**TUGAS PRAKTIKUM MATA KULIAH**

**PEMROGRAMAN PERANGKAT BERGERAK**

****

Disusun oleh :

Nama : Legar Suryantara

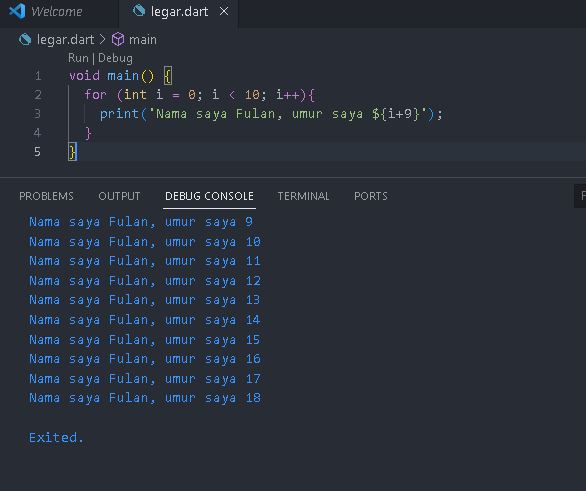
NIM : 362358302043

Kelas : 1D TRPL

**POLITEKNIK NEGERI BANYUWANGI**

**JURUSAN BISNIS DAN INFORMATIKA**

**TUGAS PRAKTIKUM SECTION 2**

1. Modifikasilah kode pada baris 3 di VS Code atau Editor Code favorit Anda berikut ini agar mendapatkan keluaran (output) sesuai yang diminta!
2. Mengapa sangat penting untuk memahami bahasa pemrograman Dart sebelum kita menggunakan framework Flutter ? Jelaskan!

**Jawab:** Memahami bahasa pemrograman Dart sangat penting sebelum menggunakan Flutter karena Dart adalah bahasa inti yang digunakan oleh Flutter. Dengan menguasai Dart, Anda dapat memahami konsep pemrograman berorientasi objek yang mendasari banyak fitur Flutter, menulis kode yang lebih efisien, dan memecahkan masalah dengan lebih mudah. Selain itu, Dart memiliki fitur-fitur seperti garbage collection dan type annotations yang meningkatkan produktivitas pengembang. Dengan dasar yang kuat dalam Dart, Anda akan lebih siap untuk memanfaatkan Flutter secara optimal dalam mengembangkan aplikasi berkualitas tinggi.

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

**Jawab:**

* 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.
* Dart bertujuan untuk menggabungkan kelebihan-kelebihan dari sebagian besar bahasa tingkat tinggi dengan fitur-fitur bahasa pemrograman terkini, antara lain sebagai berikut:
* Productive tooling: merupakan 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: Meskipun type annotations bersifat opsional, Dart tetap aman karena menggunakan fitur type-safe dan type inference untuk menganalisis types saat runtime. 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.
* Diluncurkan pada tahun 2011, Dart telah berkembang sejak saat itu. Dart merilis versi stabilnya pada tahun 2013, dengan perubahan besar termasuk dalam rilis Dart 2.0 menjelang akhir 2018, yang dapat diuraikan sebagai berikut:
* Awalnya berfokus pada pengembangan web, dengan tujuan utama menggantikan JavaScript, sekarang telah fokus pada mobile development, termasuk framework Flutter.
* Mencoba memecahkan masalah pada JavaScript: JavaScript tidak menyediakan ketahanan seperti banyak bahasa pemrograman lainnya, sehingga Dart ingin menjadi penerus daripada JavaScript.
* Menawarkan performa terbaik dan alat yang lebih baik untuk proyek berskala besar: Dart memiliki perkakas yang modern dan stabil yang telah disediakan oleh plugin IDE. Hal ini telah dirancang untuk mendapatkan performa terbaik dengan tetap menjaga nuansa bahasa yang dinamis.
* Dibentuk agar kuat dan fleksibel: Dengan tetap mempertahankan type annotations bersifat opsional dan menambahkan fitur OOP, Dart dapat menyeimbangkan dua fitur utama yaitu fleksibilitas dan ketangguhan.
* 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.
* Object orientation : Seperti kebanyakan bahasa modern, Dart dirancang untuk object-oriented (OO). Secara singkat, Bahasa OOP didasarkan pada konsep objek yang menyimpan kedua data (disebut fields) dan kode (disebut methods). Objek-objek ini dibuat dari cetak biru yang disebut class yang mendefinisikan field dan method yang akan dimiliki oleh sebuah objek.
* Dart Operators : Di Dart, operator tidak lebih dari method yang didefinisikan dalam class dengan sintaks khusus. Jadi, ketika Anda menggunakan operator seperti x == y, seolah-olah Anda sedang memanggil x.==(y) metode untuk melakukan perbandingan kesetaraan.
* Arithmetic operators : Dart hadir dengan banyak operator typical yang bekerja seperti banyak bahasa pemrograman lainnya; yaitu sebagai berikut: + 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).
* 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
* Equilty and relational operator : 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:

