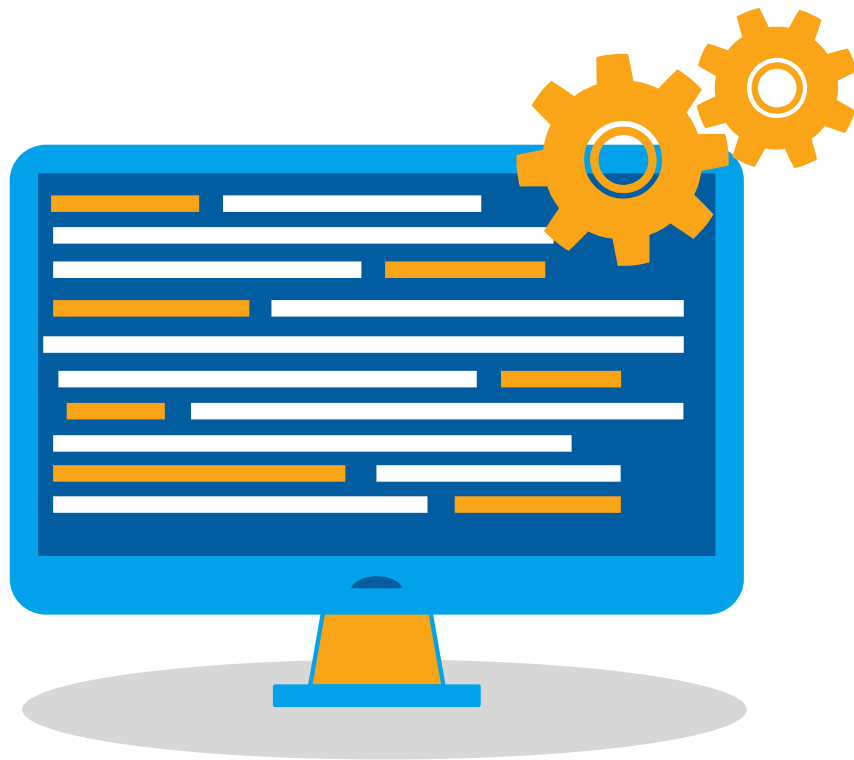


FLUTTER DAN FLUTLAB



DAFTAR ISI

Bab I Pengenalan Flutter	
A. Definisi Flutter	1
B. Dart	3
C. Framework Flutter	6
D. Manfaat Menggunakan Flutter	8
E. Mengapa Belajar Flutter	9
F. Flutlab	12



