

FUAH UIN KHAS Jember



Oleh
Maulida Dwi A

MULTIMEDIA AL-QUR'AN DAN TAFSIR

2022

LEMBAR PENGESAHAN

Modul Mata Kuliah Multimedia Al-Qur'an dan Tafsir

Nama : Maulida Dwi Agustiningsih, S.Kom., M.T.I.

NIP : 199308302020122006

NIDN : 2030089302

Dan digunakan untuk kalangan sendiri sebagai bahan ajar pada:

Mata kuliah : Multimedia Al-Qur'an dan Tafsir

Semester : Ganjil/Genap

Fakultas : Ushuluddin Adab dan Humaniora

Universitas : Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember

Disahkan pada tanggal: 5 Januari 2022

Mengesahkan

Wakil Dekan I Bidang Akademik

Fakultas Ushuluddin Adab dan Humaniora



Dr. Uin Yusuf, M.A.

NIP. 198007162011011004

DAFTAR ISI

BAB 1 APLIKASI.....	2
BAB 2 KEBUTUHAN PENGGUNA	5
BAB 3 PENGENALAN PLATFORM PENGEMBANGAN PERANGKAT LUNAK.....	8
BAB 4 LIVE DEVELOPMENT	13
BAB 5 MEMBANGUN APLIKASI PERTAMA “HELLO WORLD”.....	17
BAB 6 BACKPACK	19
BAB 7 SPESIAL TOPIK 1 “CONCEPT CARDS”	22
BAB 8 MENGGUNAKAN DYNAMIC CARDS DI APLIKASI.....	30
BAB 9 MELAKUKAN UPDATE DENGAN FIREBASE REMOTE CONFIG	38
BAB 10 BEKERJA DENGAN ATURAN FIREBASE.....	46
BAB 11 KEAMANAN FIREBASE.....	52
BAB 12 MENARGETKAN PEMBERITAHUAN PUSH ANDA.....	66
BAB 13 MENGINTEGRASIKAN POLLFISH DI APLIKASI ANDA.....	71
BAB 14 APP SURAH-SURAH PENDEK AL-QURAN.....	80

BAB 1

APLIKASI

Aplikasi seluler, paling sering disebut sebagai aplikasi, adalah jenis perangkat lunak aplikasi yang dirancang untuk berjalan di perangkat seluler, seperti ponsel cerdas atau komputer tablet. Aplikasi seluler sering berfungsi untuk menyediakan layanan serupa kepada pengguna dengan yang diakses di PC. Aplikasi umumnya kecil, unit perangkat lunak individual dengan fungsi terbatas. Penggunaan perangkat lunak aplikasi ini awalnya dipopulerkan oleh Apple Inc. dan App Store-nya, yang menawarkan ribuan aplikasi untuk iPhone, iPad, dan iPod Touch.

Aplikasi seluler juga dapat dikenal sebagai aplikasi, aplikasi web, aplikasi online, aplikasi iPhone, atau aplikasi ponsel cerdas. Aplikasi seluler adalah langkah menjauh dari sistem perangkat lunak terintegrasi yang umumnya ditemukan pada PC. Sebagai gantinya, setiap aplikasi menyediakan fungsionalitas terbatas dan terisolasi seperti game, kalkulator, atau penjelajahan web seluler. Meskipun aplikasi mungkin menghindari multitasking karena sumber daya perangkat keras yang terbatas dari perangkat seluler awal, kekhususannya sekarang menjadi bagian dari keinginan mereka karena memungkinkan konsumen untuk memilih sendiri apa yang dapat dilakukan perangkat mereka.

Aplikasi seluler paling sederhana mengambil aplikasi berbasis PC dan memindahkannya ke perangkat seluler. Karena aplikasi seluler menjadi lebih kuat, teknik ini agak kurang. Pendekatan yang lebih canggih melibatkan pengembangan khusus untuk lingkungan seluler, mengambil keuntungan dari keterbatasan dan kelebihanannya. Misalnya, aplikasi yang menggunakan fitur berbasis lokasi secara inheren dibangun dari bawah ke atas dengan memperhatikan seluler mengingat pengguna tidak terikat pada suatu lokasi, seperti pada PC.

Aplikasi dibagi menjadi dua kategori besar: aplikasi asli dan aplikasi web. Aplikasi asli dibuat untuk sistem operasi seluler tertentu, biasanya iOS atau Android. Aplikasi asli menikmati kinerja yang lebih baik dan antarmuka pengguna (UI) yang lebih halus, dan biasanya harus melewati proses pengembangan dan jaminan kualitas yang jauh lebih ketat sebelum dirilis.

Aplikasi web digunakan dalam HTML5 atau CSS dan memerlukan memori perangkat minimum karena dijalankan melalui browser. Pengguna diarahkan ke halaman web tertentu, dan semua informasi disimpan di database berbasis server. Aplikasi web memerlukan koneksi yang stabil untuk digunakan.

Ada beberapa jenis aplikasi yang tersedia saat ini.

- Aplikasi game: Setara dengan video game komputer, mereka adalah salah satu jenis aplikasi paling populer. Mereka menyumbang sepertiga dari semua unduhan aplikasi dan tiga perempat dari semua pengeluaran konsumen.
- Aplikasi produktivitas: Ini berfokus pada peningkatan efisiensi bisnis dengan memudahkan berbagai tugas seperti mengirim email, melacak kemajuan pekerjaan, memesan hotel, dan banyak lagi.
- Aplikasi gaya hidup dan hiburan: Semakin populer, ini mencakup banyak aspek gaya hidup pribadi dan sosialisasi seperti berkencan, berkomunikasi di media sosial, serta berbagi (dan menonton) video. Beberapa aplikasi yang paling banyak dikenal seperti Netflix, Facebook atau TikTok termasuk dalam kategori ini.

Jenis aplikasi lainnya termasuk aplikasi mobile commerce (M-commerce) yang digunakan untuk membeli barang secara online seperti Amazon atau eBay, aplikasi perjalanan yang membantu wisatawan dalam banyak hal (memesan tur dan tiket, menemukan jalan mereka melalui peta dan geolokasi, buku harian perjalanan, dll. .), dan aplikasi utilitas seperti aplikasi kesehatan dan pemindai kode batang.

Tiga Jenis Aplikasi: Native, Hybrid, and Web

Native Mobile Apps:

Aplikasi seluler asli dirancang untuk menjadi "asli" untuk satu platform, baik itu Apple iOS, Google Android, atau Windows Phone. Platform asli dapat menguntungkan karena cenderung mengoptimalkan pengalaman pengguna. Karena dikembangkan khusus untuk platform, ia dapat beroperasi lebih cepat dan intuitif.

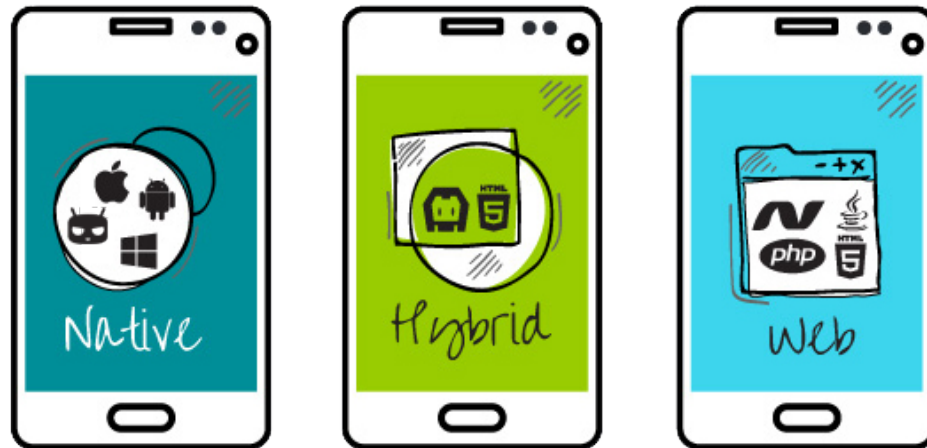
Hybrid Mobile Apps:

Aplikasi ini dapat diinstal pada perangkat seperti aplikasi asli, tetapi dijalankan melalui browser web. Semua aplikasi hybrid dikembangkan melalui bahasa pemrograman HTML5. Meskipun aplikasi hybrid tidak secepat atau dapat diandalkan seperti aplikasi asli, mereka memiliki kapasitas yang lebih besar untuk merampingkan proses pengembangan. Karena Anda tidak perlu membuat dan memelihara aplikasi untuk platform terpisah, bisnis Anda dapat menghemat waktu dan sumber daya. Ini ideal untuk aplikasi yang terutama mengirimkan konten.

Web Apps:

Situs web responsif beralih ke desain yang berbeda saat diakses dari perangkat seluler. Aplikasi web adaptif, di sisi lain, skala agar sesuai dengan ukuran layar perangkat seluler yang berbeda.

Untuk aplikasi ini, desainnya tidak berubah. Aplikasi web dibuat menggunakan bahasa pemrograman paling populer, tetapi tidak dapat menggunakan perangkat keras pada perangkat seluler atau dijual di toko aplikasi mana pun.



Daftar Pengembangan Aplikasi Seluler Lintas Platform Terbaik

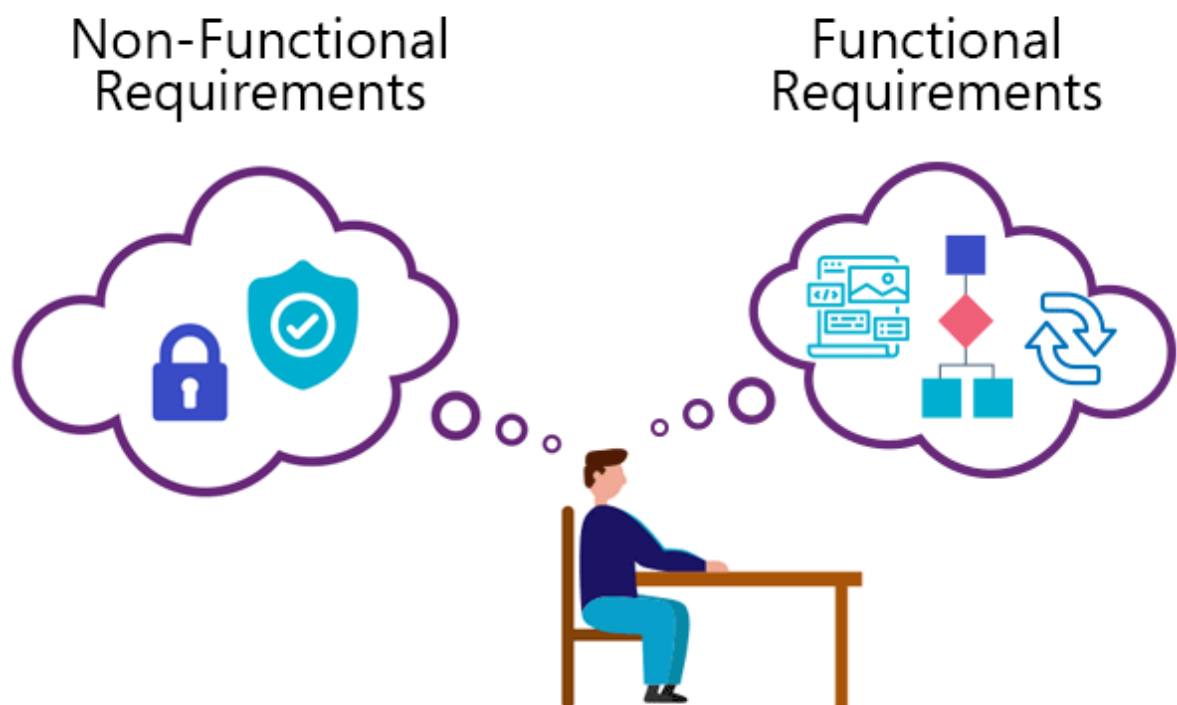
- Xamarin menghadirkan aplikasi Android, iOS, dan Windows asli, menggunakan keterampilan, tim, dan kode yang ada.
- Adobe PhoneGap adalah kerangka kerja pengembangan aplikasi seluler berdasarkan proyek sumber terbuka (Apache Cordova).
- Appcelerator adalah platform pengembangan yang berfokus pada perusahaan yang memungkinkan pengembang menulis JavaScript dengan kerangka kerja Alloy MVC-nya.
- Kony menawarkan berbagai alat sebagai bagian dari Platform Mobilitas perusahaannya untuk membantu bisnis membuat aplikasi dari satu basis kode.
- Sencha Touch adalah kerangka kerja aplikasi web seluler lintas platform terkemuka berdasarkan HTML5 dan JavaScript untuk membuat aplikasi seluler universal.
- Appery.io adalah platform pembuat aplikasi seluler berbasis cloud terbaik yang dapat digunakan oleh pengembang aplikasi ponsel untuk membuat aplikasi terbaik untuk ponsel Android, iOS, dan Windows, dan menyertakan Apache Cordova dengan akses ke komponen bawaannya.

LATIHAN

1. Sebutkan jenis-jenis dan fungsi atau fitur utama aplikasi yang populer di Indonesia
2. Sebutkan platform pengembangan aplikasi perangkat bergerak

Sebelum mengembangkan aplikasi perangkat bergerak, yang harus dilakukan adalah mendefinisikan kebutuhan calon pengguna aplikasi. Pengguna disini dapat diartikan sebagai seorang atau sekelompok orang dengan maksud tertentu menggunakan aplikasi. Pengguna dapat dikategorikan di beberapa aspek seperti pekerjaan, umur, instansi, dan jenis kelamin. Sebagai contoh adalah aplikasi Instagram, yang cocok digunakan untuk seseorang yang usianya lebih dari 12 tahun.

Kebutuhan pengguna membantu pengembang aplikasi untuk mendapatkan gambaran bagaimana suatu aplikasi digunakan. Pendefinisian kebutuhan pengguna sebaiknya terukur, spesifik, tidak ambigu, dan jelas. Kebutuhan pengguna dibedakan menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.



A. Kebutuhan Fungsional

Kebutuhan fungsional didefinisikan sebagai “what the system should do”, artinya apa yang harus dilakukan oleh sistem. Kebutuhan fungsional dapat dilihat dari fitur-fiturnya. Contohnya pada aplikasi instagram, terdapat fitur untuk dapat mengunggah foto/video.

B. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional didefinisikan sebagai “how the system should do it”, artinya bagaimana sistem itu arus berjalan. Terdapat aturan atau standar yang harus ada pada suatu sistem. Berikut adalah contoh kebutuhan non-fungsional:

- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data integrity
- Usability
- Interoperability
- Performance

Berikut adalah metode-metode untuk mendapatkan kebutuhan pengguna:

- Field Research
Melakukan wawancara dan observasi
- Focus Groups



Wawancara secara berkelompok, biasanya menggunakan prototype, untuk mengeksplorasi konsep dan pendapat user.

- User Modeling
Menganalisis karakteristik dari key audiences sebuah produk, website, atau jasa, dan membuat karakter atau persona yang sesuai dengan profil
- Information Architecture Analysis
Mengevaluasi informasi tentang produk atau website dan mengidentifikasi komponen, taEvaluation of the information space for a product or website and identifying the components, taxonomy, and relationships.
- Task Analysis—Penyimpanan dan memprioritaskan tasks yang dilakukan oleh user terhadap produk atau aplikasi.

LATIHAN

1. Diskusian dengan kelompok dalam menganalisis kebutuhan fungsional dan kebutuhan non-fungsional calon pengguna aplikasi masing-masing.
2. Gambarkan rancangan aplikasi dalam bentuk flow screen.

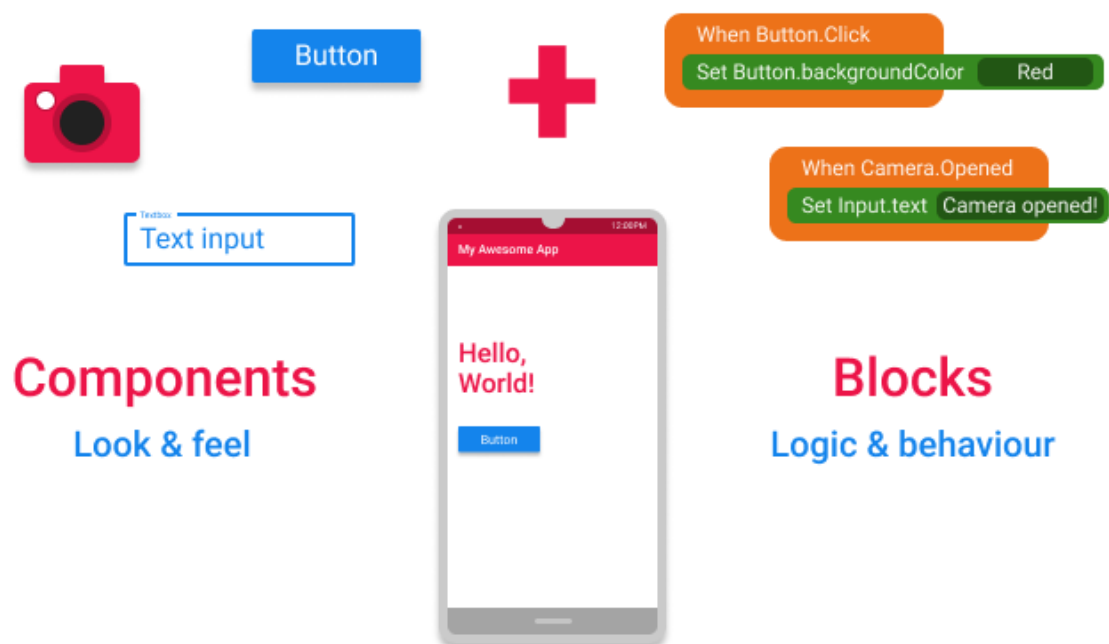
PENGENALAN PLATFORM PENGEMBANGAN PERANGKAT LUNAK

MIT Inventor

MIT App Inventor merupakan platform online yang didesain untuk membantu dalam memahami konsep *computational thinking* melalui aplikasi perangkat bergerak. MIT Inventor termasuk platform yang mudah karena menggunakan konsep tarik dan lepas (*drag and drop*) dalam mendesain layout kemudian menggunakan visualisasi *blocks language* untuk mengatur program. Platform MIT Inventor mengusung konsep *what you see is what you get* (WYSIWYG), pengguna dapat membangun aplikasi menggunakan MIT Inventor untuk menyelesaikan masalah yang ada di kehidupan sehari-hari. Format aplikasi yang diekspor dari MIT Inventor dapat dipilih untuk dijalankan di perangkat android/iOS.

Kodular

Kodular adalah perangkat lunak yang menyediakan beberapa "modul" (layanan) untuk membuat aplikasi Android tanpa coding. Infrastruktur Kodular didukung oleh mitra kita ArubaCloud, yang menjamin waktu kerja dan kinerja yang sangat baik. Perusahaan ini didirikan pada tahun 2017 dan berbasis di Dover, Delaware. Kodular memungkinkan untuk membuat aplikasi Android dengan mudah dengan editor tipe blocks.



Aplikasi di Kodular dibuat sebagai kombinasi dari berbagai Komponen , dengan masing-masing Komponen digunakan untuk tujuan tertentu. Perilaku Komponen dikonfigurasi menggunakan Blocks

Komponen

Komponen adalah elemen pembangun dasar dari aplikasi apa pun di Kodular. Segala sesuatu di aplikasi dilakukan dengan bantuan satu komponen atau yang lain. Komponen yang berbeda digunakan untuk tujuan yang berbeda. Satu komponen dapat digunakan untuk mendesain Antarmuka Pengguna (UI) aplikasi, misalnya komponen Tombol, sementara komponen lainnya dapat digunakan untuk melakukan tindakan seperti berkomunikasi ke database, menyimpan gambar ke folder perangkat Android, dll. komponen FirebaseDB.

Blocks

Blocks adalah yang digunakan untuk menjelaskan bagaimana melakukan suatu tugas. Cara Komponen merespons berbagai tindakan dan peristiwa di aplikasi dirancang menggunakan Blocks. Misalnya, bagaimana aplikasi harus merespons ketika Tombol diklik, data apa yang akan dikomunikasikan ke database menggunakan komponen FirebaseDB, dll. Semuanya dikonfigurasi menggunakan Blocks. Jadi, dengan menggunakan berbagai Komponen dan dengan mengonfigurasi perilaku dan responsnya terhadap tindakan dari pengguna dengan bantuan Blocks, sebuah aplikasi "dikodekan" di Kodular.

