

# LAPORAN PRAKTIKUM



**NIM** : 2003071

**Nama** : Hakim Asrori

**Kelas** : D3TI.2C

**Mata Kuliah** : **Pemrograman Perangkat Bergerak (TIU3403)**

**Praktikum ke / Judul** : 2 / Dasar pemrograman Dart

**Tanggal Praktikum** : 17 Februari 2022

**Dosen Pengampu** : Fachrul Pralienka Bani Muhamad,  
S.ST., M.Kom

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU  
2022**



## A. Tujuan Praktikum

### Tujuan Umum

Mahasiswa memahami konsep dasar dari bahasa pemrograman Dart serta mengimplementasikannya dalam kode program untuk menyelesaikan suatu permasalahan komputasional yang sederhana.

### Tujuan Khusus

Mahasiswa mampu:

1. Menjelaskan pengertian bahasa pemrograman Dart
2. Menjelaskan cara kerja dan karakteristik bahasa pemrograman Dart
3. Membuat variabel menggunakan bahasa pemrograman Dart
4. Membuat konstanta menggunakan bahasa pemrograman Dart
5. Mengetahui dan menggunakan tipe data dalam bahasa pemrograman Dart
6. Melakukan parsing tipe data dalam bahasa pemrograman Dart
7. Mengetahui dan menggunakan collection dalam bahasa pemrograman Dart
8. Mengetahui penggunaan nullable & non-nullable dalam bahasa pemrograman Dart
9. Mengetahui dan menggunakan tipe inputan dalam bahasa pemrograman Dart
10. Mengetahui dan menggunakan jenis-jenis operator dalam bahasa pemrograman Dart

## B. Rangkuman Teori Singkat

Bahasa Pemrograman Dart dikembangkan oleh Google yang ditujukan untuk multiplatform mobile, desktop dan web. Bersifat opensource dan general purpose. Dart menerapkan konsep OOP dimana struktur kode berada dalam class yang didalamnya berisi method maupun variable. Dart menggunakan mekanisme C-Style sehingga mirip dengan Bahasa pemrograman C, Java, Javascript, Swift.

Teknologi compiler yang fleksibel memungkinkan Dart dapat dijalankan dengan cara berbeda, tergantung platform yang dituju.

- Dart Native, ditujukan untuk program yang menargetkan perangkat seluler, desktop, server, dan lainnya. Dart native mencakup Dart VM dengan kompilasi JIT (just-in-time) dan kompiler AOT (ahead-of-time) untuk menghasilkan kode mesin. metode AOT sistem di compile sedari awal menjadi sistem native, sehingga membantu developer lebih cepat pada saat men-develop dan running aplikasi dan developer tidak perlu melakukan kompilasi atau build ulang untuk melihat hasilnya.
- Dart WEB, ditujukan untuk web. Dart web menyertakan compiler development (dartdevc) dan compiler production (dart js).

Karakteristik Bahasa Pemrograman Dart

- Statically Typed, artinya perlu mendefinisikan variable
- Type Inference, artinya tipe data akan terdeteksi secara otomatis ketika suatu variable diinisialisasikan
- String Interpolation, artinya dapat menyisipkan variable kedalam string tanpa concatenation



- Multi-Paradigm, artinya Dart bisa dituliskan sebagai object-oriented programming ataupun sebagai functional programming.