1

PENGENALAN FLUTTER DAN FLUTLAB

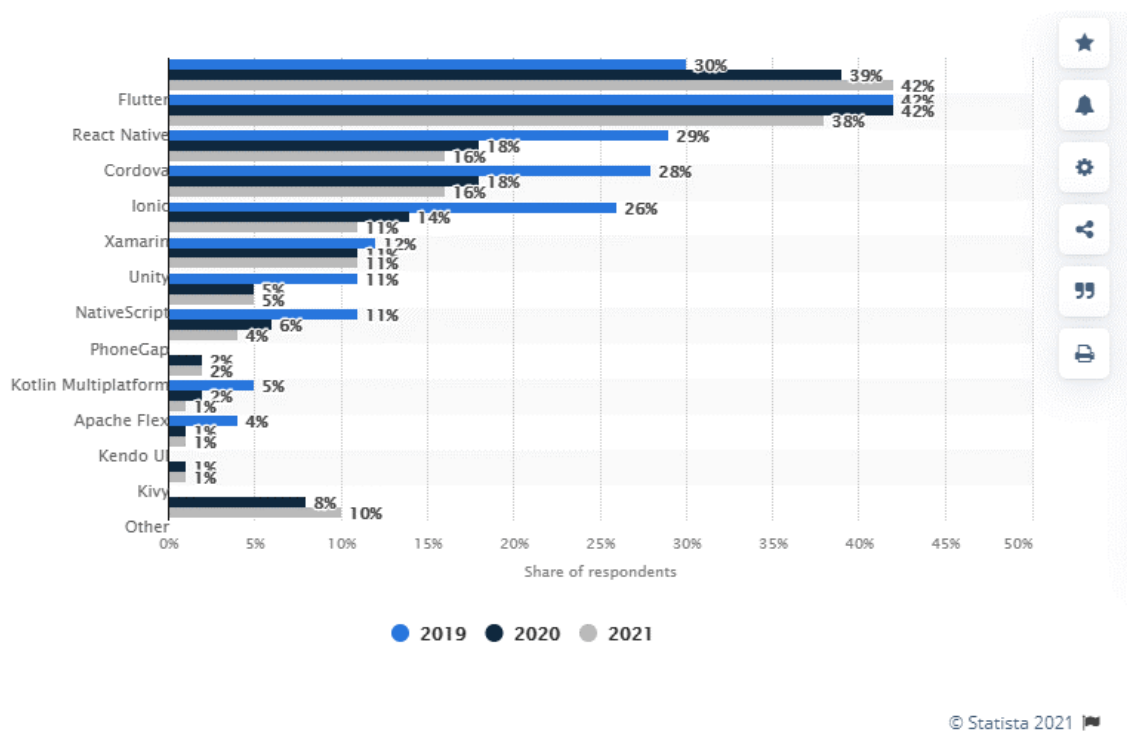
A. Definisi Flutter

Flutter adalah open source framework yang dibuat oleh Google untuk membangun aplikasi multi-platform yang indah, dikompilasi secara native, dari satu basis kode yang sama. Flutter adalah framework pengembangan perangkat lunak lintas platform yang dipresentasikan oleh Google pada tahun 2015 yang dirilis pertama kalinya pada Mei 2017 (versi 0.0.6). Versi pertama Flutter dikenal sebagai codename "Sky", hanya berjalan di sistem operasi Android. Di Dart Developer Summit Pada tanggal 4 Desember 2018, Flutter 1.0 dirilis di acara tersebut, yang menunjukkan versi "stabil" pertama dari framework tersebut. Saat ini versi flutter sudah mencapai 2.10.4. Flutter saat ini telah berkembang dengan mantap dan memberikan kemungkinan tidak hanya untuk pengembangan seluler iOS dan Android tetapi juga untuk aplikasi web dan desktop dengan baik.

Flutter terdiri dari dua bagian penting:

- a) SDK (Software Development Kit): Kumpulan tool yang akan membantu Anda mengembangkan aplikasi. Ini termasuk tool untuk mengkompilasi kode Anda menjadi kode mesin native (kode untuk iOS dan Android).
- b) Framework (library UI berdasarkan widget): Kumpulan elemen UI yang dapat digunakan kembali (tombol, input teks, bilah geser, dan sebagainya) yang dapat Anda sesuaikan untuk kebutuhan Anda sendiri.


Untuk mengembangkan dengan Flutter, Anda akan menggunakan bahasa pemrograman yang disebut Dart. Bahasa ini dibuat oleh Google pada Oktober 2011, tetapi telah meningkat pesat selama beberapa tahun terakhir ini.




B. Dart

Sebelum kita mulai mendekonstruksi framework Flutter, mari kita pelajari bahasa yang digunakan untuk membuatnya - Dart. Dart adalah bahasa pemrograman berorientasi objek yang pertama kali diperkenalkan oleh Google pada tahun 2011. Sejak itu Dart terus berkembang dengan merilis fitur yang berbeda. Antara lain, perlu disebutkan fitur "dart2native" yang memungkinkan kompilasi untuk platform Windows, Linux, dan macOS sebagai aplikasi desktop. Pada penulisan artikel ini, solusi desktop belum siap produksi tetapi prospeknya tampaknya menjanjikan. Selain itu, program Dart dapat disusun menjadi file executable mandiri atau dikompilasi ke JavaScript. Yang terakhir sangat penting karena dengan kompilasi ke JavaScript Anda dapat menjalankan program Dart di setiap browser web modern. Tetapi perhatikan bahwa solusi web belum siap produksi dan saat ini dalam rilis beta.


Secara umum, bahasa Dart mudah dipelajari. Sintaksnya sangat mirip dengan bahasa Java, Swift, atau Kotlin. Selain itu, kit pengembangan perangkat lunak Dart (SDK) dikirimkan dengan Dart Virtual Machine (VM) yang berdiri sendiri yang memungkinkan Anda membuat kode di lingkungan antarmuka baris perintah (CLI) dan jika Anda tidak terbiasa dengan CLI maka Anda dapat bermain-main dengan Dart di DartPad. DartPad adalah editor online yang menyediakan akses ke API Dart dan memungkinkan Anda untuk mengkompilasi kode Dart. Mempelajari bahasa Dart tidak sulit jika Anda memiliki pengalaman dengan Java atau JavaScript.




Dart adalah bahasa yang dioptimalkan untuk klien untuk mengembangkan aplikasi cepat di platform apa pun. Tujuannya adalah untuk menawarkan bahasa pemrograman yang paling produktif untuk pengembangan multi-platform, dipasangkan dengan platform runtime eksekusi yang fleksibel untuk kerangka kerja aplikasi.



Bahasa ditentukan oleh technical envelope – pilihan yang dibuat selama pengembangan yang membentuk kemampuan dan kekuatan bahasa. Dart dirancang untuk lingkup teknis yang sangat cocok untuk pengembangan klien, memprioritaskan pengembangan dan pengalaman produksi berkualitas tinggi di berbagai target kompilasi (web, seluler, dan desktop). Dart juga membentuk dasar Flutter. Dart menyediakan bahasa dan waktu proses yang mendukung aplikasi Flutter, tetapi Dart juga mendukung banyak tugas pengembang inti seperti memformat, menganalisis, dan menguji kode.