Mari kita pertimbangkan analogi Mobil untuk lebih memahami konsep komponen dan balok. Sebuah Mobil dibangun menggunakan berbagai bagian, seperti roda, pintu, mesin, kemudi dll masing-masing melakukan fungsi uniknya sendiri. Beberapa bagian ini terlihat oleh pengemudi (yang merupakan pengguna mobil) dan dia berinteraksi dengan mereka secara langsung, sementara beberapa di antaranya tidak terlihat. Dan bagian-bagian tersebut mengetahui apa fungsinya dan bagaimana cara menjalankannya, misalnya fungsi velg untuk memberikan dukungan dan menggerakkan mobil, fungsi lingkar kemudi untuk membantu menyetir mobil dan lain sebagainya.

Sebuah aplikasi Android seperti Mobil. Seperti bagaimana mobil memiliki berbagai bagian, aplikasi dibangun menggunakan berbagai Komponen, seperti Tombol, Label, File, dll. Dengan setiap komponen digunakan untuk tujuan tertentu. Seperti bagian dari Mobil, beberapa Komponen ini terlihat oleh pengguna aplikasi dan berinteraksi dengannya, sementara beberapa komponen tidak berinteraksi langsung dengan pengguna. Perilaku atau tujuan komponen ini

dikonfigurasi menggunakan Blok. Bagian "cara menjalankan fungsi komponen" dikonfigurasi dengan menggunakan Blok

Jenis-jenis komponen

- **Visible Components**
Komponen yang dapat dilihat di panel Viewer pada halaman Designer dikenal sebagai Visible Component. Mereka terlihat oleh pengguna di aplikasi dan pengguna dapat berinteraksi dengannya secara langsung dan membantu dalam membangun Antarmuka Pengguna aplikasi.
- **Non-Visible Components**
Komponen yang tidak dilihat di panel Viewer pada halaman Designer dikenal sebagai Non-Visible Component. Mereka adalah komponen yang biasanya melakukan hal lain seperti menyimpan file ke penyimpanan perangkat, mendapatkan data dari berbagai sensor perangkat, dll. Namun, terkadang, komponen ini juga dapat digunakan untuk membuat elemen yang terlihat di aplikasi. Misalnya, komponen Notifier, yang merupakan Komponen Tidak Terlihat dapat digunakan untuk membuat dialog yang terlihat di layar perangkat.

Tipe-tipe Blocks

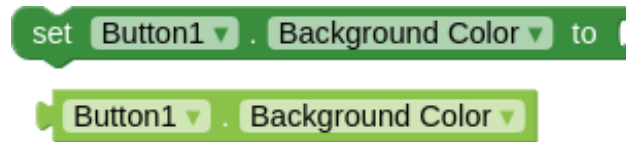
Setiap Komponen memiliki kumpulan bloknnya sendiri, dibagi menjadi tiga jenis:

- Property blocks
- Method blocks
- Event blocks

Property blocks

Setiap komponen dilengkapi dengan kumpulan Propertinya sendiri, yang menggambarkan karakteristiknya. Misalnya, komponen Tombol memiliki properti seperti Warna Latar, Warna Teks, Ukuran Font Teks, dll. yang merupakan karakteristik yang menggambarkan fitur tombol tertentu.

Beberapa Properti ini dapat diatur dari halaman Desainer, atau melalui Blok, sementara beberapa dapat diatur hanya melalui Blok dan beberapa hanya dari Desainer.



Kapan saja di aplikasi, properti komponen yang diberikan dapat dibaca melalui Blok Pengambil. Ini memberikan nilai saat ini dari properti itu dari komponen tertentu.

Demikian pula, properti komponen yang diberikan dapat dimodifikasi selama runtime aplikasi melalui Setter Blocks.

Method blocks

Metode adalah fungsi tertentu yang dapat dilakukan komponen. Misalnya, komponen Suara harus dapat memutar musik saat diperlukan. Ini dilakukan dengan menggunakan Metode yang disebut Play, yang menginstruksikan komponen untuk melakukan tugas memainkan musik. Jadi, Metode tidak lain adalah instruksi untuk melakukan tugas tertentu, seperti memainkan musik.

Beberapa metode memerlukan beberapa data tambahan, yang juga akan digunakan dalam menjalankan tugas itu. Misalnya, komponen Firebase DB menyimpan data menggunakan metode yang disebut Store Value. Tetapi bagaimana komponen mengetahui data apa yang akan disimpan dan dengan nama apa? Di sinilah Parameter atau hanya Params digunakan.

Metode Store Value memiliki dua parameter (yaitu menggunakan dua data berbeda yang perlu ditentukan) yaitu tag dan nilai To Store. Jadi metode ini dapat dipanggil (yaitu digunakan dalam blok) bersama dengan dua data yang dibutuhkan seperti yang ditunjukkan di bawah ini.



Event blocks

Aplikasi Kodular "dikodekan" dalam pendekatan yang digerakkan oleh peristiwa. Artinya, blok atau perilaku aplikasi dijalankan berdasarkan berbagai peristiwa yang terjadi. Ada berbagai macam peristiwa yang terjadi sepanjang waktu aplikasi berjalan, dan respons aplikasi terhadap peristiwa ini dapat dikonfigurasi.

Misalnya, pengguna mengklik Tombol, aplikasi dimulai adalah semua peristiwa. Respons aplikasi terhadap peristiwa tersebut dikonfigurasi dengan mengelompokkan perilaku yang diinginkan di bawah blok Peristiwa. Metode dan blok properti yang dikelompokkan di bawah setiap peristiwa dieksekusi setiap kali peristiwa tertentu itu terjadi.

Mari kita ambil contoh.

Jika Anda ingin mengubah teks Tombol saat pengguna mengkliknya, Anda harus mengelompokkan logika untuk mengubah teks di bawah acara Button.Clicked. Ini ditunjukkan di bawah ini.



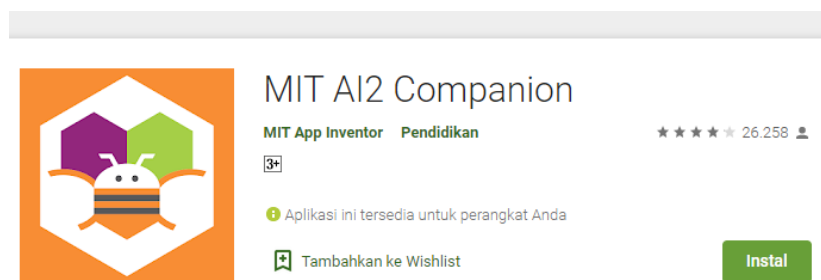
Latihan:

A. Perangkat yang harus disiapkan adalah:

1. Jaringan internet
2. Browser
3. Android smartphone

B. Instalasi MIP App Inventor Companion

1. Buka Google Play store atau Apple app store di perangkat bergerak
2. Ketik di menu cari “MIT AI2 Companion”



3. Unduh dan install di perangkat bergerak

KETENTUAN

- Maksimal 1 lembar dikumpulkan maksimal 1 minggu setelah pemberian tugas.
- Dikumpulkan perwakilan oleh ketua kelompok di sister.
- Bentuk file .pdf. Nama file D1_WorksheetX_KelX contoh: D1_Worksheet2_Kel1.pdf

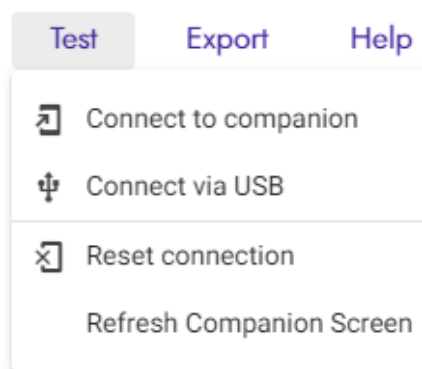
Untuk membangun dan menginstal file APK proyek Anda di perangkat Anda setiap kali Anda membuat perubahan. Kodular memiliki satu fitur yang membuat proses mendesain aplikasi Anda menjadi lebih mudah - pendamping Kodular. Pendamping dapat diinstal pada perangkat Anda seperti aplikasi Android biasa. Setelah Anda menghubungkan pengiring ke pembuat, baik melalui WiFi atau USB, perubahan yang Anda buat di perancang, akan langsung ditampilkan di perangkat Anda.

WiFi

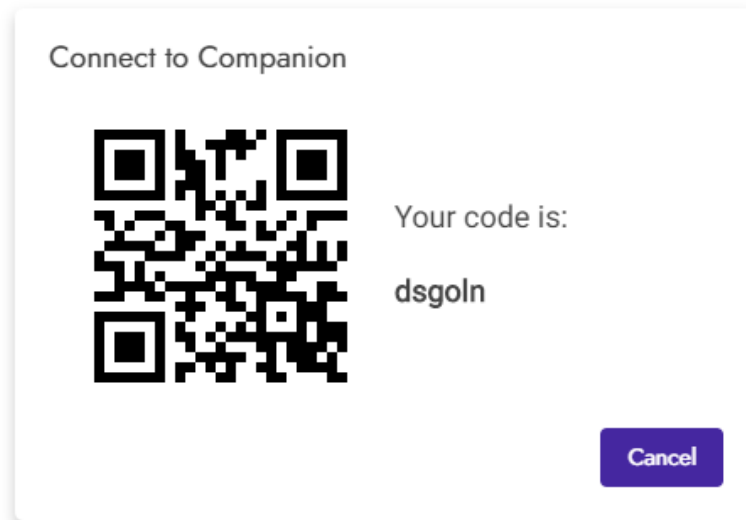
Langkah 1: Dari App Bar, pilih "Test".



Langkah 2: Klik "Companion" untuk mendapatkan Kode QR yang akan digunakan untuk menghubungkan ponsel Anda ke Builder.



Langkah 3: Simpan Kode QR (atau kode yang muncul di sebelahnya untuk melakukan Kode QR jika Anda ingin memasukkan kode secara manual) yang Anda dapatkan dari Pembuat Kodular untuk langkah selanjutnya, sekarang ke fase berikutnya.



Langkah 3: Klik Pindai "Kode QR" atau "Hubungkan dengan kode" pada aplikasi pendamping dan pindai Kode QR atau ketik kode yang Anda lihat di Pembuat Kodular.

Langkah 4: Anda telah berhasil menautkan Ponsel / emulator Android Anda ke Kodular Creator.

USB

Menjalankan Kodular Starter di PC Anda

1. Unduh file yang dapat dieksekusi untuk Kodular Starter v2.0 berdasarkan OS Anda
 - Windows - KodularStarter_windows.exe
 - macOS - KodularStarter_macos
 - Linux - KodularStarter_linux
2. Untuk menjalankan, ikuti instruksi berdasarkan OS Anda
 - Windows - Klik dua kali pada file yang dapat dieksekusi untuk menjalankan Kodular Starter.
 - macOS/Linux - Buka terminal dan jalankan file seperti yang ditunjukkan di bawah ini

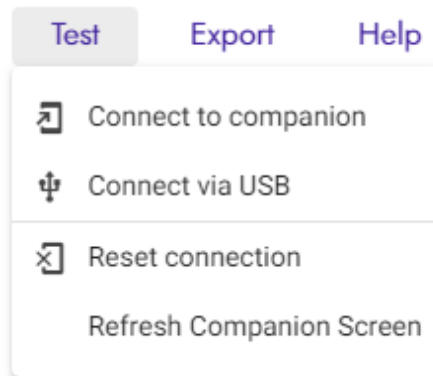
```
kodular@Kodular ~ $ chmod +x ./path/to/KodularStarter
kodular@Kodular ~ $ ./path/to/KodularStarter
```
3. Tetap jalankan Kodular Starter saat Anda menggunakan Companion melalui USB.

Connect to Companion

Langkah 1: Dari App Bar, pilih "Test".



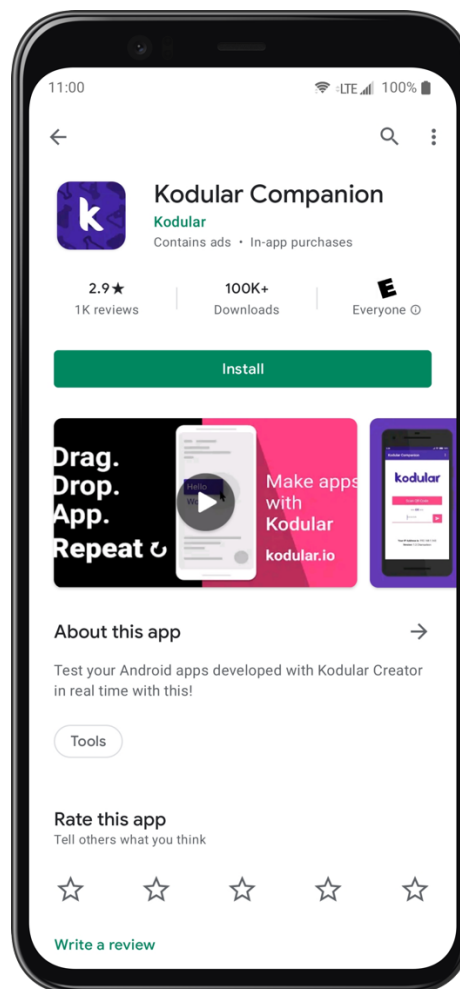
Langkah 2: Klik "USB" dan tunggu beberapa detik.



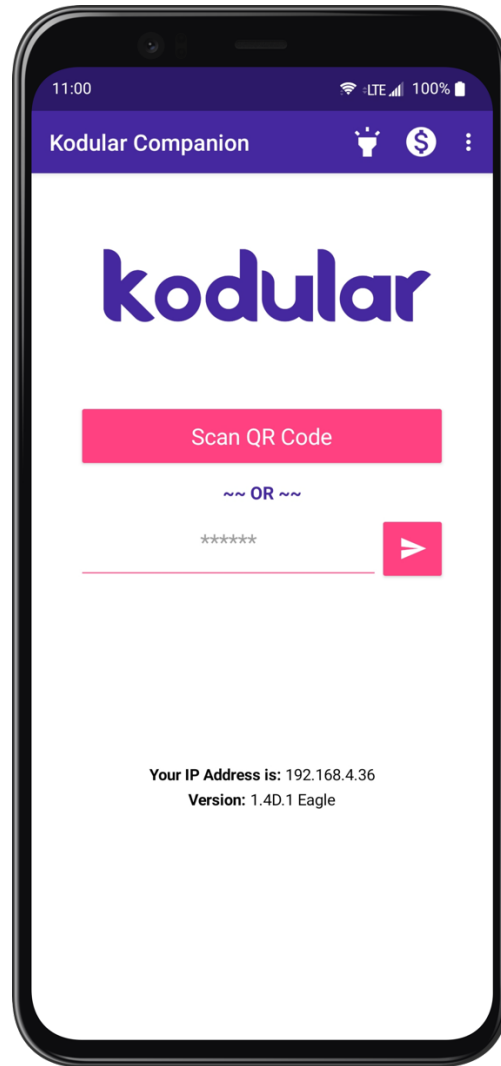
Langkah 3: Anda telah berhasil menautkan Ponsel Android Anda ke Kodular Creator.

Instalasi Kodular Companion di Ponsel / Emulator Android

Langkah 1: Unduh Companion Kodular dari Google Play Store.



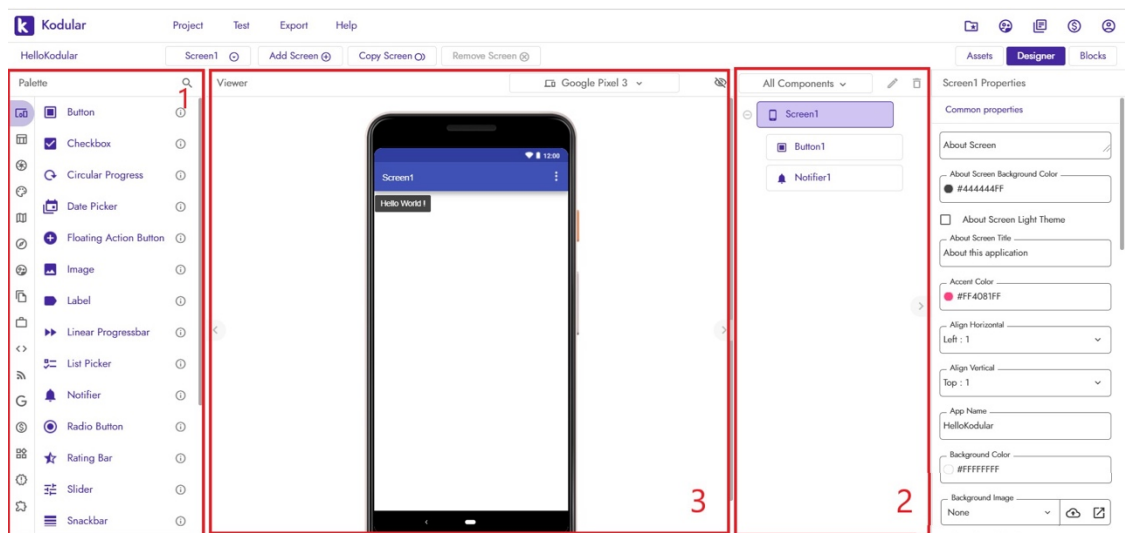
Langkah 2: Buka aplikasi



MEMBANGUN APLIKASI PERTAMA “HELLO WORLD”

Langkah-langkah membuat hello world di kodular:

1. Arahkan ke Kodular Creator. Akan ditampilkan dengan layar Login. Masuk untuk melanjutkan.
2. Kemudian setelah itu Anda akan dibawa ke layar Projects. Proyek tidak lain adalah sebuah aplikasi.
3. Click on + New Project untuk membuat Proyek.
4. Anda dapat memasukkan nama apa pun yang Anda inginkan. Dalam tutorial ini, kita akan memberi nama aplikasi HelloKodular.
5. Klik Berikutnya. Proyek akan dibuat dan Anda akan dibawa ke halaman Desainer proyek Anda.
6. Untuk menguji aplikasi Anda saat mengembangkan, silakan lihat Live Development.



Layar yang ditampilkan di atas disebut Designer.

- Dari halaman ini, Anda dapat menambah/menghapus Komponen dan memodifikasi propertinya.
- Di sisi kiri halaman Anda, nomor 1 adalah Components Palette atau hanya Palette . Ini terdiri dari berbagai komponen yang dikelompokkan dalam kategori yang berbeda. Saat Anda perlu menambahkan komponen, Anda dapat mengklik komponen di Palette, tahan dan seret ke area Viewer di halaman Designer.
- Di sisi kanan, Anda memiliki Hierarki Komponen dan Panel Properti Desainer (yang terlihat saat Anda memilih komponen). Hirarki Komponen menunjukkan komponen

yang terlihat ditambahkan ke proyek saat ini. Panel Properti Desainer akan terlihat saat mengklik komponen dan menampilkan properti komponen tersebut.

- Di tengah adalah Penampil yang memberikan tampilan aplikasi yang diolok-olok seperti yang terlihat di perangkat nyata. Untuk menambahkan komponen ke proyek, seret komponen dari Palette dan letakkan di dalam Viewer. Di bawah Penampil, Panel Komponen Tidak Terlihat hadir, yang menunjukkan semua Komponen Tidak Terlihat yang ditambahkan ke proyek saat ini.
- Di bagian atas halaman, terdapat berbagai menu dan opsi.

Membangun Aplikasi

Sekarang kita sudah familiar dengan antarmuka, kita bisa mulai bekerja dan membuat aplikasi sederhana, dengan Tombol yang menampilkan teks "Hello World".

- Langkah 1 Temukan komponen Button dari Palette di bawah kategori User Interface. Klik di atasnya, dan seret ke Viewer, sambil tetap menekan tombol mouse ke bawah. Ini adalah bagaimana Anda menambahkan komponen ke proyek.
- Langkah 2 Sekarang klik pada Button1 yang ditampilkan pada Hirarki Komponen. Secara default, ini adalah bagaimana komponen diberi nama, Anda dapat mengganti namanya jika diperlukan. Panel Properti untuk Tombol akan terlihat sekarang.
- Langkah 3 Sekarang, temukan properti Text di panel itu dan ubah menjadi Hello World!. Jika Anda telah menghubungkan pengiring Anda dan mengujinya, maka sekarang perubahan tersebut akan terlihat pada perangkat.
- Langkah 4 Sekarang, Anda dapat memilih menu Ekspor dari bilah menu atas untuk membangun proyek Anda menjadi APK, yang dapat diinstal di perangkat Android.
- Langkah 5 Setelah APK dibuat, APK akan diunduh ke komputer Anda, atau kode batang ke tautan akan ditampilkan berdasarkan opsi yang Anda pilih. Anda dapat mentransfer APK ke perangkat Android dan Anda dapat menginstal aplikasi Anda sendiri di perangkat!

