**Jobsheet 2 – Pengantar Bahasa Pemrograman Dart**

Disusun untuk memenuhi Tugas Mata Kuliah Praktikum Pemrograman Mobile

Dosen Pembimbing : Ade Ismail, S.Kom., M.TI.



Disusun oleh

Liya Novitasari 2241760006

Kelas 2E

|  |
| --- |
|  |
|  |

**Jurusan Teknologi Informasi**

**Program Studi D-4 Sistem Informasi Bisnis**

**POLITEKNIK NEGERI MALANG**

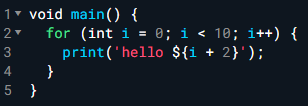
**MALANG**

**2024**

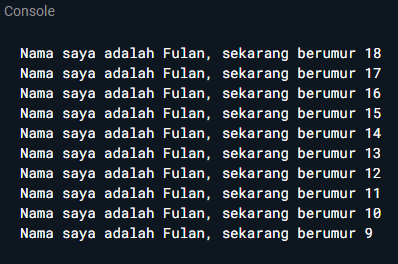
**Tugas Praktikum**

**Soal 1**

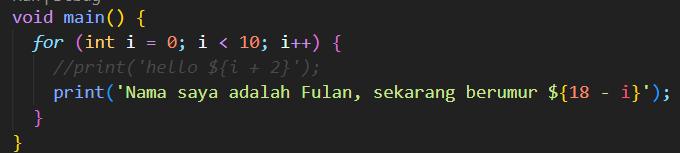
Modifikasilah kode pada baris 3 di VS Code atau Editor Code favorit Anda berikut ini agar mendapatkan keluaran (*output*) sesuai yang diminta!

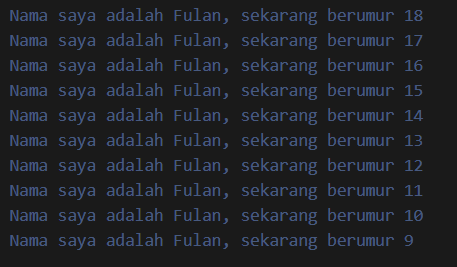


Output yang diminta:



**Jawab:**





**Soal 2**

Mengapa sangat penting untuk memahami bahasa pemrograman Dart sebelum kita menggunakan framework Flutter ? Jelaskan!

**Jawab:**

Karena Bahasa Dart adalah inti dari framework Flutter. Kerangka kerja modern seperti Flutter membutuhkan bahasa modern tingkat tinggi agar bisa memberikan pengalaman terbaik kepada pengembang, serta memungkinkan untuk membuat aplikasi seluler yang luar biasa. Memahami Dart adalah dasar untuk bekerja dengan Flutter; pengembang perlu mengetahui asal-usul bahasa Dart, bagaimana komunitas mengerjakannya, kelebihannya, dan mengapa itu adalah bahasa pemrograman yang dipilih untuk Flutter.

**Soal 3**

Rangkumlah materi dari codelab ini menjadi poin-poin penting yang dapat Anda gunakan untuk membantu proses pengembangan aplikasi mobile menggunakan framework Flutter.

**Jawab:**

1. **Geeting started with Dart**

Dart bertujuan untuk menggabungkan kelebihan-kelebihan dari sebagian besar bahasa tingkat tinggi dengan fitur-fitur bahasa pemrograman terkini, antara lain sebagai berikut:

* **Productive tooling**: fitur kakas (*tool*) untuk menganalisis kode, plugin IDE, dan ekosistem paket yang besar.
* **Garbage collection**: untuk mengelola atau menangani dealokasi memori (terutama memori yang ditempati oleh objek yang tidak lagi digunakan).
* **Type annotations (opsional)**: untuk keamanan dan konsistensi dalam mengontrol semua data dalam aplikasi.
* **Statically typed**: Fitur ini penting untuk menemukan *bug*selama kompilasi kode.
* **Portability**: bahasa Dart tidak hanya untuk web (yang dapat diterjemahkan ke JavaScript) tetapi juga dapat dikompilasi secara *native* ke kode**Advanced RISC Machines (ARM)** dan x86.