#### Dasar Pemrograman Dart

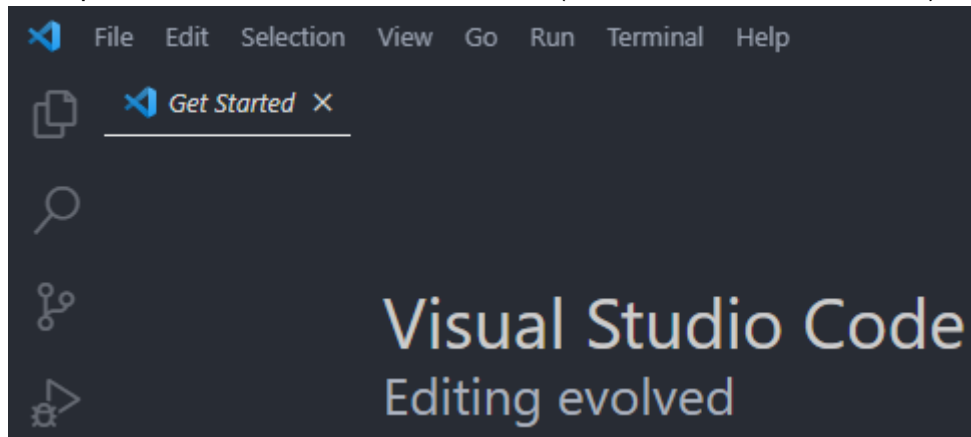
- Variable, adalah suatu tempat pada memory yang digunakan untuk menampung sebuah data (nilai). Memiliki karakteristik strongly typed.
- Konstanta, adalah variable yang digunakan nilai tetap.
- Tipe Data, adalah jenis data yang memiliki batasan tempat (pengalokasian memori di komputer) dan karakteristik sesuai dengan interpretasi data dan dapat diolah oleh komputer untuk memenuhi kebutuhan dalam pemrograman komputer.
  - o Number, mewakili literal numerik baik integer maupun double
  - o String, pada Dart tersusun atas urutan unit kode UTF-16
  - o Boolean, hanya memiliki 2 nilai yaitu true dan false
  - o List, adalah objek koleksi (collection) yang menyimpan daftar objek.
  - o Set, adalah container generik di mana tidak ada duplikasi objek.
  - o Maps, dalam dart adalah objek koleksi yang setiap elemennya berupa pasangan kunci (key) dan nilai (value).



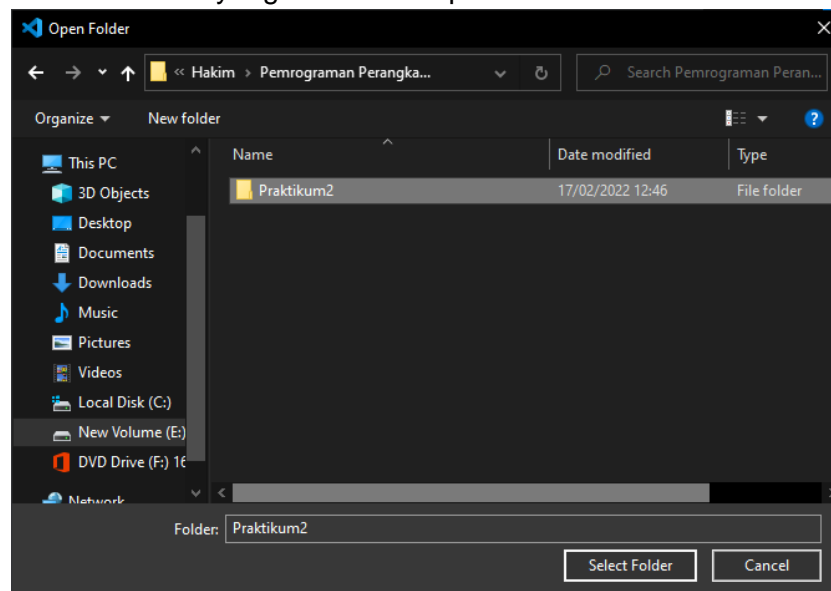
## C. Langkah dan Hasil Pelaksanaan Praktikum

