**JOBSHEET 2**

**PEMROGRAMAN MOBILE**



**SERLI PUTRI MAHARANI**

**TI-3E / 26**

**2141720023**

**PROGRAM STUDI D – IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

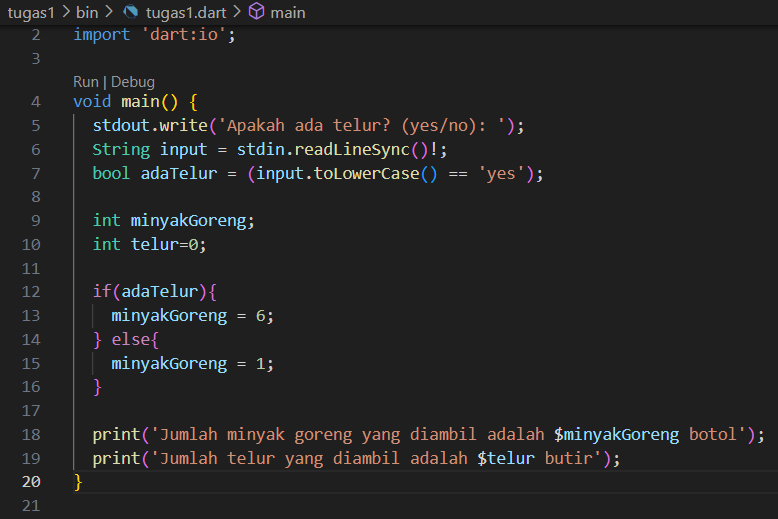
**2023**

**TUGAS 1**

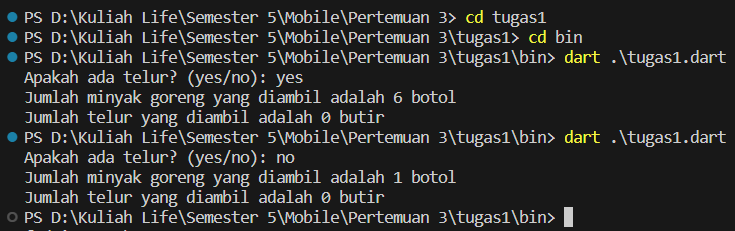
Berbeda dengan modul-modul sebelumnya yang menggunakan flowchart, bisakah kali ini Anda membuat program atau memahami maksud gambar berikut?



* **Source Code**



* **Output**

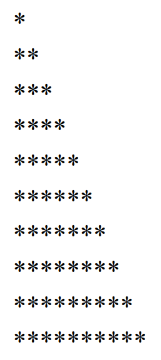


* **Penjelasan**

1. import 'package:tugas1/tugas1.dart' as tugas1; 🡪 Mengimpor file tugas1.dart dari paket tugas1 dengan menggunakan alias tugas1. Ini memungkinkan kita untuk menggunakan fungsi atau kelas yang didefinisikan dalam file tugas1.dart.
2. import 'dart:io'; 🡪 Mengimpor pustaka dart:io yang menyediakan fasilitas input/output dasar, seperti membaca input dari pengguna melalui keyboard dan menampilkan output ke konsol.
3. void main() 🡪 Ini adalah fungsi utama yang akan dieksekusi saat program dijalankan.
4. stdout.write('Apakah ada telur? (yes/no): '); 🡪 Baris ini menampilkan teks "Apakah ada telur? (yes/no): " ke konsol tanpa karakter baris baru.
5. String input = stdin.readLineSync()!; 🡪 Baris ini membaca input dari pengguna melalui keyboard menggunakan stdin.readLineSync(). Input yang diberikan oleh pengguna disimpan dalam variabel input sebagai string.
6. bool adaTelur = (input.toLowerCase() == 'yes'); 🡪 Baris ini memeriksa apakah input pengguna (dalam huruf kecil) sama dengan string 'yes'. Jika iya, variabel adaTelur akan bernilai true, jika tidak, variabel adaTelur akan bernilai false.
7. int minyakGoreng; dan int telur=0; 🡪 Mendeklarasikan dua variabel, minyakGoreng yang akan menyimpan jumlah minyak goreng yang diambil, dan telur yang akan menyimpan jumlah telur yang diambil. Variabel telur diinisialisasi dengan nilai 0.
8. if(adaTelur){ ... } else{ ... }🡪 Ini adalah struktur percabangan (if-else) yang memeriksa nilai dari variabel adaTelur. Jika adaTelur bernilai true, maka minyakGoreng akan diatur ke nilai 6. Jika adaTelur bernilai false, maka minyakGoreng akan diatur ke nilai 1.
9. print('Jumlah minyak goreng yang diambil adalah $minyakGoreng botol'): 🡪 Baris ini menampilkan output yang berisi jumlah minyak goreng yang diambil ke konsol. Variabel minyakGoreng digabungkan ke dalam string menggunakan sintaksis $minyakGoreng.
10. print('Jumlah telur yang diambil adalah $telur butir'); 🡪 Baris ini menampilkan output yang berisi jumlah telur yang diambil ke konsol. Variabel telur digabungkan ke dalam string menggunakan sintaksis $telur.

