

JOBSHEET 7 DASAR PEMROGRAMAN

Nama : Jonathan Abdiel Haryono

NIM : 254107020036

Absen : 14

Kelas : 1G

Percobaan 1

```
1  import java.util.Scanner;
2
3  public class SiakadFor14 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          double nilai, tertinggi = 0, terendah = 100;
8
9          for (int i = 1; i <= 10; i++) {
10             System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
11             nilai = sc.nextDouble();
12
13             if (nilai > tertinggi) {
14                 tertinggi = nilai;
15             }
16             if (nilai < terendah) {
17                 terendah = nilai;
18             }
19         }
20         System.out.println("\nNilai tertinggi: " + tertinggi);
21         System.out.println("Nilai terendah: " + terendah);
22     }
23 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS +

```
Masukkan nilai mahasiswa ke-1: 76,5
Masukkan nilai mahasiswa ke-2: 82,3
Masukkan nilai mahasiswa ke-3: 62,1
Masukkan nilai mahasiswa ke-4: 88,4
Masukkan nilai mahasiswa ke-5: 65,9
Masukkan nilai mahasiswa ke-6: 67,9
Masukkan nilai mahasiswa ke-7: 90,1
Masukkan nilai mahasiswa ke-8: 55,3
Masukkan nilai mahasiswa ke-9: 73,7
Masukkan nilai mahasiswa ke-10: 78,6
Nilai tertinggi: 90.1
Nilai terendah: 55.3
PS C:\Users\ACER\Documents\Politeknik Negeri Malang\TUGAS KAMPUS\Prakt. Dasar Pemrograman\Praktiku
spro\Jobsheet 7> █
```

Pertanyaan 1

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!
2. Mengapa variabel **tertinggi** diinisialisasi 0 dan **terendah** diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?
3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!

```
if (nilai > tertinggi) {  
    tertinggi = nilai;  
}  
if (nilai < terendah) {  
    terendah = nilai;  
}
```

4. Modifikasi kode program sehingga terdapat perhitungan untuk menentukan berapa mahasiswa yang lulus dan yang tidak lulus berdasarkan batas kelulusan (nilai minimal 60). Tampilkan jumlah mahasiswa lulus dan tidak lulus setelah menampilkan nilai tertinggi dan terendah!
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 1"

Jawaban 1

1.