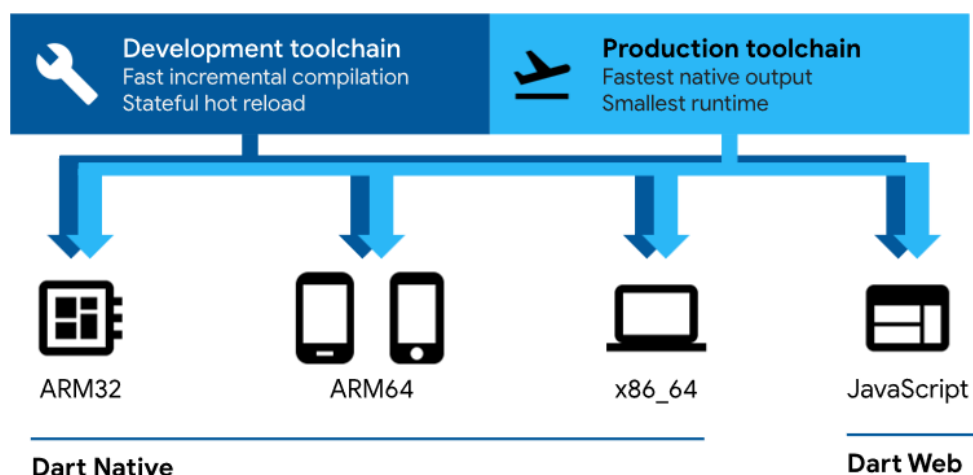
Bahasa Dart adalah tipe aman; ia menggunakan pemeriksaan tipe statis untuk memastikan bahwa nilai variabel selalu cocok dengan tipe statis variabel. Terkadang, ini disebut sebagai pengetikan suara. Meskipun tipe bersifat wajib, anotasi tipe bersifat opsional karena inferensi tipe. Sistem pengetikan Dart juga fleksibel, memungkinkan penggunaan tipe dinamis yang dikombinasikan dengan pemeriksaan runtime, yang dapat berguna selama eksperimen atau untuk kode yang perlu sangat dinamis.



Dart menawarkan keamanan null yang baik, artinya nilai tidak boleh null kecuali Anda mengatakannya bisa. Dengan keamanan null, Dart dapat melindungi Anda dari pengecualian null saat runtime melalui analisis kode statis. Tidak seperti banyak bahasa null-safe lainnya, ketika Dart menentukan bahwa suatu variabel tidak dapat dibatalkan, variabel itu selalu tidak dapat dibatalkan. Jika Anda memeriksa kode yang sedang berjalan di debugger, Anda akan melihat bahwa non-nullability dipertahankan pada saat runtime (karenanya terdengar tidak aman).

Teknologi kompiler Dart memungkinkan Anda menjalankan kode dengan cara yang berbeda:

- a) Platform native: Untuk aplikasi yang menargetkan perangkat seluler dan desktop, Dart menyertakan VM Dart dengan kompilasi just-in-time (JIT) dan compiler sebelumnya (AOT) untuk memproduksi kode mesin.
- b) Platform web: Untuk aplikasi yang menargetkan web, Dart menyertakan kompiler waktu pengembangan (dartdevc) dan kompiler waktu produksi (dart2js). Kedua kompiler menerjemahkan Dart ke dalam JavaScript.




C. Framework Flutter

Framework ini ditulis menggunakan bahasa C, C++, dan Dart dan menggunakan Mesin Grafis Skia Google untuk rendering antarmuka pengguna. Mesin grafis ini digunakan untuk produk yang dikenal seperti Google Chrome, Chrome OS, Chromium OS, Mozilla Firefox, Mozilla Thunderbird, Android, Firefox OS, dan sekarang Flutter.

Flutter berjalan menggunakan mesin virtual Dart (VM) di sistem operasi Windows, Linux, dan macOS. Dart VM menggunakan kompilasi kode just-in-time (JIT) yang menyediakan fitur hemat waktu pengembangan seperti hot-reload. Saat pengembang menulis dan men-debug aplikasi seluler, kompilasi JIT menyuntikkan kode baru ke dalam aplikasi yang sedang berjalan. Ini menyediakan fitur hot-reload stateful, di mana, dalam banyak kasus, perubahan kode sumber dapat segera tercermin dalam aplikasi yang sedang berjalan tanpa memerlukan restart atau kehilangan status. Ini menghemat banyak waktu pengembang pada akhirnya.


Saat merilis aplikasi, Dart VM menggunakan kompilasi sebelumnya (AOT) yang selanjutnya mengonversi kode Dart menjadi kode mesin asli yang bergantung pada platform, yang memungkinkan kinerja tinggi Flutter di perangkat seluler. Kerangka kerja Flutter dirancang menggunakan beberapa prinsip yang harus dijelaskan secara terpisah. Prinsip-prinsip ini adalah "Everything's a widget", "Composition > Inheritance", "Widget tree".