**TUGAS 2**

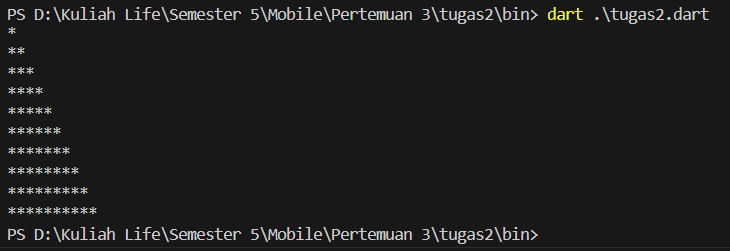
Kini saatnya menguji pehamanan Anda tentang materi for loops. Bisakah Anda membuat program Dart yang menampilkan output seperti berikut?



* **Source Code**



* **Output**

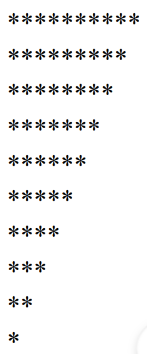


* **Penjelasan**

1. import 'package:tugas2/tugas2.dart' as tugas2; 🡪 Mengimpor file tugas2.dart dari paket tugas2 dengan menggunakan alias tugas2. Ini memungkinkan kita untuk menggunakan fungsi atau kelas yang didefinisikan dalam file tugas2.dart.
2. import 'dart:io'; 🡪 Mengimpor pustaka dart:io yang menyediakan fasilitas input/output dasar, seperti membaca input dari pengguna melalui keyboard dan menampilkan output ke konsol.
3. void main() 🡪 Ini adalah fungsi utama yang akan dieksekusi saat program dijalankan.
4. for(int i=1; i<=10; i++) 🡪 ini merupakan loop pertama yang menggunakan variabel I dan loop ini akan berjalan dari 1 hingga 10 dengan increment 1 setiap iterasinya
5. for(int j=1; j<=1; j++) 🡪 ini adalah loop kedua yang menggunakan variabel J. loop ini dimulai dari 1 hingga J kurang dari sama dengan variabel I dengan increment 1 setiap iterasi
6. stdout.write(‘\*’); 🡪 baris ini nantinya akan mencetak karakter ‘\*’ tanpa karakter baris baru dan akan mencetak sesuai dengan sejumlah nilai J pada setiap iterasi
7. print(‘’); 🡪 akan mencetak karakter baris baru ke output menggunakan print() hal ini bertujuan untuk mencetak baris baru setelah mencetak bintang untuk setiap iterasi loop pertama

**TUGAS 3**

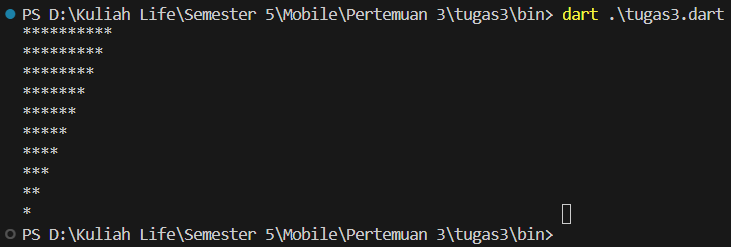
Melihat tugas sebelumnya, bisakah Anda membuat program Dart yang menampilkan output seperti di bawah ini menggunakan perulangan while atau do-while?



