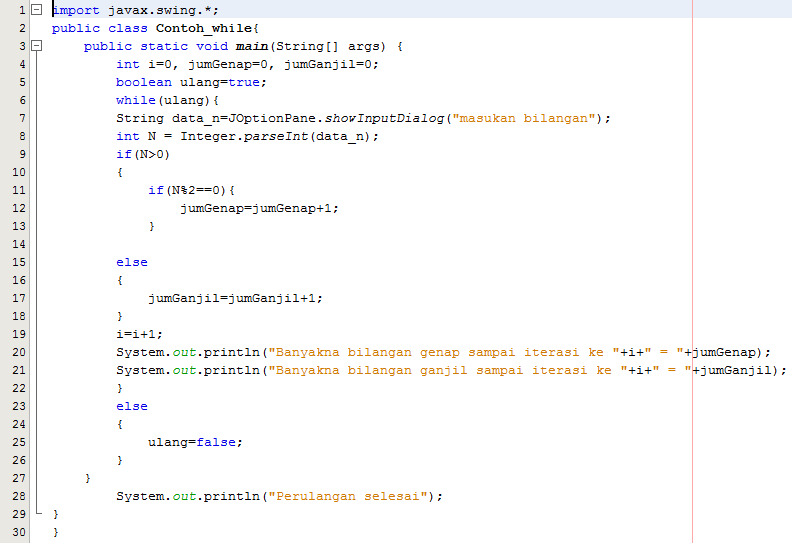
Contoh 4.2:

Berikut akan dihitung banyaknya bilangan ganjil dan bilangan genap yang dimasukkan melalui dialog box. Proses penentuan banyaknya bilangan akan terus diulang hingga pengguna memasukkan bilangan nol (0). Gambar menunjukkan flowchart untuk permasalahan tersebut.

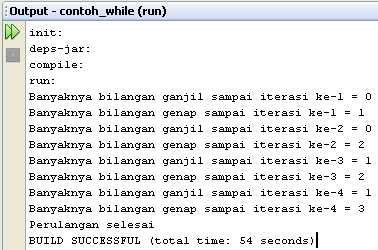


**Gambar 4.4** Diagram blok untuk contoh 4.2.

Kode program untuk permasalahan pada contoh 4.2 tersebut dapat dilihat pada **Gambar 4.5**. Apabila program tersebut dieksekusi, apabila diberikan input 6, 4, 5, 2, dan 0, maka hasilnya seperti pada **Gambar 4.6**.



**Gambar 4.5** Kode program untuk contoh 4.2.



**Gambar 4.6** Output yang dihasilkan oleh program pada contoh 4.2.

1. **Perintah: do-while**

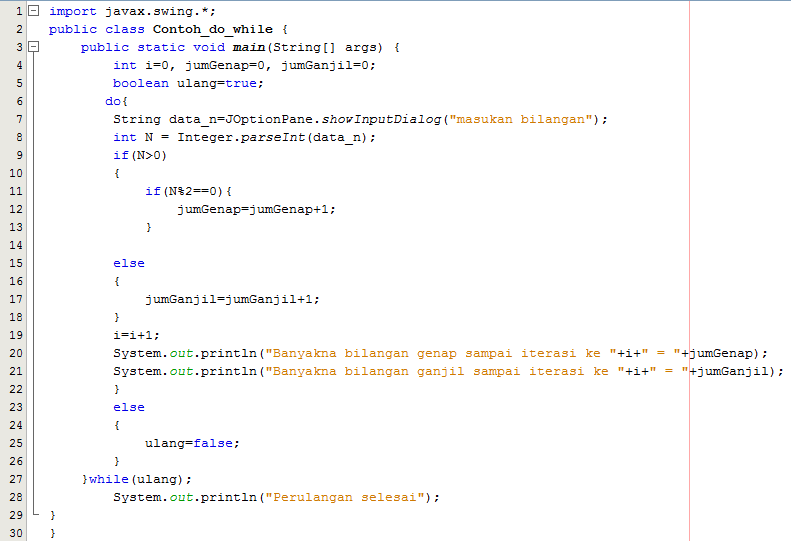
Pada pernyataan do-while, blok pernyataan dieksekusi hingga suatu kondisi yang dinyatakan oleh ekspresi logika pada while bernilai salah. Sedikit berbeda denga pernyataan while yang baru akan mengeksekusi blok pernyataan setelah diketahui kondisi benar, pada do-while ini blok akan dieksekusi setidaknya satu kali sebelum kondisi bernilai salah.

Sintaks:

|  |
| --- |
| **do**  **{**  **<blok pernyataan>**  **}**  **while (<ekspresi\_boolean>);** |

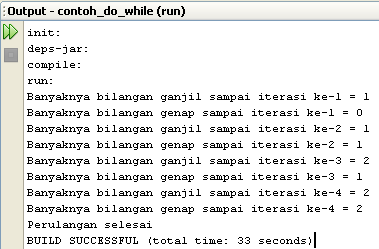
Contoh 4.3:

Pada contoh 4.2 akan coba dibuat dengan menggunakan dowhile. Kode program untuk permasalahan tersebut dapat dilihat pada **Gambar 4.7**.



**Gambar 4.7** Kode program untuk contoh 4.3.

Apabila program tersebut dieksekusi, dengan input bilangan 5, 8, 3, 10, dan 10, maka hasilnya seperti pada **Gambar 4.8**.



**Gambar 4.8** Output yang dihasilkan oleh program pada contoh 4.3.

1. **Latihan**

Lakukan ujicoba pada contoh program diatas dan lakukan analisa

1. **Tugas**
2. Buatlah sebuah program untuk menampilkan output berikut:

30 29 28 27 26.......16 1 2 3 4 5 6 7 8 9.....15

1. Buatlah sebuah program untuk menampilkan sederetan angka genap dan ganjil beserta jumlahnya

Contoh :

1 3 5 7 9 = 25

2 4 6 8 10 = 30

1. Buatlah sebuah program untuk menampilkan output berikut dengan menggunakan konsep perulangan do-while:

1

2

3

4

5

6

7

8

10 11 12 13 14 15 16 17 18 19 20

*Referensi :*

1. Sariadin Siallagan, *Pemrograman Java*, penerbit Andi, 2009