### Langkah-langkah praktikum pengenalan dart

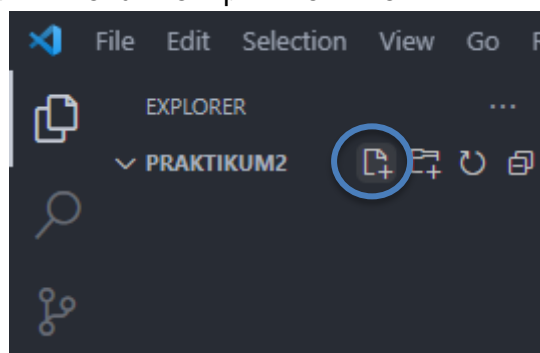
1. Buka aplikasi text editor VisualStudio Code (diasumsikan sudah terinstal)



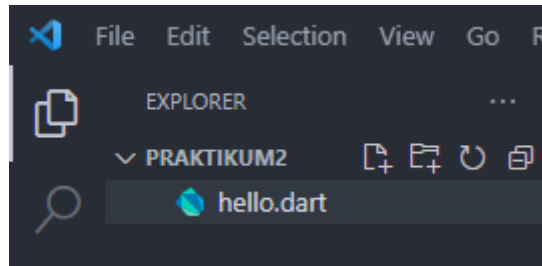
2. Buatlah sebuah folder yang diberi nama praktikum2



3. Open folder praktikum2, kemudian buatlah file baru dengan cara klik icon New File atau dengan cara klik menu File > pilih New File



4. Beri nama file menjadi hello.dart



5. Tulis dan simpan kode scripting 2.1 berikut

```
Praktikum2 - hello.dart

1 main() {
2     print("Hello World!");
3 }
```

6. Jalankan kode script hello.dart melalui menu Terminal > New Terminal (Ctrl+Shift+`), lalu ketik perintah dart hello.da

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

polindra@Hame Praktikum2 % dart hello.dart
Hello World!
polindra@Hame Praktikum2 %
```

7. Selesai



## E. Langkah dan Hasil Latihan

Latihan dasar pemrograman dart Buatlah sebuah file dengan ekstensi \*.dart di dalam folder praktikum2, lalu ketik dan simpanlah beberapa kode scripting berikut, serta tampilkan (screenshot) hasilnya!

Kode scripting 2.2 praktikum2/deklarasiVariabel.dart

```
Praktikum2 - deklarasiVariabel.dart
1  enum Status {disetujui, ditolak, diperbaiki, ditinjau}
2
3  void main() {
4      String nama = "Fulan";
5      int umur = 19;
6      double tinggi = 180.43;
7      bool isLulus = true;
8
9      var alamat = "Indramayu, Jawa Barat, Indonesia";
10
11     print("Nama saya $nama. umur saya $umur");
12     print("Tinggi badan saya sekitar $tinggi cm");
13     print("Saat ini saya tinggal di $alamat");
14     print("Status proposal Tugas Akhir saya saat ini adalah $Status.disetujui ");
15     print("Kelulusan saya tahun ini diharapkan " + isLulus.toString());
16 }
```

Hasil scripting 2.2

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Nama saya Fulan. umur saya 19
Tinggi badan saya sekitar 180.43 cm
Saat ini saya tinggal di Indramayu, Jawa Barat, Indonesia
Status proposal Tugas Akhir saya saat ini adalah Status.disetujui
Kelulusan saya tahun ini diharapkan true
Exited
```

Kode scripting 2.3 praktikum2/konstanta.dart

```
Praktikum2 - konstanta.dart
1  void main() {
2      const PHI = 3.14;
3      final GRAVITASI = 9.8;
4      const DATABASE = "db_akademik";
5
6      print("Nilai konstan phi = $PHI");
7      print("Nilai konstan gravitasi = " + GRAVITASI.toString());
8      print("Koneksi basis data $DATABASE");
9  }
```

Hasil code scripting 2.3.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Nilai konstan phi = 3.14
Nilai konstan gravitasi = 9.8
Koneksi basis data db_akademik
Exited
```



#### Kode scripting 2.4 praktikum2/parsingDariString.dart

