**3. Basic Dart**

**A. Pengenalan Dart**



Dart 1.0 telah dirilis pada tanggal 14, November 2013 oleh Google dan didirikan oleh Lars Bak dan Kasper Lund. Ini bertujuan untuk membantu pengembang membangun aplikasi web dan mobile modern. Ini mencakup klien, server, dan sekarang seluler dengan Flutter. Hadir dengan berbagai alat termasuk mesin virtual, dan repositori manajemen paket, alat ini memberikan cukup amunisi bagi Anda untuk memulai proyek berikutnya.

Sebagai bahasa yang berorientasi object (object Oriented) dengan sintaksis (Syntax) C-style yang dapat diubah secara opsional menjadi JavaScript. Keunggulan yang sangat terlihat pada Dart yaitu mendukung berbagai macam alat bantu pemrograman seperti antarmuka (interface), class, collection, generics, dan opsional typing.

Alasan mengapa bahasa dart begitu cepat populer yaitu karena kita bisa menggunakan Dart untuk membuat aplikasi Web Android iOS dan juga menjalankan Server. Mudahnya saat kita sedang menggunakan dart yaitu kita dapat membuat UI(User Interface) yang indah serta berkualitas pada setiap device dengan menggunakan:

* Bahasa yang mengoptimalkan client

Dart pertama kali dioptimalkan untuk web apps dan berevolusi untuk membantu pengembangan Mobile App. Dart juga dapat kita gunakan untuk menjalankan Command Line dan Server-Side.

* Kaya akan Framework

Baik untuk pembuatan aplikasi berbasis android maupun iOS secara maksimal fungsi yang tersedia pada framework flutter dapat digunakan dengan baik di 2 device tersebut.

* Tool yang Fleksibel dan menyenangkan

Flutter adalah tool yang sangat direkomendasikan oleh dart karena pada flutter bisa digunakan untuk berbagai tujuan.

**B. Kelebihan Dart**

Beberapa kelebihan bahasa pemrograman Dart menjadikan bahasa ini menjadi salah satu bahasa baru yang dapat dipelajari oleh para developer maupun calon developer. Developers yang bekerja di Google dan perusahaan besar lainnya menggunakan dart untuk membangun aplikasi Android iOS dan Web yang berkualitas. Dart memberikan fitur yang Client Side Development (Pengembangan dari sisi client) yang oleh karena inilah banyak developer yang memilih menggunakan Dart.

1. Mudah dipelajari

* Dart memiliki banyak kemiripan dengan bahasa pemrograman yang banyak digunakan oleh developers(Java, C++, PHP , Java Script...dll).
* Kita bisa jadi tanpa disadari sudah dapat menggunakan dart karena kemiripan nya.
* Dart akan lebih mudah dipelajari jika kita sebelum nya sudah memiliki pengalaman dalam menggunakan bahasa pemrograman yang bersifat Object Oriented seperti Java ataupun C++.

1. CodeBase yang sudah di compile Natively (bawaan)

* Framework lain memberikan kita sedikit akses untuk menggunakan codingan kita pada Platform yang berbeda. Berbeda dengan Dart.
* Dart memberikan kita izin penuh untuk membuat satu aplikasi yang codingannya dapat digunakan di berbagai platform. Aplikasi yang kita buat akan dapat digunakan pada Android juga iOS.
* Dart tidak hanya dapat kita gunakan untuk mobile develop kita juga dapat menggunakan Dart untuk Web Development.

1. Produktif

* Cepat dan mudah dalam Layouting dan menambahkan Feature pada Project.
* Layout juga dapat kita buat dengan menggunakan Codingan.

1. Compile AoT dan JiT.

* Perubahan pada project dapat kita lihat secara Instan, di aplikasi.
* Tidak perlu melakukan Recompile yang memakan banyak waktu.
* Melihat perubahan tidak perlu untuk menunggu project di load.
* Anda hanya perlu untuk save dan perubahan akan terlihat.
* Ini dikarenakan Framework yang dapat meng compile Ahead of Time (AoT) / Lebih Cepat dan Just in Time(JiT) / Tepat waktu.