* **Source Code**



* **Output**



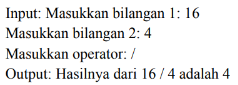
* **Penjelasan**

1. import 'package:tugas3/tugas3.dart' as tugas1; 🡪 Mengimpor file tugas3.dart dari paket tugas3 dengan menggunakan alias tugas3. Ini memungkinkan kita untuk menggunakan fungsi atau kelas yang didefinisikan dalam file tugas3.dart.
2. import 'dart:io'; 🡪 Mengimpor pustaka dart:io yang menyediakan fasilitas input/output dasar, seperti membaca input dari pengguna melalui keyboard dan menampilkan output ke konsol.
3. void main() 🡪 Ini adalah fungsi utama yang akan dieksekusi saat program dijalankan.
4. int i = 10; 🡪 menginisialisasi i = 10. Variabel i nantinya akan digunakan untuk mengontrol loop luar
5. do {…} while (I >= 1); 🡪 berikut merupakan loop luar yang menggunakan variabel i. loop ini nantinya akan menjalankan kode di dalamnya terlepas dari kondisi dan akan terus berjalan selama nilai i lebih besar dari atau sama dengan 1
6. int j = 1; 🡪 mendeklarasikan variabel J dan menginisialisasi dengan nilai 1. Variabel J akan digunakan untuk mengontrol loop dalam
7. do {…} while (j <= 1); 🡪 loop ini akan menjalankan kode di dalam, kondisi ini akan terus berjalan selama nilai j kurang dari atau sama dengan nilai i
8. stdout.write(‘\*’); 🡪 baris ini nantinya akan mencetak karakter ‘\*’ tanpa karakter baris baru dan akan mencetak sesuai loop dalam setiap iterasi
9. j++; 🡪 Baris ini akan menginkrementasi nilai j dengan 1 setiap iterasi loop dalam, sehingga nilai j akan bertambah satu setiap kali bintang dicetak.
10. print('');: Baris ini mencetak karakter baris baru ke output menggunakan print(). Ini bertujuan untuk mencetak baris baru setelah mencetak bintang untuk setiap iterasi loop luar.
11. i--;: Baris ini akan mengurangi nilai i dengan 1 setiap iterasi loop luar, sehingga nilai i akan berkurang satu setiap kali satu baris bintang selesai dicetak.

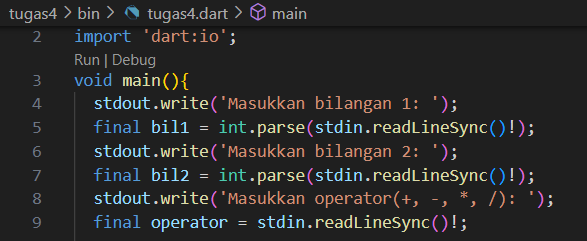
**TUGAS 4**

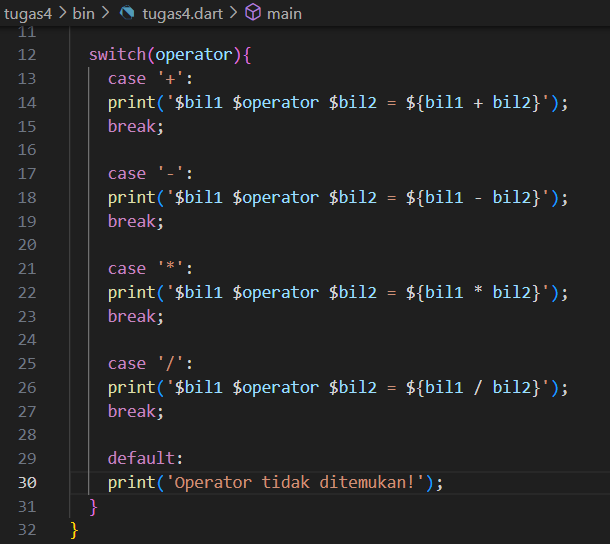
Jadikan contoh program diatas bisa menerima masukkan dari user.

Contoh:



