

Nama : Alfina Salsabilla  
NIM : 2141720044  
No. Absen : 06  
Kelas : TI-3D

**LAPORAN PRAKTIKUM PEMOGRAMAN MOBILE**

**Minggu Ke - 4**



**JURUSAN TEKNOLOGI INFORMASI  
PROGRAM STUDI D4 TEKNIK INFORMATIKA  
POLITEKNIK NEGERI MALANG  
TAHUN 2023**

## 1) Tugas 1 :

- **Kode Program :**

```
tugas1.dart > mainCode
1 import 'dart:io';
2
3 int mainCode() {
4     // declaration variable
5     int oilThatShouldBuy = 0;
6     // ---[ Tulis kodemu setelah baris ini ]---
7
8     print("Apakah Anda perlu membeli minyak goreng? (ya/tidak)");
9     String buyOil = stdin.readLineSync()!;
10
11     if (buyOil.toLowerCase() == "ya") {
12         oilThatShouldBuy += 1;
13     }
14
15     print("Apakah Anda perlu membeli telur? (ya/tidak)");
16     String buyEggs = stdin.readLineSync()!;
17
18     if (buyEggs.toLowerCase() == "ya") {
19         oilThatShouldBuy += 5;
20     }
21
22     // ---[ Jangan menulis atau mengubah kode di bawah ini ]---
23     return oilThatShouldBuy;
24 }
25
```

- **Penjelasan :**

- 1) import 'dart:io';: Ini adalah pernyataan impor yang digunakan untuk mengimpor pustaka 'dart:io' yang memungkinkan kita untuk berinteraksi dengan masukan (input) dari pengguna melalui konsol dan mencetak keluaran (output) ke konsol.
- 2) int mainCode() { ... }: Ini adalah fungsi utama yang akan dijalankan. Fungsi ini mengembalikan nilai bertipe integer, yang akan digunakan untuk menentukan berapa banyak botol minyak goreng yang harus dibeli.
- 3) int oilThatShouldBuy = 0;: Ini adalah deklarasi variabel oilThatShouldBuy yang akan digunakan untuk menghitung jumlah botol minyak goreng yang harus dibeli.
- 4) Pertanyaan kepada pengguna dimulai dengan perintah print dan pengguna diminta untuk menjawab dengan "ya" atau "tidak". Jawaban dari pengguna akan disimpan dalam variabel yang sesuai.

- 5) Misalnya, pengguna ditanya apakah perlu membeli minyak goreng dengan perintah `print("Apakah Anda perlu membeli minyak goreng? (ya/tidak)");`, dan jawaban pengguna disimpan dalam variabel `buyOil`.
- 6) Setelah itu, jawaban pengguna diubah menjadi huruf kecil menggunakan `.toLowerCase()` agar tidak sensitif terhadap huruf besar/kecil.
- 7) Jika jawaban pengguna adalah "ya", maka nilai `oilThatShouldBuy` akan ditambahkan dengan jumlah tertentu. Dalam kasus ini, jika pengguna menjawab "ya", maka `oilThatShouldBuy` akan ditambah 1.
- 8) Hal yang sama berlaku untuk pertanyaan apakah perlu membeli telur. Jika pengguna menjawab "ya", maka `oilThatShouldBuy` akan ditambah 5.
- 9) Setelah semua pertanyaan telah dijawab dan nilai `oilThatShouldBuy` telah dihitung berdasarkan jawaban pengguna, nilai ini akan dikembalikan oleh fungsi `mainCode()`.
- 10) Pada fungsi `void main() { ... }`, hasil dari `mainCode()` akan disimpan dalam variabel `result` dan kemudian dicetak ke layar dengan pesan yang menyatakan jumlah botol minyak goreng yang harus dibeli.