Di Flutter hampir semuanya adalah widget dan itu adalah blok bangunan dasar aplikasi. Dibandingkan dengan kerangka kerja lain, Flutter tidak memiliki pengontrol, tampilan, atau tata letak yang terpisah. Hampir setiap aspek pengembangan Flutter dicakup oleh blok penyusun terpadu - widget. Widget dapat berupa tombol unik, elemen gaya, atau layar pop-up terpisah, dll.







Pendekatan komposisi lebih baik daripada pewarisan. Seringkali widget terdiri dari widget lain yang lebih kecil dan itu adalah pendekatan berbasis komposisi. Menggunakan Flutter API memungkinkan Anda untuk menggabungkan beberapa widget untuk mendapatkan perilaku yang tepat yang Anda butuhkan.

Konsep widget tree pada dasarnya adalah implementasi dari widget bersarang yang mewakili komponen antarmuka pengguna. Widget ini bisa stateless atau stateful dan perbedaan di antara keduanya sesuai dengan status widget. Fitur berguna yang membantu dalam mengelola status aplikasi.



Hal yang paling menarik tentang Flutter adalah komponen antarmuka pengguna. Google membuat dua set widget, Cupertino (iOS) dan Material (Android). Kumpulan widget ini bertanggung jawab atas antarmuka pengguna (UI) dan mencakup setiap komponen yang mungkin Anda perlukan untuk pengembangan Android dan iOS. Widget ini tidak terhubung dengan API asli iOS atau Android seperti di React Native tetapi berfungsi sebagai komponen Flutter mandiri dengan kecepatan rendering dan animasi yang sesuai.

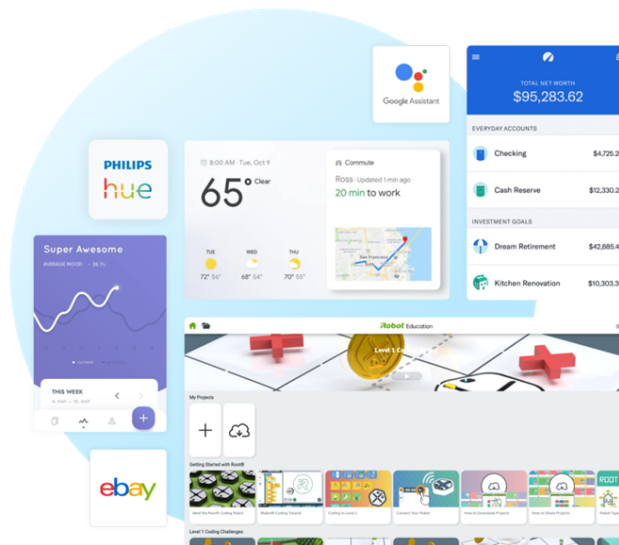


D. Manfaat Menggunakan Flutter

Tanpa membuat perbandingan apa pun dengan platform lain, berikut adalah daftar beberapa fitur dan kualitas yang mungkin membuat Anda mempertimbangkan untuk mencoba Flutter:

1. **Produktivitas tinggi.** Karena Flutter adalah lintas platform, Anda dapat menggunakan basis kode yang sama untuk aplikasi iOS dan Android Anda. Ini pasti dapat menghemat waktu dan sumber daya Anda.
2. **Performa bagus.** Dart mengkompilasi ke dalam kode asli dan tidak perlu mengakses widget OEM karena Flutter memilikinya sendiri. Ini berarti lebih sedikit komunikasi yang dimediasi antara aplikasi dan platform. Flutter adalah satu-satunya SDK seluler yang menyediakan tampilan reaktif tanpa memerlukan jembatan JavaScript. Semua ini berkontribusi pada waktu startup aplikasi yang cepat dan masalah kinerja yang lebih sedikit.
3. **Perkembangan yang cepat dan sederhana.** Salah satu fitur Flutter yang paling dipuji adalah hot reload yang memungkinkan Anda untuk langsung melihat perubahan yang dibuat dalam kode pada emulator, simulator, dan perangkat keras. Dalam waktu kurang dari satu detik, kode yang diubah dimuat ulang saat aplikasi berjalan tanpa perlu memulai ulang. Ini bagus tidak hanya untuk membangun UI atau menambahkan fitur tetapi juga untuk memperbaiki bug. Sejauh menyangkut kesederhanaan, Flutter mengklaim bahwa pemrograman dengan Flutter sangat mudah sehingga tidak diperlukan pengetahuan pemrograman sebelumnya: "Pengalaman dengan bahasa berorientasi objek sangat membantu, tetapi bahkan non-programmer telah membuat aplikasi Flutter!"
4. **Kesesuaian.** Karena widget adalah bagian dari aplikasi dan bukan platform, Anda mungkin akan mengalami lebih sedikit atau tidak ada masalah kompatibilitas pada versi OS yang berbeda. Ini pada gilirannya berarti lebih sedikit waktu yang dihabiskan untuk pengujian.
5. **Sumber terbuka.** Flutter dan Dart keduanya open-source dan gratis untuk digunakan, serta menyediakan dokumentasi ekstensif dan dukungan komunitas untuk membantu mengatasi masalah apa pun yang mungkin Anda alami