**C. Kenapa perlu belajar Dart?**

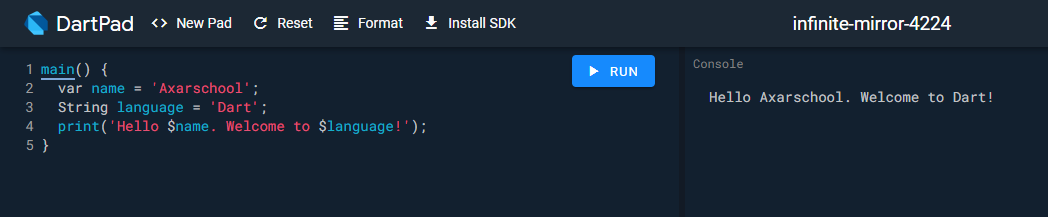
Dalam buku “***The Pragmatic Programmer***” disebutkan bahwa untuk menjadi professional software developer kita perlu belajar setidaknya satu bahasa baru setiap tahunnya. Lalu adakah alasan yang bagus untuk memulai belajar pemrograman dengan Dart?

* Pertama, ***Dart adalah bahasa pemrograman yang fleksibel***. Dart bisa berjalan di mana pun baik itu Android, iOS, maupun web. Sebagai developer, tentunya sebuah keuntungan jika bisa menuliskan kode dan bisa berjalan di mana saja.
* ***Dart adalah project open-source***. Dart dibuat oleh Google, lalu bersama dengan komunitas developer Dart mengembangkan teknologi dan fitur-fitur menarik yang bisa ditambahkan pada Dart. Jika Anda menemukan bug atau masalah pada Dart, Anda dapat melaporkannya atau bahkan memperbaikinya sendiri. Selain itu Anda tidak perlu khawatir dalam masalah lisensi ketika menggunakan bahasa Dart. Anda dapat ikut berkontribusi pada bahasa Dart pada repositori berikut: <https://github.com/dart-lang>.
* ***Dart digunakan oleh Flutter***. Sejak kemunculan Flutter, Dart kembali menjadi perhatian. Saat ini ada banyak perusahaan yang menggunakan Flutter pada aplikasinya. Flutter bisa dibilang merupakan proyek yang revolusioner dari Google untuk mengembangkan aplikasi multiplatform dengan tampilan UI yang menarik. Untuk itu, jika Anda tertarik mengembangkan aplikasi dengan Flutter, maka menguasai Dart adalah hal yang fundamental.
* ***Dart memiliki dukungan tools yang lengkap***. Hampir setiap teks editor atau IDE memiliki dukungan besar untuk Dart. Anda dapat menggunakan IDE seperti IntelliJ IDEA, Webstorm, Android Studio maupun editor sederhana seperti VS Code, Sublime text, Atom, atau yang lainnya sesuai kenyamanan Anda.
* ***Dart mudah dipelajari dan bagus sebagai first language***. Anda akan bisa memahami Dart dengan cepat khususnya jika sudah familiar dengan bahasa pemrograman populer lain seperti Java, Python, JavaScript, dll. Bahkan jika Anda baru memulai pemrograman, Dart adalah bahasa yang bagus. Anda tidak perlu menginstal apapun, cukup memanfaatkan online compiler dari Dart, Anda sudah bisa menulis dan menjalankan aplikasi Dart. Selain itu, dokumentasi dan tutorial Dart yang disediakan Google cukup mudah untuk diikuti, ditambah dengan sintaks yang sederhana, dan komunitas yang bersahabat dalam membantu kita mempelajari Dart.

**D. Karakteristik Dart**

Dart merupakan bahasa modern dan berfitur lengkap. Dart juga memiliki banyak kemiripan dengan bahasa lain yang sudah banyak dikenal seperti Java, C#, Javascript, Swift, dan Kotlin. Salah satu rancangan utama dari Dart adalah supaya bahasa ini familiar bagi developer Javascript dan Java/C#. Artinya,  yang telah familiar dengan kedua bahasa tersebut dapat memulai belajar bahasa Dart dengan lebih mudah. Namun, jika Anda adalah calon developer yang baru memulai belajar pemrograman dan memutuskan Dart sebagai first language, tenang saja. Dart adalah bahasa yang nyaman dan mudah dipelajari untuk memulai pemrograman.