- **Output :**

```
Apakah Anda perlu membeli minyak goreng? (ya/tidak)
ya
Apakah Anda perlu membeli telur? (ya/tidak)
ya
Jumlah minyak goreng yang harus Anda beli: 6 botol
```

## 2) Tugas 2 :

- **Kode Program :**

```
import 'dart:io';

Run | Debug
void main() {
  var a = '';
  // var b = 5;

  for (var i = 0; i < 10; i++) {
    for (var j = 0; j <= i; j++) {
      a += '*';
    }
    a += '\n';
  }
  print('$a');
}
```

- **Penjelasan :**

Kode program di atas adalah program Dart yang menghasilkan pola segitiga dengan menggunakan karakter '\*' di konsol. Berikut adalah penjelasan langkah demi langkah:

- a) `var a = "";` Ini adalah deklarasi variabel `a` yang digunakan untuk menyimpan pola segitiga yang akan dicetak ke konsol. Pada awalnya, variabel ini diinisialisasi sebagai string kosong. Kita memiliki dua perulangan bersarang (`for`) di dalam kode:
- b) Perulangan pertama (`for (var i = 0; i < 10; i++)`): Ini adalah perulangan luar yang mengontrol jumlah baris dalam segitiga. Perulangan ini akan berjalan dari 0 hingga 9, sehingga ada total 10 baris dalam segitiga.
- c) Perulangan kedua (`for (var j = 0; j <= i; j++)`): Ini adalah perulangan dalam yang mengontrol jumlah karakter `"` yang akan dicetak pada setiap baris. Variabel `j` akan berjalan dari 0 hingga nilai `i`, sehingga setiap baris akan memiliki jumlah `"` yang berbeda.
- d) Di dalam perulangan dalam, karakter `"` ditambahkan ke dalam variabel `a` pada setiap iterasi. Oleh karena itu, setiap baris akan memiliki sejumlah karakter `"` yang sesuai dengan nilai `j`.
- e) Setelah selesai mencetak karakter `*` untuk satu baris, kita menambahkan karakter newline `\n` ke variabel `a`. Ini akan membuat baris baru untuk baris berikutnya dalam segitiga.
- f) Setelah perulangan dalam selesai untuk satu baris, kita akan melanjutkan ke baris berikutnya dalam perulangan luar.
- g) Setelah seluruh segitiga telah dibangun dalam variabel `a`, kita menggunakan pernyataan `print('$a');` untuk mencetak isi variabel `a` ke konsol. Menggunakan `$a` dalam string interpolasi akan mencetak isi variabel `a` dengan format segitiga yang sesuai. Hasil akhir dari kode ini adalah mencetak segitiga yang terdiri dari karakter `"` dengan 10 baris pertama, dan setiap baris memiliki jumlah karakter `"` yang sesuai dengan nomor baris (baris ke-1 memiliki 1 `"`, baris ke-2 memiliki 2 `"`, dan seterusnya).

- **Output :**

```
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas2.dart
*
**
***
****
*****
*****
*****
*****
*****
*****
```

### 3) Tugas 3 :

- **Kode Program :**

```
import 'dart:io';

Run | Debug
void main() {
  var a = '';
  // var b = 5;

  for (var i = 10; i > 0; i--) {
    for (var j = 0; j < i; j++) {
      a += '*';
    }
    a += '\n';
  }
  print('$a');
}
```

- **Penjelasan :**

Kode program di atas mirip dengan kode sebelumnya, namun kali ini menghasilkan pola segitiga terbalik dengan karakter '\*' di konsol. Mari kita jelaskan langkah-langkahnya:

- var a = "": Ini adalah deklarasi variabel a yang akan digunakan untuk menyimpan pola segitiga terbalik yang akan dicetak ke konsol. Pada awalnya, variabel ini diinisialisasi sebagai string kosong. Kita memiliki dua perulangan bersarang (for) di dalam kode:
- Perulangan pertama (for (var i = 10; i > 0; i--)): Ini adalah perulangan luar yang mengontrol jumlah baris dalam segitiga terbalik. Perulangan ini akan berjalan dari 10 hingga 1, sehingga ada total 10 baris dalam segitiga terbalik.