LATIHAN

Buatlah aplikasi pertama "Hello World" dengan tutorial yang sudah dijelaskan sebelumnya.

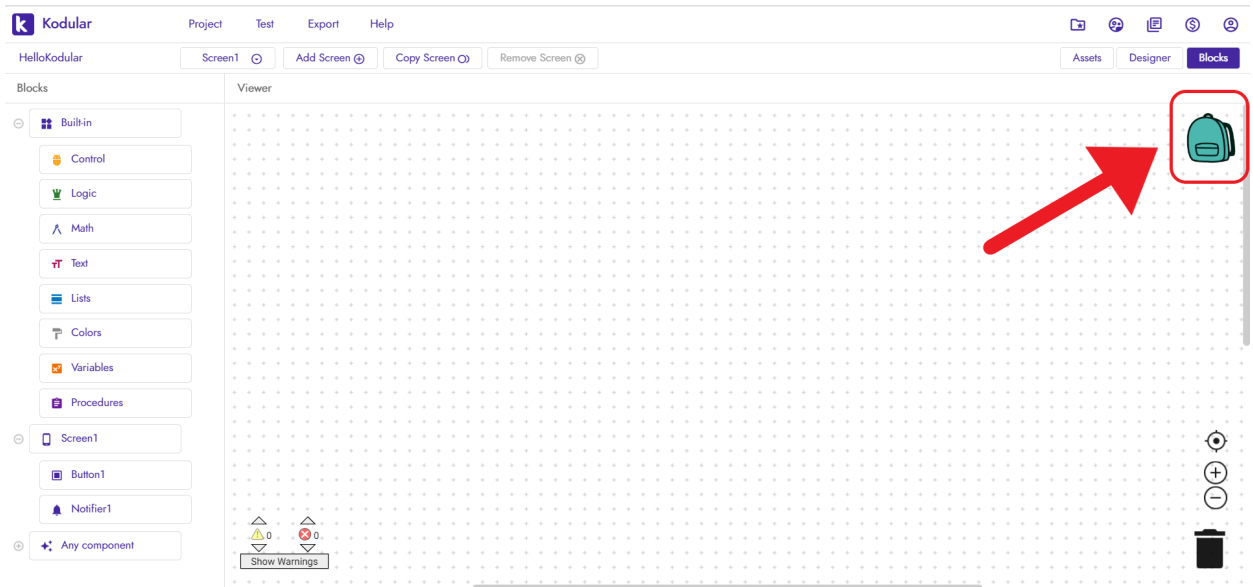


Backpack: Salin dan Tempel Blok ke Layar dan Proyek yang Berbeda

TheBackpack memungkinkan Anda untuk membawa blok di seluruh repositori proyek Anda, memungkinkan blok untuk ditransfer antar proyek dan antar layar. Isi Backpackare terus-menerus di seluruh login. Saat Anda masuk ke MIT App Inventor, Anda akan menemukan blok yang Anda tinggalkan dari sesi terakhir Anda.

Bagaimana Backpack Bekerja?

Backpack adalah fitur salin dan tempel yang berfungsi antara proyek dan layar. Ikon Backpack terletak di sudut kanan atas ruang kerja.



Menyalin Blok. Blok dari ruang kerja dapat diseret dan dijatuhkan ke dalam Backpack. Ini adalah operasi penyalinan -- blok disalin (digandakan) di dalam ransel. Saat balok dijatuhkan ke dalam Ransel, animasi dan suara akan muncul untuk mengonfirmasi pengguna bahwa operasi itu berhasil. Blok tidak dihapus dari ruang kerja.

Melihat Isi Ransel. Isi Ransel dapat dilihat dengan mengklik ikonnya (sudut kanan atas ruang kerja). Flyout yang dapat digulir akan ditarik keluar dari tepi kanan ruang kerja, menampilkan konten Ransel. Flyout Ransel bekerja dengan cara yang sama seperti flyout di laci Block.

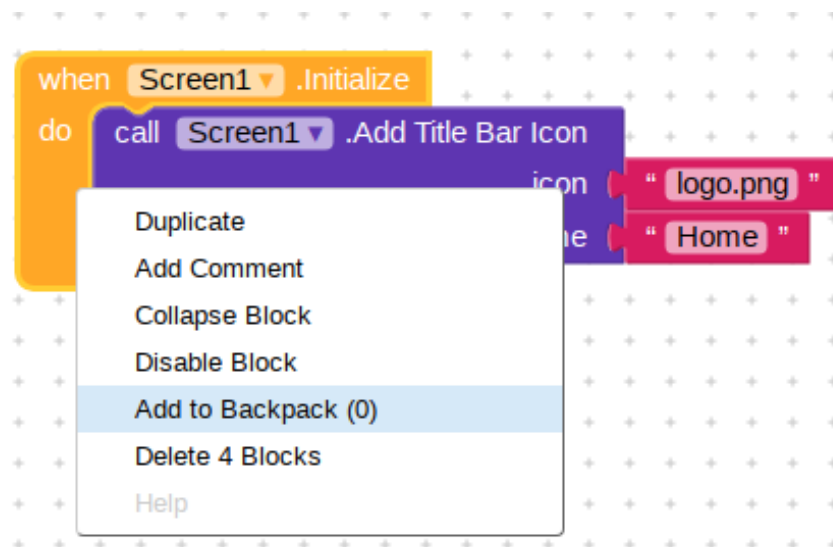
Untuk menutup flyout tanpa menempelkan blok apa pun, Anda dapat mengklik ruang kerja atau mengklik ruang kosong apa pun di Ransel tanpa blok yang disorot. Perhatikan bahwa jika Anda mengklik blok yang disorot di flyout, blok itu akan ditempelkan ke ruang kerja.

Menempelkan Blok. Blok dapat ditempelkan dari ransel ke ruang kerja saat ini dengan mengklik ikon Ransel (sudut kanan atas ruang kerja) dan menyeret blok dari flyout ke lokasi yang diinginkan di ruang kerja.

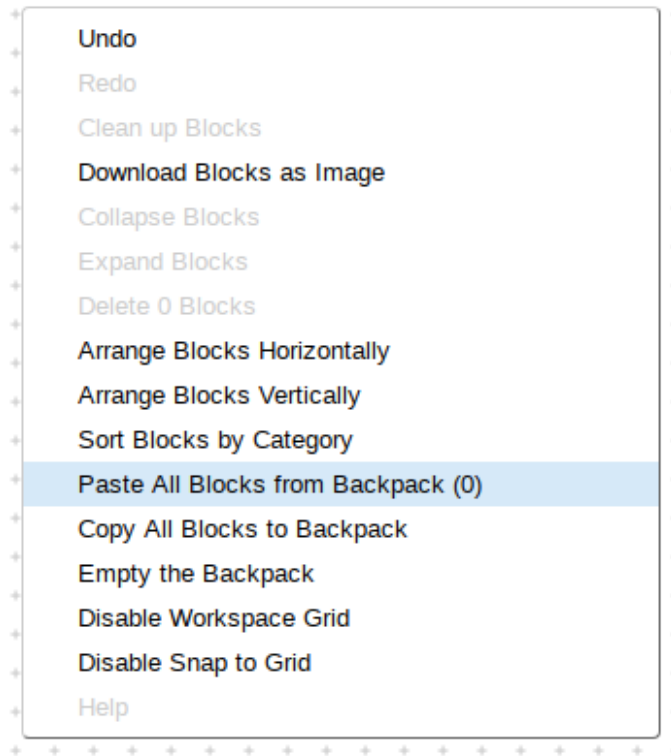
Menu Background

Selain fungsionalitas drag-and-drop, Anda dapat memilih fungsi Backpack dengan mengklik kanan pada blok atau pada ruang kerja, seperti yang ditunjukkan pada gambar berikut.

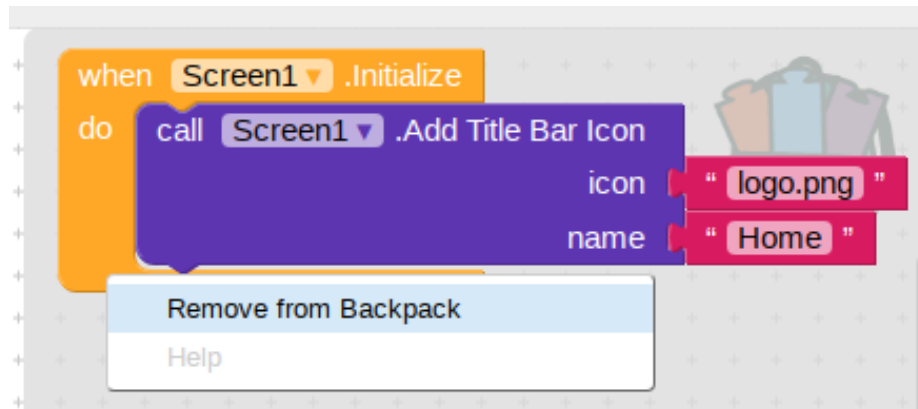
klik kanan pada sebuah blok



klik kanan pada workspace



klik kanan pada blok di backpack



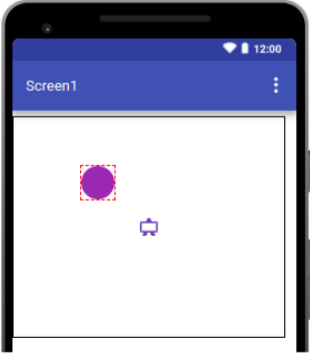
K

Bounce Sprite Off Canvas Edge

Make a ball bounce when it hits an edge of a Canvas.

Edge = 1

Edge = -3



Edge = 3

Edge = -1

Edges are assigned values (1,3,-1,-3)
For top, right, bottom, left

Getting Ready

You will need the following components in your design screen:

- * Canvas
- * Sprite

Blocks Editor

```

when Ball1 .Edge Reached
  edge
do call Ball1 .Bounce
  edge get edge
                    
```

What Does It Mean?


The **Ball1.EdgeReached** event will detect when the Ball sprite hits the edge of the Canvas and provides an argument **edge**. Each **edge** of a Canvas holds a numeric value. So just feed back the same value **edge** into the **Bounce** call.

Ball1.Bounce causes the ball to bounce and move in the opposite direction off the wall (**Ball1.Heading** is changed by 180 degrees).

K

Adding Sound

Add sound by either adding short audio files like a crash sound when two objects collide or longer audio files like background music.



Getting Ready

You will need the following components in your design screen:

- * Button
- * Sound
- * Player

Blocks Editor

```

when Button1 .Click
do call Meow .Play

when Screen1 .Initialize
do call BackgroundMusic .Start
                    
```

What Does It Mean?

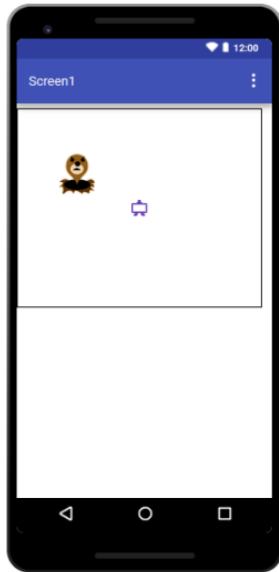
When **Button1** is clicked the **Meow** sound will play.

When **Screen1** comes up on your phone the **BackgroundMusic** will start playing.

k

Movement on a Timer

Move an ImageSprite with the passing of time.



Getting Ready

You will need the following components in your design screen:

- * Canvas
- * ImageSprite
- * Clock

Blocks Editor

```
when Clock1.Timer
do call Image_Sprite1.Move To
  x Image_Sprite1.X + 10
  y Image_Sprite1.Y
```

What Does It Mean?

MoveTo moves an object to an absolute location on the canvas, not a relative amount.

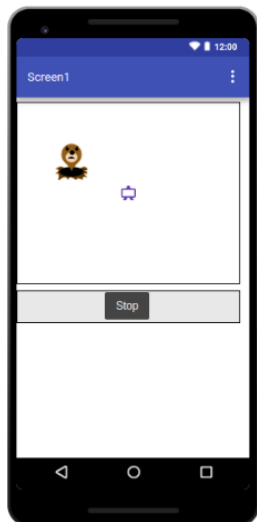
Clock1 has an Interval that specifies how often the Timer should go off and the Timer event should be called.

Whenever the **Clock1.Timer** fires the Sprite will move to the right 10 pixels since the Sprite's Interval is 1 second.

k

Start/Stop Timed Movement

This allows the end user to touch a button to start and stop an ImageSprite moving with the passing of time.



Getting Ready

You will need the following components in your design screen:

- * Canvas
- * ImageSprite
- * Clock
- * Button

Blocks Editor

```
when Clock1.Timer
do call Image_Sprite1.Move To
  x Image_Sprite1.X + 10
  y Image_Sprite1.Y

when StartStopButton.Click
do if Clock1.Timer Enabled
then set Clock1.Timer Enabled to false
set StartStopButton.Text to "Start"
else set Clock1.Timer Enabled to true
set StartStopButton.Text to "Stop"
```

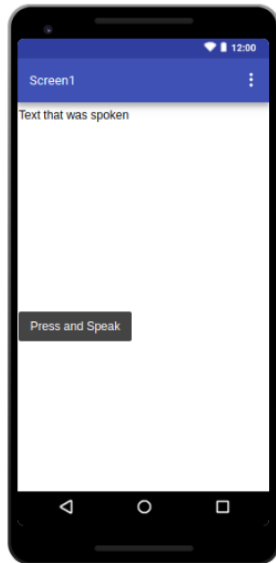
What Does It Mean?

When the StartStopButton.Click is touched, if the clock is enabled then stop the timer and display start on the button. This will stop the Sprite's movement. The opposite will happen when the clock is disabled.

k

Speech Recognition

Display the text of what is being said on the phone screen.



Getting Ready

You will need the following components in your design screen:

* Label, Speechrecognizer and Button

Blocks Editor

```

when PressAndSpeakButton Click
do call Speech_Recognizer1 Get Text

when Speech_Recognizer1 After Getting Text
result
do set Label1 Text to Speech_Recognizer1 Result

when Speech_Recognizer1 Before Getting Text
do set Label1 Text to ""
    
```

What Does It Mean?

When the **PressAndSpeakButton** is clicked the **SpeechRecognizer** event is called and is ready for you to speak.

The **BeforeGettingText** event will be triggered before speech has been received and recognized.

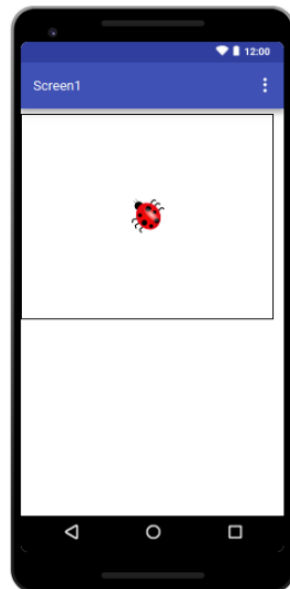
Then the **Label** will display no text on the screen.

The **AfterGettingText** event will be triggered once speech has been received and recognized. Then the **Label** will display the text on the screen.

k

Movement With Sensors

Move an ImageSprite by tilting your phone.



Getting Ready

You will need the following components in your design screen:

* Canvas, ImageSprite, OrientationSensor, Clock

Blocks Editor

```

to MoveBug
do set Bug Heading to Orientation_Sensor1 Angle
do set Bug Speed to Orientation_Sensor1 Magnitude * 100

when Clock1 Timer
do call MoveBug
    
```

What Does It Mean?

A procedure called **MoveBug** was created that moves the bug in the direction that the phone is tilted.

The **OrientationSensor.Angle** is used to tell the bug which direction to move based on what angle your phone is tilted.

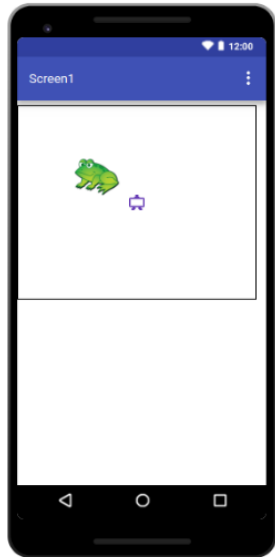
The **OrientationSensor.Magnitude** is used to tell the bug what speed to move based on how much tilt you are putting on your phone.

Whenever the **Clock1.Timer** fires, the event **MoveBug** will be called.

k

Random Numbers

Generate random numbers to make ImageSprites appear in random (x.y) coordinate locations.



Getting Ready

You will need the following components in your design screen:

- * Canvas
- * ImageSprite
- * Clock

Blocks Editor

```

when Clock1.Timer
do call Frog.Move To
    x random integer from 1 to 300
    y random integer from 1 to 400
    
```

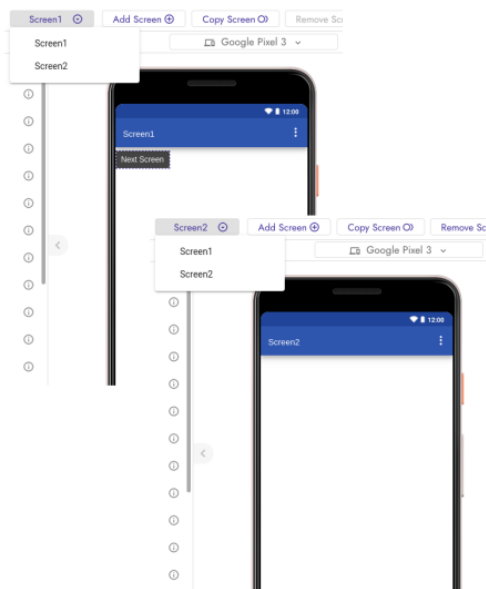
What Does It Mean?

When the **Clock1.Timer** event is triggered, then **Frog.MoveTo** moves the frog to a random coordinates between the values of 1 and 300 for the x coordinate and 1 and 400 for the y coordinate.

k

Multiple Screens

Use multiple screens in your app. Get the next screen by clicking a button.



Getting Ready

You will need the following components in your design screen:

- * Screen1: Button
- * Screen2: Label

Blocks Editor

(For screen1)

```

when Button1.Click
do open another screen screenName "Screen2"
    
```

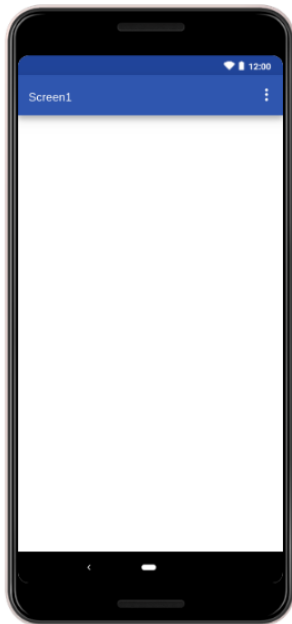
What Does It Mean?

Open another screen takes in a text block. The text inside this block is the name of another screen. When the button is clicked, Screen2 will be opened.

k

Creating Your Own Color

Create your own colors using the make a color block.



Getting Ready

No components are necessary to use make color.

Blocks Editor

```
initialize global Purple to make color make a list 157 57 252
```

```
when Screen1 .Initialize do set Screen1 . Background Color to get global Purple
```

What Does It Mean?

Make color takes in a list of 4 numbers. The first three numbers represent the RGB values. The last is the alpha or how strong the color is.

Purple is made from using 157 as R, 57 as G, 252 as B and 100 as alpha.

When Screen1 is initialized, the background color is set the color we created in the variable Purple.

k

Fling Movement

Change the heading and speed of a Sprite by flinging your fingers.



Getting Ready

You will need the following components in your design screen:

- * Canvas
- * ImageSprite

Blocks Editor

```
when PirateSprite .Fling do set PirateSprite . Heading to get heading set PirateSprite . Speed to get speed
```

What Does It Mean?

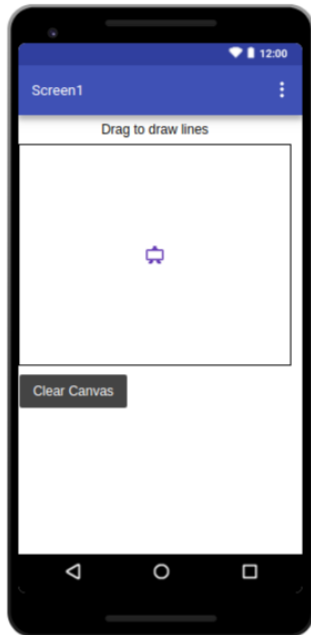
Fling detects when the user makes a fling motion with the sprite across the screen.

The user sets the heading and speed from the fling to the PirateSprite's **Heading** and **Speed**.

k

Drawing on a Canvas

Change the heading and speed of a Sprite by flinging your fingers.