```
1  for (int i = 1; i <= 10; i++) {  
2      System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");  
3      nilai = sc.nextDouble();  
4  
5      if (nilai > tertinggi) {  
6          tertinggi = nilai;  
7      }  
8      if (nilai < terendah) {  
9          terendah = nilai;  
10     }  
11 }
```

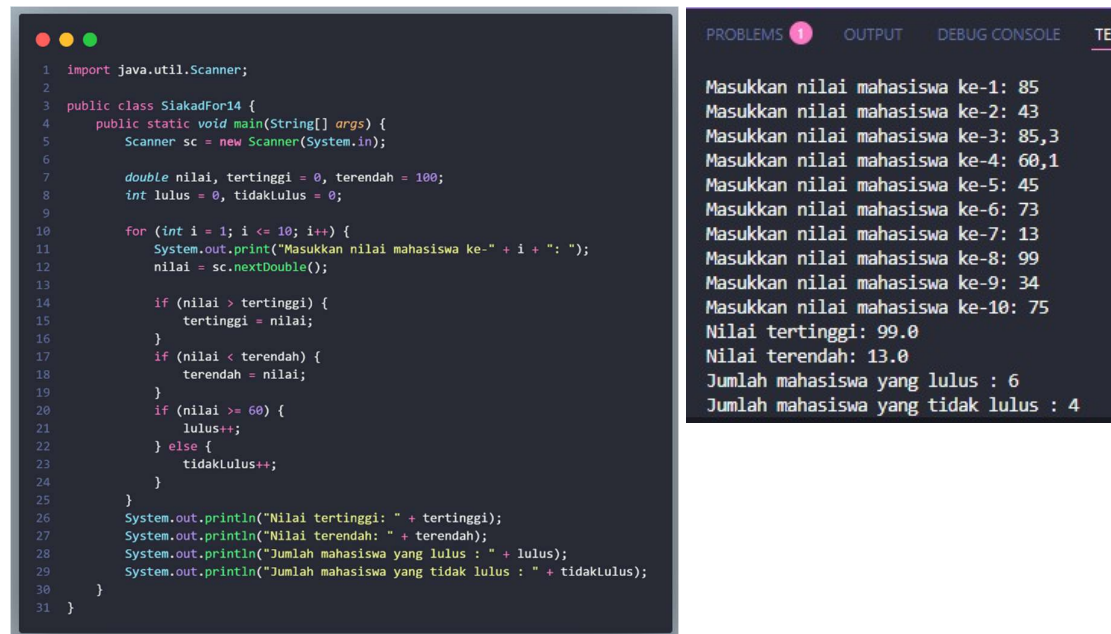
Komponen FOR

- `int i = 1` → Inisialisasi
 - `i <= 10` → Kondisi
 - `i++` → Update
 - `{ ... }` → Statement (yang di dalam kurung kurawal)
2. Variabel **tertinggi** diinisialisasi 0 dan variabel **terendah** diinisialisasi 100 karena kedua variabel tersebut akan digunakan sebagai pembanding dengan masukan sehingga dibutuhkan kemungkinan terburuknya yakni ketika nilai tertingginya 0 dan nilai terendahnya 100 supaya ketika dibandingkan, masukan akan dicek apakah ia lebih dari nilai tertinggi (0), kemudian apakah kurang dari nilai terendah (100). Demikian seterusnya hingga ditemukan nilai yang paling tepat. Apabila variabel

tertinggi diinisialisasi 100 dan variabel **terendah** diinisialisasi 0, maka berapapun masukan yang diterima, program akan selalu mengeluarkan 0 untuk nilai terendah dan 100 untuk nilai tertinggi karena ketika kedua nilai tersebut dibandingkan dengan masukan, tak akan ditemukan kondisi dimana masukan lebih dari 100 atau kurang dari 0 (kecuali masukan tidak sesuai, misalnya -1, 101).

3. Fungsi potongan kode tersebut adalah untuk mengecek apakah variabel **nilai** merupakan yang terendah atau tertinggi. Alurnya adalah sebagai berikut. Pertama, program akan mengecek apakah nilai variabel **nilai** lebih dari nilai variabel **tertinggi** saat ini (mula-mula 0). Jika ya, maka variabel **tertinggi** akan diisi dengan nilai dari variabel **nilai**. Kemudian, program akan mengecek apakah nilai variabel **nilai** kurang dari nilai variabel **terendah** saat ini (mula-mula 100). Jika ya, maka variabel **terendah** akan diisi dengan nilai dari variabel **nilai**. Begitu seterusnya untuk setiap masukan yang diterima sehingga akan ditemukan nilai terendah dan tertinggi dari suatu kumpulan bilangan yang dimasukkan.

4.



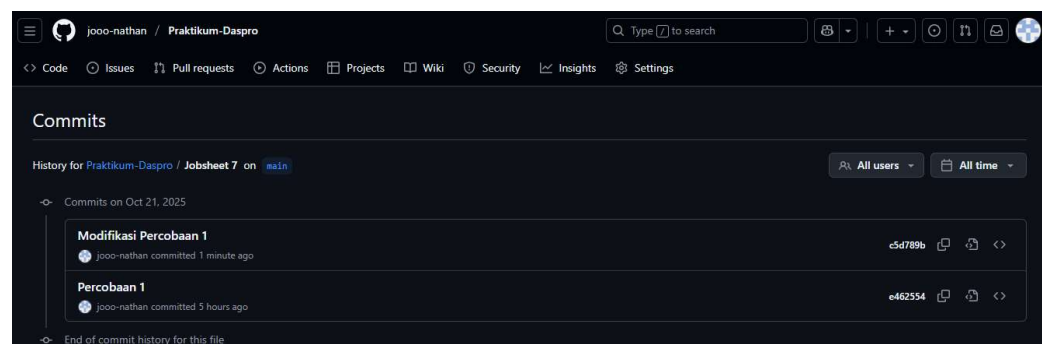
The screenshot shows a Java IDE with a code editor on the left and a console on the right. The code is a Java program named `SiakadFor14` that takes 10 student scores as input. It calculates the highest score (`tertinggi`), the lowest score (`terendah`), and the number of students who passed (`lulus`) and failed (`tidakLulus`). The output in the console shows the input scores, the calculated highest and lowest scores, and the counts of students who passed and failed.