E. Mengapa Belajar Flutter



1. Sederhana untuk dipelajari dan digunakan

Flutter adalah framework modern, dan Anda bisa merasakannya! Jauh lebih mudah untuk membuat aplikasi seluler dengannya. Jika Anda telah menggunakan Java, Swift, atau React Native, Anda akan melihat perbedaan Flutter. Anda dapat membuat aplikasi asli tanpa banyak kode.

2. Kompilasi cepat: produktivitas maksimum

Berkat Flutter, Anda dapat mengubah kode dan melihat hasilnya secara real-time. Ini disebut Hot-Reload. Hanya perlu waktu singkat setelah Anda menyimpan untuk memperbarui aplikasi itu sendiri. Modifikasi signifikan memaksa Anda untuk memuat ulang aplikasi. Tetapi jika Anda bekerja seperti desain, misalnya, dan mengubah ukuran elemen, itu dalam waktu nyata!

3. Ideal untuk MVP startup

Jika Anda ingin menunjukkan produk Anda kepada investor sesegera mungkin, Flutter adalah pilihan yang baik.

- a) Lebih murah mengembangkan aplikasi seluler dengan Flutter karena Anda tidak perlu membuat dan memelihara dua aplikasi seluler (satu untuk iOS dan satu untuk Android).
- b) Hanya satu pengembang yang Anda butuhkan untuk membuat MVP Anda.
- c) Performanya luar biasa – Anda tidak akan melihat perbedaan antara aplikasi asli dan aplikasi Flutter.
- d) Sangat bagus – Anda dapat dengan mudah menggunakan widget yang disediakan oleh Flutter dan mempersonalisasikannya untuk membuat UI yang berharga bagi pelanggan Anda (Anda dapat menemukan contoh aplikasi yang dibuat dengan Flutter di bawah).

4. Dokumentasi yang bagus

Penting bagi teknologi baru untuk memiliki dokumentasi yang baik. Tapi itu tidak selalu terjadi! Anda dapat belajar banyak dari dokumentasi Flutter, dan semuanya sangat detail dengan contoh mudah untuk kasus penggunaan dasar.

5. Komunitas yang berkembang







Flutter memiliki komunitas yang kuat, dan ini baru permulaan! Banyak komunitas yang berbagi pengetahuan dan konten bermanfaat tentang pemrograman Flutter. Beberapa diantaranya:

- a) Flutter Awesome: Daftar mengagumkan yang mengkurasi library dan alat Flutter terbaik. Situs web ini menerbitkan konten harian dengan banyak contoh, templat aplikasi, saran, dan sebagainya.
- b) Awesome Flutter: Repositori GitHub (ditautkan ke Flutter Awesome) dengan daftar artikel, video, komponen, utilitas, dan sebagainya.
- c) Ini semua widget!: Daftar aplikasi terbuka yang dibuat dengan Flutter.
- d) Komunitas Flutter: Publikasi Medium tempat Anda dapat menemukan artikel, tutorial, dan banyak lagi.



6. Didukung oleh Android Studio dan VS Code

Flutter tersedia di IDE yang berbeda. Dua editor kode utama untuk pengembangan dengan teknologi ini adalah Android Studio (IntelliJ) dan VS Code. Android Studio adalah perangkat lunak yang lengkap dengan semuanya sudah terintegrasi. Anda harus mengunduh plugin Flutter dan Dart untuk memulai. VS Code adalah alat yang ringan, dan semuanya dapat dikonfigurasi melalui plugin dari pasar.



F. Flutlab

Jika anda tidak memiliki device yang mumpuni, cukup dengan koneksi internet yang memadai, anda sudah bisa belajar flutter dan membuat aplikasi sederhana dengan Flutlab. Flutlab adalah IDE online yang terbaik untuk pemula di flutter, yang ingin belajar & membangun aplikasi lintas platform menggunakan editor online flutter.