```
Praktikum2 - parsingDariString.dart

1 void main() {
2   var nilai = "20";
3   int bilanganBulat = int.parse(nilai);
4   print(bilanganBulat + 5);
5
6   double bilanganDesimal = double.parse("7.894");
7   print(bilanganDesimal + 22);
8 }
9
```

Hasil kode scripting 2.4.

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

25
29.894
Exited
```

#### Kode scripting 2.5 praktikum2/parsingKeString.dart

```
Praktikum2 - parsingKeString.dart

1 main() {
2   var idNumber = 123;
3   var idString = idNumber.toString();
4
5   print("Nilai idNumber = " + idNumber.toString());
6
7   double bobotDouble = 50.98;
8   String bobotString = bobotDouble.toString();
9
10  print("nilai bobotDouble = $bobotString");
11 }
```

Hasil kode scripting 2.5.

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

Nilai idNumber = 123
nilai bobotDouble = 50.98
Exited
```

#### Kode scripting 2.6 praktikum2/collectionList.dart



```
Praktikum2 - collectionList.dart

1 void main() {
2   List<String> daftarNama = <String>[];
3
4   daftarNama.add("Abdi");
5   daftarNama.add("Banu");
6   daftarNama.add("Carita");
7
8   print("Nama Pertama $daftarNama[0]");
9   print("Nama Kedua $daftarNama[1]");
10  print("Nama Ketiga $daftarNama[2]");
11  print(daftarNama);
12  var daftarNilai = {90, 78, 84};
13  print(daftarNilai);
14 }
```

Hasil kode scripting 2.6.

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

Nama Pertama [Abdi, Banu, Carita][0]
Nama Kedua [Abdi, Banu, Carita][1]
Nama Ketiga [Abdi, Banu, Carita][2]
{90, 78, 84}
Exited
```

Kode scripting 2.7 praktikum2/collectionSet.dart

```
Praktikum2 - collectionSet.dart

1 void main() {
2   Set<String> mataKuliah = new Set<String>();
3
4   mataKuliah.add("PBO");
5   mataKuliah.add("PPB");
6   mataKuliah.add("BD");
7   mataKuliah.add("PBO");
8
9   print(mataKuliah);
10
11  Set<int> nilai = { 34, 78, 12, 34, 78 };
12  print(nilai);
13 }
```

Hasil kode scripting 2.7.

```
PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

{PBO, PPB, BD}
{34, 78, 12}
Exited
```

Kode scripting 2.8 praktikum2/collectionMap.dart





```
Praktikum2 - collectionMap.dart

1 void main() {
2   var mahasiswa = <int, String> {
3     1: "Fulan",
4     2: "Fulanah",
5   };
6   print(mahasiswa[1]);
7
8   Map nilai = {'Joko': 89, 'Udin': 90};
9   print(nilai);
10
11  var mataKuliah = {1: "PBO", 2: "PPB"};
12  print(mataKuliah[2]);
13
14  var email = new Map();
15  email[1] = 'fulan@gmail.com';
16  email[2] = 'fulanah@gmail.com';
17  print(email);
18 }
```

Hasil kode scripting 2.8.

```
PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

Fulan
{Joko: 89, Udin: 90}
PPB
{1: fulan@gmail.com, 2: fulanah@gmail.com}
Exited
```

Kode scripting 2.9 praktikum2/nullable.dart

```
Praktikum2 - nullable.dart

1 void main() {
2   String? data;
3
4   var nilai = int.parse(data!);
5
6   print(nilai);
7 }
```

Hasil kode scripting 2.9.

```
PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

Unhandled exception:
Null check operator used on a null value
#0      main
#1      _delayEntrypointInvocation.<anonymous closure> (dart:isolate-patch/isolate_patch.dart:297:19)
#2      _RawReceivePortImpl._handleMessage (dart:isolate-patch/isolate_patch.dart:192:12)

Exited (255)
```