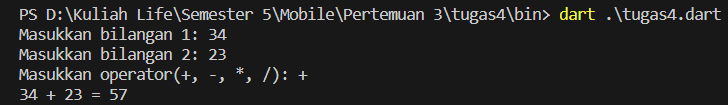
* **Source Code**



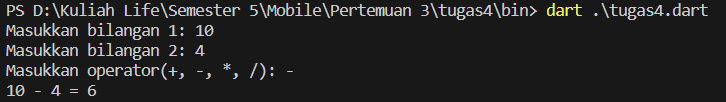


* **Output**

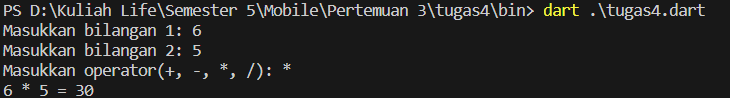
1. **+**



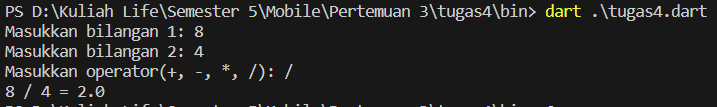
1. **–**



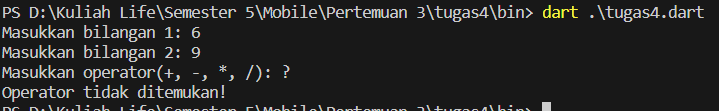
1. **\***



1. **/**



1. **Default**

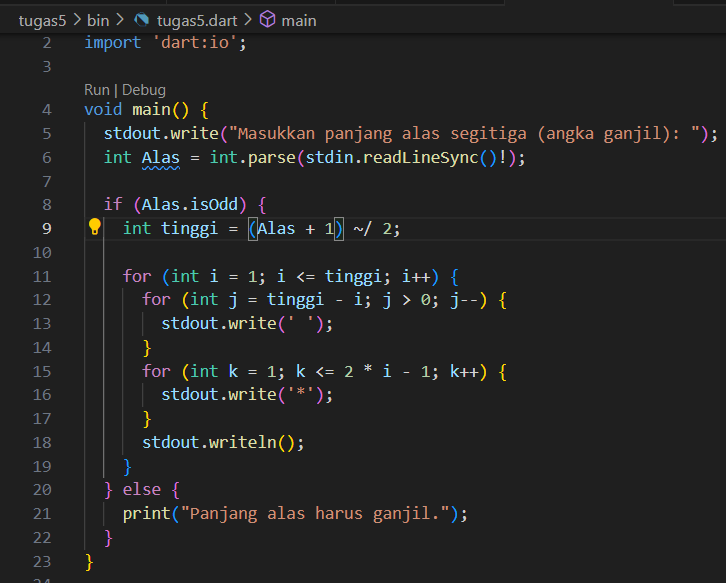


* **Penjelasan**

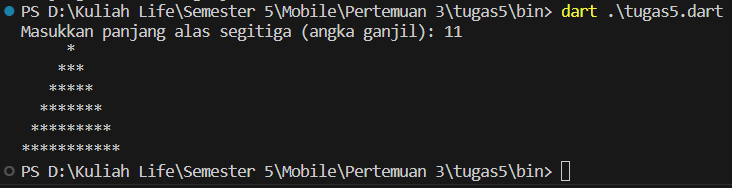
1. stdout.write('Masukkan bilangan 1: ');: Baris ini menampilkan teks "Masukkan bilangan 1: " ke konsol tanpa karakter baris baru.
2. final bil1 = int.parse(stdin.readLineSync()!);: Baris ini membaca input dari pengguna melalui keyboard menggunakan stdin.readLineSync() dan mengubahnya menjadi integer menggunakan int.parse(). Input yang diberikan oleh pengguna disimpan dalam variabel bil1.
3. stdout.write('Masukkan bilangan 2: ');: Baris ini menampilkan teks "Masukkan bilangan 2: " ke konsol tanpa karakter baris baru.
4. final bil2 = int.parse(stdin.readLineSync()!);: Baris ini membaca input dari pengguna melalui keyboard menggunakan stdin.readLineSync() dan mengubahnya menjadi integer menggunakan int.parse(). Input yang diberikan oleh pengguna disimpan dalam variabel bil2.
5. stdout.write('Masukkan operator(+, -, \*, /): ');: Baris ini menampilkan teks "Masukkan operator(+, -, \*, /): " ke konsol tanpa karakter baris baru.
6. final operator = stdin.readLineSync()!;: Baris ini membaca input dari pengguna melalui keyboard menggunakan stdin.readLineSync(). Input yang diberikan oleh pengguna disimpan dalam variabel operator sebagai string.
7. switch(operator){ ... }: Ini adalah struktur switch yang memeriksa nilai dari variabel operator. Struktur switch digunakan untuk membandingkan nilai operator dengan beberapa kasus dan menjalankan kode yang sesuai dengan kasus yang cocok.
8. case '+': ... break;: Ini adalah kasus pertama dalam struktur switch yang cocok dengan operator '+' (penjumlahan). Jika operator adalah '+', maka baris kode di dalamnya akan dieksekusi. Dalam hal ini, akan dicetak hasil penjumlahan bilangan bil1 dan bil2.
9. case '-': ... break;: Ini adalah kasus kedua dalam struktur switch yang cocok dengan operator '-' (pengurangan). Jika operator adalah '-', maka baris kode di dalamnya akan dieksekusi. Dalam hal ini, akan dicetak hasil pengurangan bilangan bil1 dan bil2.
10. case '\*': ... break;: Ini adalah kasus ketiga dalam struktur switch yang cocok dengan operator '' (perkalian). Jika operator adalah '', maka baris kode di dalamnya akan dieksekusi. Dalam hal ini, akan dicetak hasil perkalian bilangan bil1 dan bil2.
11. case '/': ... break;: Ini adalah kasus keempat dalam struktur switch yang cocok dengan operator '/' (pembagian). Jika operator adalah '/', maka baris kode di dalamnya akan dieksekusi. Dalam hal ini, akan dicetak hasil pembagian bilangan bil1 dan bil2.
12. default: ...: Ini adalah kasus default dalam struktur switch. Jika operator tidak cocok dengan kasus-kasus sebelumnya, maka baris kode di dalamnya akan dieksekusi. Dalam hal ini, akan dicetak teks "Operator tidak ditemukan!".