```
1 import java.util.Scanner;
2
3 public class SiakadFor14 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         double nilai, tertinggi = 0, terendah = 100;
8         int lulus = 0, tidakLulus = 0;
9
10        for (int i = 1; i <= 10; i++) {
11            System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
12            nilai = sc.nextDouble();
13
14            if (nilai > tertinggi) {
15                tertinggi = nilai;
16            }
17            if (nilai < terendah) {
18                terendah = nilai;
19            }
20            if (nilai >= 60) {
21                lulus++;
22            } else {
23                tidakLulus++;
24            }
25        }
26        System.out.println("Nilai tertinggi: " + tertinggi);
27        System.out.println("Nilai terendah: " + terendah);
28        System.out.println("Jumlah mahasiswa yang lulus : " + lulus);
29        System.out.println("Jumlah mahasiswa yang tidak lulus : " + tidakLulus);
30    }
31 }
```

OUTPUT:

```
Masukkan nilai mahasiswa ke-1: 85
Masukkan nilai mahasiswa ke-2: 43
Masukkan nilai mahasiswa ke-3: 85,3
Masukkan nilai mahasiswa ke-4: 60,1
Masukkan nilai mahasiswa ke-5: 45
Masukkan nilai mahasiswa ke-6: 73
Masukkan nilai mahasiswa ke-7: 13
Masukkan nilai mahasiswa ke-8: 99
Masukkan nilai mahasiswa ke-9: 34
Masukkan nilai mahasiswa ke-10: 75
Nilai tertinggi: 99.0
Nilai terendah: 13.0
Jumlah mahasiswa yang lulus : 6
Jumlah mahasiswa yang tidak lulus : 4
```

5.



Percobaan 2

```
1 import java.util.Scanner;
2 public class SiakadWhile14 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int jml, nilai, i = 0;
6
7         System.out.print("Masukkan jumlah mahasiswa: ");
8         jml = sc.nextInt();
9
10        while (i < jml) {
11            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
12            nilai = sc.nextInt();
13
14            if (nilai < 0 || nilai > 100) {
15                System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid!");
16                continue;
17            }
18            if (nilai > 80 && nilai <= 100) {
19                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
20            }
21            else if (nilai > 73 && nilai <= 80) {
22                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
23            }
24            else if (nilai > 65 && nilai <= 73) {
25                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
26            }
27            else if (nilai > 60 && nilai <= 65) {
28                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
29            }
30            else if (nilai > 50 && nilai <= 60) {
31                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
32            }
33            else if (nilai > 39 && nilai <= 50) {
34                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
35            }
36            else {
37                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
38            }
39            i++;
40        }
41    }
42 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Masukkan nilai mahasiswa ke-2: 63
Masukkan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukkan nilai mahasiswa ke-3: 101
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukkan nilai mahasiswa ke-4: -15
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukkan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C
```

Pertanyaan 2

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut:

