

Nama : Akmal Muhamad Firdaus

NIM : 1301204188

Praktikum ABP Modul 8

1. Pengenalan Dart

Dart adalah bahasa pemrograman modern yang dikembangkan oleh Google. Dart dapat digunakan untuk membuat berbagai macam aplikasi, mulai dari aplikasi web hingga aplikasi mobile dan desktop. Dart juga merupakan bahasa pemrograman resmi dari framework Flutter yang digunakan untuk membangun aplikasi mobile multi-platform.

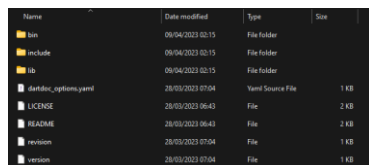
Dart memiliki sintaks yang mirip dengan bahasa pemrograman C dan JavaScript, sehingga mudah dipelajari oleh programmer yang sudah menguasai bahasa-bahasa tersebut. Dart juga memiliki fitur-fitur modern seperti tipe data yang kuat, pemrograman berorientasi objek, pengumpulan sampah (garbage collection), dan lain-lain.

Selain itu, Dart juga dilengkapi dengan berbagai macam library dan framework yang memudahkan pengembangan aplikasi, seperti Flutter untuk pembuatan aplikasi mobile, AngularDart untuk pembuatan aplikasi web, dan banyak lagi.

Untuk belajar flutter, tidak perlu terlalu fasih untuk mempelajari bahasa dart. Terdapat fundamental yang perlu dipelajari seperti variable, statement control, looping, array, fungsi, dsb. Karakteristik bahasa dart mirip dengan bahasa C ataupun Java. Wajib menggunakan titik koma diakhir codingan.

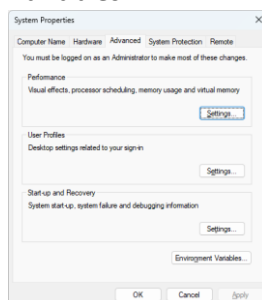
2. Instalasi Dart

1. Download dart-sdk
2. Extract sdk tersebut

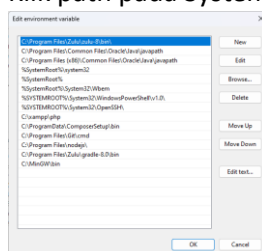


Name	Date modified	Type	Size
bin	09/04/2023 02:15	File folder	
include	09/04/2023 02:15	File folder	
lib	09/04/2023 02:15	File folder	
dartdevc_options.yaml	28/03/2023 07:04	Text Source File	1 KB
LICENSE	28/03/2023 06:43	File	2 KB
README	28/03/2023 06:43	File	2 KB
version	28/03/2023 07:04	File	1 KB
version	28/03/2023 07:04	File	1 KB

3. Buka windows, lalu cari **Edit the system environment variables**, lalu klik **Environment Variables...**



4. Klik path pada System Variables lalu klik edit



5. Tambahkan file path dart-sdk tadi yang sudah di extract

`C:\dart-sdk\bin`

6. Buka cmd lalu ketikkan perintah “dart --version”, jika cmd menampilkan versi dart yang digunakan, maka dart berhasil di install

```
C:\Users\Akmal Muhamad Firdau>dart --version
Dart SDK version: 2.19.6 (stable) (Tue Mar 28 13:41:04 2023 +0000) on "windows_x64"
```

3. Variable

Untuk penggunaan variable di dart, terdapat beberapa cara, yaitu dengan var, type annotation dan multiple variable.

```
1 void main() {
2   // var
3   //Cara penggunaan : var <variable_name>;
4   //Cara penggunaan : var <name> = <expression>;
5   var name = "Akmal Muhamad Firdaus";
6   var nim = 1301204188;
7   var isStudent = true;
8   print("Nama : $name");
9   print("NIM : $nim");
10  print("Is Student ? $isStudent");
11
12  // type annotation
13  //Cara penggunaan : <type> <variable_name>;
14  //Cara penggunaan : <type> <name> = <expression>;
15  String username = "akmalmf24";
16  int grade = 100;
17  bool isPassing = true;
18
19  // multiple variable
20  //Cara penggunaan : <type> <var1,var2...varN>;
21  String firstName, lastName;
22  int age, height, weight;
23
24 }
```

```
Nama : Akmal Muhamad Firdaus
NIM : 1301204188
Is Student ? true
```