Getting Ready

You will need the following components in your design screen:

- * Canvas
- * Button
- * Label

Blocks Editor

```

when Canvas1.Dragged
do call Canvas1.Draw Line
    x1 get prev X
    y1 get prev Y
    x2 get current X
    y2 get current Y

when ClearButton.Click
do call Canvas1.Clear
    
```

What Does It Mean?

When the **Canvas1.Dragged** event is triggered, a curved line will be drawn from where you finger started on the screen to where it was dragged. When the **Clear.Click** button is touched the canvas will be cleared.

k

Shaking Phone

Make something to happen when you shake your phone.



Getting Ready

You will need the following components in your design screen:

- * Image
- * Sound
- * AccelerometerSensor
- * Label

Blocks Editor

```

when Accelerometer_Sensor1.Shaking
do call Meow.Play
    call Meow.Vibrate
        millisecs 20
    
```

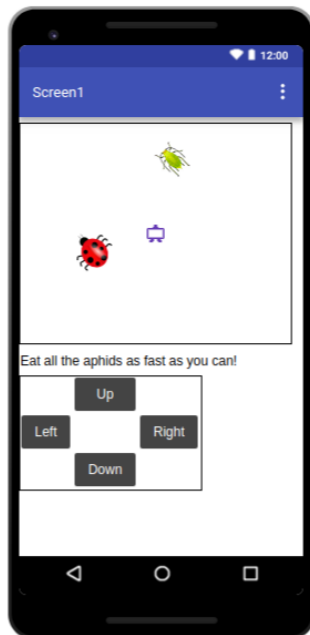
What Does It Mean?

The **AccelerometerSensor.Shaking** event will detect when the phone is shaking and then the Meow sound will play and the phone will vibrate for 20 milliseconds.

k

Collision Detection

Make something happen when one Sprite collides with another.



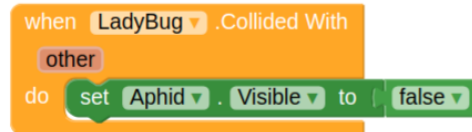
Getting Ready

You will need the following components in your design screen:

- * Canvas
- * Sprite
- * Button
- * Label

Blocks Editor

HINT: To make your ladybug move by clicking buttons, check out the Movement cards.



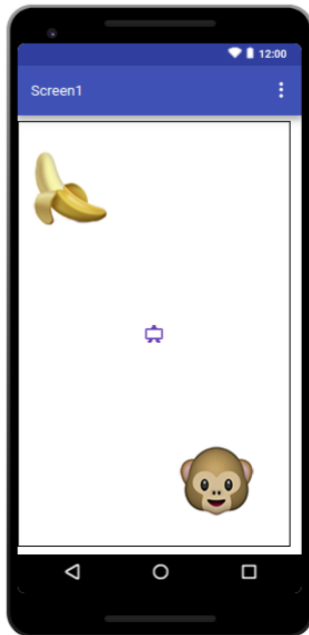
What Does It Mean?

The **LadyBugCollidedWith** event is triggered when the Ladybug touches the Aphid. Then this will make the Aphid disappear.

k

Drag A Sprite

Move a sprite side to side by dragging your finger



Getting Ready

You will need the following components in your design screen:

* Canvas, Sprite

Blocks Editor

```

when MonkeySprite .Dragged
  start X start Y prev X prev Y current X current Y
do
  call MonkeySprite .Move To
    x get current X
    y MonkeySprite . Y
    
```

What Does It Mean?

While the user is dragging the sprite, **MonkeySprite.Dragged** gets called multiple times. Each call has 6 arguments:

* **startX** and **startY**, where the user initially touched the screen.

* **currentX** and **currentY**, where the user is currently touching

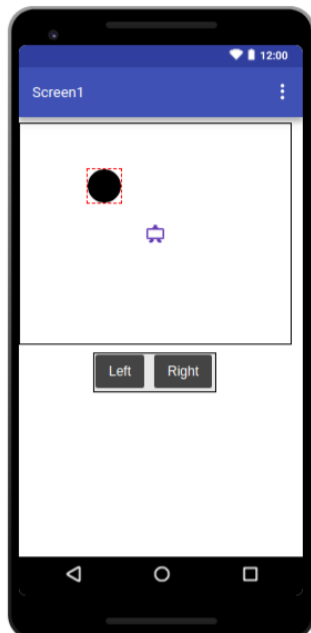
* **prevX** and **prevY** hold whatever values were in **currentX** and **currentY** on the previous call to the event. (On the first call of this event, **prevX** and **prevY** are the same as **startX** and **startY**.)

When the user drags the MonkeySprite, it will be moved to the new X location, **currentX**, of the drag. The Y stays the same so the monkey can only move in the X direction.

k

Movement With Buttons

Move a sprite by touching a button.



Getting Ready

You will need the following components in your design screen:

* Canvas, Sprite and Button

Blocks Editor

```

initialize global speed to 1
when Left .Click
do
  set Ball1 . X to Ball1 . X - get global speed
when Right .Click
do
  set Ball1 . X to Ball1 . X + get global speed
    
```

What Does It Mean?

Define a speed variable to 1 to set the how far the sprite will move each time the button is clicked.

The Left.Click event moves the ball to the left every time the button is touched.

The Right.Click event moves the ball to the right every time the button is touched.

MENGGUNAKAN DYNAMIC CARDS DI APLIKASI

Dalam panduan ini, Anda akan mempelajari cara menggunakan kartu dinamis di aplikasi Anda. Anda akan membuat aplikasi berita sederhana yang berisi daftar 20 artikel berita teratas yang sedang tren secara global.

Pre-requisites

Berikut adalah beberapa hal yang harus Anda ketahui sebelum Anda dapat memulai dengan panduan ini:

- Bekerja dengan API dan Komponen Web
- Bekerja dengan komponen JSON
- Bekerja dengan Label, Tombol, dan Gambar Dinamis

Langkah pertama

Kita akan menggunakan News API untuk mengambil berita utama yang sedang tren. Buka <https://newsapi.org/> untuk mendapatkan kunci API.

Seperti inilah contoh responsnya:

```
{
  "status": "ok",
  "totalResults": TOTAL_NUMBER_OF_ARTICLES,
  "articles": [
    {
      "source": {
        "id": "SOURCE_ID",
        "name": "SOURCE_NAME"
      },
      "author": AUTHOR_NAME,
      "title": "ARTICLE_TITLE",
      "description": "ARTICLE_SUMMARY",
      "urlToImage": "ARTICLE_IMAGE",
      "publishedAt": "ARTICLE_PUBLISH_DATE",
      "content": "ARTICLE_CONTENT"
    },
    {
      ...
    }
  ]
}
```

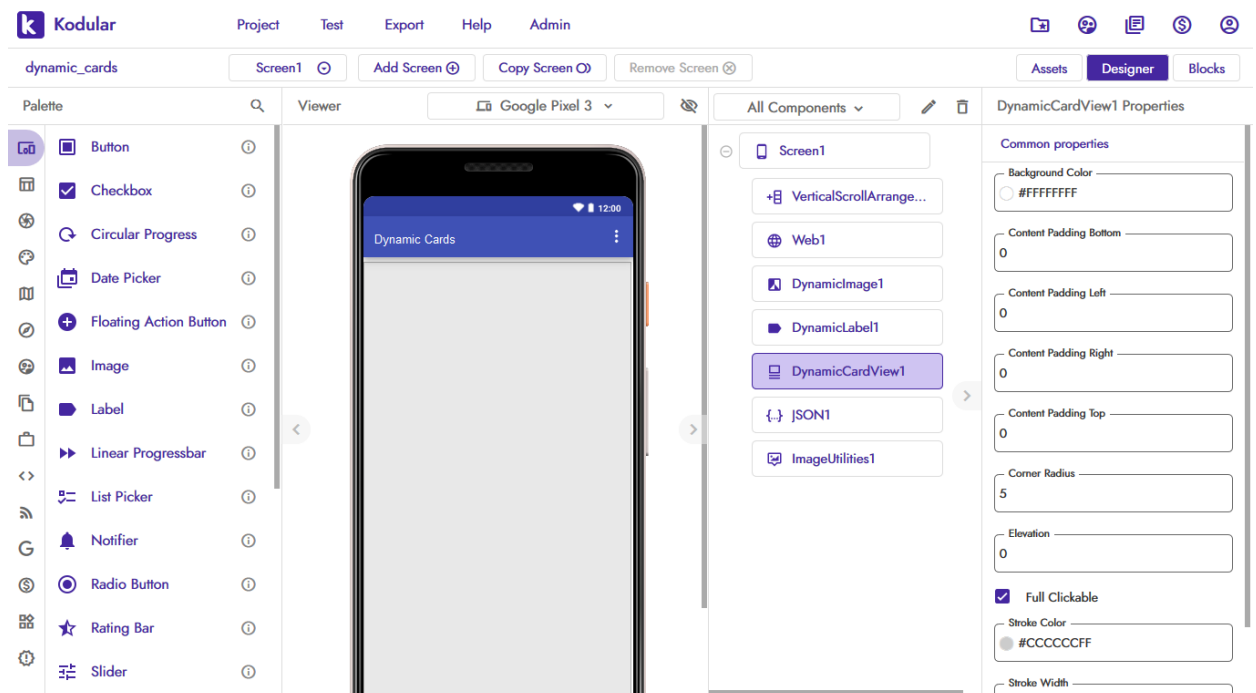
Kita akan membutuhkan judul artikel, deskripsi, dan Gambar url untuk mengisi kartu. Kita akan menggunakan komponen JSON untuk mengekstrak data dari API.

Mendesain Aplikasi

Seperti biasa, kita akan mulai dengan proyek baru. Seret dan lepas komponen Pengaturan Gulir Vertikal, Web, JSON, Label Dinamis, Gambar Dinamis, dan Tampilan Kartu Dinamis.

Atur properti desainer tinggi dan lebar Vertical Scroll Arrangement ke Fill Parent, dan properti Align Vertical ke Center. Di properti desainer Tampilan Kartu Dinamis, atur Radius Sudut ke 5 dan Elevasi ke 0. Terakhir, aktifkan properti desainer Penuh yang Dapat Diklik. Terakhir, tambahkan komponen Utilitas Gambar yang akan kita gunakan untuk memuat gambar secara tidak sinkron

Seperti inilah seharusnya desainer Anda terlihat



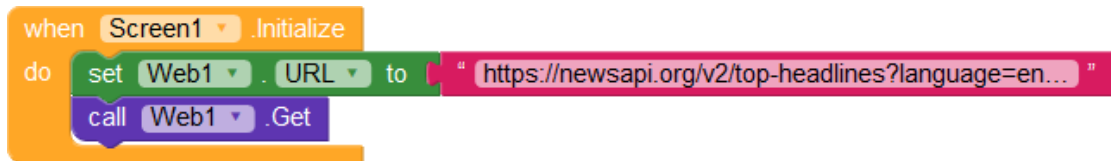
20 artikel berita teratas akan ditampilkan sebagai Tampilan Kartu Dinamis, yang akan dimuat di dalam Pengaturan Gulir Vertikal. Setiap kartu akan menyertakan judul artikel, deskripsi singkat, dan gambar header.

Coding the blocks

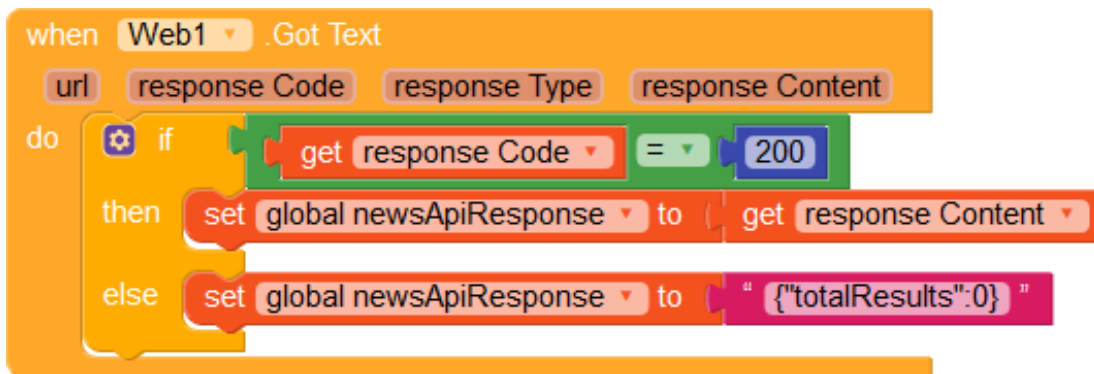
Kita akan membutuhkan variabel untuk menyimpan respons JSON dari API. Ini bisa menjadi variabel Teks sederhana seperti itu.

initialize global `newsApiResponse` to `""`

Data API akan diambil saat layar diinisialisasi. Setel url komponen Web ke url API, dan lakukan permintaan GET.

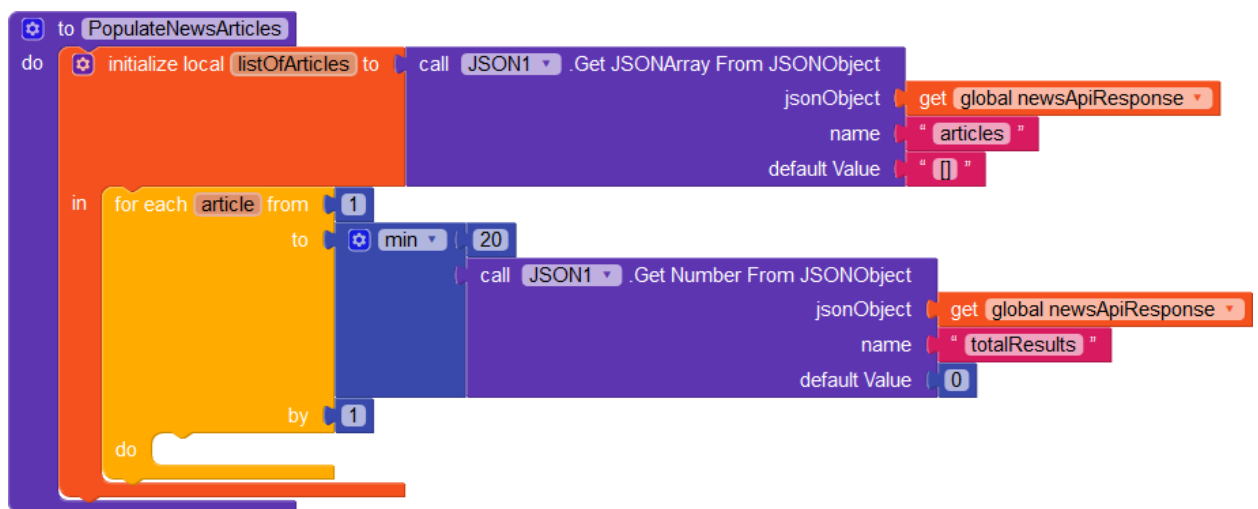


Saat komponen Web telah mengambil data, atur konten respons ke variabel global yang Anda buat. Jika ada masalah saat mendapatkan konten, kita menyetel variabel ke JSON kesalahan khusus. Ini agar penguraian JSON, yang harus dilakukan nanti, tidak menimbulkan kesalahan.

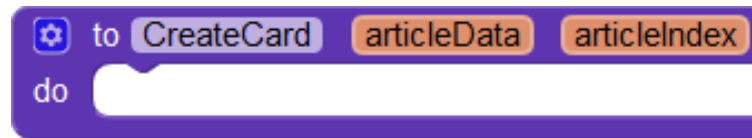


Selanjutnya, kita membutuhkan prosedur yang mengisi Vertical Scroll Arrangement dengan kartu. Prosedur ini akan diulang 20 kali jika ada lebih dari 20 artikel berita dalam variabel respon api. Jika ada kurang dari 20 artikel, prosedur hanya akan berulang melalui jumlah artikel dalam respons api.

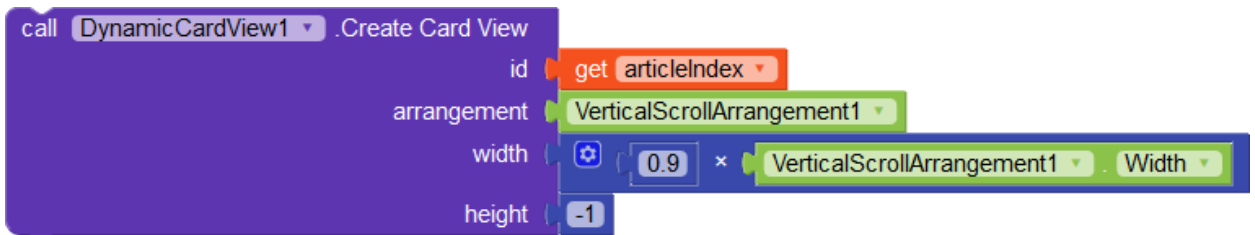
Kita menyebut prosedur ini di dalam blok Got Text komponen Web.



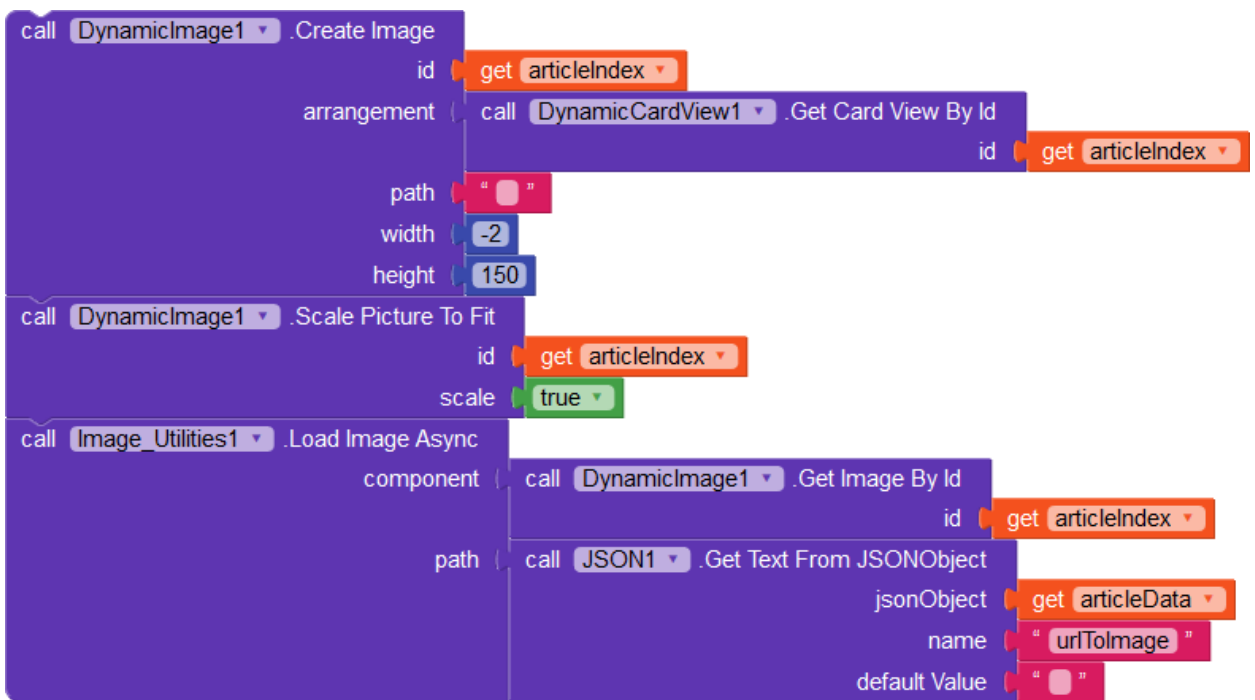
Selanjutnya, kita membuat prosedur lain yang mengambil objek JSON dan indeks array sebagai parameter. Prosedur ini akan membuat kartu untuk kita dan menambahkannya ke Vertical Scroll Arrangement.



