

**TUGAS PENDAHULUAN
PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL I
PENGENALAN DART**



Disusun Oleh :

Dwi Candra Pratama/2211104035

SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

A. Guided

1) Variabel dengan var

Var adalah cara untuk mendeklarasikan variabel tanpa menentukan tipe datanya secara eksplisit. Dart akan secara otomatis menentukan tipe data berdasarkan nilai yang diberikan. Berikut adalah tipe dari Var, yaitu Type Annotation & Multiple Variable

a) Type Annotation

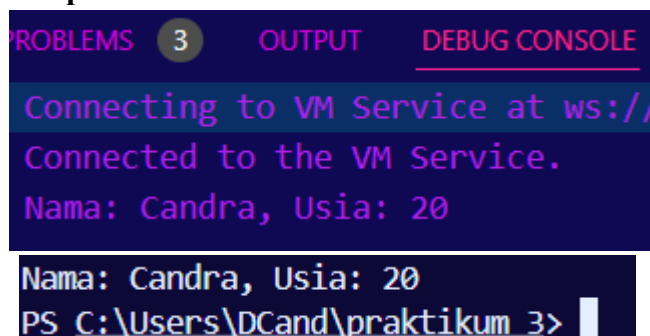
Soal Studi Case

```
void main() {  
  
    // Menggunakan var  
    var name = "Alice"; // Tipe data String  
    var age = 25; // Tipe data Integer  
  
    print("Nama: $name, Usia: $age");  
}
```

SourceCode

```
void main() {  
    // Menggunakan var  
    var name = "Candra"; // Tipe data String  
    var age = 20; // Tipe data Integer  
  
    print("Nama: $name, Usia: $age");  
}
```

Screenshoot Output



The screenshot shows the IDE's output window with three tabs: PROBLEMS, OUTPUT, and DEBUG CONSOLE. The DEBUG CONSOLE tab is active, displaying the following text: "Connecting to VM Service at ws://", "Connected to the VM Service.", and "Nama: Candra, Usia: 20". Below this, a terminal window shows the same output: "Nama: Candra, Usia: 20" and the command prompt "PS C:\Users\DCand\praktikum_3>".

b) Multiple Variable

Soal Studi Case

```
void main() {  
  
    // Multiple variable  
    String firstName, lastName; // Tipe data String  
    firstName = "Charlie";  
    lastName = "Brown";  
  
    print("Nama Lengkap: $firstName $lastName");  
}
```

SourceCode

```
void main() {  
    // Multiple variable  
    String firstName, lastName; // Tipe data String  
    firstName = "Dwi Candra ";  
    lastName = "Pratama";  
    print("Nama Lengkap: $firstName $lastName");  
} //Output: Nama Lengkap: Dwi Candra Pratama
```

Deskripsi Program

(Program ini bertujuan menampilkan informasi nama dan usia pengguna. Dengan mendeklarasikan dua variabel, **name** (tipe *String*) yang berisi "Candra" dan **age** (tipe *Integer*) yang berisi 20, program menggunakan fungsi print untuk mencetak teks yang menggabungkan nilai kedua variabel tersebut. Hasilnya, program menampilkan "Nama: Candra, Usia: 20" di konsol. Program ini sederhana dan mempraktikkan penggunaan variabel serta interpolasi string.)

2) Statemen Control

a) If-Else

If-Else adalah alur yang menjalankan blok kode tertentu berdasarkan kondisi true atau false.

```
void main() {  
    var openHours = 8;  
    var closedHours = 21;  
    var now = 17;  
    if (now > openHours && now < closedHours) {  
        print("Hello, kita sekarang Buka");  
    } else {  
        print("Sorry, kita sekarang tutup");  
    } //Output: Hello, kita sekarang Buka  
}
```

b) Switch-Case

Digunakan ketika ada banyak kemungkinan nilai yang harus dibandingkan. Setiap nilai yang cocok memiliki blok eksekusi tersendiri.

```
void main() {  
    var Nilai = 'b'; // Misalkan 1 = Senin, 2 = Selasa, dst.  
    switch (Nilai) {  
        case 'a':  
            print("Sangat Baik");  
            break;  
        case 'b':  
            print("Baik");  
            break;  
        case 'c':  
            print("Cukup Baik");  
            break;  
        default:  
            print("Nilai Tidak Tersedia");  
    } //Output: Baik  
}
```

Deskripsi Program

Program ini digunakan untuk memeriksa nilai suatu variabel dan menampilkan pesan berdasarkan statusnya. Saat Anda menggunakan If-Else, program terlebih dahulu mengevaluasi apakah nilai variabel memenuhi kondisi tertentu (misalnya apakah angkanya lebih besar dari 5) dan menampilkan pesan terkait. Selain itu, program switch-case memetakan nilai variabel lain ke case yang telah ditentukan sebelumnya, seperti huruf 'a', 'b', atau 'c'. Jika nilainya cocok dengan salah satu kasus, pesan terkait akan ditampilkan. Jika tidak ada kecocokan, program akan menampilkan pesan standar yang menunjukkan bahwa nilainya tidak tersedia. Kombinasi kedua struktur pengendalian ini memudahkan pengambilan keputusan dan memberikan hasil sesuai dengan kondisi tertentu.