Pada contoh di atas, terdapat tiga jenis variabel yang dideklarasikan. Variabel pertama dideklarasikan dengan menggunakan keyword var tanpa type annotation, sehingga tipe datanya akan ditentukan oleh nilai awal yang diberikan. Variabel kedua dan ketiga dideklarasikan dengan menggunakan type annotation untuk menentukan tipe datanya secara eksplisit.

Selain itu, contoh terakhir menunjukkan cara mendeklarasikan multiple variable dengan tipe data yang sama pada satu baris dengan memisahkan setiap nama variabel menggunakan koma.

4. Statement Control / Control Flow

Statement Control / Control Flow pada Dart digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi. Ada dua jenis statement control pada Dart, yaitu:

1. If-else

Statement ini digunakan untuk mengevaluasi sebuah kondisi dan melakukan tindakan berbeda tergantung pada hasil evaluasi. Bentuk umum dari if-else adalah sebagai berikut:

If Statement

```
1 if (kondisi) {
2   // kode yang dijalankan jika kondisi true
3   // 1301204188 - Akmal Muhamad Firdaus
4 }
```

If-Else Statement

```
1 if (kondisi) {
2   // kode yang dijalankan jika kondisi true
3   // 1301204188 - Akmal Muhamad Firdaus
4 } else {
5   // kode yang dijalankan jika kondisi false
6   // 1301204188 - Akmal Muhamad Firdaus
7 }
```

IF ELSE IF Statement

```

1  if (kondisi) {
2      // kode yang dijalankan jika kondisi true
3      // 1301204188 - Akmal Muhamad Firdaus
4  } else if(kondisi2) {
5      // kode yang dijalankan jika kondisi2 true
6      // 1301204188 - Akmal Muhamad Firdaus
7  }
8  .
9  . Kondisi ke n-2
10 . Kondisi ke n-1
11 else if(kondisi ke-n){
12     // kode yang dijalankan jika kondisi ke-n true
13     // 1301204188 - Akmal Muhamad Firdaus
14 } else {
15     // kode yang dijalankan jika tidak ada kondisi yang true
16     // 1301204188 - Akmal Muhamad Firdaus
17 }

```

Contoh Penggunaan

```

1  void main() {
2      // var
3      //Cara penggunaan : var <variable_name>;
4      //Cara penggunaan : var <name> = <expression>;
5      var name = "Akmal Muhamad Firdaus";
6      var nim = 1301204188;
7      var isStudent = true;
8      if(isStudent){
9          print("Halo $name, kamu adalah student");
10     } else if(nim == 1301204188){
11         print("Halo $name, ternyata kamu adalah student");
12     } else {
13         print("Maaf $name, kamu bukan student");
14     }
15 }

```

Halo Akmal Muhamad Firdaus, kamu adalah student

2. Switch Case

Statement ini digunakan untuk membandingkan nilai dari sebuah variabel dengan beberapa nilai yang mungkin, dan melakukan tindakan berbeda tergantung pada nilai yang cocok. Bentuk umum dari switch adalah sebagai berikut:

```

1  switch (nilai) {
2      case nilai1:
3          // kode yang dijalankan jika nilai sama dengan nilai1
4          // 1301204188 - Akmal Muhamad Firdaus
5          break;
6      case nilai2:
7          // kode yang dijalankan jika nilai sama dengan nilai2
8          // 1301204188 - Akmal Muhamad Firdaus
9          break;
10     default:
11         // kode yang dijalankan jika nilai tidak sama dengan nilai1 atau nilai2
12         // 1301204188 - Akmal Muhamad Firdaus
13 }