- c) Perulangan kedua (for (var j = 0; j < i; j++)): Ini adalah perulangan dalam yang mengontrol jumlah karakter " yang akan dicetak pada setiap baris. Variabel j akan berjalan dari 0 hingga nilai i - 1, sehingga setiap baris akan memiliki jumlah karakter " yang berbeda.
- d) Di dalam perulangan dalam, karakter " ditambahkan ke dalam variabel a pada setiap iterasi. Oleh karena itu, setiap baris akan memiliki sejumlah karakter " yang sesuai dengan nilai j.
- e) Setelah selesai mencetak karakter '\*' untuk satu baris, kita menambahkan karakter newline '\n' ke variabel a. Ini akan membuat baris baru untuk baris berikutnya dalam segitiga terbalik.
- f) Setelah perulangan dalam selesai untuk satu baris, kita akan melanjutkan ke baris berikutnya dalam perulangan luar.
- g) Setelah seluruh segitiga terbalik telah dibangun dalam variabel a, kita menggunakan pernyataan print('\$a'); untuk mencetak isi variabel a ke konsol. Menggunakan \$a dalam string interpolasi akan mencetak isi variabel a dengan format segitiga terbalik yang sesuai.
- h) Hasil akhir dari kode ini adalah mencetak segitiga terbalik yang terdiri dari karakter " dengan 10 baris pertama, dan setiap baris memiliki jumlah karakter " yang berkurang seiring dengan penurunan nomor baris (baris pertama memiliki 10 ", baris kedua memiliki 9 ", dan seterusnya).

- **Output :**

```
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas3.dart
*****
*****
*****
*****
*****
****
***
**
*
```

#### 4) Tugas 4 :

- **Kode Program :**

```
import 'dart:io';

Run | Debug
void main() {

  stdout.write("Masukkan Bilangan 1: ");
  int firstNumber = int.parse(stdin.readLineSync()!);
  stdout.write("Masukkan Bilangan 2: ");
  int secondNumber = int.parse(stdin.readLineSync()!);
  stdout.write("Masukkan Operator: ");
  String operator = stdin.readLineSync()!;

  switch(operator){
    case "+":
      //print('$firstNumber $operator $secondNumber = ${firstNumber + secondNumber}');
      print('Hasilnya dari $firstNumber $operator $secondNumber adalah ${firstNumber + secondNumber}');
      break;
    case "-":
      //print('$firstNumber $operator $secondNumber = ${firstNumber - secondNumber}');
      print('Hasilnya dari $firstNumber $operator $secondNumber adalah ${firstNumber - secondNumber}');
      break;
    case "*":
      //print('$firstNumber $operator $secondNumber = ${firstNumber * secondNumber}');
      print('Hasilnya dari $firstNumber $operator $secondNumber adalah ${firstNumber * secondNumber}');
      break;
    case "/":
      //print('$firstNumber $operator $secondNumber = ${firstNumber / secondNumber}');
      print('Hasilnya dari $firstNumber $operator $secondNumber adalah ${firstNumber / secondNumber}');
      break;
    default:
      print("Operasi tidak ditemukan");
      break;
  }
}
```

- **Penjelasan :**

- a) `stdout.write("Masukkan Bilangan 1: ");`: Ini adalah pernyataan untuk mencetak teks "Masukkan Bilangan 1: " ke konsol tanpa baris baru, sehingga pengguna dapat memasukkan bilangan pertama setelah teks ini.
- b) `int firstNumber = int.parse(stdin.readLineSync()!);`: Ini adalah pernyataan untuk membaca input dari pengguna dengan `stdin.readLineSync()` dan kemudian mengonversinya menjadi bilangan bulat (`int`) dengan `int.parse()`. Input ini kemudian disimpan dalam variabel `firstNumber`.

- c) Langkah yang sama berlaku untuk mengambil input pengguna untuk bilangan kedua: `stdout.write("Masukkan Bilangan 2: ");` dan `int secondNumber = int.parse(stdin.readLineSync()!);`.
- d) `stdout.write("Masukkan Operator: ");`; Ini adalah pernyataan untuk meminta pengguna memasukkan operator matematika (misalnya, +, -, \*, /).
- e) `String operator = stdin.readLineSync()!;` Ini adalah pernyataan untuk membaca input operator matematika dari pengguna dan menyimpannya dalam variabel operator sebagai string.
- f) Program menggunakan pernyataan `switch` untuk mengevaluasi nilai dari variabel operator dan melakukan operasi matematika yang sesuai berdasarkan operator yang dimasukkan oleh pengguna.
- g) Setiap kasus (case) dalam `switch` mengatasi operator matematika yang mungkin dimasukkan oleh pengguna. Misalnya, jika operator adalah "+", program akan menambahkan `firstNumber` dan `secondNumber`, dan mencetak hasilnya ke layar.
- h) Pada setiap kasus, program mencetak hasil operasi matematika ke layar dengan format "Hasilnya dari [bilangan pertama] [operator] [bilangan kedua] adalah [hasil]".
- i) Jika pengguna memasukkan operator yang tidak dikenali, program akan mencetak pesan "Operasi tidak ditemukan."
- j) Program selesai dan keluar setelah mencetak hasil atau pesan kesalahan sesuai dengan operator yang dimasukkan oleh pengguna.