Kode scripting 2.10 praktikum2/input.dart

```
Praktikum2 - input.dart
1 import 'dart:io';
2
3 main(List<String> args) {
4   print("Masukan nama anda : ");
5   String? nama = stdin.readLineSync();
6
7   print(nama);
8 }
```

Hasil kode scripting 2.10

```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
Masukan nama anda :
→ Hakim Asrori
Global evaluation requires a thread to have been loaded
```

Kode scripting 2.11 praktikum2/arithmeticOperator.dart

```
Praktikum2 - arithmeticOperator.dart
1 main(List<String> args) {
2   const PHI = 3.14;
3   double r = 7;
4   double luas = PHI * r * r;
5
6   print("Luas lingkaran : $luas");
7 }
```

Hasil kode scripting 2.11.

```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
Luas lingkaran : 153.86
Exited
```

Kode scripting 2.12 praktikum2/relationalOperator.dart



```
Praktikum2 - relationalOperator.dart

1 main(List<String> args) {
2   var num1 = 5;
3   var num2 = 9;
4   var res = num1 > num2;
5   print("num1 greater than num2 :: " + res.toString());
6
7   res = num1 < num2;
8   print("num1 lesser than num2 :: " + res.toString());
9
10  res = num1 >= num2;
11  print("num1 greater than or equal to num2 :: " + res.toString());
12
13  res = num1 <= num2;
14  print("num1 lesser than or equal to num2 :: " + res.toString());
15
16  res = num1 != num2;
17  print("num1 not equal to num2 :: " + res.toString());
18
19  res = num1 == num2;
20  print("num1 not equal to num2 :: " + res.toString());
21 }
```

Hasil kode scripting 2.12

```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

num1 greater than num2 :: false
num1 lesser than num2 :: true
num1 greater than or equal to num2 :: false
num1 lesser than or equal to num2 :: true
num1 not equal to num2 :: true
num1 not equal to num2 :: false
Exited
```

Kode scripting 2.13 praktikum2/typeTestOperator.dart

```
Praktikum2 - typeTestOperator.dart

1 void main(List<String> args) {
2   double n = 2.0;
3   var num = n is! int;
4   print(num);
5 }
```

Hasil kode scripting 2.13

```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

true
Exited
```

Kode scripting 2.14 praktikum2/logicalOperator.dart



```
Praktikum2 - logicalOperator.dart

1 main(List<String> args) {
2   var a = 10;
3   var b = 12;
4   var res = (a<b)&&(b>10);
5   print(res);
6 }
```

Hasil kode scripting 2.14

```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

true
Exited
```

Kode scripting 2.15 praktikum2/bitwiseOperator.dart

```
Praktikum2 - bitwiseOperator.dart

1 main(List<String> args) {
2   var a = 2;
3   var b = 3;
4
5   var result = (a & b);
6   print("(a & b) => ${result}");
7
8   result = (a | b);
9   print("(a | b) => ${result}");
10
11  result = (a ^ b);
12  print("(a ^ b) => ${result}");
13
14  result = (~b);
15  print("(~b) => ${result}");
16 }
```

Hasil kode scripting 2.15



```
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
(a & b) => 2
(a | b) => 3
(a ^ b) => 1
(~b) => -4
Exited
```

## F. Hasil dan Penjelasan Tugas

1. Buatlah program dart yang dapat menghitung total nilai dari mata kuliah, dengan ketentuan berikut:
  - Input: nama mata kuliah, nilai uts, nilai uas, nilai tugas
  - Output: nama mata kuliah, total nilai
  - Rumus total nilai:  $(\text{nilai uts} + \text{nilai uas} + \text{nilai tugas}) / 3$

