

PERCOBAAN 1

- 1. Jika pada perulangan for, inisialisasi $i=1$ diubah menjadi $i=0$, apa akibatnya? Mengapa bisa demikian?**

Jika inisialisasi i diubah menjadi 0, maka ketika nilai $n = 5$, jumlah bintang yang keluar menjadi 6. Hal ini karena looping berjalan 6 kali, dari 0 hingga sama dengan n (5). jadi loop berjalan dari 0 - 5

- 2. Jika pada perulangan for, kondisi $i \leq n$ diubah menjadi $i < n$, bagaimana bentuk outputnya jika input $n = 5$? Mengapa hasilnya berbeda?**

Jika kondisi diubah menjadi $i < n$ dengan input $n = 5$ dan $i = 1$, maka output yang dihasilkan adalah 4 bintang. Hal ini terjadi karena looping berjalan 4 kali, dari 1 hingga kurang dari n , yaitu 4. Jadi loop hanya berjalan dari 1 - 4.

- 3. Jika pada perulangan for, kondisi $i \leq n$ diubah menjadi $i > n$, apa akibatnya? Mengapa bisa demikian?**

Jika kondisi diubah menjadi $i > n$, maka tidak akan ada output yang keluar dengan nilai $n = 5$ dan $i = 1$. Karena 1 kurang dari 5, jadi tidak memenuhi kondisi $i > n$ dan update $i++$ tidak akan pernah dijalankan.

- 4. Jika pada perulangan for, kondisi step $i++$ diubah menjadi $i--$ apa akibatnya? Mengapa bisa demikian?**

Jika step/update diubah menjadi $i--$ dengan nilai $n = 5$ dan $i = 1$, maka kondisi $i \leq n$ tidak akan pernah False, sehingga menciptakan infinite loop yang tidak akan pernah selesai.

- 5. Jika pada perulangan for, step $i++$ diubah menjadi $i += 2$, bagaimana pola outputnya jika input $n = 6$? Apa yang menyebabkan perubahan tersebut?**

Jika step/update diubah menjadi $i += 2$ dengan nilai $n = 6$, $i = 1$, dan kondisi $i \leq n$, maka output yang dihasilkan adalah 3 bintang. Hal ini karena looping hanya terjadi 3 kali, ketika $i = 1$, $i = 3$, dan $i = 5$.

PERCOBAAN 2

- 1. Perhatikan perulangan luar. Jika pada sintaks for, inisialisasi iOuter=1 diubah menjadi iOuter=0, apa yang akibatnya? Mengapa bisa demikian?**

Jika inisialisasi iOuter diubah menjadi 0, maka perulangan luar akan mengulang sebanyak $n + 1$. Karena kondisi $iOuter \leq n$, mengakibatkan perulangan akan berjalan ketika $iOuter = 0$ hingga $iOuter = n$. Jika $n = 5$, maka perulangan akan berjalan ketika $iOuter = 0$ hingga $iOuter = 5$ (berjalan 6 kali).

- 2. Kembalikan program semula dimana inisialisasi iOuter=1. Kemudian perhatikan perulangan dalam. Jika pada sintaks for, inisialisasi i=1 diubah menjadi i=0, apa yang akibatnya? Mengapa bisa demikian?**

Jika inisialisasi i diubah menjadi 0, maka perulangan dalam akan mengulang sebanyak $n + 1$. Karena kondisi $i \leq n$, mengakibatkan perulangan akan berjalan ketika $i = 0$ hingga $i = n$. Jika $n = 5$, maka perulangan akan berjalan ketika $i = 0$ hingga $i = 5$ (berjalan 6 kali).

- 3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?**

Fungsi dari perulangan luar adalah mengatur jumlah baris sesuai dengan jumlah N, dan mengganti baris menggunakan (println) setelah perulangan dalam berjalan. Sementara perulangan dalam berfungsi untuk mencetak bintang sesuai dengan jumlah N dan mencetaknya hanya dalam 1 baris menggunakan (print) saat mencetak bintang.