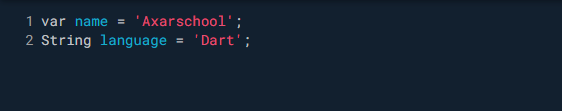
Kita ambil contoh potongan kode Dart berikut:



Bahasa pemprograman dart juga termasuk dalam Bahasa yang memiliki beberapa karakteristik, antara lain :

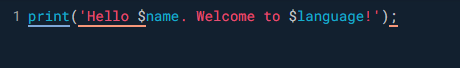
* *Type Inference*
* *String Expressions*
* *Statically Typed*
* *Multi-Paradigm : OOP & Functional.*

Bahasa pemrprograman dart merupakan Bahasa yang ***Statically Typed,*** yang berarti kita perlu mendefinisikan variabel sebelum bisa menggunakannya. Berikut contoh kode yang mendeklarasikan variabel pada Bahasa dart.



Dapat dilihat pada kode diatas bahwa dart tidak memerlukan definisi tipe data variabel secara eksplisit. Ini dikarenakan dart mendukung ***Type Inference,*** yang mana tipe data akan secara otomatis mendeteksi ketika ada variabel diinisialisasi. Seperti variabel ***name***  akan mendeteksi sebagai ***String.***

Pada Bahasa dart juga memiliki fitur ***String Interpolation***. Yang dimana fitur tesebut bisa menyisipkan variabel ke dalam sebuah objek ***String*** tanpa ***Concatenantion*** (penggabungan objek  ***String*** menggunakan **+)**. Dengan fitur tesebut, tentu dapat menjadi mempermudah dalam membuat objek ***String*** yang dinamis. Sebagai contoh dapat dilihat sebagai berikut :



**E. Instalasi Tools**

Sebelum kita mulai mencoba bahasa pemrograman Dart, Beritkut ini beberapa tools yang harus di-install. Seperti meng-install Dart SDK, Dart Tool (DartPad, IDEs, Command-line tools) dan lainnya.

**Instalasi Dart**

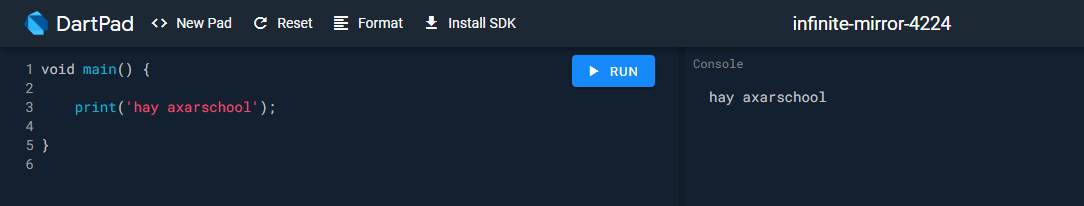
* Click Link Berikut ini Ini : https://gekorm.com/dart-windows/



* Download fully tested version
* Jalankan file yang telah didownload, tekan tombol “next” seterusnya hingga “finish”

**First Code on Dart**

* Dart menyediakan editor online di <https://dartpad.dev/?null_safety=true>. Editor Dart dapat menjalankan skrip dan menampilkan HTML serta Console Output.



* **Fungsi Main()** Setiap aplikasi terdapat point dalam programnya yang berfungsi untuk masuk ke aplikasi. Ketika sebuah aplikasi dijalankan, dimulai dari point masuk yang ditentukan itu. Di dart, point untuk masuknya adalah fungsi main ().
* **print( )** adalah fungsi yang telah ditetapkan untuk mencetak string atau nilai tertentu ke output.

Berikut ini beberapa penjelasan tentang **Dart Fundamental** yang perlu diketahui sebelum memulai Coding menggunakan Dart

**4. Dart Fundamental**

Sebelum mempelajari lebih dalam tentang Dart dan Mobile Apps, terlebih dahulu kita mempelajari hal yang paling dasar pada pemprograman, yaitu ‘**Fundamental**’. Menurut Paul Barry dan David Griffiths “Banyak orang tahu cara menggunakan komputer, tapi hanya sedikit saja yang melanjutkan untuk belajar mengendalikannya”.