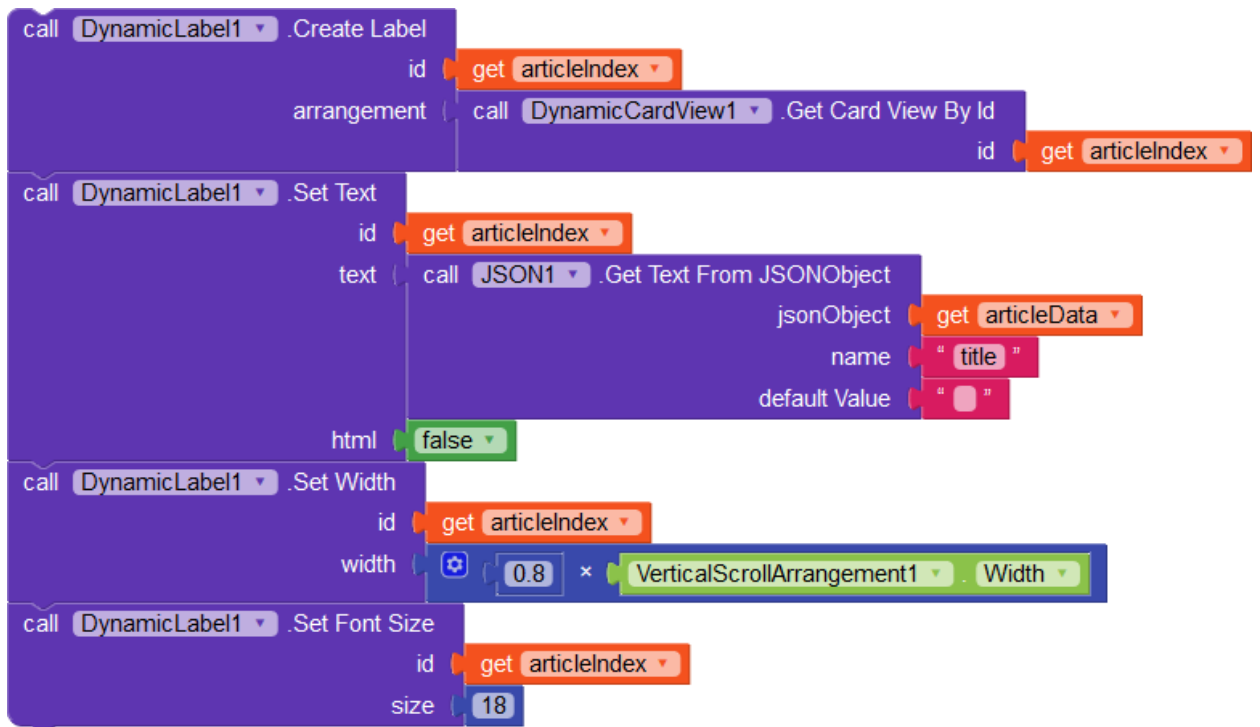
Untuk setiap artikel berita, pertama-tama kita membuat tampilan kartu yang menampung semua konten yang ingin kita tampilkan. Gunakan blok Buat Tampilan Kartu di dalam prosedur untuk membuat kartu dinamis untuk setiap artikel. Setiap kartu akan menjadi 90% dari lebar layar dan akan memiliki tinggi otomatis sehingga dapat diubah ukurannya sesuai dengan konten di dalamnya.



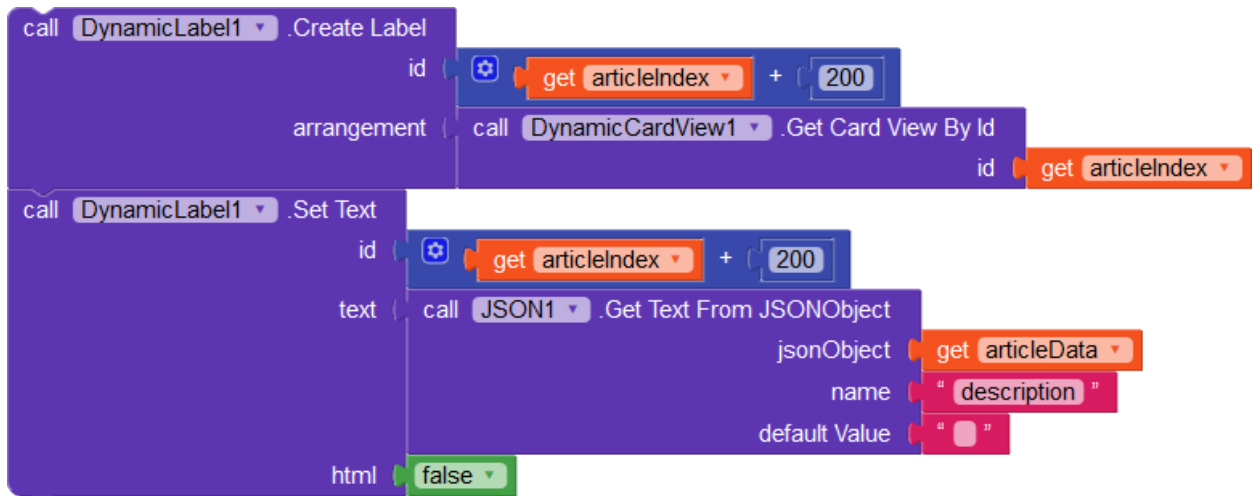
Selanjutnya, kita membuat gambar dinamis yang berisi gambar header artikel. Kita menggunakan komponen Utilitas Gambar untuk memuat gambar secara asinkron dari URL, sehingga UI tidak ketinggalan saat data sedang diambil.



Menggunakan komponen Label Dinamis, kita membuat label dinamis untuk setiap kartu yang berisi judul artikel. Karena ini judul artikel, kita mengatur ukuran font menjadi 18.



Kita menggunakan komponen Label Dinamis sekali lagi untuk membuat label untuk deskripsi. Kita menetapkan id ke nilai yang kita yakin belum digunakan. Jika Anda menampilkan 200 artikel berita teratas, maka Anda mungkin ingin memulai id Anda dari 1000 (atau nomor yang sesuai).



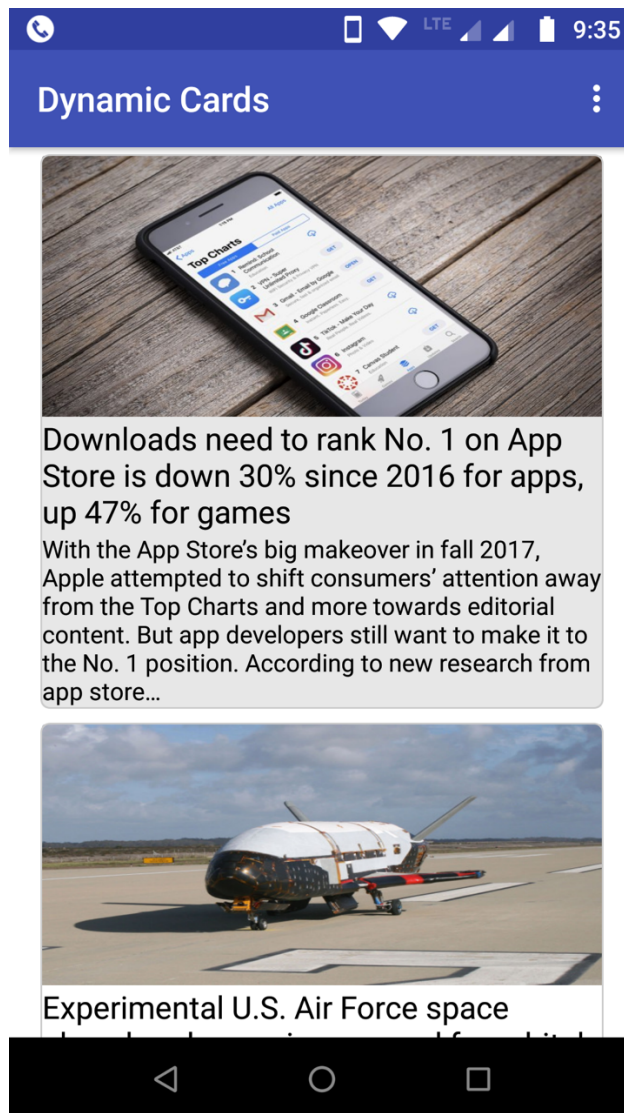
Akhirnya, kita menempatkan blok panggilan prosedur di dalam prosedur PopulateNewsArticles.

```

to PopulateNewsArticles
do
  initialize local listOfArticles to call JSON1 .Get JSONArray From JSONObject
  jsonObject get global newsApiResponse
  name "articles"
  default Value "[]"

  in for each article from 0
  to min 19
  do call CreateCard
  articleData call JSON1 .Get JSONObject From JSONArray
  jsonArray get listOfArticles
  index get article
  default Value ""
  articleIndex get article
  
```

HASIL



```

initialize global newsApiResponse to ""

when Screen1.Initialize
do
  set Web1.URL to "https://newsapi.org/v2/top-headlines?sources=tec..."
  call Web1.Get

when Web1.Got Text
  uri response Code response Type response Content
do
  if get response Code == 200
  then set global newsApiResponse to get response Content
  else set global newsApiResponse to '{"totalResults":0}'
  call PopulateNewsArticles

to PopulateNewsArticles
do
  initialize local listOfArticles to call JSON1.Get JSONArray From JSONObject
  jsonObject get global newsApiResponse
  name articles
  default Value {}

  in for each article from 0 to min 19
  do
    call JSON1.Get Number From JSONObject
    jsonObject get global newsApiResponse
    name totalResults
    default Value 0

    call CreateCard
    articleData call JSON1.Get JSONObject From JSONArray
    jsonArray get listOfArticles
    index get article
    default Value {}
    articleIndex get article

to CreateCard articleData articleIndex
do
  call DynamicCardView1.Create Card View
  id get articleIndex
  arrangement VerticalScrollArrangement1
  width 0.9 * VerticalScrollArrangement1.Width
  height 1

  call DynamicImage1.Create Image
  id get articleIndex
  arrangement call DynamicCardView1.Get Card View By Id
  id get articleIndex
  path ""
  width -2
  height 150

  call DynamicImage1.Scale Picture To Fit
  id get articleIndex
  scale true

  call ImageUtilities1.Load Image Async
  component call DynamicImage1.Get Image By Id
  id get articleIndex
  path call JSON1.Get Text From JSONObject
  jsonObject get articleData
  name imageUrlImage
  default Value ""

  call DynamicLabel1.Create Label
  id get articleIndex
  arrangement call DynamicCardView1.Get Card View By Id
  id get articleIndex

  call DynamicLabel1.Set Text
  id get articleIndex
  text call JSON1.Get Text From JSONObject
  jsonObject get articleData
  name title
  default Value ""
  html false

  call DynamicLabel1.Set Width
  id get articleIndex
  width 0.8 * VerticalScrollArrangement1.Width

  call DynamicLabel1.Set Font Size
  id get articleIndex
  size 18

  call DynamicLabel1.Create Label
  id get articleIndex + 200
  arrangement call DynamicCardView1.Get Card View By Id
  id get articleIndex

  call DynamicLabel1.Set Text
  id get articleIndex + 200
  text call JSON1.Get Text From JSONObject
  jsonObject get articleData
  name description
  default Value ""
  html false
  
```

Anda telah berhasil membuat aplikasi yang menggunakan tampilan kartu dinamis!

Berikut adalah ringkasan dari apa yang telah Anda pelajari dalam panduan ini.

- Cara membuat tampilan kartu dinamis.
- Cara menambahkan komponen dinamis lainnya di dalam tampilan kartu dinamis.
- Cara memuat gambar secara asinkron di dalam gambar dinamis.
- Cara mendekode JSON yang diambil dari API.

Langkah selanjutnya

Kita memiliki beberapa hal untuk Anda coba.

- Buka artikel lengkap dalam tampilan web saat kartu diklik.
- Alihkan visibilitas deskripsi kartu saat diklik
- Bereksperimenlah dengan berbagai ukuran font dan tipografi untuk judul dan deskripsi.
- Tambahkan lebih banyak informasi ke setiap kartu, seperti nama penulis dan tanggal penerbitan.

MELAKUKAN UPDATE DENGAN FIREBASE REMOTE CONFIG

Dalam panduan ini, Anda akan mempelajari cara memperbarui aplikasi secara bertahap menggunakan komponen Firebase Remote Config. Dengan pembaruan bertahap, Anda dapat mengontrol pengguna mana yang melihat pembaruan berdasarkan negara, usia, dll., atau memilih untuk menampilkan pembaruan hanya kepada persentase pengguna yang dipilih secara acak.

Pembaruan bertahap berguna saat memperkenalkan fitur baru, terutama untuk proyek tanpa program pengujian beta khusus. Anda akan dapat mengumpulkan umpan balik yang berguna dari grup pengujian acak Anda sebelum rilis Anda tersedia untuk semua orang. Selain itu, Anda akan dapat mengembalikan pembaruan "gagal" tanpa menyebabkan masalah besar

Mengapa Firebase Remote Config?

Firebase Remote Config memungkinkan Anda mengaktifkan dan menonaktifkan fitur untuk pengguna tertentu tanpa harus memperbarui aplikasi. Fitur-fitur ini dapat berupa minor (seperti penyedia login sosial baru) dan mayor (seperti desain ulang total aplikasi Anda). Remote Config sangat mirip dengan sistem pembaruan OTA yang digunakan produsen perangkat Android untuk memperbarui versi Android Anda.

Anda akan membuat aplikasi sederhana dengan tata letak tab. Dengan menggunakan komponen Firebase Remote Config, Anda kemudian akan menampilkan navigasi bawah alih-alih tata letak tab kepada 20% pengguna Anda. Anda juga akan mempelajari cara merilis pembaruan ke semua pengguna, dan juga mengembalikan perubahan.

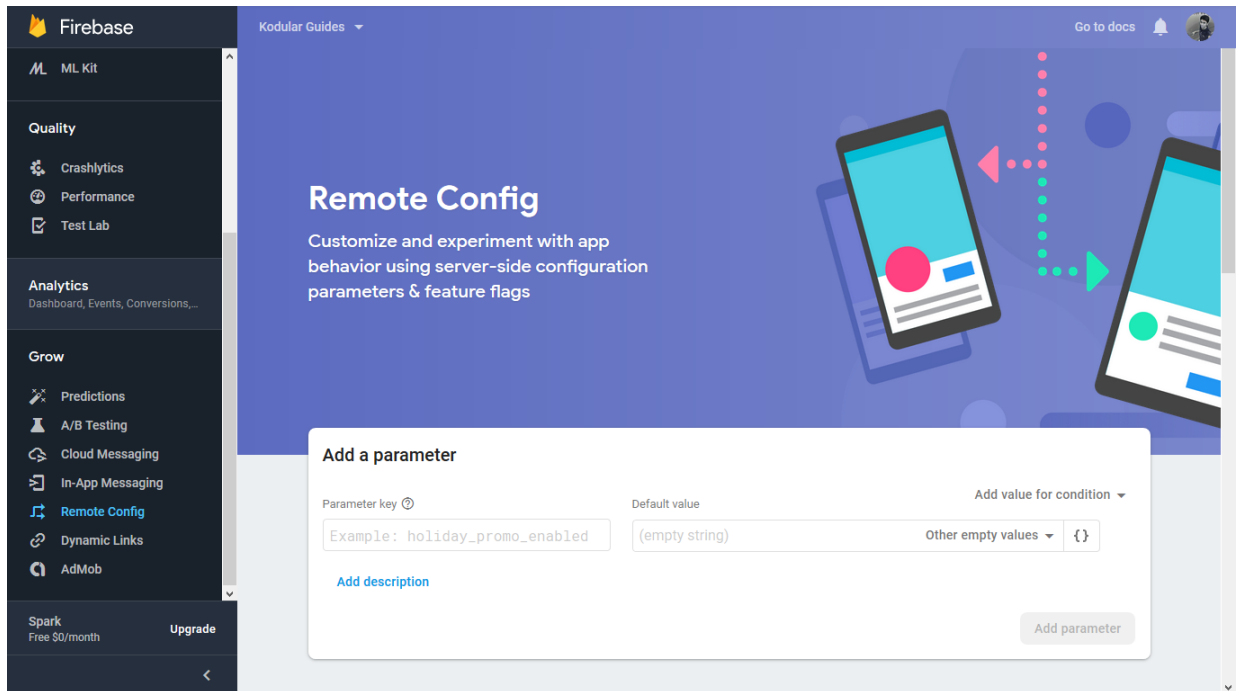
Berikut adalah beberapa hal yang harus Anda ketahui sebelum Anda dapat memulai dengan panduan ini.

- Bekerja dengan Firebase
- Bekerja dengan komponen Tata Letak Tab dan Navigasi Bawah

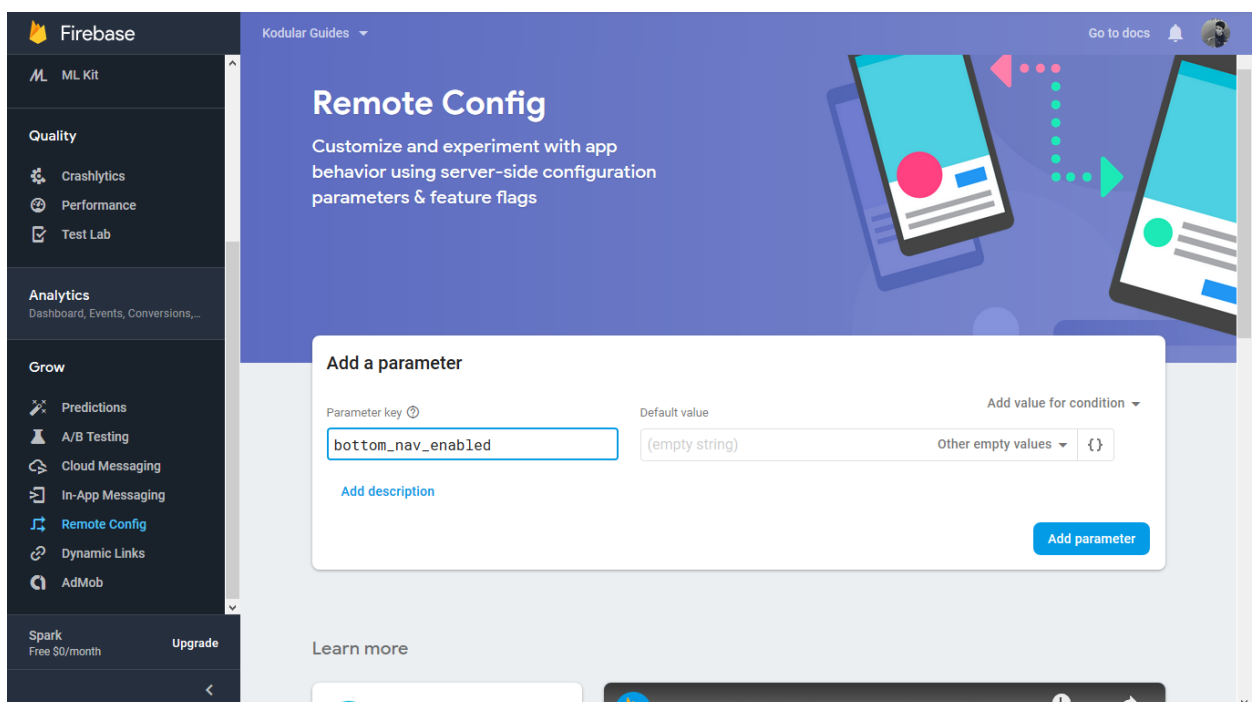
Langkah pertama

Buka Konsol Firebase Anda untuk menyiapkan Remote Config untuk proyek Firebase Anda. Jika Anda tidak memiliki proyek yang sudah ada, buat yang baru.

Gunakan menu navigasi di sebelah kiri untuk menuju ke bagian Remote Config.