1. **The evolution of Dart**

* Awalnya berfokus pada pengembangan web, dengan tujuan utama menggantikan JavaScript, sekarang telah fokus pada mobile development, termasuk framework Flutter.
* Mencoba memecahkan masalah pada JavaScript
* Menawarkan performa terbaik dan alat yang lebih baik untuk proyek berskala besar
* Dibentuk agar kuat dan fleksibel

1. **How Dart works**

Untuk memahami dari mana fleksibilitas bahasa dart itu berasal, kita perlu tahu bagaimana cara mengeksekusi kode Dart. Ini dapat dilakukan dengan dua cara, seperti yang dijelaskan berikut ini:

* Dart virtual machines (VMs)
* JavaScript compilations

**Dart VM and JavaScript compilation**

Kode Dart dapat dieksekusi pada lingkungan yang mendukung bahasa Dart. Lingkungan yang mendukung bahasa Dart perlu memperhatikan fitur-fitur penting seperti berikut:

* *Runtime systems*
* *Dart core libraries*
* *Garbage collectors*

Eksekusi kode Dart dapat beroperasi dalam dua mode — kompilasi Just-In-Time (JIT) atau Kompilasi Ahead-Of-Time (AOT). Dijelaskan secara lebih rinci sebagai berikut:

* Kompilasi JIT adalah tempat kode sumber dikompilasi sesuai kebutuhan—*Just in time*. Dart VM memuat dan mengkompilasi kode sumber ke kode mesin asli (*native*). Pendekatan ini digunakan untuk menjalankan kode pada *command line* atau selama proses pengembangan aplikasi mobile yang dapat memanfaatkan fitur seperti debugging dan *hot reload*.
* Kompilasi AOT adalah dimana Dart VM dan kode Anda dikompilasi sebelumnya, VM bekerja lebih seperti sistem runtime Dart, yang menyediakan *garbage collector* dan metode-metode *native* dari Dart software development kit (SDK) pada aplikasi. Pendekatan ini memiliki keuntungan performa yang sangat besar dibandingkan kompilasi JIT, tetapi fitur lain seperti debugging dan hot reload tidak tersedia.

1. **Introducing the structure of the Dart language**
   * **Object orientation**

Dart dirancang dengan prinsip **Object-Oriented Programming (OOP)**. OOP berbasis konsep objek yang menyimpan data (fields) dan kode (methods). Objek-objek ini dibuat dari class, yang berfungsi sebagai cetak biru untuk mendefinisikan fields dan methods. Sesuai prinsip OO memastikan bahwa Dart memiliki fitur encapsulation, inheritance, composition, abstraction, dan polymorphism.

* + **Dart operators**

Di Dart, operator adalah method yang didefinisikan dalam class dengan sintaks khusus. Misalnya, saat menggunakan x == y, sebenarnya Dart memanggil method x.==(y) untuk perbandingan kesetaraan. Tidak seperti Java yang memiliki tipe data primitif, semua tipe data di Dart merupakan objek dari suatu class, sehingga memiliki method. Ini memungkinkan operator diubah atau di-override sesuai logika yang diinginkan.

* + **Arithmetic operators**

Dart hadir dengan banyak operator typical yang bekerja seperti banyak bahasa pemrograman lainnya; yaitu:

* + untuk tambahan.
* - untuk pengurangan.
* \* untuk perkalian.
* / untuk pembagian.
* ~/ untuk pembagian bilangan bulat. Di Dart, setiap pembagian sederhana dengan / menghasilkan nilai double. Untuk mendapatkan nilai bilangan bulat, Anda perlu membuat semacam transformasi (yaitu, typecast) dalam bahasa pemrograman lain; namun Dart sudah mendukung untuk operasi ini.
* % untuk operasi modulus (sisa bagi dari bilangan bulat).
* -expression untuk negasi (yang membalikkan suatu nilai).

Beberapa operator memiliki perilaku yang berbeda tergantung pada jenis operan di sisi kiri; Misalnya, operator + dapat digunakan untuk menjumlahkan variabel dari tipe num, tetapi juga dapat digunakan untuk menggabungkan string. Karena method yang dirujuk diimplementasikan secara berbeda pada kelas yang berbeda.

Dart juga menyediakan shortcut operator untuk menggabungkan variabel setelah operasi lainnya. Operator aritmatika atau shortcut operator adalah +=, -=, \*=, /=, dan ~/=.

* + **Increment and decrement operators**

Operator penambahan dan pengurangan juga merupakan operator umum dan diimplementasikan pada angka, sebagai berikut:

* ++var atau var++ untuk menambah nilai variabel var sebesar 1
* --var atau var-- untuk mengurangi nilai variabel var sebesar 1

Operator Dart increment dan decrement berperilaku mirip dengan bahasa lain. Penerapan operator increment dan decrement sangat baik untuk operasi perhitungan pada perulangan.