Apakah belajar mengoperasikan komputer adalah sesuatu yang membanggakan? Tidak, sampai kita tahu ad acara untuk membuat komputer mengikuti semua perintah kita. Gimana caranya? Membuat program komputer atau **programming**!

**Kenapa harus belajar Fundamental Pemprograman?**

Umumnya, fundamental pemprograman yaitu antara Bahasa pemprograman yang satu dengan Bahasa pemprograman yang lain adalah sama. Perbedaan terletak pada bentuk syntax pada Bahasa pemprograman tersebut

Kita tidak perlu rakus belajar semua Bahasa pemprograman, karena pasti bakal ketemu konsep-konsep yang itu-itu saja. Makanya orang-orang yang belajar pemprograman secara otodidak bakal sering kali mudah megikuti tutorial yang ada, tapi kerepotan ketika harus membuat variasi program yang lain.

Itu karena kebanyakan tutorial memberikan contoh kasus yang sifatnya umum dan kurang tepat guna. Ya, walaupun juga banyak tutorial yang sudah bagus dan layak untuk diikuti. Sehingga bagi kita yang baru belajar (apalagi otodidak) butuh bertahun-tahun untuk mengerti tentang Fundamental Pemprograman ini.

Tapi, dengan memahami tentang Fundamental Pemprograman ini, kita akan lebih mudah untuk belajar hal-hal baru terkait pemprograman dalam Bahasa pemprograman apa saja. Menarik kan? Ya, karena Fundamental Pemprograman bersifat umum dan tidak terikat dengan Bahasa pemprogramannya. Jadi, kita bisa gunakan Java, PHP, Javascript, Switft atau bahkan Python.

**Apakah setelah belajar Fundamental Pemprograman sudah bisa membuat program?**

Hmmmm, ketawa batin.

Mempelajari Fundamental Pemprograman **tidak sama** dengan membuat program. Sama halnya seperti mempelajari cara menggunakan kamera digital **tidak sama** dengan membuat film. Sama halnya seperti mempelajari tentang cara menggunakan gergaji **tidak sama** dengan membuat rumah.

Tapi, ketika kita sudah bisa membuat rumah nantinya, kita bisa bekreasi sesuka hati ketika ingin membuat rangka atap rumah kita, karena sebelumnya kita belajar teknik memotong kayu yang benar menggunakan gergaji. Dan, **kita tidak harus** mengikuti gaya rangka atap pada umumnya. Sampai disini paham, kan?

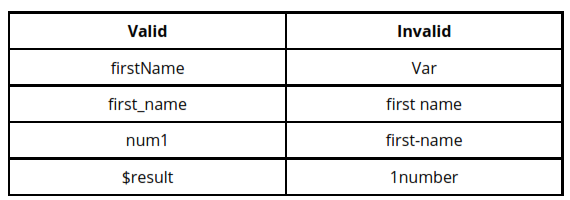
**Apa saja yang dipelajari dalam Fundamental Pemprograman?**

Pada Fundamental Dart akan mempelajari seperti syntax sederhana identifier, comment, Data Type dan sebagainya. Kita mulai dahulu dengan identifier pada Dart.

**Identifier pada Dart**

Pengidentifikasi (identifier) adalah nama yang diberikan kepada elemendalam program seperti variabel, fungsi, dll. Rules (Aturan) untuk identifier adalah

* + Identifier dapat menyertakan karakter, dan digit(angka). Namun, identifier tidak dapat dimulai dengan digit.
  + Identifier tidak dapat berisikan simbol-simbol spesial selain underscore (\_) dan tanda dolar ( $ ).
  + Identifier tidak dapat berisikan Keyword \*
  + Identifier harus unik - Identifier tidak dapat berisikan spasi(space)
  + Berikut beberapa contoh identifier yang valid(dapat digunakan) dan invalid identifier (yang tidak dapat digunakan).



