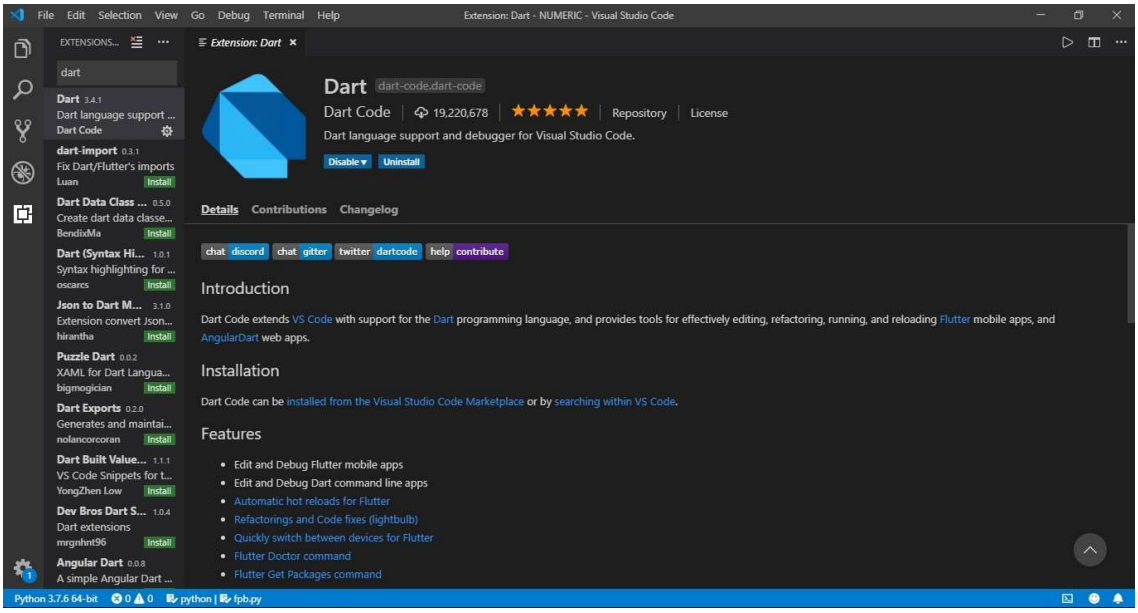
BAHASA PEMOGRAMAN DART

Dart adalah bahasa pemrograman dikembangkan oleh google sejak tahun 2007 oleh sebuah tim yang dipimpin Lars Bak dan Kasper Lund yang berfokus untuk optimalisasi sisi client. Tidak hanya digunakan untuk pengembangan aplikasi seluler, Dart juga dapat digunakan untuk mengembangkan berbagai macam aplikasi seperti *web,* micro service, desktop dan aplikasi lain yang mengusung teknologi Internet of Things (IoT).

Dart merupakan bahasa pemrograman yang berorientasi objek (OOP) dengan syntax yang mirip dengan C++, Java dan Javascript. Jadi apabila anda pernah belajar java atau javascript maka seharusnya mempelajari dart akan menjadi lebih mudah. Dart merupakan salah satu bahasa pemrograman yang sering digunakan untuk membangun sebuah aplikasi mobile.

Cara Penggunaan Dart

Peralatan dasar yang dibutuhkan saat menggunakan bahasa pemrograman dart adalah teks editor serta Dart SDK (Software Development Kit). Teks editor yang paling umum dan lengkap adalah Visual Studio Code. Jika kita menggunakan VS Code sebagai teks editor maka perlu untuk menginstal ekstensi Dart terlebih dahulu sebelum menggunakannya. Dart SDK diperlukan untuk mempermudah dalam melakukan coding dalam bahasa Dart. Di dalamnya terdapat library, compiler dan lainnya, yang dapat diinstal lewat Archive | Dart.



Dart Extension di VS Code

Syntax, Variabel dan Tipe Data

Bahasa pemrograman Dart menggunakan standar ECMA-408 sehingga memiliki sintaks yang serupa dengan bahasa C++ dan Java. Jadi, jika sudah cukup familiar dengan C++ dan Java maka akan cukup mudah mempelajari Dart kedepannya.

Dalam Dart terdapat beberapa tipe data dasar yaitu: var yang dapat menyimpan tipe data apa saja, int dan double yang dapat menyimpan tipe angka, String yang menyimpan tipe teks, dan bool yang menyimpan tipe boolean.