Kode program



```
Praktikum2 - tugas_1.dart

1  import 'dart:io';
2
3  double rumus(double uts, double uas, double tugas) {
4      var hasil = (uts + uas + tugas) / 3;
5      return hasil;
6  }
7
8  main() {
9      print("");
10     stdout.write("Masukan Nama Mata Kuliah = ");
11     String nama = stdin.readLineSync();
12
13     print("");
14
15     stdout.write("Masukan Nilai UTS = ");
16     double uts = double.parse(stdin.readLineSync());
17
18     print("");
19
20     stdout.write("Masukan Nilai UAS = ");
21     double uas = double.parse(stdin.readLineSync());
22
23     print("");
24
25     stdout.write("Masukan Nilai Tugas = ");
26     double tugas = double.parse(stdin.readLineSync());
27
28     print("");
29
30     double totalNilai = rumus(uts, uas, tugas);
31
32     print("Nama Mata Kuliah = $nama");
33     print("Total Nilai = " + totalNilai.toStringAsFixed(2));
34 }
35
```



## Hasil

```
MINGW64:/e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2
Lenovo@DESKTOP-8PNCJ3A MINGW64 /e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2
$ dart tugas/tugas_1.dart

Masukan Nama Mata Kuliah = PBO

Masukan Nilai UTS = 78

Masukan Nilai UAS = 87

Masukan Nilai Tugas = 77

Nama Mata Kuliah = PBO
Total Nilai = 80.67

Lenovo@DESKTOP-8PNCJ3A MINGW64 /e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2
$ |
```

2. Buatlah susunan data mahasiswa sebagai berikut dengan menggunakan gabungan beberapa collection

```
mahasiswa
0
  nim = "09030015"
  nama = "Udin"
  matkul
    0 = "PW"
    1 = "PBO"
    2 = "PPB"
1
  nim = "09030016"
  nama = "Joko"
  matkul
    0 = "BASDAT"
    1 = "RPL"
```

## Kode program



```
Praktikum2 - tugas_2.dart

1 void main() {
2   var data = {
3     "mahasiswa": <int, Map>{
4       0: {
5         "nim": "09030015",
6         "nama": "Udin",
7         "matkul": <int, String>{
8           0: "PW",
9           1: "PBO",
10          2: "PPB",
11        }
12      },
13      1: {
14        "nim": "09030016",
15        "nama": "Joko",
16        "matkul": <int, String>{
17          0: "Basdat",
18          1: "RPL",
19        }
20      },
21    },
22  };
23
24  print(data);
25 }
26
```

Hasil

```
MINGW64:/e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2

Lenovo@DESKTOP-8PNCJ3A MINGW64 /e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2
$ dart tugas/tugas_2.dart
{mahasiswa: {0: {nim: 09030015, nama: Udin, matkul: {0: PW, 1: PBO, 2: PPB}}, 1: {nim: 09030016, nama: Joko, matkul: {0: Basdat, 1: RPL}}}}

Lenovo@DESKTOP-8PNCJ3A MINGW64 /e/Hakim/Pemrograman Perangkat Bergerak/Praktikum2
$
```





## G. Kesimpulan

Bahasa Pemrograman Dart merupakan Bahasa yang dibangun Google yang diperuntukan multiplatform, dengan arsitektur OOP atau functional programming.

## H. Referensi

- Dieter Meiller. "Modern App Development with Dart and Flutter 2". The Deutsche Nationalbibliothek. Berlin. 2021.
- Alberto Miola. "Flutter Complete Reference Create Beautiful, Fast and Native Apps for Any Device". Independently Published. 2020.
- Kode TR. "Belajar Flutter Mulai dari Dasar menggunakan Pemrograman Dart". 2019. <https://www.kodetr.com/belajar-flutter-mulai-dari-dasar-menggunakan-pemrograman-dart/>. Diakses tanggal 15 Februari 2022.
- Omadi Jaya. "Pengenalan Syntax dan Data Type pada Dart Lang". 2020. <https://belajarflutter.com/dart-pengenalan-syntax-dan-data-type-pada-dart/>. Diakses tanggal 16 Februari 2022.