- **Output :**

```
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas4.dart
Masukkan Bilangan 1: 16
Masukkan Bilangan 2: 4
Masukkan Operator: /
Hasilnya dari 16 / 4 adalah 4.0
```



## 5) Tugas 5 :

- **Kode Program :**

```
void main() {
    stdout.write("Masukkan tinggi segitiga (bilangan ganjil): ");
    int tinggi = int.parse(stdin.readLineSync()!);

    if (tinggi.isOdd) {
        for (int i = 0; i < tinggi; i++) {
            for (int j = 0; j < tinggi - i - 1; j++) {
                stdout.write(" ");
            }
            for (int k = 0; k < 2 * i + 1; k++) {
                stdout.write("*");
            }
            print(""); // Pindah ke baris berikutnya
        }
    } else {
        print("Tinggi segitiga harus bilangan ganjil.");
    }
}
```

- **Penjelasan :**

- a) `stdout.write("Masukkan tinggi segitiga (bilangan ganjil): ");`: Pernyataan ini digunakan untuk mencetak pesan ke layar dan meminta pengguna untuk memasukkan tinggi segitiga. Pesan ini tidak mengikuti baris baru sehingga pengguna dapat memasukkan input di samping pesan.
- b) `int tinggi = int.parse(stdin.readLineSync()!);`: Ini adalah pernyataan untuk membaca input dari pengguna dengan `stdin.readLineSync()` dan mengonversinya menjadi bilangan bulat (`int`) dengan `int.parse()`. Input ini kemudian disimpan dalam variabel `tinggi`.
- c) `if (tinggi.isOdd) { ... }`: Ini adalah kondisi `if` yang memeriksa apakah nilai `tinggi` adalah bilangan ganjil. Jika benar (ganjil), maka kode di dalam blok `if` akan dijalankan. Jika salah (genap), program akan mencetak pesan bahwa tinggi segitiga harus ganjil.
- d) Dalam perulangan pertama (`for (int i = 0; i < tinggi; i++)`), program mengatur tingkat baris dalam segitiga. Loop ini akan berjalan sebanyak `tinggi` kali, dan pada setiap iterasi, program akan mencetak baris dari segitiga. Di dalam perulangan pertama, ada dua perulangan bersarang:
- e) Perulangan kedua (`for (int j = 0; j < tinggi - i - 1; j++)`): Ini digunakan untuk mencetak spasi sebelum karakter `"` pada setiap baris. Jumlah

spasi yang dicetak bergantung pada nilai j dan i, sehingga karakter " akan ditampilkan dengan benar dalam pola segitiga.

- f) Perulangan ketiga (for (int k = 0; k < 2 \* i + 1; k++)): Ini digunakan untuk mencetak karakter " pada setiap baris. Jumlah karakter " yang dicetak bergantung pada nilai k dan i, sehingga membentuk pola segitiga yang sesuai.
- g) Setelah mencetak spasi dan karakter '\*' pada baris tertentu, program mencetak baris kosong (dengan print("")) untuk pindah ke baris berikutnya dalam segitiga.
- h) Jika tinggi segitiga bukan bilangan ganjil (genap), program akan mencetak pesan "Tinggi segitiga harus bilangan ganjil."
- i) Dengan cara ini, program ini akan mencetak pola segitiga dengan karakter '\*' sesuai dengan tinggi segitiga yang dimasukkan oleh pengguna, asalkan tinggi segitiga adalah bilangan ganjil. Jika tidak, program akan memberikan pesan kesalahan.

- **Output :**

```
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas5.dart
Masukkan tinggi segitiga (bilangan ganjil): 3
*
***
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas5.dart
Masukkan tinggi segitiga (bilangan ganjil): 5
*
***
*****
*****
*****
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> dart --enable-experiment=non-nullable tugas5.dart
Masukkan tinggi segitiga (bilangan ganjil): 7
*
***
*****
*****
*****
*****
*****
*****
PS D:\SEMESTER 5\MOBILE\Pertemuan 4> |
```