```
if (nilai < 0 || nilai > 100) {  
    System.out.println(x: "Nilai tidak valid. Masukkan lagi nilai yang valid!");  
    continue;  
}
```

 - a. `nilai < 0 || nilai > 100`
 - b. `continue`
2. Mengapa sintaks `i++` dituliskan di akhir perulangan `WHILE`? Apa yang terjadi jika posisinya dituliskan di awal perulangan `WHILE`?
3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan `WHILE` akan berjalan?
4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!
5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 2"**

Jawaban 2

- 1a Sintaks `nilai < 0 || nilai > 100` maksudnya adalah kondisi tersebut bernilai **true** jika variabel `nilai` bernilai kurang dari 0 atau variabel `nilai` bernilai lebih dari 100, yang mana artinya akan terpenuhi (bernilai **true**) jika salah satu atau kedua kondisi bernilai **true**. Kegunaannya adalah untuk menentukan apakah variabel `nilai` sebagai nilai mahasiswa ternyata valid atau tidak (valid jika berada dalam rentang 0 hingga 100).
 - 1b Sintaks `continue` maksudnya adalah melewati 1 iterasi saat ini dengan tidak melakukan seluruh arahan tepat setelah sintaks `continue` yang ada dalam perulangan melainkan langsung melanjutkan ke iterasi selanjutnya. Kegunaannya adalah untuk melewati suatu iterasi tertentu sesuai dengan kondisi yang diberikan, misalnya pada potongan kode tersebut adalah untuk melewati iterasi jika nilainya tidak valid.
2. Sintaks `i++` dituliskan di akhir perulangan `WHILE` karena sebelum melakukan *update* (melalui sintaks `i++`), program perlu melakukan pengecekan terlebih dahulu apakah nilai yang dimasukkan sudah valid atau belum. Jika sudah, program akan berjalan sebagaimana mestinya, tanpa peringatan bahwa nilai tidak valid, dan melakukan *update* di akhir program. Namun jika belum, program akan mengeluarkan peringatan bahwa nilai tidak valid, tidak akan melakukan *update*, dan mengulangi iterasi pada `i` saat itu hingga nilai yang dimasukkan valid.

Jika posisinya dituliskan di awal perulangan WHILE, ada 2 hal yang akan terjadi. Pertama, keluaran untuk $(i + 1)$ akan dimulai dari 2, bukan 1 sehingga menyebabkan kebingungan oleh pengguna (kecuali jika sintaks $(i + 1)$ juga disesuaikan dengan menghilangkan $+1$). Kedua, apabila nilai yang dimasukkan tidak valid, *update* akan terus berjalan tanpa mempedulikan apakah nilai tersebut valid atau tidak sehingga tidak ada iterasi tambahan untuk nilai yang tidak valid (jika nilai yang dimasukkan valid semua, tidak akan ada perbedaan antara $i++$ di awal atau akhir). Intinya, ada error, namun bila sintaks $(i + 1)$ diubah menjadi (i) dan masukan semuanya valid, tidak akan ditemukan error.

3. Dengan jumlah mahasiswa yang dimasukkan adalah sebanyak 19, apabila nilai yang dimasukkan valid semua, maka perulangan WHILE akan berjalan sebanyak 19 kali pula. Namun, apabila nilai yang dimasukkan ada yang tidak valid, maka perulangan WHILE akan berjalan sebanyak $19 +$ banyaknya nilai yang tidak valid.

4.

```

1 import java.util.Scanner;
2 public class SiakadWhile14 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int jml, nilai, i = 0;
6
7         System.out.print("Masukkan jumlah mahasiswa: ");
8         jml = sc.nextInt();
9
10        while (i < jml) {
11            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
12            nilai = sc.nextInt();
13
14            if (nilai < 0 || nilai > 100) {
15                System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid!");
16                continue;
17            }
18            if (nilai > 80 && nilai <= 100) {
19                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
20                System.out.println("Bagus, pertahankan nilainya");
21            }
22            else if (nilai > 73 && nilai <= 80) {
23                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
24            }
25            else if (nilai > 65 && nilai <= 73) {
26                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
27            }
28            else if (nilai > 60 && nilai <= 65) {
29                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
30            }
31            else if (nilai > 50 && nilai <= 60) {
32                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
33            }
34            else if (nilai > 39 && nilai <= 50) {
35                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
36            }
37            else {
38                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
39            }
40            i++;
41        }
42    }
43 }

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Masukkan jumlah mahasiswa: 3
Masukkan nilai mahasiswa ke-1: 90
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya
Masukkan nilai mahasiswa ke-2: 54
Nilai mahasiswa ke-2 adalah C
Masukkan nilai mahasiswa ke-3: 81
Nilai mahasiswa ke-3 adalah A
Bagus, pertahankan nilainya

```

5.

The screenshot shows a GitHub repository named 'Praktikum-Daspro' by user 'joou-nathan'. The 'Commits' tab is active, showing a history for 'Jobsheet 7 on main'. Two commits are listed:

- Modifikasi Percobaan 2**: Committed by joou-nathan 3 minutes ago. Commit hash: b284b05.
- Percobaan 2**: Committed by joou-nathan 10 minutes ago. Commit hash: cbffed9.

Percobaan 3