- 4. Mengapa perlu ditambahkan sintaks System.out.println(); di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?**

Fungsi dari sintaks (println) yang ada di bawah perulangan dalam adalah mengganti baris setelah perulangan dalam dijalankan. Jika sintaks tersebut dihilangkan, maka perulangan dalam akan mencetak bintang hanya di 1 baris saja.

- 5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”**

PERCOBAAN 3

1. Perhatikan, apakah output yang dihasilkan dengan nilai $n = 5$ sesuai dengan tampilan berikut?

```
*  
**  
***  
****  
*****
```

Output yang dihasilkan belum sesuai karena hanya menampilkan bintang pada 1 baris.

2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan.

Bagian yang harus ditambahkan adalah enter. Tepat setelah inner loop harus ditambahkan (`println`) agar kursor pindah ke baris berikutnya. Sehingga setelah inner loop mencetak bintang sesuai dengan nilai i , iterasi outer loop berikutnya, inner loop akan mencetak bintang di baris bawahnya.

3. Jelaskan peran masing-masing variabel i dan j dalam program ini. Mengapa j di-set ulang ke 0 di awal setiap iterasi outer loop? Apa yang akan terjadi jika j tidak di-reset?

Variabel i berperan untuk mengatur berapa jumlah baris yang akan dicetak sesuai dengan kondisi $i \leq n$, ketika $n = 5$, maka outer loop akan berjalan 6 kali, ketika $i = 0$, hingga $i = 5$. Variabel i juga berperan untuk membatasi berapa jumlah bintang yang dicetak oleh inner loop sesuai dengan kondisi $j < i$, ketika $i = 0$, tidak akan ada bintang yang dicetak, ketika $i = 1$, 1 bintang dicetak, ketika $i = 5$, 5 bintang dicetak.

Variabel j berperan untuk mengatur jumlah bintang yang dicetak oleh inner loop, hal ini pula yang mengharuskan j di set ulang ke 0 di setiap awal iterasi outer loop. Sesuai dengan kondisi $j < i$, ketika $j = 0$ dan $i = 0$, maka inner loop tidak berjalan dan tidak ada bintang yang dicetak, di iterasi outer loop berikutnya, ketika $j = 0$ dan $i = 1$, maka inner loop berjalan 1 kali dan mencetak 1 bintang. Begitu pula ketika iterasi ke 5, ketika $j = 0$ dan $i = 5$, inner loop akan berjalan 5 kali dan mencetak 5 bintang. Hal ini lah yang mengharuskan j di set ulang ke 0 ketika awal outer loop, jika j tidak di set 0 kembali, maka setiap iterasi kecuali iterasi pertama, hanya mencetak 1 bintang. Karena ketika iterasi ke 2, $j = 0$ dan $i = 1$, bintang dicetak 1

kali, kemudian $j++$ dan $i++$, tanpa mereset j kembali ke 0. Sehingga iterasi berikutnya, $j = 1$ dan $i = 2$, hanya 1 bintang yang dicetak karena kondisi berjalannya inner loop adalah $j < i$. Begitu pula untuk iterasi berikutnya.

4. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”**

PERCOBAAN 4

- 1. Jelaskan apa yang terjadi pada variabel totalNilai di setiap iterasi outer loop dan mengapa inisialisasinya (total = 0) berada di dalam outer loop, bukan di luar.**

Di setiap iterasi outer loop, variabel totalNilai akan di reset menjadi 0, hal ini dilakukan agar total nilai setiap kelompok sebelum ditambah variabel nilai selalu dimulai dari 0, bukan dimulai dari jumlah total kelompok sebelumnya. Jika variabel totalNilai inisialisasi nya berada di luar outer loop, maka semua kelompok selain kelompok pertama, total nilai mereka tidak akan sesuai, karena tidak dimulai dari 0, namun dimulai dari totalNilai kelompok sebelumnya.

- 2. Modifikasi program diatas, sehingga dapat mencari kelompok dengan rata-rata nilai tertinggi dan tampilkan nomor kelompok tersebut.**
- 3. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 4”**