LAPORAN PRAKTIKUM
Modul 3
“Pengenalan Dart”

Disusun Oleh:
Muhammad Ralfi - 2211104054
Kelas SE-06-2

Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO

2024

1. **Tujuan**
   a. Mahasiswa mengetahui dasar-dasar bahasa pemrograman dart.
   b. Mahasiswa mampu memahami konsep layout pada Flutter.
   c. Mahasiswa dapat mengimplementasikan desain user interface pada Flutter.
2. **Landasan Teori**

Dart merupakan bahasa pemrograman open source yang bersifat umum. Bahasa ini dikembangkan oleh Google dengan tujuan untuk membangun aplikasi yang dapat dijalankan di berbagai platform, termasuk mobile, desktop, dan web. Dart pertama kali diperkenalkan pada konferensi GOTO tahun 2011. Proyek ini dipimpin oleh Lars Bak dan Kasper Lund dari Google, dan versi 1.0 dari Dart diluncurkan pada 14 November 2013. Pada Agustus 2018, Dart 2.0 diumumkan dengan sejumlah perubahan dalam sistem tipenya.

Sebelum adanya Flutter, Dart digunakan untuk pengembangan berbagai aplikasi web di Google. Tujuan awal dari penciptaan Dart adalah untuk menggantikan JavaScript yang dianggap memiliki berbagai kekurangan. Seiring waktu, peluncuran Flutter SDK untuk pengembangan aplikasi iOS, Android, dan web semakin memperkuat perhatian terhadap bahasa Dart.

Dart sebagai bahasa pemrograman memiliki beberapa karakteristik, diantaranya yaitu:

a. *Statically typed*, artinya kita perlu mendefinisikan variabel sebelum bisa menggunakannya

```
var name = 'Dicoding';
String language = 'Dart';
```

Bisa dilihat bahwa pada Dart kita tidak perlu mendefinisikan tipe data variabel secara eksplisit.

b. *Type Interface.*
tipe data akan secara otomatis terdeteksi ketika suatu variabel diinisialisasi. Sebagai contoh variabel `name` di atas akan terdeteksi sebagai String

c. *String Interpolation.*

```
print('Hello $name. Welcome to $language!');
```

Fitur di mana kita bisa menyisipkan variabel ke dalam sebuah objek String tanpa *concatenation* (penggabungan objek String menggunakan +)

d. *Multi paradigm, OOP & Functional.*
Dart mendukung paradigma Pemrograman Berorientasi Objek (OOP) dan Pemrograman Fungsional (FP). Melalui OOP, dapat mengembangkan program yang terstruktur dengan menggunakan konsep objek, kelas, properti, konstruktor, dan lainnya. Di sisi lain, dengan FP, dapat memanfaatkan fitur-fitur seperti *high-order function, recursion, first-class citizen.*

3. Guided

a. Membuat Variabel

Untuk membuat sebuah variabel dapat dideklarasikan dengan dianotasikan dengan atau tanpa tipe datanya.

Contoh membuat variabel tanpa anotasi tipe data menggunakan var.

```
var name = 'Udin';
var umur = 12;

print('Nama saya $name, umur $umur');
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Nama saya Udin, umur 12
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

Contoh membuat variabel dengan anotasi tipe datanya.

```
// dideklarasikan dengan tipe data
String name = 'Budiono Siregar';
String sekolah = 'sd mulia insain';
print ('Hallo nama saya $name, saya bersekolah di $sekolah');
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Hallo nama saya Budiono Siregar, saya bersekolah di sd mulia insain
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

b. Percabangan

If Else

```
var umur = 18;
if (umur >= 17) {
  print('Kamu sudah boleh menaiki wahana ini!');
} else {
  print('Kamu belum boleh mainiki wahana ini!');
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Kamu sudah boleh menaiki wahana ini!
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

Switch Case

```
var day = 4;
switch(day) {
  case 1:
    print('Hari ini adalah hari senin');
  break;
  case 2:
    print('Hari ini adalah hari selasa');
  break;
  case 3:
    print('Hari ini adalah hari rabu');
  break;
  case 4:
    print('Hari ini adalah hari kamis');
  break;
  case 5:
    print('Hari ini adalah hari jumat');
  break;
  case 6:
    print('Hari ini adalah hari sabtu');
  break;
  case 7:
    print('Hari ini adalah hari minggu');
  break;
  default:
    print('Masukan input hari yg benar!');
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Hari ini adalah hari kamis
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

c. Perulangan

For Loop

```
// LOOPING
void main() {    The declaration 'main' isn't
// For loop untuk mencetak angka 1 sampai 5
for (int i = 1; i <= 5; i++) {
print(i);
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
1
2
3
4
5
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

While loop