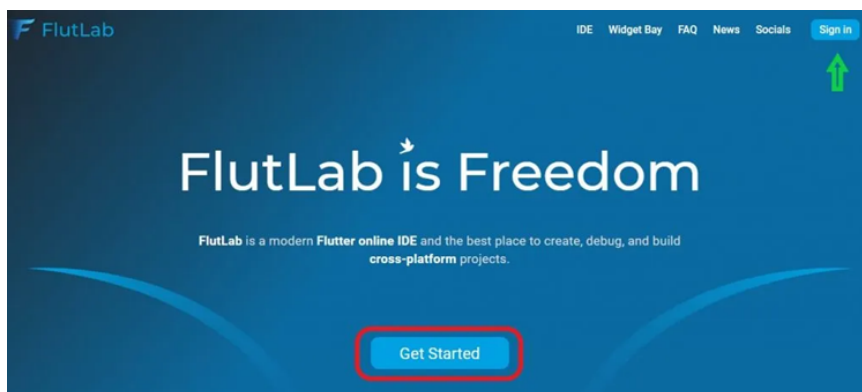
FlutLab adalah IDE online yang sangat berguna, untuk pengembang flutter yang tidak memiliki konfigurasi laptop/komputer yang baik, dan IDE Android Studio atau IDE lainnya berjalan sangat lambat dan tidak dapat menangani Flutter SDK karena konfigurasi PC yang rendah. Tool paling baik untuk mempelajari pengembangan aplikasi flutter secara online tanpa menginstal IDE apa pun.

Flutlab.io pada dasarnya adalah IDE online untuk mengembangkan aplikasi flutter online. Ini adalah proyek bootstrap flutter yang dikembangkan oleh tim CodeGemz S.R.O. Pengembang flutter dapat mengembangkan aplikasi lintas platform lengkap bahkan tanpa menginstal perangkat lunak apa pun di PC-nya kecuali browser. UI Flutlab ini sangat mulus sehingga Anda mendapatkan pengalaman seperti sedang mengembangkan aplikasi flutter di komputer lokal Anda.

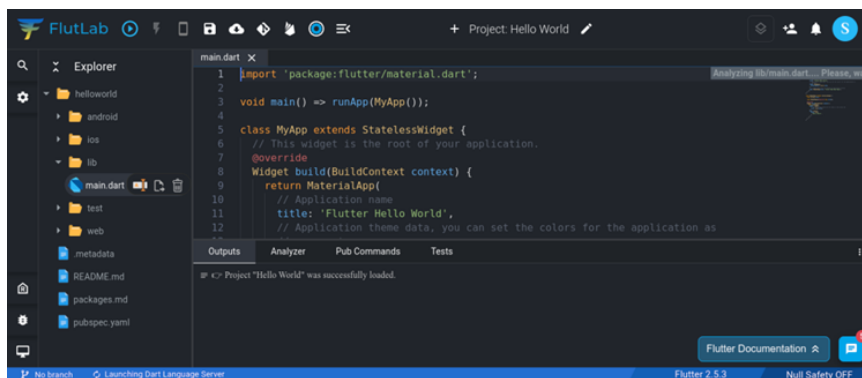
Secara umum, flutlab gratis untuk digunakan, namun memiliki limitasi, yaitu satu akun hanya bisa membuat maksimum 2 project flutter dengan kapasitas penyimpanan maksimum 250 Mb. Jika anda seorang programmer profesional, tidak ada salahnya untuk menjadi akun premium dengan biaya bulanan tertentu, namun memiliki fitur dan kapasitas yang jauh daripada akun gratisnya.

Membuat Akun Flutlab

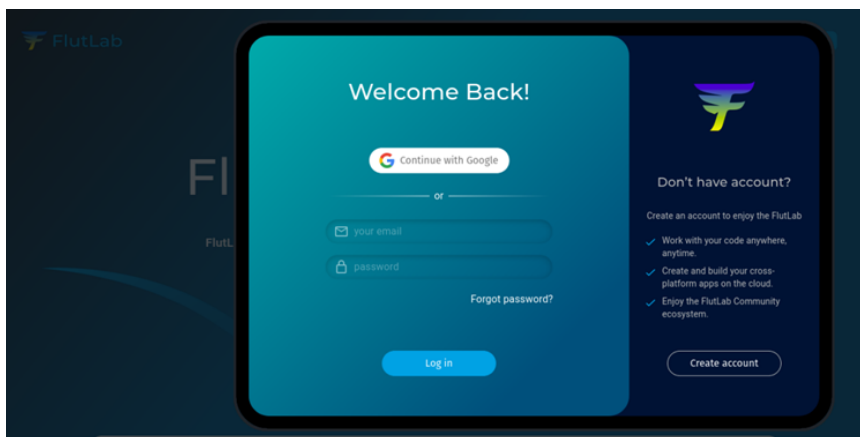
1. Kunjungi web flutlab dengan url: <https://flutlab.io/>