**Whitespace dan Line Breaks**

Dart mengabaikan spasi, tab, dan baris baru yang muncul dalam program. Kita dapat menggunakan spasi, tab, dan baris baru secara bebas dalam programkita dan kita bebas untuk memformat dan memasukkan program kita dengancara yang rapi dan konsisten yang membuat kode mudah dibaca dan dipahami.

**Dart adalah case-sensitive**

Ini berarti bahwa Dart membedakan antara huruf besar dan huruf kecil. Kesalahan penggunaan huruf besar dan huruf kecil sering terjadi pada pembuatan identifier dan class. Untuk itu harap lebih diperhatikan lagi dalam penamaan pada Dart!.

**Pernyataan (statement) diakhiri dengan Semicolon**

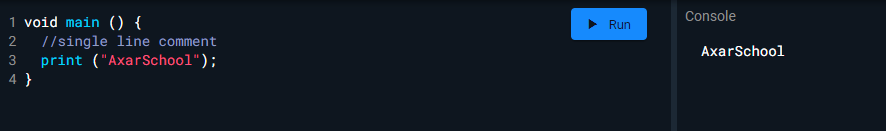
Setiap baris instruksi disebut pernyataan. Setiap pernyataan Dart harus diakhiri dengan titik koma (;). Satu baris dapat berisi beberapa pernyataan. Namun, pernyataan ini harus dipisahkan dengan titik koma.

**Comment di Dart**

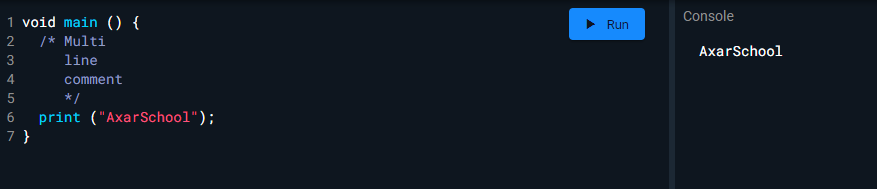
Komentar (Comment) adalah cara untuk meningkatkan keterbacaan suatu program. Comment dapat digunakan untuk memasukkan informasi tambahan tentang program seperti penulis kode, petunjuk tentang fungsi / konstruksi dll. Comment akan diabaikan oleh compiler. Dart dapat berisikan jenis komentar berikut

* **Single Line Comment ( // )** ⇒ Teks apapun yang berada setelah " // " dalam satu line diperlakukan sebagai komentar
* **Multi Line Comments ( /\* \*/ )** ⇒ Teks apapun yang berada diantara “/\* \*/” diperlakukan sebagai komentar, Komentar ini dapat mencakup banyak line.

**Single Line Comment**



**Multi Line Comment**



**Syntax pada Dart**

Syntax mendefinisikan seperangkat aturan untuk menulis program. Setiap spesifikasi bahasa mendefinisikan sintaksnya sendiri. Sintaks Dart terdiri dari:

* Variable
* Operators
* Class
* Function
* Loop
* Comments
* Library
* Package
* Typedefs
* Decision making
* Expression dan Programming Construct (konstruksi pemrograman)
* Data Structure yang direpresentasikan sebagai Generics/Collections

**Tipe Data Pada Dart**

Salah satu karakteristik paling mendasar dari bahasa pemrograman adalah himpunan tipe data yang dimiliki oleh bahasa pemrograman. Ini adalah jenis nilai yang dapat direpresentasikan dan dimanipulasi dalam bahasa pemrograman. Dart memiliki tipe-tipe data berikut :

* Numbers
* Strings
* Booleans
* Lists
* Maps
* **Numbers**

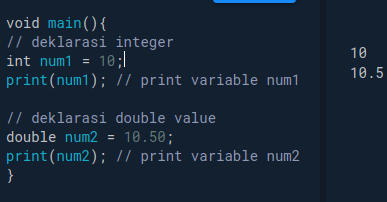
Numbers (Angka) dalam Dart digunakan untuk mewakili literal numerik. Number pada Dart memiliki dua tipe :

* **Integer** sebuah tipe data dari Number yang nilainya berbentuk bilangan bulat. Nilainya harus bersifat angka dan tidak boleh ada koma.
* **Double** digunakan untuk merepresentasikan angka float / decimal.