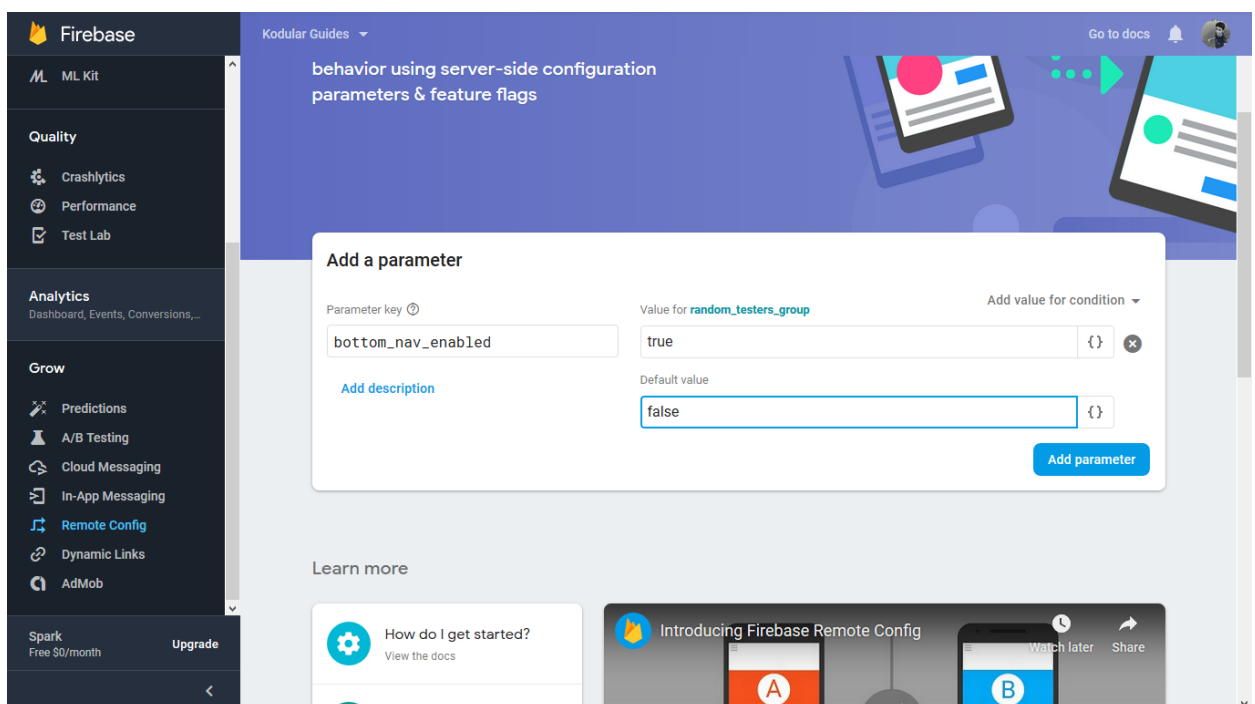


Firestore Remote Config berfungsi menggunakan parameter dan nilai. Aplikasi wasiat Anda akan mengambil nilai parameter dan kemudian memutuskan bagaimana melanjutkan. Firestore dapat mengirim nilai yang berbeda ke pengguna yang berbeda, berdasarkan beberapa kondisi yang dapat Anda tetapkan. Dalam panduan ini, kita akan membuat parameter yang disebut `bottom_nav_enabled`, yang mengembalikan `false` untuk semua pengguna kecuali 20% yang dipilih secara acak.

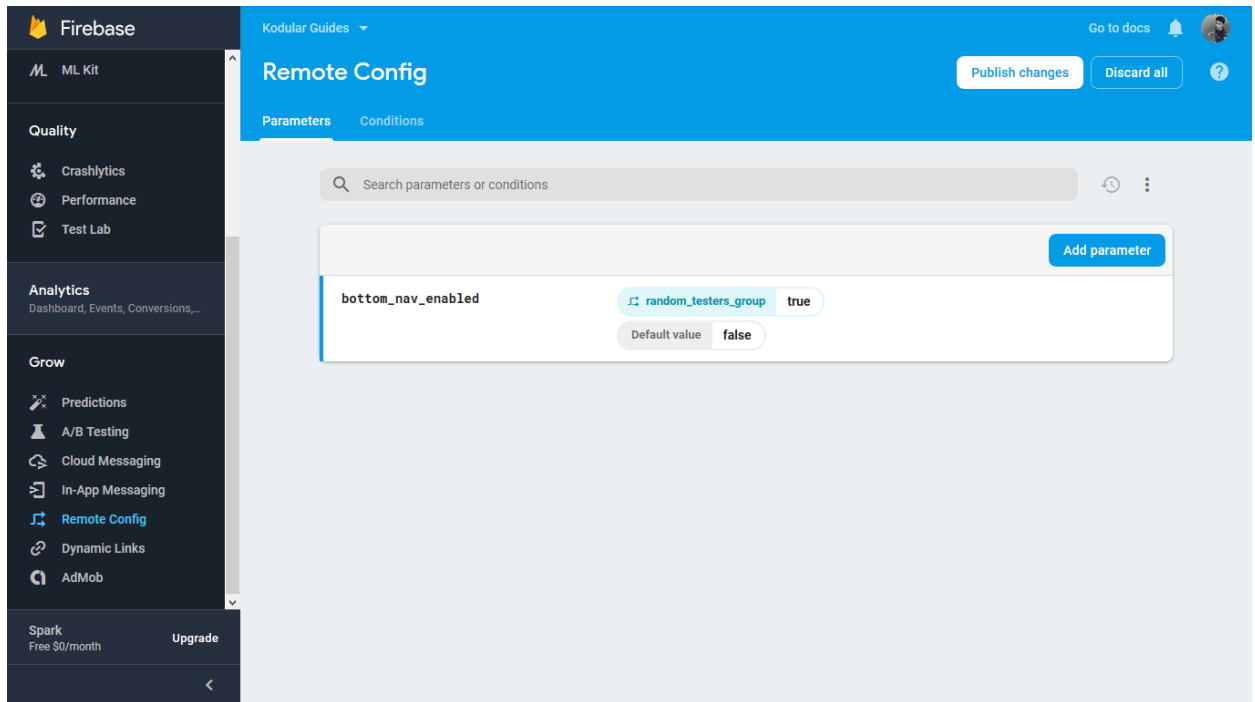


Selanjutnya, tambahkan kondisi ke parameter Anda. Setiap kondisi memiliki nama, sehingga dapat diidentifikasi dengan mudah ketika beberapa digunakan. Kita akan memberi nama kondisi kita `random_testers_group`. Ada beberapa batasan untuk dipilih, tetapi untuk panduan ini, kita akan membatasi diri pada "Pengguna dalam persentil acak".

Your parameter can now have two values: one for those in the `random_testers_group`, and another for everybody else. Set the value to "true" for the testers, and "false" for others.



Finally, click on "Add parameter", and then click "Publish changes".



Itu saja untuk bagian Firebase. Sekarang mari kita beralih ke membangun aplikasi Anda.

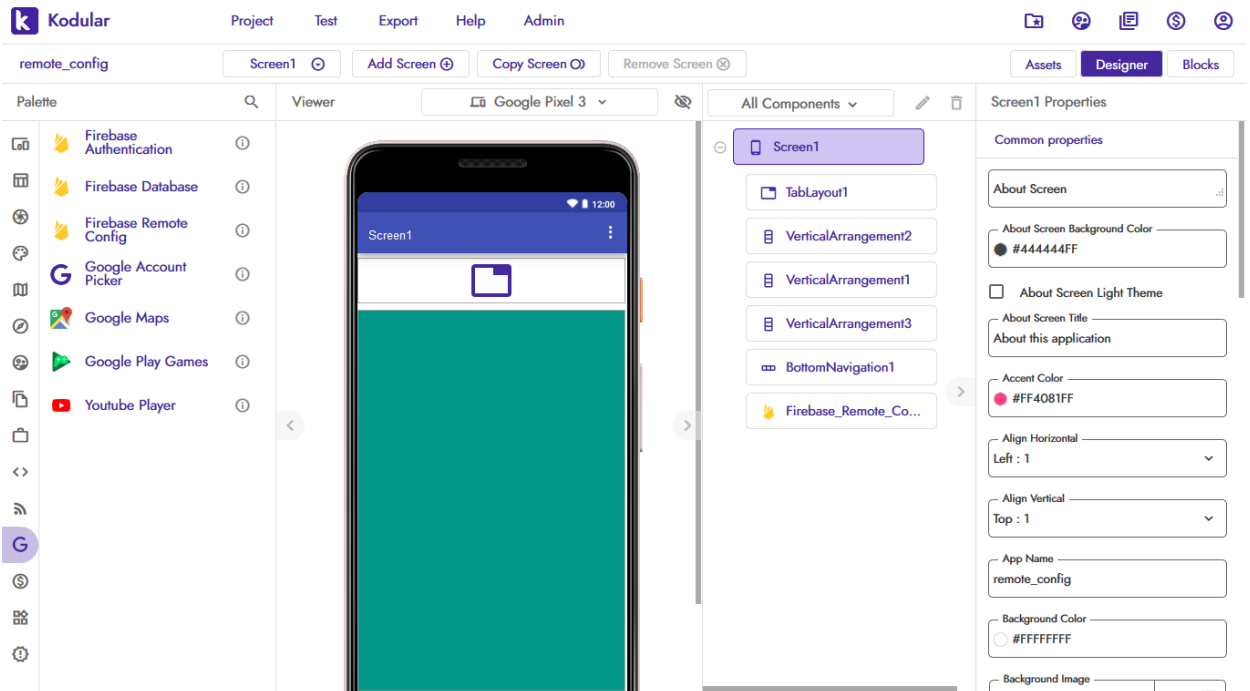
Mendesain aplikasi

Seperti biasa, kita akan mulai dengan proyek baru. Seret dan lepas komponen Tata Letak Tab, Navigasi Bawah, dan Firebase Remote Config. Juga tambahkan tiga komponen Pengaturan Vertikal ke Layar. Mereka akan berfungsi sebagai "sub-tampilan" aplikasi yang berbeda yang dapat diakses menggunakan Tata Letak Tab atau Navigasi Bawah.

Berikan setiap Pengaturan Vertikal warna latar belakang yang berbeda, dan atur tinggi dan lebarnya ke Fill Parent. Pastikan hanya satu pengaturan yang terlihat pada satu waktu, jadi atur properti Visible dari dua pengaturan ke False.

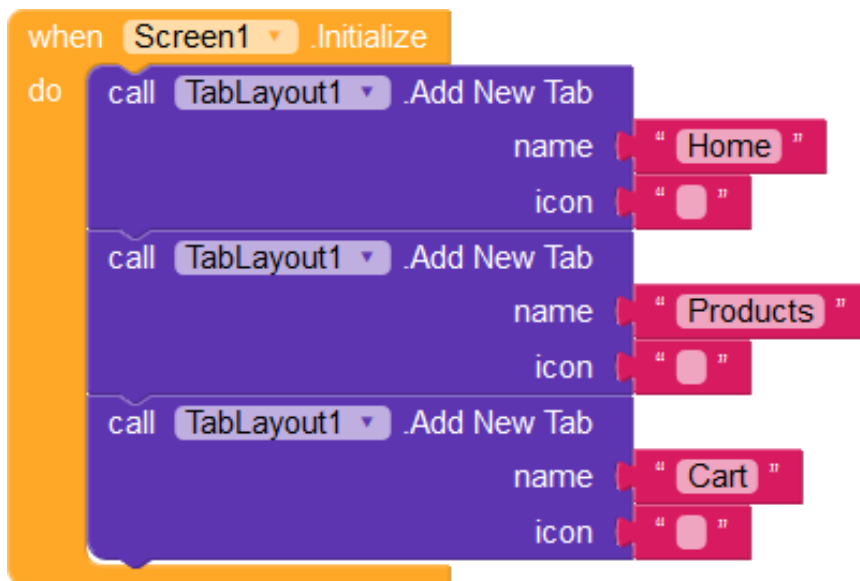
Terakhir, atur properti Visible dari komponen Navigasi Bawah ke False. Secara default, pengguna akan menavigasi aplikasi Anda menggunakan tata letak tab. Kita mengubah perilaku ini untuk grup penguji acak dari Editor Blok.

Seperti inilah seharusnya desainer Anda terlihat

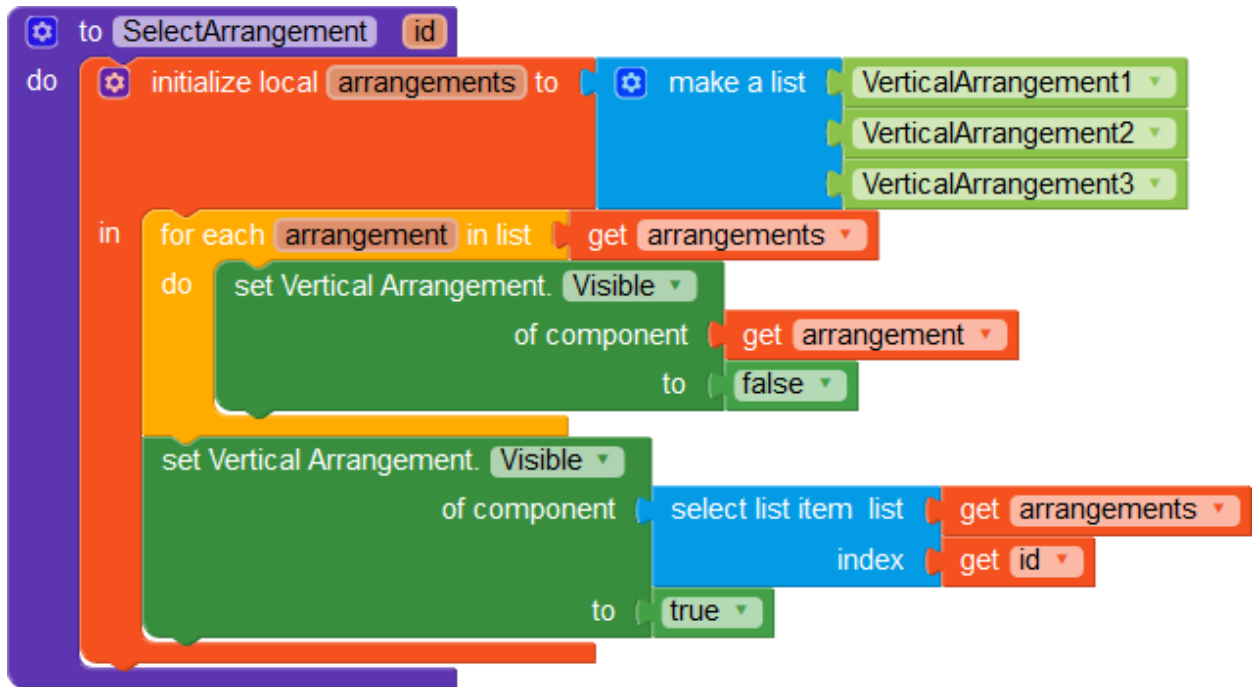


Mengkodekan blok

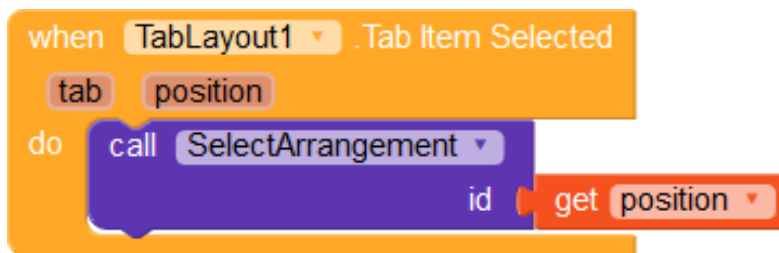
Kita pertama-tama membuat semua tab saat layar diinisialisasi. Dalam panduan ini, kita memiliki tiga tab: Beranda, Produk, dan Keranjang.



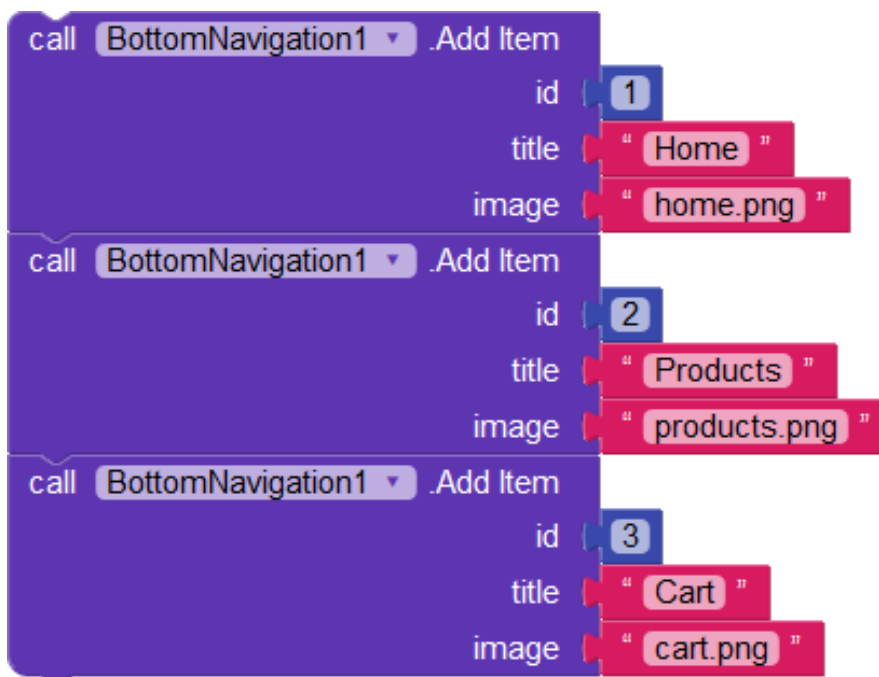
Selanjutnya, kita menangani "pilihan" pada tab menggunakan prosedur yang mengambil id tab sebagai input. Kita mengulang semua pengaturan vertikal dan mengatur properti Visible mereka ke False. Kemudian, kita mendapatkan item tab yang dipilih dan pengaturannya yang sesuai, dan mengatur properti Visible ke True. Blok umum memudahkan untuk memanipulasi visibilitas komponen, terutama di aplikasi dengan beberapa tab.



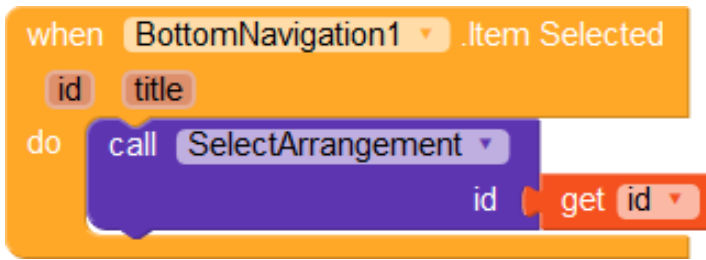
Kita menyebut prosedur ini dari acara Tab Item Selected komponen Tab Layout.



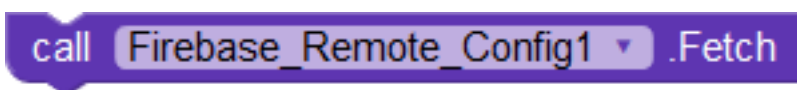
Kita melakukan hal yang sama untuk komponen Navigasi Bawah: pertama-tama tambahkan item menu (dalam acara Inisialisasi Layar), lalu tangani acara pemilihan.



Kita melakukan hal yang sama untuk komponen Navigasi Bawah: pertama-tama tambahkan item menu (dalam acara Inisialisasi Layar), lalu tangani acara pemilihan.



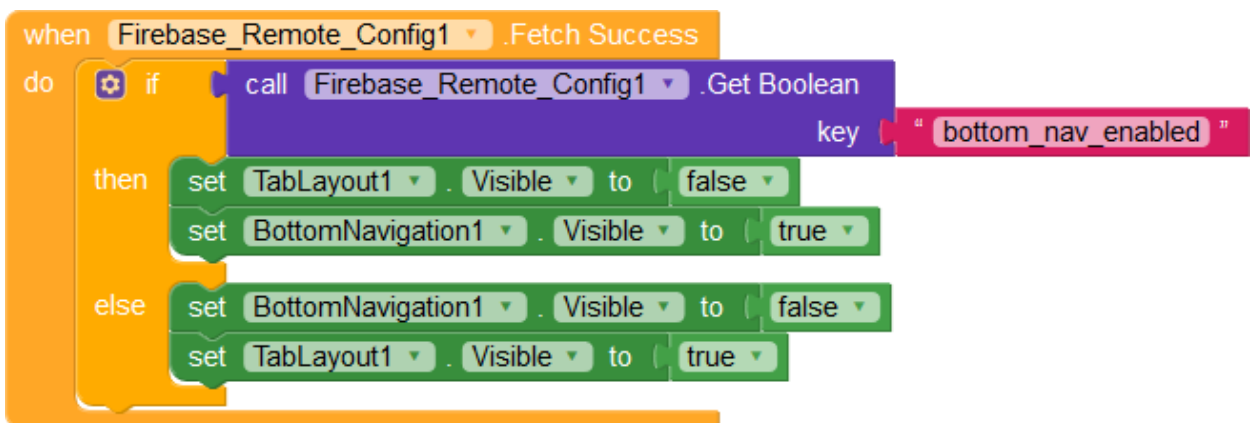
Sekarang untuk bagian yang penting. Kita menggunakan komponen Firebase Remote Config untuk mendapatkan status visibilitas komponen Navigasi Bawah. Sebelum kita menggunakan salah satu parameter, kita perlu mengambil semua data dari Firebase. Tempatkan blok pengambilan di acara Inisialisasi Layar untuk melakukannya.