```

Contoh penggunaan

```

1  void main() {
2      var nilai = 'A';
3      switch (nilai) {
4          case 'A':
5              // 1301204188 - Akmal Muhamad Firdaus
6              print('Excellent');
7              break;
8          case 'B':
9              // 1301204188 - Akmal Muhamad Firdaus
10             print('Good');
11             break;
12          case 'C':
13              // 1301204188 - Akmal Muhamad Firdaus
14              print('Average');
15              break;
16          case 'D':
17              // 1301204188 - Akmal Muhamad Firdaus
18              print('Below Average');
19              break;
20          case 'E':
21              // 1301204188 - Akmal Muhamad Firdaus
22              print('Fail');
23              break;
24          default:
25              // 1301204188 - Akmal Muhamad Firdaus
26              print('Invalid Grade');
27      }
28 }

```

Excellent

5. Looping

Pada bahasa pemrograman Dart, ada tiga jenis looping yang tersedia untuk melakukan perulangan yaitu:

1. For loop

For loop digunakan untuk melakukan perulangan sejumlah kali tertentu. For loop terdiri dari tiga bagian: inisialisasi, kondisi, dan perubahan. Bentuk umum dari for loop adalah sebagai berikut:

```
1 for (inisialisasi; kondisi; perubahan) {
2     // kode yang dijalankan selama perulangan
3     // 1301204188 - Akmal Muhamad Firdaus
4 }
```

Contoh penggunaan

```
1 void main() {
2     for (var i = 1; i <= 5; i++) {
3         print("Cloningan akmal ke-$i");
4     }
5 }
```

```
Cloningan akmal ke-1
Cloningan akmal ke-2
Cloningan akmal ke-3
Cloningan akmal ke-4
Cloningan akmal ke-5
```

2. While loop

While loop digunakan untuk melakukan perulangan selama kondisi yang ditentukan bernilai true. Bentuk umum dari while loop adalah sebagai berikut:

```
1 while (kondisi) {
2     // kode yang dijalankan selama perulangan
3     // 1301204188 - Akmal Muhamad Firdaus
4 }
```

Contoh penggunaan

```
1 void main() {
2     var i = 1;
3     while (i <= 5) {
4         print("Cloningan akmal ke-$i, menggunakan while");
5         i++;
6     }
7 }
```

```
Cloningan akmal ke-1, menggunakan while
Cloningan akmal ke-2, menggunakan while
Cloningan akmal ke-3, menggunakan while
Cloningan akmal ke-4, menggunakan while
Cloningan akmal ke-5, menggunakan while
```

3. Do-While loop

Do-while loop mirip dengan while loop, namun perbedaannya adalah kode yang ada di dalam blok do akan dijalankan setidaknya satu kali bahkan jika kondisi tidak terpenuhi. Bentuk umum dari do-while loop adalah sebagai berikut:

```
1 do {
2     // kode yang dijalankan selama perulangan
3     // 1301204188 - Akmal Muhamad Firdaus
4 } while (kondisi);
```

Contoh penggunaan

```
1 void main() {
2     var i = 1;
3     do {
4         print("Jalan dulu, baru cek kondisi (Iterasi ke-$i)");
5         // 1301204188 - Akmal Muhamad Firdaus
6         i++;
7     } while (i < 5);
8 }
```

```
Jalan dulu, baru cek kondisi (Iterasi ke-1)
Jalan dulu, baru cek kondisi (Iterasi ke-2)
Jalan dulu, baru cek kondisi (Iterasi ke-3)
Jalan dulu, baru cek kondisi (Iterasi ke-4)
```

Setiap jenis looping memiliki kegunaan yang berbeda, tergantung pada kebutuhan penggunaannya.