```dart
int i = 1; // Inisialisasi variabel
// While loop untuk mencetak angka 1 sampai 5
while (i <= 5) {
  print('Angka: $i');
  i++; // Increment untuk menghindari loop tak berujung
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Angka: 1
Angka: 2
Angka: 3
Angka: 4
Angka: 5
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

d. List

Fixed List

Fixed list adalah list yang tidak dapat diubah setelah dibuat.

```dart
// List
// Fixed List
List<int> fixedList =
    List.filled(3, 0); // List dengan 3 elemen, semua diinisialisasi ke 0

// Mengubah elemen dalam list
fixedList[0] = 10;
fixedList[1] = 20;
fixedList[2] = 30;
print('Fixed Length List: $fixedList'); // Output: [10, 20, 0, 0, 0]

// Menambah atau menghapus elemen tidak diperbolehkan
// fixedList.add(30); // Ini akan menimbulkan error
// fixedList.removeAt(0); // Ini juga akan menimbulkan error
```

Growable list

List yang dapat diubah setekah dibuat.

```dart
List<int> growableList = [];
// Menambahkan elemen baru ke dalam list
growableList.add(10);
growableList.add(20);
growableList.add(30);
print( 'Growable List setelah menambah elemen: $growableList'); //Output: [10, 20, 30]
// Menambahkan lebih banyak elemen
growableList.add(40);
growableList.add(50);
print(growableList); // Output: [10, 20, 30, 40, 50]
// Menghapus elemen dari list
growableList.remove(20);
print( 'Growable List setelah menghapus elemen: $growableList'); //Output: [10, 30, 40, 50]
```

e. Fungsi

Mendefinisikan fungsi

```dart
// mendefiniskan fungsi
void greet(String pesan){
  print(pesan);
}
```

Mengembalikkan nilai

```dart
int multiply(int a, int b){
  return a * b;  // mengembalikkan nilai a dan b
}
```

Memanggil fungsi

```dart
Run | Debug
void main(){
  int hasil = multiply(3, 4); // memanggil fungsi multiply

  print(hasil);
  greet('Hello World!'); // memanggil fungsi greet
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
12
Hello World!
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

4. **Unguided**

a. Unguided 1

Pada code berikut, bisa menggunakan percabangan if else atau switch, hasilnya juga akan sama.

```dart
void main() {
  var nilai = 50;
  if (nilai > 70) {
    print('Nilai A');
  } else if(nilai <= 70){
    print('Nilai B');
  } else if(nilai <= 40){
    print('Nilai C');
  } else{
    print('Nilai Tidak Valid!');
  }

  switch(nilai){
    case > 70:
      print('Nilai A');
    break;
    case <= 70:
      print('Nilai B');
    break;
    case <= 40:
      print('Nilai C');
    break;
    default:
      print('Nilai Tidak Valid!');
  }
}
```

Output :

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Nilai B
Nilai B
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

b. Unguided 2

```dart
import 'dart:io';

Run | Debug
void main() {
  stdout.write('Enter your number: ');
  int n = int.parse(stdin.readLineSync()!);
  for(int i = 1; i <= n; i++){
    for (int j = n; j > i; j--) {
      stdout.write(" ");
    }
    for (int k = 1; k <= (2 * i - 1); k++) {
      stdout.write("*");
    }
    print('');
  }
}
```

Output:

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Enter your number: 3
  *
 ***
*****
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

c. Unguided 3

```dart
Run | Debug
void main() {
  stdout.write('Masukkan bilangan bulat: ');
  int? number = int.tryParse(stdin.readLineSync() ?? '');

  if (number == null) {
    print('Input tidak valid. Harap masukkan bilangan bulat.');
  } else {
    // Memeriksa apakah bilangan tersebut bilangan prima
    bool isPrime = true;

    if (number <= 1) {
      isPrime = false; // Bilangan kurang dari 2 bukan bilangan prima
    } else {
      for (int i = 2; i * i <= number; i++) {
        if (number % i == 0) {
          isPrime = false; // Dapat dibagi oleh bilangan lain, bukan prima
          break;
        }
      }
    }

    if (isPrime) {
      print('$number adalah bilangan prima.');
    } else {
      print('$number bukan bilangan prima.');
    }
  }
}
```

Output :

```
PS D:\Semester_5\PPB_Praktikum\praktikum_3> dart index.dart
Masukkan bilangan bulat: 12
12 bukan bilangan prima.
PS D:\Semester_5\PPB_Praktikum\praktikum_3>
```

5. Kesimpulan

Dengan mempelajari dan memahami dasar-dasar bahasa pemrograman dart akan memudahkan dalam melanjutkan pembelajaran tingkat lanjut, karena sudah memiliki bekal terkait variabel, percabangan, perulangan, list dan fungsi. Meskipun masih banyak yang perlu dipelajari, namun untuk tahapan dasar materi tersebut cukup untuk bisa melanjutkan ke pembelajaran tingkat lanjut.