Input, Output

Dalam menginput data menggunakan Dart, penulisan dasar yang selalu digunakan adalah stdin.readLineSync() yang secara otomatis akan membaca input sebagai string. Lalu, jika ingin mengubah tipe data dari String ke int maka diperlukan tryParse. Namun perlu diingat, dalam versi terbaru Dart sudah terdapat null safety sehingga diperlukan statement yang menyatakan bahwa variabel yang dimaksud tidak null.

```
void main(){
    stdout.write("contoh data = ");
    var contoh = stdin.readLineSync();
    print(" tipe datanya: ${contoh.runtimeType}");
}
```

Contoh pengecekan tipe data

```
PS D:\KP\NUMERIC> dart contoh.py
contoh data = 15
 tipe datanya: String
PS D:\KP\NUMERIC>
```

```
PS D:\KP\NUMERIC> dart contoh.py
contoh data = ini kalimat
 tipe datanya: String
PS D:\KP\NUMERIC>
```

Tampilan tipe data

Setelah diubah tipe datanya menggunakan tryParse():

```
void main(){
    stdout.write("contoh data = ");
    var contoh = stdin.readLineSync();
    var ubah = int.tryParse(contoh ?? "");
    if(ubah != null) print(" tipe datanya: ${ubah.runtimeType}");
}
```

Mengubah String ke int

```
PS D:\KP\NUMERIC> dart contoh.py
contoh data = 16
 tipe datanya: int
PS D:\KP\NUMERIC>
```

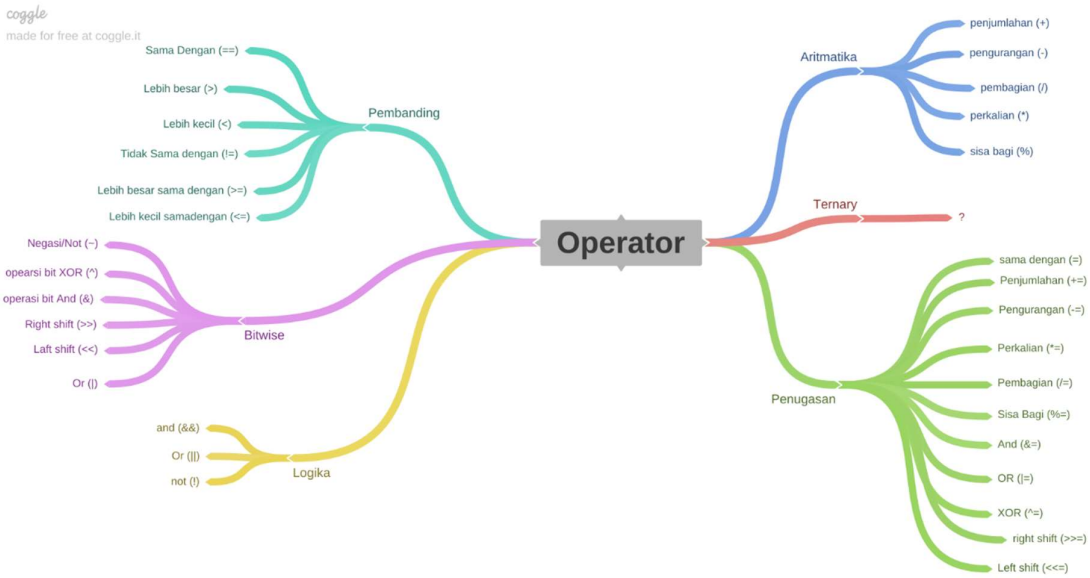
Hasil setelah diubah

Untuk output datanya, ada dua metode yaitu stdout.write() serta print(). Perbedaan dasar keduanya hanya terletak di penggunaan \n dalam mengakhiri proses. Jika menggunakan stdout.write() tidak terdapat \n, berbeda dengan print() yang sudah mencakup \n dalam penggunaannya.

stdout.write("ini kalimat\n") = print("ini kalimat")

Operator

Operator di Dart pada dasarnya serupa dengan penggunaan operator pada C++, Java.



Operator

Dalam Dart, penggunaan operator hanya dapat dilakukan pada tipe data yang sama seperti int dengan int, num dengan num, double dengan double. Sedangkan pada String, perlu diubah tipe datanya dulu agar bisa diterapkan ke operator. Perlu diingat terdapat perbedaan besar antara int dengan int?, String dengan String?, yang mana didasari pada tipe data mana saja yang nullable dan mana yang tidak.