6. List

List pada Dart merupakan kumpulan elemen yang dipesan (ordered) dan dapat diakses menggunakan index. List dapat berisi elemen dengan tipe yang sama atau berbeda. Terdapat dua jenis list pada Dart yaitu:

1. Fixed-length List

Fixed-length list adalah list yang ukurannya tetap dan tidak dapat diubah setelah dibuat. Untuk membuat fixed-length list, digunakan constructor `List(n)` dengan `n` adalah jumlah elemen yang diinginkan. Contohnya:

```
1 var fixedList = List(3);
2 fixedList[0] = 'Akmal';
3 fixedList[1] = 'Muhamad';
4 fixedList[2] = 'Firdaus';
```

Contoh penggunaan

```
1 void main() {
2     var fixedList = List(3);
3     fixedList[0] = 'Akmal';
4     fixedList[1] = 'Muhamad';
5     fixedList[2] = 'Firdaus';
6     for (var name in fixedList) {
7         print(name);
8     }
9 }
```

```
Akmal
Muhamad
Firdaus
```

2. Growable list

Growable list adalah list yang ukurannya dapat berubah setelah dibuat. Terdapat dua jenis growable list, yaitu `List` dan `List<E>`. Untuk membuat growable list menggunakan constructor `List()`. Contohnya:

```
1 var growableList = List();
2 growableList.add('Akmal');
3 growableList.add('Muhamad');
4 growableList.add('Firdaus');
```

Dapat juga menggunakan syntax literal [...] untuk membuat list. Contohnya:

```
1 var names = ["Akmal", "Muhamad", "Firdaus"];
```

Kita juga dapat melakukan operasi seperti mengakses elemen berdasarkan index, menambahkan elemen, menghapus elemen, mencari elemen tertentu, mengurutkan elemen, dan banyak lagi. Berikut beberapa contoh operasi yang dapat dilakukan pada list:

```
1 var names = ["Akmal", "Muhamad", "Firdaus"];
2 print(names[1]); // mengakses elemen pada index ke-1 (menghasilkan 'Muhamad')
3 names.add('1301204188'); // menambahkan elemen baru
4 names.remove('Muhamad'); // menghapus elemen 'Muhamad'
5 print(names.indexOf('Firdaus')); // mencari index elemen 'Firdaus'
6 names.sort(); // mengurutkan elemen secara ascending
7 print(names); // ['1301204188', 'Akmal', 'Firdaus']
```

Contoh penggunaan:

```
1 void main() {
2     var names = ["Akmal", "delete", "Muhamad", "Firdaus"];
3     names.remove('delete');
4     for (var name in names) {
5         print(name);
6     }
7 }
```

```
Akmal
Muhamad
Firdaus
```

7. Function

Function atau fungsi pada Dart adalah sebuah blok kode yang dapat dipanggil untuk melakukan tugas tertentu. Di Dart, setiap fungsi harus didefinisikan terlebih dahulu sebelum dapat digunakan.

Berikut adalah bentuk umum dari sebuah fungsi pada Dart:

```
1  return_type function_name(parameters) {  
2      // kode yang dijalankan oleh fungsi  
3      // 1301204188 - Akmal Muhamad Firdaus  
4      return value;  
5  }
```

Penjelasan dari tiap komponen di atas adalah sebagai berikut:

- 'return_type' adalah tipe data dari nilai yang akan dikembalikan oleh fungsi. Jika fungsi tidak mengembalikan nilai, maka tipe data yang digunakan adalah 'void'.
- 'function_name' adalah nama dari fungsi yang didefinisikan. Nama fungsi harus unik di dalam program.
- 'parameters' adalah variabel yang digunakan sebagai masukan untuk fungsi. Jika fungsi tidak memiliki masukan, maka tanda kurung kosong () harus tetap ditulis.
- "kode yang dijalankan oleh fungsi" adalah blok kode yang akan dieksekusi ketika fungsi dipanggil.
- 'return value' adalah statement yang mengembalikan nilai dari fungsi. Statement ini hanya perlu ditulis jika fungsi mengembalikan nilai.

Contoh penggunaan

```
1  int factorial(number) {  
2      if (number <= 0) {  
3          // termination case  
4          return 1;  
5      } else {  
6          return (number * factorial(number - 1));  
7          // function invokes itself  
8      }  
9  }  
10  
11 void main() {  
12     print( factorial(6));  
13 }
```

720

Full Source Code: <https://github.com/codernewbie04/CII3H4-ABP-Praktikum>