Sintaks untuk menyatakan number adalah seperti yang diberikan di bawah ini :

int nama\_var; // mendeklarasikan variable integer

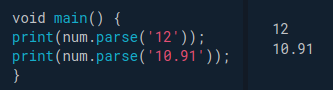
double nama\_var; // mendeklarasikan variable variable



*NOTE: Dart akan throw exception (memberikan error) jika nilai pecahan diletakkan pada variabel integer.*

**Parsing Number**

Fungsi statis parse() memungkinkan penguraian string yang berisi numerik literal menjadi angka. contoh berikut menunjukkan hal yang sama :



**Property dari Number**

Tabel berikut mencantumkan properti yang dimiliki oleh Number pada Dart

|  |  |  |
| --- | --- | --- |
| **Property** | **Deskripsi** | **Contoh** |
| **hashcode** | Me-return kode hash untuk  nilai numerik | *void main() {*  *int n = 5000;*  *print(n.hashCode);*  *}* |
| **isFinite** | **true**, jika jumlahnya terbatas.  **false**, jika jumlahnya tidak terbatas. | *void main() { int n = 5000;*  *print(n.isFinite);}* |
| **isInfinite** | **true**, jika jumlahnya adalah infinity (tidak terbatas) positif atau tidak terbatasnya negatif.  **false**, jika tidak infinity. | *void main() { int n = 5000;*  *print(n.isInfinite);}* |
| **isNegative** | **true**, jika jumlahnya negatif.  **false**, jika sebaliknya. | *void main() { int posNum = 10;*  *int posNeg = -10;*  *print(posNum.isNegative);*  *print(posNeg.isNegative); }* |
| **sign** | Return minus satu, nol atau plus satu tergantung pada tanda dan nilai numerik dari angka tersebut. | *void main() { int posNum = 10;*  *int posNeg = -12; int valZero = 0;*  *print(posNum.sign); print(posNeg.sign);*  *print(valZero.sign); }* |
| **isEven** | Me-return nilai true jika nomornya adalah bilangan genap. | *void main() {int posNum = 10; print(posNum.isEven); }* |
| **isOdd** | Me-return nilai true jika nomor tersebut adalah angka ganjil. | *void main() { int posNum = 10;*  *print(posNum.isOdd);}* |

**Method dari Number**

Dibawah ini adalah List Method yang dapat kita gunakan pada Numbers

|  |  |  |
| --- | --- | --- |
| **Method** | **Deskripsi** | **Contoh** |
| **abs** | Return nilai absolute dari sebuah number. | *void main() {*  *var a = -2;*  *print(a.abs());* *}* |
| **ceil** | Return bilangan bulat terkecil yang tidak lebih kecil dari angka. | *void main() {*  *var a = 2.4;*  *print(“ceiling value dari 2.4 = + ${a.ceil()}”);* *}* |
| **compare** | Membandingkan number ini dengan  number lain. | *void main() {*  *var a = 2.4;*  *print(a.compareTo(12));*  *print(a.compareTo(2.4));*  *print(a.compareTo(0));* *}* |
| **Floor** | Return bilangan bulat terbesar tidak lebih besar dari angka saat ini. | *void main() {*  *var a = 2.9;*  *print(“floor value 2.9 =*  *${a.floor()}”);* *}* |
| **remainder** | Return sisa terpotong setelah  membagi dua angka. | *void main() {*  *var a = 10;*  *var b = 17;*  *print(a.remainder(2));*  *print(b.remainder(2));* *}* |
| **Round** | Return integer terdekat dari number  saat ini. | *void main() {*  *double n1 = 12.023;*  *double n2 =*  *12.89;*  *var value = n1.round();*  *print(value);*  *value = n2.round();*  *print(value);* *}* |
| **toDouble** | Return double yang setara dengan number saat ini. | *void main() {*  *int n1 = 2;*  *var value =*  *n1.toDouble();*  *print(“Output = ${value}”);* *}* |
| **toInt** | Return int yang setara dengan number  saat ini. | *void main() {*  *int n1 = 2.0;*  *var value = n1.toInt();*  *print(“Output = ${value}”);* *}* |
| **toString** | Return string yang setara dengan  number saat ini. | *void main() {*  *int n1 = 2;*  *var value = n1.toString();*  *print(value is String);* *}* |
| **Truncate** | Return integer setelah membuang digit  pecahan apapun | *void main() {*  *double n1 = 2.123;*  *var value =*  *n1.truncate();*  *print(“truncate value dari 2.123 =*  *${value}”);* *}* |