```
1  import java.util.Scanner;
2  public class KafeDowWhile14 {
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5
6          int kopi, teh, roti;
7          String namaPelanggan;
8          int hargaKopi = 12000, hargaTeh = 7000, hargaRoti = 20000;
9
10         do {
11             System.out.print("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
12             namaPelanggan = sc.nextLine();
13
14             if (namaPelanggan.equalsIgnoreCase("batal")) {
15                 System.out.println("Transaksi dibatalkan.");
16                 break;
17             }
18
19             System.out.print("Jumlah kopi: ");
20             kopi = sc.nextInt();
21
22             System.out.print("Jumlah teh: ");
23             teh = sc.nextInt();
24
25             System.out.print("Jumlah roti: ");
26             roti = sc.nextInt();
27
28             int totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
29             System.out.println("Total yang harus dibayar: Rp " + totalHarga);
30             sc.nextLine(); // Membersihkan newline dari buffer
31         } while (true);
32     }
33 }
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukkan nama pelanggan (ketik 'batal' untuk keluar): Rena
Jumlah kopi: 3
Jumlah teh: 0
Jumlah roti: 1
Total yang harus dibayar: Rp 56000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Yuni
Jumlah kopi: 1
Total yang harus dibayar: Rp 56000
Jumlah kopi: 1
Jumlah teh: 1
Jumlah roti: 1
Jumlah roti: 1
Total yang harus dibayar: Rp 56000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Yuni
Jumlah kopi: 1
Jumlah teh: 4
Jumlah roti: 2
Total yang harus dibayar: Rp 80000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Transaksi dibatalkan.

SOURCE CONTROL ...

CHANGES

Percobaan 3

Commit

Committing Changes... 1

KafeDowWhile14.java 1, A

Changes 2

hs_err_pid3028.log C:\... U

Tugas2.java 1, U

Pertanyaan 3

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!
3. Apa fungsi dari penggunaan nilai **true** pada kondisi DO-WHILE?
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?

Jawaban 3

1. Dengan memasukkan “batal” sebagai nama pelanggan, perulangan terhitung sudah dilakukan sebanyak 1x karena perulangan DO-WHILE akan melakukan perintah terlebih dahulu baru melakukan pengecekan yang artinya apapun kondisinya nanti, dengan melakukan perulangan DO-WHILE, program terhitung sudah melakukan 1x perulangan.
2. Kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut adalah ketika pengguna memasukkan kata “batal” pada masukan nama pengguna (termasuk jenis menambahkan pertanyaan karena perulangan berdasarkan jawaban pengguna).
3. Nilai **true** digunakan untuk mengulang program dengan terus-menerus tanpa berhenti (kecuali dihentikan oleh **break**) karena nilai **true** akan tetap selamanya **true**.
4. Karena komponen inisialisasi dan *update* memang sudah tidak diperlukan untuk kasus ini, karena kasus ini ingin agar pengguna bisa menjalankan program terus menerus selama nama yang dimasukkan bukan kata “batal”. Namun, sebenarnya, komponen-komponen ini masih ada, hanya saja fungsinya sudah digantikan oleh hal yang lain, yakni oleh aturan bahwa **true** akan selamanya **true**.

TUGAS

1. Seorang pengelola bioskop ingin membuat program untuk menghitung total penjualan tiket dalam satu hari. Tiket dijual dengan harga Rp 50.000 per tiket. Program harus menghitung total tiket yang terjual dan total harga penjualan tiket selama satu hari dengan ketentuan sebagai berikut:
 - Jika pelanggan membeli lebih dari 4 tiket, pelanggan mendapatkan diskon 10%.
 - Jika pelanggan membeli lebih dari 10 tiket, pelanggan mendapatkan diskon 15%.
 - Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input tersebut dan meminta input ulang.

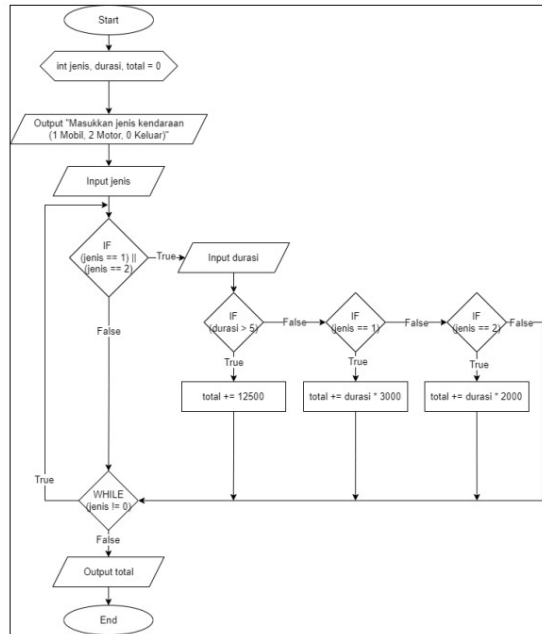
Commit dan push program Anda ke Github dengan pesan “Tugas 1”