* + **Equality and relational operators**

Persamaan operator Dart dijelaskan sebagai berikut:

* == untuk memeriksa apakah operan sama
* != untuk memeriksa apakah operan berbeda

Untuk melakukan pengujian relasional, maka gunakan operator sebagai berikut:

* > memeriksa apakah operan kiri lebih besar dari operan kanan
* < memeriksa apakah operan kiri lebih kecil dari operan kanan
* >= memeriksa apakah operan kiri lebih besar dari atau sama dengan operan kanan
* <= memeriksa apakah operan kiri kurang dari atau sama dengan operan kanan

Di Dart, tidak seperti Java dan bahasa lainnya, operator == tidak membandingkan referensi/alamat memori melainkan isi dari variabel tersebut.

Juga, tidak seperti JavaScript, tidak ada operator === yang diperlukan pada Dart karena telah memiliki fitur type safety yang memastikan bahwa operator persamaan == hanya digunakan pada objek dengan tipe yang sama.

* + **Logical operators**

Operator logika di Dart adalah operator yang diterapkan pada operan bool; bisa berupa variabel, ekspresi, atau kondisi. Selain itu, dapat dikombinasikan dengan ekspresi kompleks dengan menggabungkan nilai ekspresi yang dievaluasi. Operator logika yang disediakan adalah sebagai berikut:

* !expression negasi atau kebalikan hasil ekspresi—yaitu, true menjadi false dan false menjadi true.
* || menerapkan operasi logika OR antara dua ekspresi.
* && menerapkan operasi logika AND antara dua ekspresi.

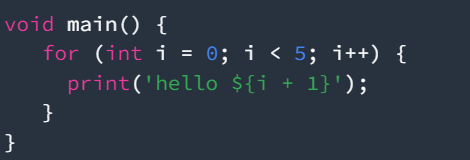
1. **Hands-on with Dart**

**DartPad**

Cara termudah untuk memulai pemrograman Dart adalah dengan menggunakan kakas DartPad. Ini adalah kakas online yang bagus untuk belajar dan bereksperimen dengan fitur bahasa Dart.

Kakas ini mendukung *core library* Dart, kecuali untuk *library VM* seperti dart:io.

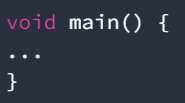
**Hello world Dart style**



**Main function**

Dart menggunakan *function*dan *method* sebagai cara untuk memisahkan kode. Sebuah *function*atau *method*adalah potongan kode yang menerima beberapa data, eksekusi kode, dan kemudian mengembalikan beberapa data (*return*).

*Function* dari contoh hello world terlihat seperti ini:



Baris pertama memiliki beberapa informasi penting, yang dijelaskan sebagai berikut:

* Tipe data yang dikembalikan dari *method* perlu didefinisikan terlebih dahulu. Pada kasus ini, void menunjukkan bahwa *method* tidak mengembalikan data apa pun ketika telah selesai eksekusi. void adalah *keyword* dalam bahasa Dart yang hanya dapat digunakan secara spesifik. Kita akan mempelajari tipe data pada pertemuan berikutnya.
* Berikutnya adalah nama *function* —dalam hal ini, main. Nama tersebut digunakan oleh kode lain untuk merujuk pada nama *method* ini. Dalam kasus ini, main adalah nama *function* utama yang dicari oleh Dart VM saat pertama kali mengeksekusi kode. Setiap aplikasi Dart harus memiliki *function*main sehingga Dart VM tahu di mana harus memulai eksekusi kode.
* Tanda kurung kosong ( ) adalah tempat *function*untuk mendefinisikan data yang akan diterima. *Function*main ini tidak menerima data apa pun, oleh karena itu tanda kurung ini kosong. Kita akan lihat cara sebuah *function* dapat menerima data sebagai parameter pada pertemuan berikutnya.
* Terakhir, kurung kurawal { } di akhir baris pertama menentukan di mana kode *function*main dimulai, dan kurung kurawal penutup setelah beberapa baris kemudian untuk menentukan di mana kode *function*main berakhir. Berbeda dengan beberapa bahasa seperti Python, dia tidak menggunakan kurung kurawal.

**Soal 4**

Buatlah slide yang berisi penjelasan dan contoh eksekusi kode tentang perbedaan Null Safety dan Late variabel ! **(Khusus soal ini kelompok berupa link google slide)**

Kumpulkan jawaban Anda di spreadsheet pada tautan yang telah disediakan di grup telegram. Untuk soal nomor 1 sampai 3 push repo GitHub Anda.