**Strings**

Tipe data String mewakili urutan karakter. String Dart adalah urutan unit kode UTF 16 1 . Nilai string di Dart dapat direpresentasikan menggunakan kutip tunggal (‘ ’) atau ganda ( “ ” ) atau tiga (‘’’ ‘’’). String baris tunggal direpresentasikan menggunakan tanda kutip tunggal atau ganda. Kutipan rangkap(kutip tiga) digunakan untuk mewakili string multi-line.

//kutip satu dan dua untuk string satu line (tunggal)

String variable\_name = 'value';

String variable\_name = ''value'';

//kutip rangkap (tiga) untuk string multi-line

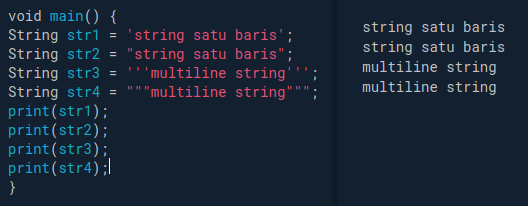
String variable\_name = '''line1

line2''';

String variable\_name= ''''''line1

Line2'''''';

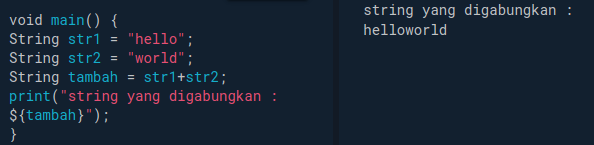
**Contoh**



**String Interpolation (interpolasi)**

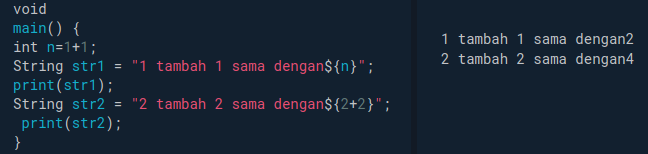
Proses membuat string baru dengan menambahkan nilai ke string statis disebut sebagai penggabungan atau interpolasi. Dengan kata lain, itu adalah proses menambahkan string ke string lain.

Operator plus ( + ) adalah mekanisme yang biasa digunakan untuk menyatukan / menyisipkan string.



**Contoh 2**

Kita dapat menggunakan " $ { } ", dapat digunakan untuk menginterpolasi nilai ekspresi Dart dalam string. Contoh berikut menggambarkan hal yang sama.



**String Properties**

Properti string pada tabel berikut hanya dapat dibaca(read-only properties).

|  |  |  |
| --- | --- | --- |
| **Properti** | **Deskripsinya** | **Contoh** |
| **codeUnits** | Me-return daftar unit kode UTF-16 yang tidak dapat dimodifikasi dari  string ini. | *void main() {*  *String str = “Hello”;*  *print(str.codeUnits); }* |
| **isEmpty** | Return true jika string tidak memiliki value atau empty (kosong). | *void main() {*  *String str = “Hello”;*  *print(str.isEmpty); }* |
| **Length** | Return panjang (jumlah karakter) pada string termasuk spasi, tab dan karakter  baris baru (enter). | *void main() {*  *String str = “Hello All”;*  *print(“Length dari string*  *adalah ${str.length}”); }* |

**5. Perkenalan Flutter**



**A. Sejarah Flutter**

Flutter adalah framework untuk pengembangan aplikasi mobile open source yang dikembangkan oleh Google, versi pertama dari flutter dinamai dengan codename Sky dan berjalan pada sistem operasi Android, lalu pada Desember 2018 flutter stabil versi 1 dirilis.