Catatan: Perulangan dapat menggunakan for, while, atau do-while. Penambahan break atau continue jika diperlukan

Jawaban

```
1  import java.util.Scanner;
2
3  public class Tugas1 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          int hargaTiket = 50000, jumlahTiket, totalTiket = 0;
8          float hargaPerOrang, totalPendapatan = 0;
9          String namaPembeli;
10
11          System.out.print("\n----- PROGRAM PENGHITUNG PENJUALAN TIKET BIOSKOP ----- \n");
12
13          do {
14              System.out.println("\n=====");
15              System.out.print("Masukkan nama pembeli (ketik 'keluar' untuk berhenti): ");
16              namaPembeli = sc.nextLine();
17
18              if (namaPembeli.equalsIgnoreCase("keluar")) {
19                  System.out.println("\n=====");
20                  System.out.println("Terima kasih telah menggunakan layanan kami. |");
21                  System.out.println("===== \n");
22                  break;
23              }
24
25              System.out.print("\nMasukkan jumlah tiket yang dibeli: ");
26              jumlahTiket = sc.nextInt();
27
28              if (jumlahTiket < 0) {
29                  System.out.println("Jumlah tiket tidak valid. Silakan masukkan jumlah yang benar!");
30                  sc.nextLine(); // Membersihkan newline dari buffer
31                  continue;
32              }
33
34              totalTiket += jumlahTiket;
35
36              if (jumlahTiket > 10) {
37                  hargaPerOrang = (float)((jumlahTiket * hargaTiket) * 0.85); // Diskon 15%
38              }
39              else if (jumlahTiket > 4) {
40                  hargaPerOrang = (float)((jumlahTiket * hargaTiket) * 0.9); // Diskon 10%
41              }
42              else {
43                  hargaPerOrang = (float)(jumlahTiket * hargaTiket); // Harga normal
44              }
45
46              totalPendapatan += hargaPerOrang;
47
48              System.out.printf("Total harga yang harus dibayar oleh " + namaPembeli + " adalah: Rp %.2f\n", hargaPerOrang);
49              sc.nextLine(); // Membersihkan newline dari buffer
50
51          } while(true);
52
53          System.out.println("=====");
54          System.out.println("TOTAL TIKET TERJUAL\t: " + totalTiket);
55          System.out.printf("TOTAL PENDAPATAN\t: %.2f", totalPendapatan);
56          System.out.println("\n===== \n");
57      }
58  }
```

2. Perhatikan flowchart berikut!



Sebuah tempat parkir ingin membuat program untuk menghitung total pembayaran parkir dari beberapa kendaraan. Tarif parkir adalah Rp 3.000 per jam untuk mobil dan Rp 2.000 per jam untuk motor. Namun, jika durasi parkir lebih dari 5 jam, diberikan tarif tetap sebesar Rp 12.500 untuk semua kendaraan. Program akan terus meminta masukan selama input bukan 0. Implementasikan flowchart tersebut ke dalam bentuk kode program Java! **Commit dan push program Anda ke Github dengan pesan “Tugas 2”**.

Jawaban :

```
1 import java.util.Scanner;
2 public class Tugas2 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5
6         int jenis = 1, durasi, total = 0;
7
8         while (jenis != 0) {
9             System.out.println("\nJenis Kendaraan : \n1. Mobil\n2. Motor\n0. Keluar");
10            System.out.print("\nMasukkan jenis kendaraan (1/2/0): ");
11            jenis = sc.nextInt();
12
13            if (jenis == 1 || jenis == 2) {
14                System.out.print("Masukkan durasi parkir (dalam jam): ");
15                durasi = sc.nextInt();
16
17                if (durasi > 5) {
18                    total += 12500;
19                } else if (jenis == 1) { // Mobil
20                    total += durasi * 3000;
21                } else if (jenis == 2) { // Motor
22                    total += durasi * 2000;
23                }
24            }
25            else {
26                System.out.println("\nTerima kasih sudah menggunakan layanan kami.");
27                break;
28            }
29            System.out.println("Total biaya parkir: Rp " + total);
30        }
31    }
32 }
```