3) Looping

a) For Loop

Loop ini digunakan untuk mengulang eksekusi blok kode dengan jumlah pengulangan yang diketahui.

```
void main() {  
    for (var i = 0; i < 5; i++) {  
        print("Looping$i");  
    }  
} //Output: Looping 0, Looping 1, Looping 2, Looping 3, Looping 4
```

b) While Loop

Loop ini menjalankan blok kode selama kondisi tertentu bernilai true.

```
void main() {  
    var i = 0;  
    while (i < 5) {  
        print("Looping$i");  
        i++;  
    }  
} //Output: Looping 0, Looping 1, Looping 2, Looping 3, Looping 4
```

Deskripsi Program

Program ini menggunakan struktur loop untuk mengulang eksekusi blok kode secara otomatis hingga kondisi tertentu terpenuhi. Dengan for loop, program menjalankan perulangan dengan jumlah iterasi yang telah ditentukan, seperti mencetak nilai dari 0 hingga 4. Pada while loop, program akan terus mengulang selama kondisi masih bernilai true, misalnya mencetak angka hingga kondisi $i < 5$ tidak lagi terpenuhi. Loop mempermudah tugas berulang tanpa perlu menulis kode yang sama berkali-kali.

4) List

adalah struktur data yang digunakan untuk menyimpan kumpulan elemen secara teratur. Elemen dalam list dapat diakses berdasarkan indeks (dimulai dari 0). Dart menyediakan beberapa metode untuk memanipulasi data di dalam list, seperti menambah, menghapus, atau mengubah elemen.

a) Fixed Length List

```
void main() {  
  // Membuat fixed-length list dengan panjang 3  
  List<int> fixedList = List.filled(3, 0); // List dengan 3 elemen,  
  // yang semua elemennya diisi dengan nilai 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]  
  //OutPut Fixed Length List: [10, 20, 30]  
  
  // Menambah atau menghapus elemen tidak diperbolehkan  
  // fixedList.add(30); // Ini akan menimbulkan error  
  // fixedList.removeAt(0); // Ini juga akan menimbulkan error  
}
```

b) Growable List

```
void main() {  
  // Membuat growable list (panjangnya bisa berubah)  
  List<int> growableList = [];  
  // Menambahkan elemen baru ke dalam list  
  growableList.add(10);  
  growableList.add(20);  
  growableList.add(30);  
  
  print(growableList);  
  // Menambahkan lebih banyak elemen  
  growableList.add(50);  
  growableList.add(80);  
  growableList.add(90);  
  
  print(growableList);  
  
  growableList.remove(20);  
  
  print(growableList);  
  // Output: [10, 30, 40, 50]  
}
```

Note: Dapat diubah ukurannya secara dinamis, kita bisa menambah atau menghapus elemen kapan saja.


5) Fungsi

blok kode yang dirancang untuk melakukan tugas tertentu dan dapat dipanggil berkali-kali. Fungsi membantu mengorganisasi kode agar lebih modular dan mudah dibaca. Dart mendukung fungsi dengan parameter dan juga dapat mengembalikan nilai.

```
void cetakPesan(String pesan) {  
    print(pesan);  
}  
int perkalian(int a, int b) {  
    return a * b;  
}  
void main() {  
    int hasil = perkalian(5, 4);  
    print('hasil perkalian fungsi: $hasil');  
  
    cetakPesan('Halo, guys!');  
} //Output: hasil perkalian fungsi: 20  
//Halo, guys!
```

Menambah parameter

```
void greet(String name, int age) {  
    print('Halo $name, you are $age years old.');
```



```
}  
void main() {  
    greet('Candra', 20);  
} //Output: Halo Candra, you are 20 years old.
```

B. UNGUIDED

- 1) Tugas Percabangan (Branching) Soal: Buatlah sebuah fungsi dalam Dart yang menerima sebuah nilai dari user, lalu melakukan percabangan untuk memberikan output berdasarkan kondisi berikut:

Deskripsi :

- Jika nilai lebih besar dari 70, program harus mereturn "Nilai A".
- Jika nilai lebih besar dari 40 tetapi kurang atau sama dengan 70, program harus mereturn "Nilai B".
- Jika nilai lebih besar dari 0 tetapi kurang atau sama dengan 40, program harus mereturn "Nilai C".

d. Jika nilai tidak memenuhi semua kondisi di atas, program harus mereturn teks kosong.