**B. Kelebihan Flutter**

memiliki beberapa kelebihan yaitu :

1. **Hot Reload** merupakan sebuah fitur ketika developer mengembangkan sebuah aplikasi, lalu melakukan perubahan pada kode program, program akan otomatis mengupdate perubahan, maka dengan adanya hot reload kita tidak perlu melakukan build ulang, cukup save lalu akan otomatis mengupdate sesuai dengan apa yang telah kita ubah, sehingga sangat mempercepat proses pengembangan program.
2. **Multiplatform** artinya kode yang telah kita buat maka akan bisa dijalankan baik di ios, android, web, maupun desktop tanpa perlu melakukan porting secara manual.
3. **UI yang Cantik** UI flutter telah didesain dengan sangat baik, juga untuk menggunakannya sangat mudah sekali.

**C. Instalasi**

Untuk menggunakan flutter, maka ada 2 IDE yang direkomendasikan , yaitu Visual Studio Code dan Android Studio, untuk mendapatkannya download melalui link berikut :

1. Android Studio : <https://developer.android.com/studio>

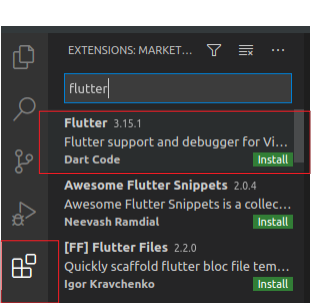
2. Visual Studio Code : <https://code.visualstudio.com/>

Setelah menginstall IDE maka diperlukan Flutter SDK, untuk mendapatkannya bisa didownload melalui link <https://flutter.dev/docs/get-started/install>

Setelah itu install plugin Flutter :

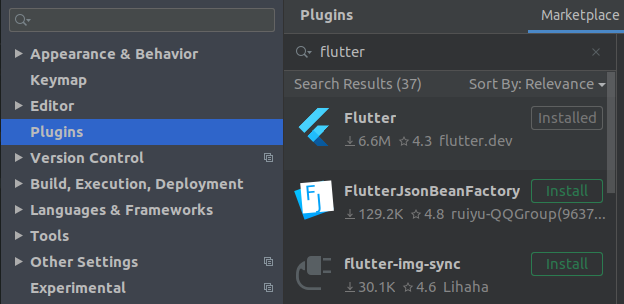
**1. Visual Studio Code :**

Install plugin melalui menu extension, lalu search Flutter dan install



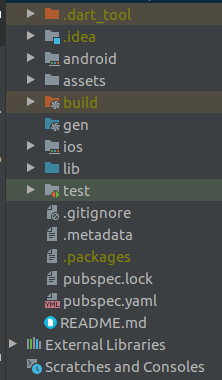
**2. Android Studio**

Buka Settings > Plugins > Marketplace, lalu search Flutter dan install



**D. Hal yang Perlu diketahui**

Hal yang perlu diketahui ketika menggunakan flutter yaitu :



1. **folder lib** pada folder lib inilah kode kita akan dibuat, dan biasanya di folder ini bisa berisi beberapa folder agar file kodingan kita bisa lebih rapi.
2. **pubspec.yaml** pada flutter semua gambar atau assets yang ditambahkan ke project maka dia wajib didaftarkan di file pubspec.yaml, file ini juga berfungsi apabila kita ingin menambah dependensi pada project kita, maka kita perlu menambahkannya disini lalu setelah itu kita tinggal menjalankan perintah flutter pub get pada terminal.
3. **build.gradle** pada umumnya fungsinya untuk menambah dependensi seperti pada project android.
4. **Info.plist** Pada file ini berisi konfigurasi permission untuk ios.

**E. Hot Reload**

Hot reload flutter mempunyai fitur pengembangan cepat / hot reload, dimana saat kita memperbarui kode kita tidak perlu membuild aplikasinya. Sama halnya saat kita koding di web, hanya save dan lihat hasilnya, Untuk melakukan build project flutter, kita dapat menggunakan sistem operasi apapun seperti Windows, Linux, dan juga MacOS. Namun apabila kita ingin build project untuk iOS, maka diperlukan perangkat berupa MacBook, dan menggunakan XCode sebagai emulatornya.