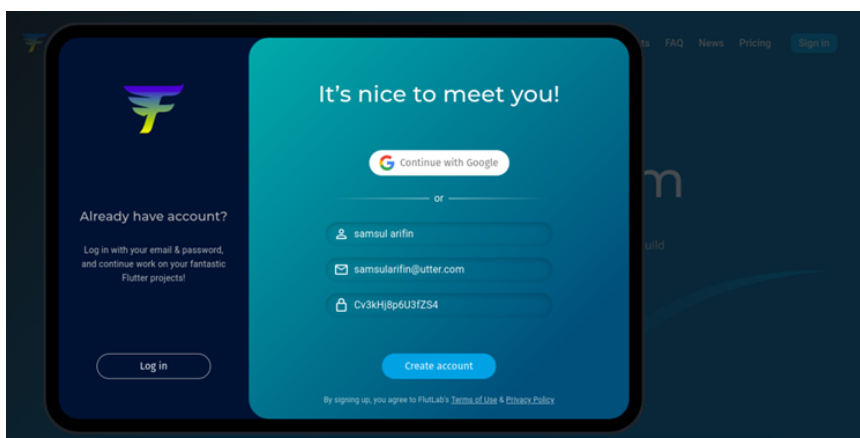
jika Anda ingin mencobanya tanpa masuk, Anda dapat melakukannya hanya dengan mengklik Get Started, tetapi perhatikan bahwa proyek Anda tidak akan disimpan untuk pembaruan di masa mendatang.



2. Kemudian membuat akun flutlab, dengan klik Sign in > Create Account. Atau bisa juga dengan login menggunakan akun google anda. Jika anda menggunakan akun google anda, skip langkah 3-4.



3. Isi data dengan lengkap dan benar, gunakan kombinasi huruf dan angka untuk membuat password yang kuat



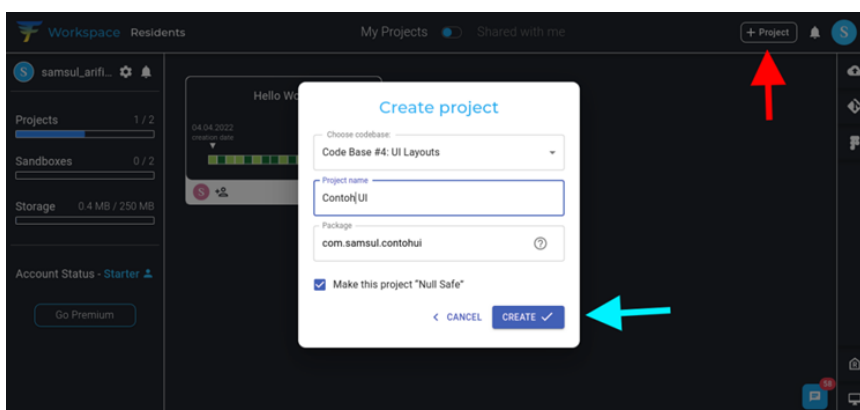
4. Cek email anda, apakah ada verifikasi akun dari flutlab dan pastikan klik tombol Confirm Registration



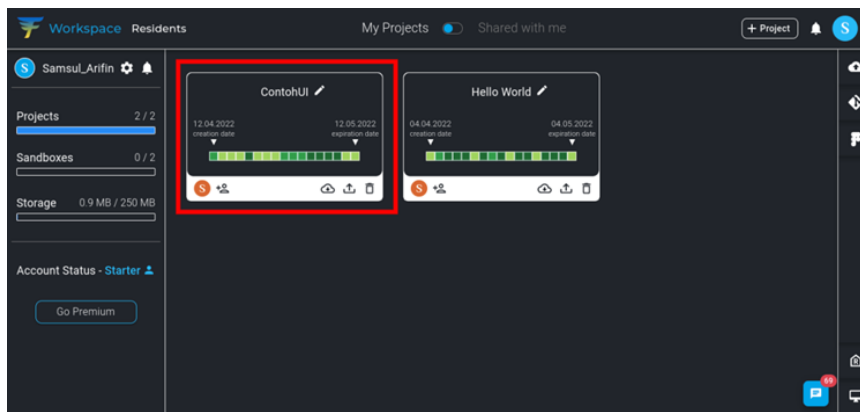
5. Selamat anda sudah sukses dan berhasil memiliki akun flutlab.

Membuat Project di Flutlab.io

1. Di halaman Workspace Residents, Klik tombol +Project yang ada di kanan atas, hingga muncul form create project.

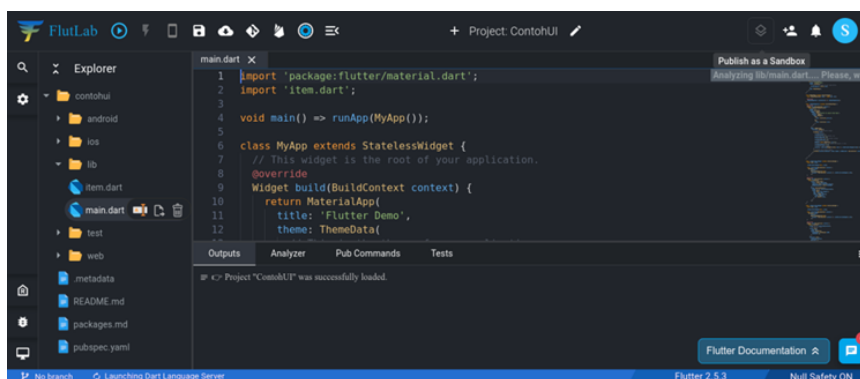


2. Anda harus memilih Code Base default dari menu dropdown untuk proyek Anda. Anda bisa memilih Stateless Hello World atau UI_Layouts
3. Beri nama project-nya dan domain package. untuk package boleh diisi atau boleh dikosongkan.
4. Selanjutnya Klik tombol Create, jika berhasil akan muncul project anda seperti gambar berikut

