

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

```
tugas1 > bin > tugas1.dart > main
2  import 'dart:io';
3
4  void main() {
5      stdout.write('Apakah ada telur? (yes/no): ');
6      String input = stdin.readLineSync();
7      bool adaTelur = (input.toLowerCase() == 'yes');
8
9      int minyakGoreng;
10     int telur=0;
11
12     if(adaTelur){
13         minyakGoreng = 6;
14     } else{
15         minyakGoreng = 1;
16     }
17
18     print('Jumlah minyak goreng yang diambil adalah $minyakGoreng botol');
19     print('Jumlah telur yang diambil adalah $telur butir');
20 }
21
```

→ Output

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3> cd tugas1
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas1> cd bin
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas1\bin> dart .\tugas1.dart
Apakah ada telur? (yes/no): yes
Jumlah minyak goreng yang diambil adalah 6 botol
Jumlah telur yang diambil adalah 0 butir
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas1\bin> dart .\tugas1.dart
Apakah ada telur? (yes/no): no
Jumlah minyak goreng yang diambil adalah 1 botol
Jumlah telur yang diambil adalah 0 butir
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas1\bin> █
```

→ 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 pemahaman Anda tentang materi for loops. Bisakah Anda membuat program Dart yang menampilkan output seperti berikut?

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

➔ Source Code

```
tugas2 > bin >  tugas2.dart >  main
1  import package:tugas2/tugas2.dart
2  import 'dart:io';
3
4  Run | Debug
5  void main(){
6      for(int i=1; i<=10; i++){
7          for (int j=1; j<=i; j++){
8              stdout.write('*');
9          }
10         print('');
11     }
12 }
```

→ Output

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas2\bin> dart .\tugas2.dart  
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****
```

PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas2\bin>

→ 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('\n');` → 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

```
tugas3 > bin >  tugas3.dart >  main
1  import package:tugas3/tugas3.dart as tugas3;
2  import 'dart:io';
3
   Run | Debug
4  void main() {
5      int i = 10;
6      do {
7          int j = 1;
8          do {
9              stdout.write('*');
10             j++;
11         } while (j <= i);
12         print('');
13         i--;
14     } while (i >= 1);
15 }
```

→ Output

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas3\bin> dart .\tugas3.dart  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas3\bin>

→ 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("\n");` 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:

Input: Masukkan bilangan 1: 16

Masukkan bilangan 2: 4

Masukkan operator: /

Output: Hasilnya dari 16 / 4 adalah 4

→ Source Code

```
tugas4 > bin > tugas4.dart > main
2  import 'dart:io';
   Run | Debug
3  void main(){
4      stdout.write('Masukkan bilangan 1: ');
5      final bil1 = int.parse(stdin.readLineSync());
6      stdout.write('Masukkan bilangan 2: ');
7      final bil2 = int.parse(stdin.readLineSync());
8      stdout.write('Masukkan operator(+, -, *, /): ');
9      final operator = stdin.readLineSync();
```

```
tugas4 > bin > tugas4.dart > main
11
12     switch(operator){
13         case '+':
14             print('$bil1 $operator $bil2 = ${bil1 + bil2}');
15             break;
16
17         case '-':
18             print('$bil1 $operator $bil2 = ${bil1 - bil2}');
19             break;
20
21         case '*':
22             print('$bil1 $operator $bil2 = ${bil1 * bil2}');
23             break;
24
25         case '/':
26             print('$bil1 $operator $bil2 = ${bil1 / bil2}');
27             break;
28
29         default:
30             print('Operator tidak ditemukan!');
31     }
32 }
```

➔ Output

a. +

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas4\bin> dart .\tugas4.dart
Masukkan bilangan 1: 34
Masukkan bilangan 2: 23
Masukkan operator(+, -, *, /): +
34 + 23 = 57
```

b. -

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas4\bin> dart .\tugas4.dart
Masukkan bilangan 1: 10
Masukkan bilangan 2: 4
Masukkan operator(+, -, *, /): -
10 - 4 = 6
```

c. *

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas4\bin> dart .\tugas4.dart
Masukkan bilangan 1: 6
Masukkan bilangan 2: 5
Masukkan operator(+, -, *, /): *
6 * 5 = 30
```

d. /

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas4\bin> dart .\tugas4.dart
Masukkan bilangan 1: 8
Masukkan bilangan 2: 4
Masukkan operator(+, -, *, /): /
8 / 4 = 2.0
```


e. Default

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas4\bin> dart .\tugas4.dart
Masukkan bilangan 1: 6
Masukkan bilangan 2: 9
Masukkan operator(+, -, *, /): ?
Operator tidak ditemukan!
```

→ 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

```
tugas5 > bin > tugas5.dart > main
2  import 'dart:io';
3
4  void main() {
5      stdout.write("Masukkan panjang alas segitiga (angka ganjil): ");
6      int Alas = int.parse(stdin.readLineSync());
7
8      if (Alas.isOdd) {
9          int tinggi = (Alas + 1) ~/ 2;
10
11         for (int i = 1; i <= tinggi; i++) {
12             for (int j = tinggi - i; j > 0; j--) {
13                 stdout.write(' ');
14             }
15             for (int k = 1; k <= 2 * i - 1; k++) {
16                 stdout.write('*');
17             }
18             stdout.writeln();
19         }
20     } else {
21         print("Panjang alas harus ganjil.");
22     }
23 }
```

→ Output

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas5\bin> dart .\tugas5.dart
Masukkan panjang alas segitiga (angka ganjil): 11
      *
     ***
    *****
   *********
  ***********
 *****
*
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas5\bin>
```

```
PS D:\Kuliah Life\Semester 5\Mobile\Pertemuan 3\tugas5\bin> dart .\tugas5.dart
Masukkan panjang alas segitiga (angka ganjil): 6
Panjang alas harus ganjil.
```

→ 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.