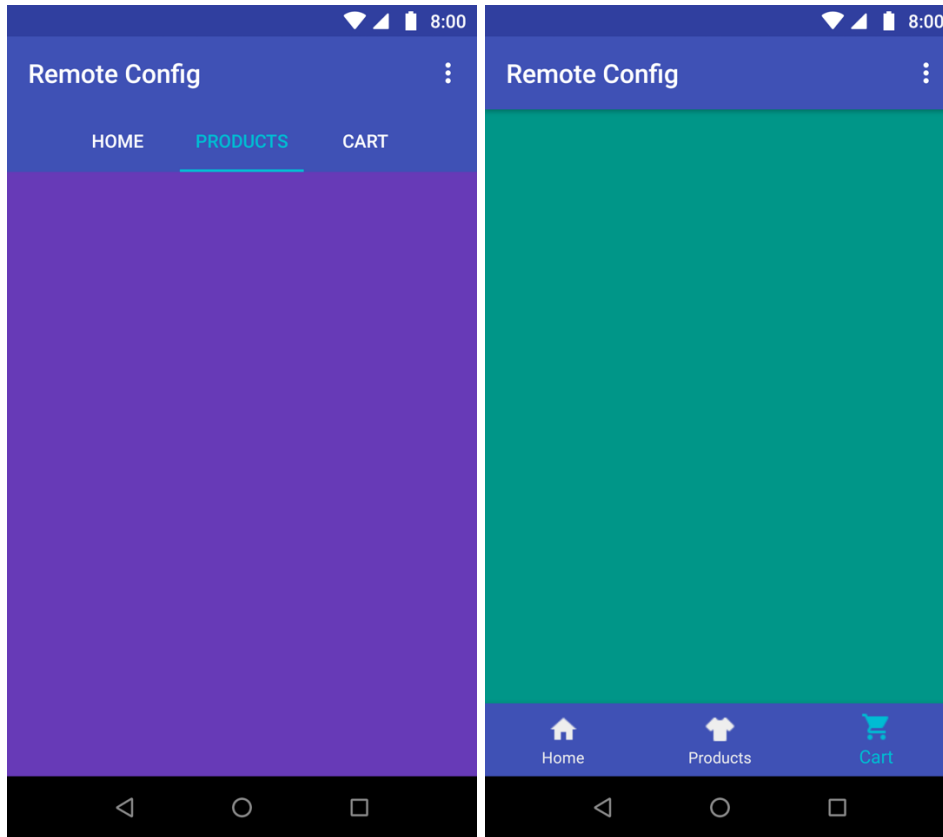
Setelah data diambil, kita mendapatkan nilai parameter `bottom_navigation_enabled`. Karena ini adalah parameter boolean, kita menggunakan blok Get Boolean untuk mengambil nilainya.

Nilai ini dapat berupa True atau False, tergantung pada status pengguna. Jika pengguna berada di 20% yang dipilih secara acak (`random_testers_group` kita), maka True dikembalikan, False sebaliknya.

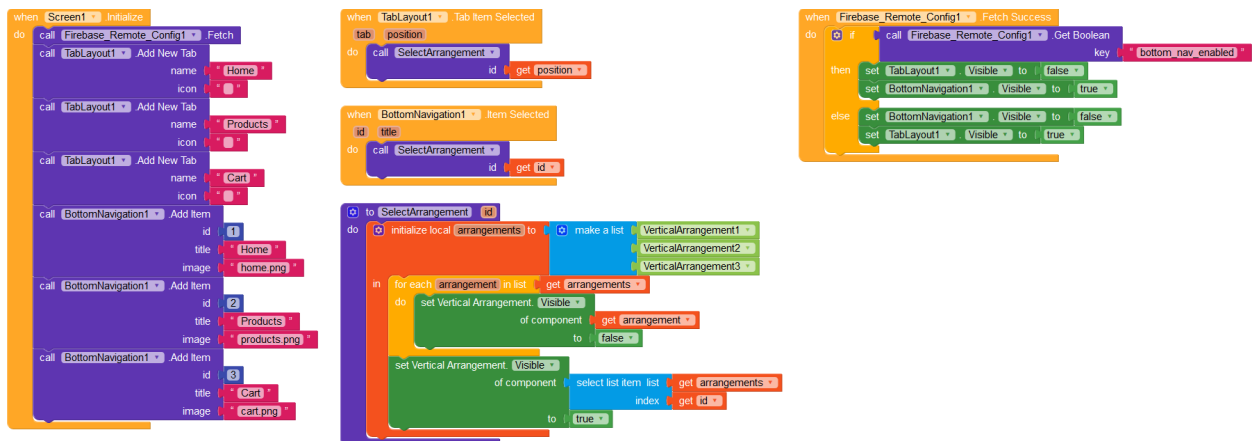
Jika nilainya Benar, kita menampilkan Navigasi Bawah alih-alih Tata Letak Tab



HASIL



Apa yang dilihat pengguna biasa (kiri) vs apa yang dilihat oleh penguji beta yang dipilih secara acak (kanan)



BEKERJA DENGAN ATURAN FIREBASE

Firestore menyediakan sistem database real-time yang kuat yang memungkinkan operasi baca dan tulis yang sangat cepat. Tidak seperti SQL atau layanan database konvensional lainnya, database Firestore tidak menggunakan sistem berbasis nama pengguna/sandi untuk memvalidasi modifikasi data. Sebaliknya, Firestore menggunakan sistem berbasis aturan yang sangat intuitif yang memastikan data Anda hanya dapat digunakan oleh mereka yang Anda beri otorisasi.

Bagaimana Cara Kerja

Aturan Firestore ditulis dalam JSON dan JavaScript. Mereka bekerja bersama Firestore Authentication untuk membatasi akses baca/tulis ke pengguna tertentu. Dimungkinkan juga untuk memvalidasi data sebelum menulisnya ke database. Dalam panduan ini, kita akan melihat berbagai aturan yang dapat digunakan seseorang untuk mengamankan datanya, dan beberapa praktik terbaik saat menulis aturan.

Gunakan kasus untuk Aturan Firestore

Testing

Menguji aplikasi Anda menjadi mudah bila Anda memiliki akses penuh ke database Anda. Untuk mengaktifkan akses baca/tulis ke semua tag, gunakan:

```
{
  "rules": {
    ".read" : true,
    ".write" : true,
  }
}
```

Kode JSON di atas menetapkan dua aturan untuk seluruh database Anda. Setiap aturan dapat dianggap sebagai kondisi yang harus mengembalikan nilai true jika operasi ingin berhasil. Dalam kasus di atas, `.read` dan `.write` selalu mengembalikan nilai true, artinya tidak diperlukan pemeriksaan sebelum mengakses database Anda.

Authenticated-only mode

Anda hanya dapat mengaktifkan pengguna yang diautentikasi untuk membaca dan menulis tag di database Anda. Pengguna dapat diautentikasi menggunakan komponen Firebase Authentication.

```
{
  "rules": {
    ".read" : "auth.uid != null",
    ".write" : "auth.uid != null",
  }
}
```

Setiap pengguna yang diautentikasi memiliki "uid" yang unik. Jika uid pengguna saat ini adalah nol, maka mereka belum diautentikasi. Pengguna tersebut tidak akan diizinkan untuk membaca/menulis data. Ini adalah setelan Firebase default.

Uid pengguna dapat diakses dari variabel ID pengguna di blok Sukses Login dan Sukses Pengguna Saat Ini di komponen Firebase Authentication.

Uid-Tag mode

Anda juga dapat menulis aturan untuk membatasi akses baca dan tulis ke tag tertentu. Dengan aturan tersebut, Anda dapat memastikan bahwa pengguna hanya dapat mengedit data yang telah mereka buat.

Mengingat database ini:

- users
- U43gnonNP9joboLnv
 - ranking : 1
 - email : "example@example.ex"
 - age : 20
- greuo80NOpwqM8100
 - ranking : 67
 - email : "something@someone.com"
 - age : 15
- kpmkmnn2YIHVI542a
 - ranking : 2

- email : "user@product.io"
- age : 45
- challenges
 - FindingNemo
 - name : "Finding Nemo"
 - recommended_age: 15
 - Marathon
 - name : "25mi Marathon"
 - recommended_age: 20

Struktur databasenya adalah:

- users
 - uid
 - ranking
 - email
 - age
- challenges
 - challengeId
 - name
 - recommended_age

Setiap pengguna memiliki tag mereka sendiri (yang diidentifikasi oleh uid mereka), dan setiap tag berisi data tentang pengguna. Kita ingin pengguna hanya dapat mengedit data mereka, artinya pengguna dengan uid U43gnonNP9joboLnv harus dapat mengedit peringkat, email, dan usia hanya untuk tag dengan nama U43gnonNP9joboLnv.

Namun, kita ingin membiarkan semua pengguna melihat tantangan, tetapi tidak mengizinkan siapa pun mengeditnya (tantangan diedit langsung dari Firebase console/CLI).

Berikut adalah bagaimana Anda akan menulis aturan untuk sistem seperti itu.

Mulailah dengan aturan baca/tulis untuk tag pengguna.

```
{
  "rules" : {
    "users" : {
```

```

    "$user_id" : {
      ".read" : "auth.uid == $user_id",
      ".write" : "auth.uid == $user_id"
    }
  }
}

```

Tag \$user_id adalah referensi wildcard untuk semua kunci yang ada di dalam tag pengguna. Kita mengizinkan operasi baca dan tulis hanya jika nama tag (seperti yang ditentukan dalam \$user_id) sama dengan uid pengguna yang diautentikasi. Dengan cara ini, pengguna tidak akan dapat mengakses tag yang bukan miliknya karena auth.uid tidak akan menjadi nama tag tersebut.

Di sisi lain, tantangan dapat dibaca oleh semua orang dan tidak ditulis oleh siapa pun.

```

"challenges" : {
  ".read" : true,
  ".write" : false
}

```

Objek JSON terakhir akan terlihat seperti

```

{
  "rules" : {
    "users" : {
      "$user_id" : {
        ".read" : "auth.uid == $user_id",
        ".write" : "auth.uid == $user_id"
      },
      "challenges" : {
        ".read" : true,
        ".write" : false
      }
    }
  }
}

```

Validasi data

Dengan Aturan Firebase, Anda dapat memvalidasi data pengguna sebelum dikirim ke database. Firebase menyediakan dua variabel untuk tujuan ini: `data` dan `newData`.

- `data`

Variabel ini menyimpan nilai tag database saat ini. Anda dapat menggunakan `data.val()` untuk mengambil nilai, atau menggunakan `data.hasChild(child)` untuk memeriksa apakah nilai tersebut memiliki turunan yang ditentukan. Firebase juga menyediakan beberapa fungsi lain yang dapat Anda gunakan pada data Anda. Informasi lebih lanjut dapat ditemukan di sini

- `newData` Variabel ini menyimpan nilai yang coba ditulis pengguna ke database. `newData` juga mendukung `val`, `hasChild`, dan fungsi lain yang dimiliki `data`.

Best Practice

- Hindari tumpang tindih

Karena sifat cara kerja database Firebase, aturan yang ditetapkan pada tag tertentu berlaku untuk semua turunannya. Dalam contoh yang digunakan dalam mode Uid-Tag, kita telah menetapkan kondisi baca/tulis ke tag "tantangan". Aturan ini berlaku untuk semua tantangan di dalam tag, dan untuk setiap nama dan rekomendasi_usia di dalam setiap tantangan. Aturan untuk tag anak tidak dapat ditimpa jika tag induk sudah memiliki aturan. Intinya, Aturan Firebase diterapkan dari atas ke bawah. Aturan yang lebih dangkal menimpa aturan yang lebih dalam. Oleh karena itu, disarankan untuk menyusun database Anda sedemikian rupa sehingga Anda mengelompokkan tag yang memiliki aturan serupa. Misalnya, database Anda dapat memiliki dua tag utama: publik dan pribadi, dan menolak akses baca ke semua tag di dalam pribadi. Dengan demikian, Anda akan menulis lebih sedikit aturan dan menghindari aturan yang berlebihan. Semakin kompleks struktur database Anda, semakin sulit untuk menulis aturan koheren yang tidak memiliki celah atau kelemahan keamanan.

- Hindari akses yang tidak diautentikasi

Itu selalu lebih baik untuk menetapkan aturan ke `auth.uid != null` daripada `true`. Meskipun keduanya hampir sama, yang pertama mencegah pengguna yang tidak sah mengakses data Anda. Jadi, jika terjadi kesalahan, Anda akan tahu pasti bahwa salah satu pengguna terautentikasi Anda yang menyebabkannya.

- Berikan akses terbatas

Tulis aturan Anda sedemikian rupa sehingga pengguna Anda hanya dapat mengakses data yang dapat mereka lihat. Tidak disarankan untuk memberi pengguna Anda akses ke seluruh database Anda.

- Kunci akses dalam pengembangan

Setel semua aturan ke false di lingkungan pengembangan di mana Anda tidak ingin siapa pun mengedit database Anda. Anda juga disarankan untuk mengunci database Anda jika produk Anda dimatikan.

KESIMPULAN

Aturan Firebase sangat bagus untuk mengamankan database Anda dan menyalurkan pengguna ke tag dan nilai yang tepat. Dengan seperangkat aturan yang tepat, basis data Anda praktis tidak dapat diretas.

Dalam panduan ini, Anda akan mempelajari cara mengamankan database real-time Firebase Anda menggunakan Aturan Firebase. Aturan adalah sistem berbasis JSON dan JavaScript yang memungkinkan Anda membatasi akses basis data ke tag dan/atau pengguna tertentu.

Anda akan membuat aplikasi sederhana yang memungkinkan pengguna masuk dengan Firebase Authentication dan menyimpan data mereka di database waktu nyata. Kemudian, Anda akan menulis beberapa aturan untuk mencegah orang lain mengakses data pengguna yang sensitif.

Prasyarat

Berikut adalah beberapa hal yang harus Anda ketahui sebelum Anda dapat memulai dengan panduan ini.

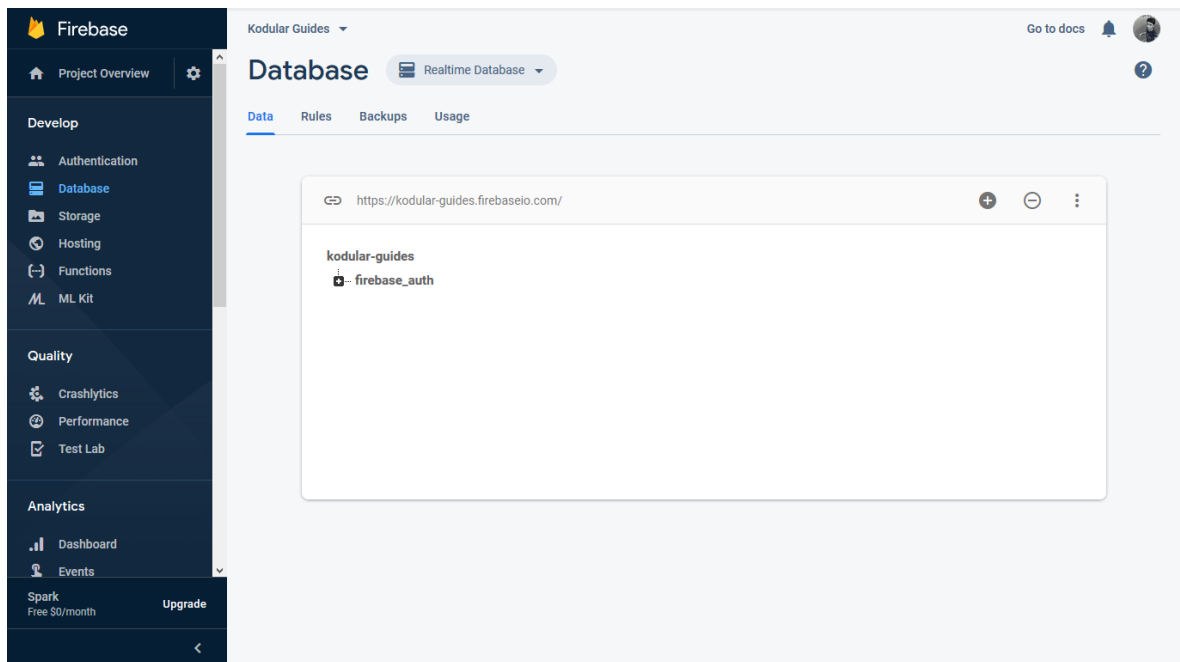
- Bekerja dengan Firebase Database dan Otentikasi
- Bekerja dengan Aturan Firebase

Lihat panduan kita tentang bekerja dengan Aturan Firebase untuk pengenalan sistem aturan.

Langkah pertama

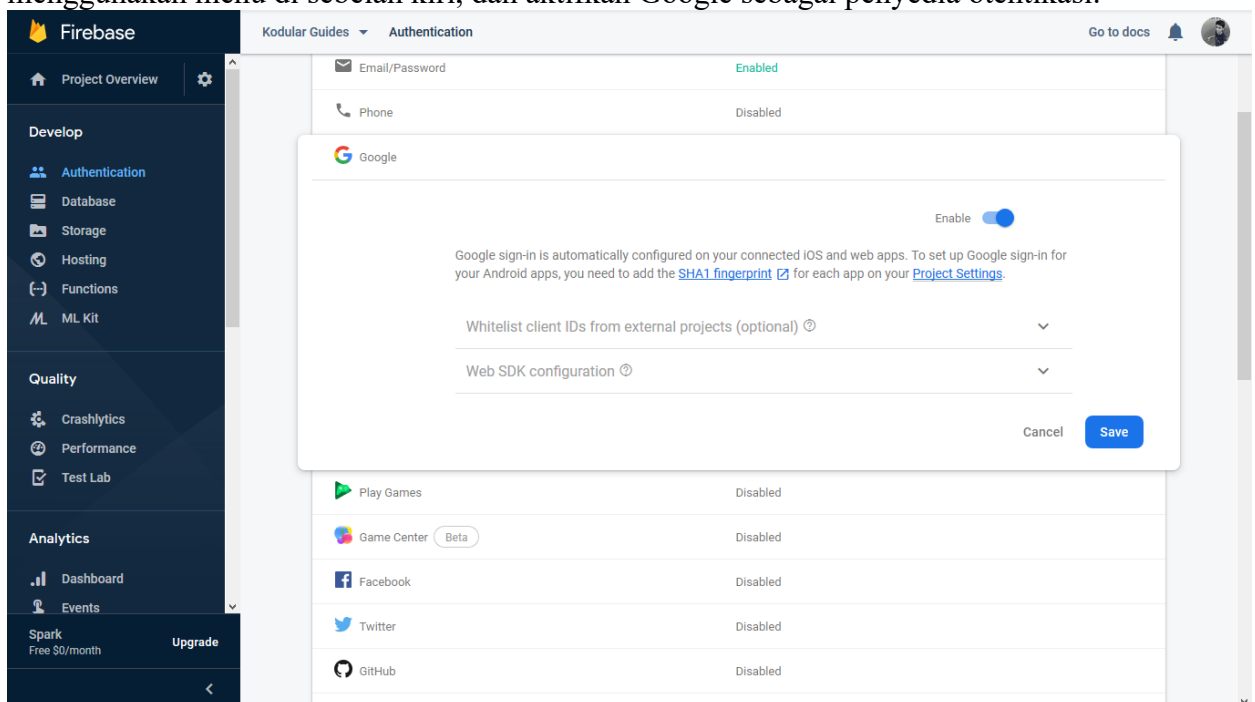
Buka Konsol Firebase Anda untuk menyiapkan database waktu nyata untuk proyek Firebase Anda. Jika Anda tidak memiliki proyek yang sudah ada, buat yang baru.

Gunakan menu navigasi di sebelah kiri untuk menuju ke bagian Database.



Jika database real-time belum diaktifkan, gulir ke bawah dan klik "Realtime database".

Anda akan menggunakan login Google untuk otentikasi pengguna. Buka bagian Otentikasi menggunakan menu di sebelah kiri, dan aktifkan Google sebagai penyedia otentikasi.



Enable google login

Struktur Database

Untuk tutorial ini, kita akan menyimpan tag untuk setiap pengguna yang mendaftar di aplikasi. Tag akan diidentifikasi oleh id unik pengguna, dan akan berisi nama, email, dan versi Android perangkat mereka. Kita ingin merahasiakan nama pengguna, tetapi email dan versi Android mereka pribadi.

Berikut adalah tampilan database kita:

- bucket_name

- uid
- name
- email
- android_version
- uids

uids akan menjadi array dari semua id unik yang ada di database. Kita akan menggunakan tag ini nanti untuk mengulangi semua pengguna dan mendapatkan nama mereka.

Itu saja Firebase untuk saat ini. Mari beralih ke membangun aplikasi Anda. Kita akan kembali ke konsol Firebase nanti untuk menulis aturan.