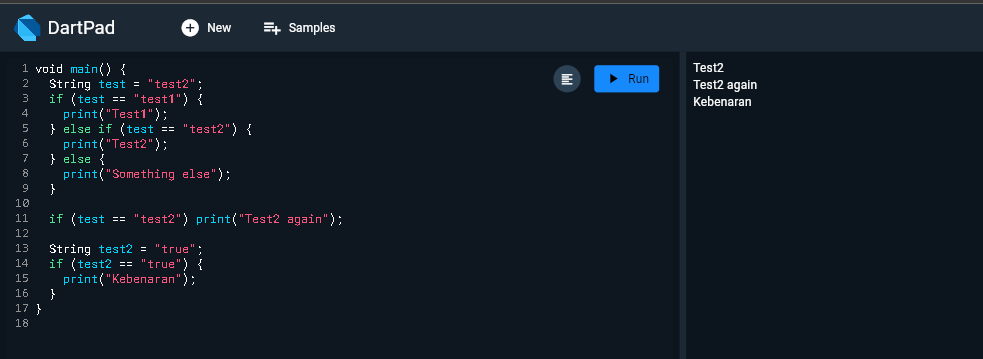
* > memeriksa apakah operan kiri lebih besar dari operan kanan
* < memeriksa apakah operan kiri lebih kecil dari operan kanan
* >= memeriksa apakah operan kiri lebih besar dari atau sama dengan operan kanan
* <= memeriksa apakah operan kiri kurang dari atau sama dengan operan kanan
* Logical operators : 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 logic OR antara dua ekspresi.
* && menerapkan operasi logika AND antara dua ekspresi.

**TUGAS PRAKTIKUM SECTION 3**

* 1. Silakan selesaikan Praktikum 1 sampai 3, lalu dokumentasikan berupa screenshot hasil pekerjaan beserta penjelasannya!

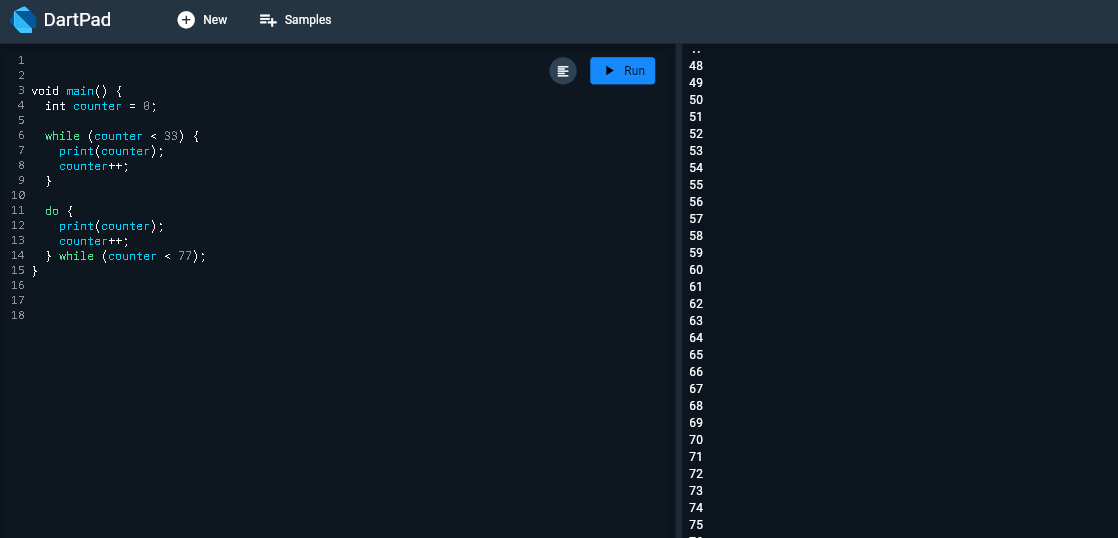
**Jawab:**

Praktikum 1

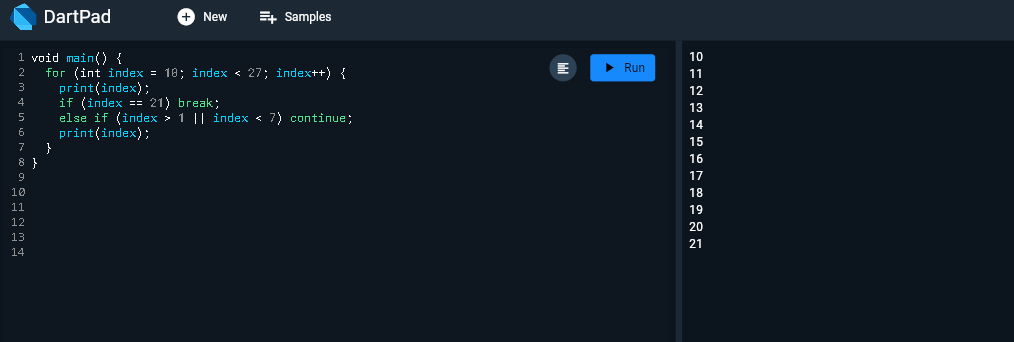


Penjelasan : Pada kode di atas semula terjadi error pada bagian penulisan else If dan Else yang seharusnya penulisan yang benar adalah else if dan else, kenapa harus menggunakan lowercase?karena dibeberapa Bahasa pemrograman khususnya dart memili case-sensitive.

Praktikum 2



Penjelasan : Pada kode di atas semula terdapat error dikarenakan variable counter belum diinisialisasi. Dalam kode ini, variabel counter diinisialisasi dengan nilai 0. Loop while akan berjalan selama counter kurang dari 33, dan setelah itu, loop do-while akan berjalan selama counter kurang dari 77.



Penjelasan :  Pada Kode di atas semula terdapat error dikarenakan ariabel Index dan index digunakan secara bergantian sehingga tidak konsisten. Bahasa Dart peka terhadap huruf besar dan kecil (case-sensitive), Pada loop for, variabel index tidak diinkrementasi, Penulisan If dan Else If harus menggunakan huruf kecil (if dan else if), dan Pernyataan if dan else if harus berada di dalam blok loop for.

* 1. Buatlah sebuah program yang dapat menampilkan bilangan prima dari angka 0 sampai 201 menggunakan Dart. Ketika bilangan prima ditemukan, maka tampilkan nama lengkap dan NIM Anda.

**Jawab :**