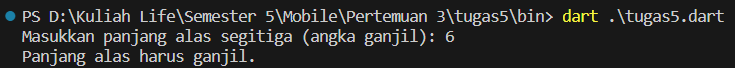
**TUGAS 5**

* **Source Code**

****

* **Output**

****

****

* **Penjelasan**

1. stdout.write("Masukkan panjang alas segitiga (angka ganjil): ");: Baris ini menampilkan teks "Masukkan panjang alas segitiga (angka ganjil): " ke konsol tanpa karakter baris baru
2. int Alas = int.parse(stdin.readLineSync()!);: Baris ini membaca input dari pengguna melalui keyboard menggunakan stdin.readLineSync() dan mengubahnya menjadi integer menggunakan int.parse(). Input yang diberikan oleh pengguna disimpan dalam variabel Alas sebagai panjang alas segitiga.
3. if (Alas.isOdd) { ... } else { ... }: Ini adalah pernyataan kondisional if-else. Jika panjang alas Alas adalah bilangan ganjil (memiliki sisa pembagian dengan 2), maka blok kode di dalam if akan dieksekusi. Jika panjang alas Alas bukan bilangan ganjil, maka blok kode di dalam else akan dieksekusi.
4. int tinggi = (Alas + 1) ~/ 2;: Baris ini menghitung tinggi segitiga berdasarkan panjang alas Alas. Tinggi segitiga dihitung dengan menjumlahkan 1 pada Alas dan membaginya dengan 2 menggunakan operator ~/ untuk melakukan pembagian bilangan bulat (mengabaikan angka desimal).
5. for (int i = 1; i <= tinggi; i++) { ... }: Ini adalah loop untuk mengontrol baris-baris segitiga. Loop ini akan berjalan dari i=1 hingga i kurang dari atau sama dengan tinggi dengan increment 1 setiap iterasi.
6. for (int j = tinggi - i; j > 0; j--) { ... }: Ini adalah loop untuk mencetak spasi sebelum karakter '\*' pada setiap baris segitiga. Loop ini akan berjalan dari j yang diinisialisasi dengan tinggi - i hingga j lebih besar dari 0 dengan decrement 1 setiap iterasi. Hal ini bertujuan untuk mencetak jumlah spasi yang berkurang setiap barisnya.
7. stdout.write(' ');: Baris ini mencetak karakter spasi ke output menggunakan stdout.write(). Ini akan mencetak spasi pada setiap iterasi loop dalam untuk mengatur posisi karakter '\*' pada baris segitiga.
8. for (int k = 1; k <= 2 \* i - 1; k++) { ... }: Ini adalah loop untuk mencetak karakter '' pada setiap baris segitiga. Loop ini akan berjalan dari k=1 hingga k kurang dari atau sama dengan 2 \* i - 1 dengan increment 1 setiap iterasi. Hal ini bertujuan untuk mencetak jumlah karakter '' yang bertambah setiap barisnya.
9. stdout.write('\*');: Baris ini mencetak karakter '' ke output menggunakan stdout.write(). Ini akan mencetak karakter '' pada setiap iterasi loop dalam untuk membentuk pola segitiga.
10. stdout.writeln();: Baris ini mencetak karakter baris baru ke output menggunakan stdout.writeln(). Ini bertujuan untuk mencetak baris baru setelah mencetak karakter '\*' pada setiap baris segitiga.
11. print("Panjang alas harus ganjil.");: Baris ini mencetak teks "Panjang alas harus ganjil." ke konsol jika panjang alas Alas tidak ganjil (bukan bilangan ganjil). Ini memberikan pesan kesalahan kepada pengguna bahwa panjang alas yang dimasukkan haruslah bilangan ganjil.