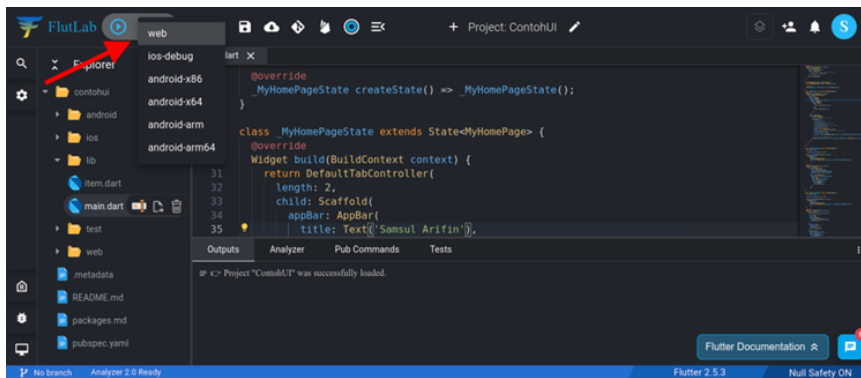


Memodifikasi dan Menjalankan Project

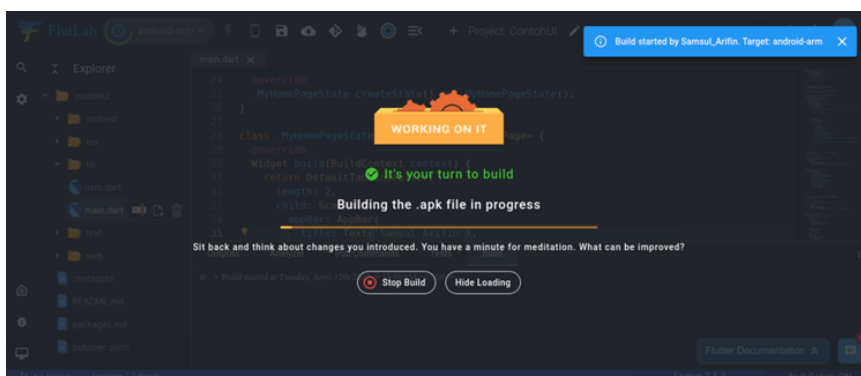
1. Dari project yang sudah dibuat sebelumnya, buka project tersebut dengan klik langsung kartu project-nya, sehingga akan muncul halaman editor seperti gambar berikut:



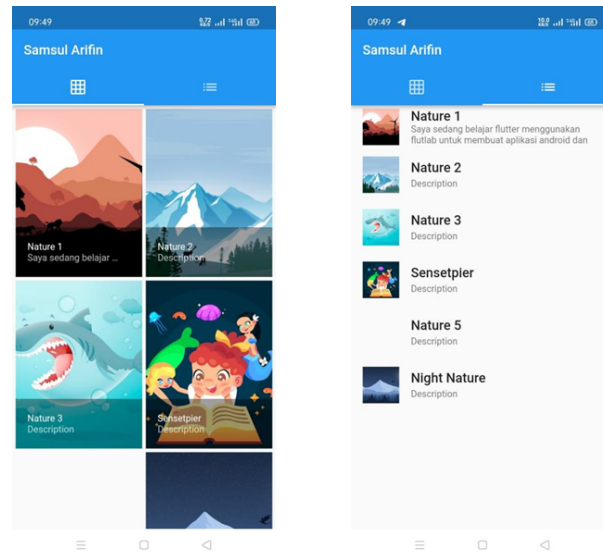
2. Kemudian edit title di baris 35 dengan nama anda masing-masing.
3. Kemudian edit juga teks “Lorem ipsum dst...” di baris 180-185 dengan deskripsi anda masing-masing.
4. Untuk menjalankan project, pertama anda arahkan kursor mouse ke tombol RUN yang ada di sebelah kiri atas, sehingga muncul menu dropdown, seperti gambar berikut:



5. Klik menu dropdown dan pilih salah satu opsi yang ingin anda gunakan untuk membangun project, misalkan pilih saja “android-arm”, setelah dipilih klik tombol RUN, sehingga muncul gambar berikut:



6. Setelah proses selesai, akan muncul hasilnya berupa apk, silahkan download apk tersebut dan install di hp anda masing-masing, hasilnya seperti gambar berikut



7. Jika memilih opsi "web", maka flutlab akan mem-build project dan hasilnya akan dimunculkan dalam sebuah emulator dari flutlab, seperti gambar berikut:



8. Nah mudah bukan...