Percabangan dan Perulangan

Percabangan pada Dart sama seperti yang terdapat pada C++, C#, dan Java. Berbagai percabangan yang dapat digunakan adalah if/ else if/ else, switch/ case. Sama halnya dengan perulangan di Dart yang juga serupa dengan perulangan di C++, C#, Java.

Berikut adalah penggunaan dasar dari percabangan dan perulangan dalam Dart.

```
void main(){
    print("================");
    var ulang = "Y";
    while(ulang == "Y"){
        print("Masukkan bilangan a dan b untuk menghitung FPB:");
        print("a = ..."); var aa = stdin.readLineSync(); var a = int.tryParse(aa ?? "");
        print("b = ..."); var bb = stdin.readLineSync(); var b = int.tryParse(bb ?? "");
        if(a == null || b == null) print("bye");
        else if(a != null && b != null){
            fpb(a,b);
            print("FPB dari ${a} dan ${b} = ${fpb(a,b)}");
        }
        print("Coba angka lain? (Y/T):...");
        var ulangg = stdin.readLineSync(); if(ulangg != null) ulang = ulangg;
    }
    print("================");
}
```

Perulangan dan Percabangan

List

Penggunaan list dalam dart memiliki aturan yang cukup kompleks sehingga perlu dideskripsikan sejelas mungkin. Ada list yang mampu menampung berapapun banyaknya data dengan menggunakan List<>.generate(1, (i) => 0) dan ada juga list yang menampung jumlah tertentu data dengan menggunakan List<>(x) dimana x adalah angka pasti banyaknya data dalam list.

Fungsi

Fungsi pada Dart sebenarnya sama dengan fungsi yang terdapat pada program C, C++, dan Java. Dengan menuliskan berbagai statement dalam satu fungsi, diakhiri dengan nilai kembalian berisi perhitungan/ penyelesaian dalam fungsi tersebut yang akan dibawa kembali ke fungsi main untuk ditampilkan. Berikut adalah contoh pengaplikasian fungsi dalam program Dart.

```
            fpb(a,b);
            print("FPB dari ${a} dan ${b} = ${fpb(a,b)}");
        }
        print("Coba angka lain? (Y/T):...");
        var ulangg = stdin.readLineSync(); if(ulangg != null) ulang = ulangg;
    }
    print("================");
}

fpb(var a, var b){
    if (a == 0)
        return b;
    return fpb(b%a, a);
}
```

Contoh Fungsi

```
PS D:\KP\NUMERIC> dart fpb.py
================
Masukkan bilangan a dan b untuk menghitung FPB:
a = ...
17
b = ...
27
FPB dari 17 dan 27 = 1
```

Hasil pemanggilan fungsi

Kelebihan Dart

Setelah membaca dan memahami lebih jauh tentang Dart, banyak kesamaan antara Dart dan beberapa bahasa pemrograman yang sudah ada. Lalu, apa kelebihan Dart dari bahasa pemrograman lainnya?

1. Dioptimalkan untuk UI
   Async-await yang lengkap untuk UI yang berisi code berbasis peristiwa, dipasangkan dengan konkurensi berbasis isolate. Kemudian, bahasa pemrograman yang dioptimalkan untuk membangun UI dengan fitur-fitur seperti operator penyebaran untuk memperluas koleksi, dan koleksi jika untuk menyesuaikan UI untuk setiap platform. Dan bahasa pemrograman yang mudah dipelajari, dengan sintaks yang familiar.

2. Perkembangan yang produktif
   Perubahan pada source-code secara berulang, menggunakan hot reload untuk langsung melihat efeknya di aplikasi yang sedang berjalan. Penulisan code menggunakan sistem tipe fleksibel dengan analisis statik yang kaya dan perkakas yang kuat dan dapat dikonfigurasi. Serta pembuatan profil, logging, dan debugging dengan teks editor pilihan.

3. Cepat di semua platform
   Menargetkan web dengan kompiler yang lengkap, matang, dan cepat untuk JavaScript. Serta menjalankan kode backend yang mendukung aplikasi, yang ditulis menggunakan satu bahasa pemrograman.

   Jika ingin mempelajari lebih jauh tentang Dart dan kegunaannya dalam memprogram suatu aplikasi mobile, ada banyak media pembelajaran yang mempermudahkan kita untuk belajar Dart.