Sampel Input: 80

Sampel Output: 80 merupakan Nilai A

Sampel Input: 5

Sampel Output: 50 merupakan Nilai B

```
03_Pengertian Dart > UNDEFINED > JawabanNo1.dart > nilaiHuruf
1  import 'dart:io';
2
3  String nilaiHuruf(int nilai) {
4      if (nilai > 70) {
5          return "Nilai A";
6      } else if (nilai > 40 && nilai <= 70) {
7          return "Nilai B";
8      } else if (nilai > 0 && nilai <= 40) {
9          return "Nilai C";
10     } else {
11         return "";
12     }
13 }
14
Run | Debug
15 void main() {
16     print("Masukkan nilai: ");
17     String? input = stdin.readLineSync();
18     if (input != null) {
19         int nilai = int.parse(input);
20         String hasil = nilaiHuruf(nilai);
21         if (hasil.isNotEmpty) {
22             print("$nilai merupakan $hasil");
23         } else {
24             print(""); // Mengembalikan teks kosong jika tidak memenuhi semua kondisi
25         }
26     } else {
27         print("Input tidak valid");
28     }
29 }
```

PROBLEMS 62 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

UIDED\JawabanNo1.dart
Masukkan nilai:
80
80 merupakan Nilai A
PS C:\Praktikum PBB\PBB_DwiCandraPratama_2211104035_SE-06-02>

```
Masukkan nilai:
50
50 merupakan Nilai B
PS C:\Praktikum PBB\PBB_DwiCandraPratama_2211104035_SE-06-02>
```

2) Tugas Looping (Perulangan)

Soal:

Buatlah sebuah program dalam Dart yang menampilkan piramida bintang dengan

menggunakan for loop. Panjang piramida ditentukan oleh input dari user.

Contoh Output:



```

1  import 'dart:io';
2
   Run | Debug
3  ▼ void main() {
4      print('Masukkan tinggi piramida: ');
5      String? input = stdin.readLineSync();
6  ▼  if (input != null) {
7          int height = int.parse(input);
8          printStarPyramid(height);
9  ▼  } else {
10         print('Input tidak valid');
11     }
12 }
13
14 ▼ void printStarPyramid(int height) {
15 ▼     for (int i = 1; i <= height; i++) {
16         // Mencetak spasi
17 ▼     for (int j = 1; j <= height - i; j++) {
18         stdout.write(' ');
19     }
20     // Mencetak bintang
21 ▼     for (int k = 1; k <= 2 * i - 1; k++) {
22         stdout.write('*');
23     }
24     // Pindah ke baris baru
25     print('\n');
26 }
27 }

```

```

UIUED\JawabanNo2.dart"
Masukkan tinggi piramida:
8

      *
    ***
  *****
*****
*****
*****
*****
*****
*****
*****
PS C:\Praktikum PBB\PBB_DwiCar
UIUED\JawabanNo2.dart"

```

3) Tugas Function

Soal:

Buatlah program Dart yang meminta input berupa sebuah bilangan bulat dari user,

kemudian program akan mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan.

Sampel Input: 23

Sampel Output: bilangan prima

Sampel Input: 12

Sampel Output: bukan bilangan prima

```
03_Pengenalan Dart > UNGUIDED > JawabanNo3.dart > main
1  import 'dart:io';
2
3  bool isPrime(int number) {
4    if (number <= 1) return false;
5    for (int i = 2; i <= number / 2; i++) {
6      if (number % i == 0) return false;
7    }
8    return true;
9  }
10
Run | Debug
11 void main() {
12   print('Masukkan sebuah bilangan bulat:');
13   String? input = stdin.readLineSync();
14
15   if (input != null) {
16     try {
17       int number = int.parse(input);
18       if (isPrime(number)) {
19         print('bilangan prima');
20       } else {
21         print('bukan bilangan prima');
22       }
23     } catch (e) {
24       print('Input tidak valid. Mohon masukkan bilangan bulat.');
```

```
PS C:\Praktikum PBB\PBB_DwiCandraPratama_2211104035_SE-06-02> dart
UNGUIDED\JawabanNo3.dart
Masukkan sebuah bilangan bulat:
23
bilangan prima
PS C:\Praktikum PBB\PBB_DwiCandraPratama_2211104035_SE-06-02> dart
UNGUIDED\JawabanNo3.dart
Masukkan sebuah bilangan bulat:
12
bukan bilangan prima
PS C:\Praktikum PBB\PBB_DwiCandraPratama_2211104035_SE-06-02> |
```