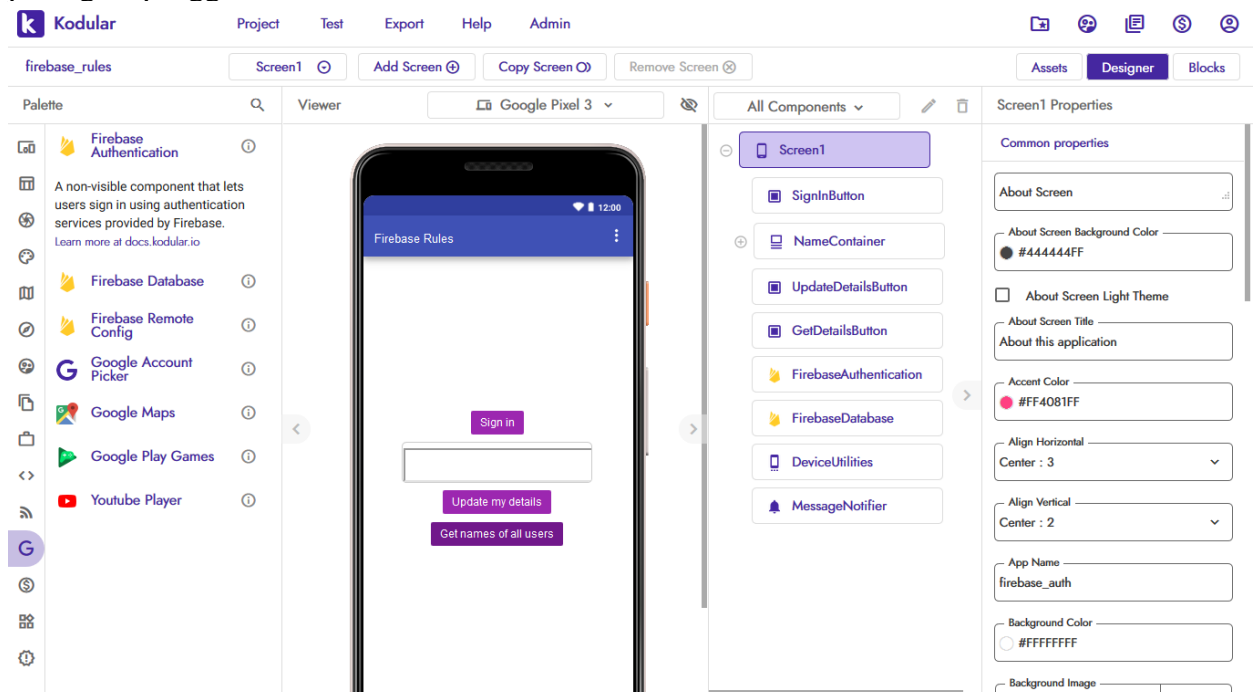
Mendesain aplikasi

Seperti biasa, kita akan mulai dengan proyek baru. Kita akan membutuhkan tiga tombol untuk proyek ini - satu untuk memasukkan pengguna, yang lain untuk mengirim detail mereka ke database, dan satu lagi untuk mengambil nama semua pengguna di database. Seret dan lepas tiga komponen Tombol dan beri nama SignInButton, UpdateDetailsButton, dan GetDetailsButton.

Selanjutnya, letakkan Kotak Teks tempat pengguna dapat memasukkan nama mereka.

Kemudian, tambahkan komponen Notifier dan ganti namanya menjadi MessageNotifier. Kita akan menggunakan komponen ini untuk menampilkan nama semua pengguna.

Terakhir, lepaskan komponen Firebase Database, Firebase Authentication, dan Device Utilities. Komponen Utilitas Perangkat akan digunakan untuk mengambil versi Android dari perangkat pengguna.



Coding pada Blocks

Pertama, kita menambahkan blok acara untuk menangani klik pada SignInButton. Pengguna akan diperlihatkan perintah untuk masuk dengan Google.


```

when SignInButton .Click
do call FirebaseAuth .Google Sign In

```

Setelah login berhasil, kita menambahkan id unik pengguna ke tag uids di database.

```

when FirebaseAuth .Login Success
  provider user ID name email phone Number profile Picture
do call FirebaseDatabase .Append Value
  tag " uids "
  value To Add get user ID

```

Selanjutnya, kita menangani klik pada UpdateDetailsButton. Saat tombol diklik, pertama-tama kita mengambil detail pengguna sehingga kita bisa mendapatkan uid mereka. Kemudian, dengan menggunakan uid, kita memperbarui nama, email, dan versi Android mereka.

```

when UpdateDetailsButton .Click
do call FirebaseAuth .Get Current User

when FirebaseAuth .Current User Success
  user ID name email phone Number profile Picture
do
  call FirebaseDatabase .Store Value
    tag join get user ID "/name "
    value To Store NameInput . Text
  call FirebaseDatabase .Store Value
    tag join get user ID "/email "
    value To Store get email
  call FirebaseDatabase .Store Value
    tag join get user ID "/android_version "
    value To Store DeviceUtilities . Android Version

```

Akhirnya, kita mengkodekan fungsi rekursif untuk mendapatkan nama semua pengguna. Perhatikan bahwa kita tidak dapat menggunakan blok Dapatkan Daftar Tag karena aturan yang akan kita tulis akan memblokir akses langsung ke tag ini.

Untuk mendapatkan nama semua pengguna, kita memerlukan dua daftar: satu untuk semua uid dan satu lagi untuk semua nama.

```

initialize global uids to create empty list
initialize global names to create empty list

```

Kita memulai rekursi menggunakan blok acara Klik untuk GetDetailsButton.

```

when GetDetailsButton .Click
do
  set global uids to create empty list
  set global names to create empty list
  call FirebaseDatabase .Get Value
  tag "uids"
  value If Tag Not There create empty list

```

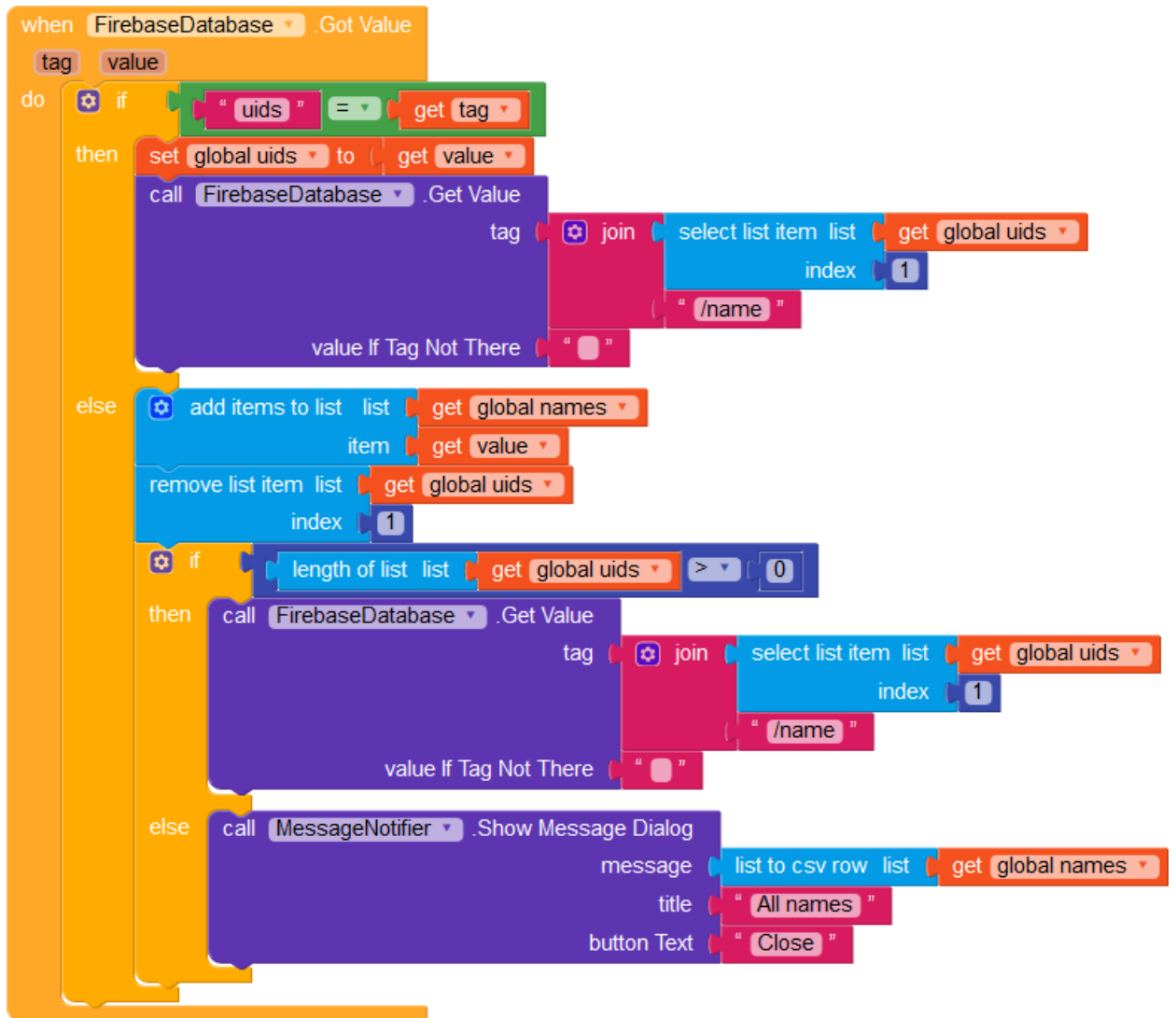
Ketika nilai telah diambil, kita mengisi daftar uid, dan kemudian mengambil nama uid pertama dalam daftar.

```

when FirebaseDatabase .Got Value
tag value
do
  if "uids" = get tag
  then
    set global uids to get value
    call FirebaseDatabase .Get Value
    tag join select list item list get global uids
    index 1
    " /name "
    value If Tag Not There " "
  else

```

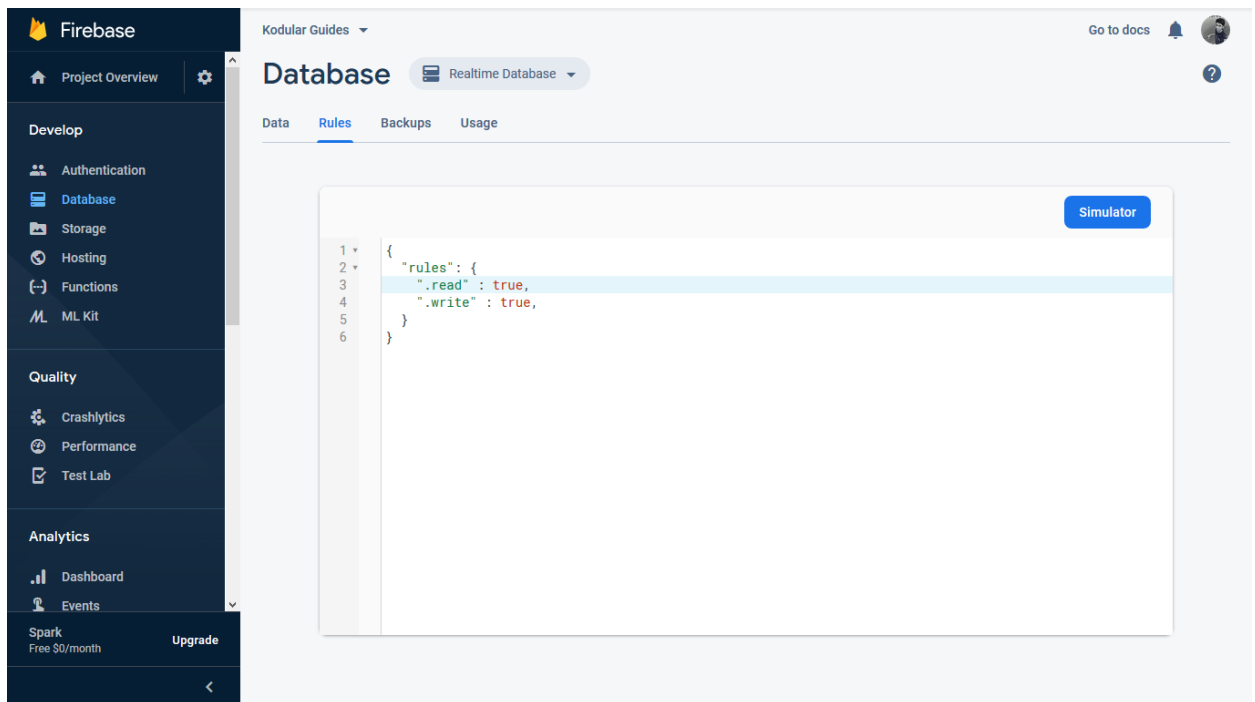
Kita melakukan hal yang sama untuk setiap uid dalam daftar. Setiap kali kita mengambil nama uid, kita menghapus uid dari daftar dan menambahkan nama ke daftar nama.



Setelah semua nama diambil, kita menampilkannya dalam dialog pesan.

Menulis Aturan

Kembali ke konsol Firebase Anda. Di bagian Realtime Database, navigasikan ke tab "aturan".



Secara default, Firebase mengizinkan akses baca/tulis penuh. Jika Anda membuat database dalam mode terkunci, aturan default hanya akan mengizinkan pengguna yang diautentikasi, dan aturan Anda akan terlihat seperti:

```
{
  "rules": {
    ".read": "auth !== null",
    ".write": "auth !== null",
  }
}
```

Keduanya tidak aman karena semua pengguna Anda memiliki akses ke semua data di database Anda. Mari kita perbaiki ini dengan beberapa aturan.

Untuk memulai, ganti aturan yang saat ini ditetapkan dengan template kosong seperti yang ditunjukkan di bawah ini:

```
{
  "rules": {
  }
}
```

Selanjutnya, ubah aturan seperti yang ditunjukkan untuk menargetkan semua tag di bucket proyek Anda. Ganti `*PROJECT_BUCKET*` dengan nama bucket Anda. (Nama bucket dapat ditemukan di properti desainer komponen Firebase Database).

```
{
  "rules": {
    "**PROJECT_BUCKET*": {
      //... to be added later
    }
  }
}
```

Aturan untuk detail pengguna

Kita berharap pengguna hanya dapat mengedit detail mereka. Setiap pengguna memiliki tag untuk diri mereka sendiri, yang dinamai sebagai uid mereka. Dengan demikian, pengguna hanya dapat mengedit tag miliknya. Kita dapat menggunakan kunci wildcard untuk memilih tag apa pun di dalam ember proyek Anda, seperti yang ditunjukkan di bawah ini:

```
{
  "rules": {
    "*PROJECT_BUCKET*": {
      "$uid": {
        }
      }
    }
  }
}
```

Seorang pengguna harus dapat mengedit hanya nama mereka dan tidak ada orang lain, tetapi semua pengguna yang diautentikasi harus dapat melihat setiap nama pengguna. Kita dapat membatasi operasi baca/tulis sesuai dengan aturan ini:

```
"name": {
  ".read": "auth != null",
  ".write": "auth.uid == $uid"
}
```

Namun, email dan versi Android seharusnya membatasi akses baca dan tulis.

```
"email": {
  ".read": "auth.uid == $uid",
  ".write": "auth.uid == $uid"
},
"android_version": {
  ".read": "auth.uid == $uid",
  ".write": "auth.uid == $uid"
}
```

Daftar aturan kita saat ini terlihat seperti ini:

```
{
  "rules": {
    "*PROJECT_BUCKET*": {
      "$uid": {
        "name": {
          ".read": "auth != null",
          ".write": "auth.uid == $uid"
        },
        "email": {
          ".read": "auth.uid == $uid",
          ".write": "auth.uid == $uid"
        },
        "android_version": {
          ".read": "auth.uid == $uid",
          ".write": "auth.uid == $uid"
        }
      }
    }
  }
}
```

}
 Mari kita ambil contoh database sampel untuk melihat cara kerja aturan. Database saat ini memiliki data berikut:

The screenshot shows the Firebase Realtime Database interface. The URL is `https://kodular-guides.firebaseio.com/`. The database structure is as follows:

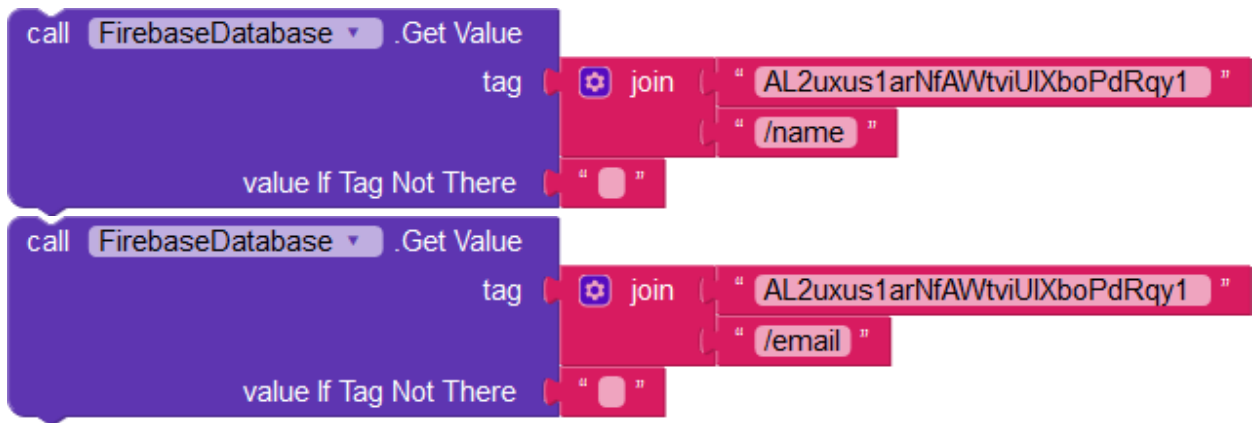
```

kodular-guides
├── firebase_auth
│   ├── 163IN5WF62eJsFeYQMmFdIIcAvl2
│   │   ├── android_version: "\8.1.0\"
│   │   ├── email: "\"vishwas@kodular.io\"
│   │   └── name: "\"Vishwas\"
│   ├── AL2uxus1arNfAWtviUIXboPdRqy1
│   │   ├── android_version: "\9.0\"
│   │   ├── email: "\"helios@helios.com\"
│   │   └── name: "\"Helios\"
│   └── KW4LkSOtG0Y1iede9KaTrwsYCx62
│       ├── android_version: "\8.1.0\"
│       ├── email: "\"vishwasadiga@gmail.com\"
│       └── name: "\"Mr. Example\"
└── uids: ["AL2uxus1arNfAWtviUIXboPdRqy1", "163IN5WF62eJsFeYQMmFdIIcAvl2", "KW4LkSOtG0Y1iede9KaTrwsYCx62"]
  
```

```

{
  "*PROJECT_BUCKET*" : {
    "163IN5WF62eJsFeYQMmFdIIcAvl2" : {
      "android_version" : "\8.1.0\"",
      "email" : "\"vishwas@kodular.io\"",
      "name" : "\"Vishwas\""
    },
    "AL2uxus1arNfAWtviUIXboPdRqy1" : {
      "android_version" : "\9.0\"",
      "email" : "\"helios@helios.com\"",
      "name" : "\"Helios\""
    },
    "KW4LkSOtG0Y1iede9KaTrwsYCx62" : {
      "android_version" : "\8.1.0\"",
      "email" : "\"vishwasadiga@gmail.com\"",
      "name" : "\"Mr. Example\""
    },
    "uids" : ["AL2uxus1arNfAWtviUIXboPdRqy1",
    "\"163IN5WF62eJsFeYQMmFdIIcAvl2\"", "\"KW4LkSOtG0Y1iede9KaTrwsYCx62\""]
  }
}
  
```

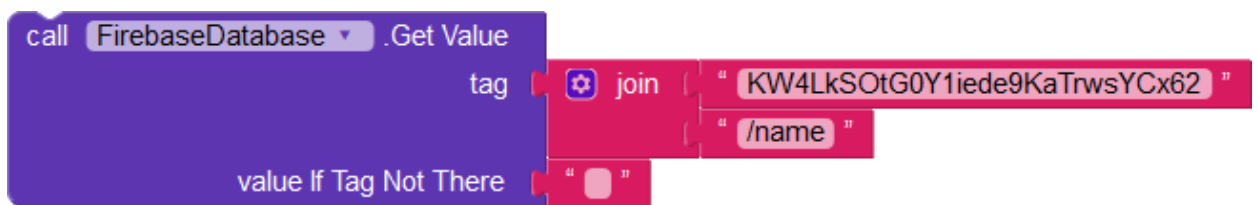
Pengguna yang saat ini diautentikasi dengan uid AL2uxus1arNfAWtviUIXboPdRqy1 dapat mengambil nama, email, dan versi Android mereka sendiri:



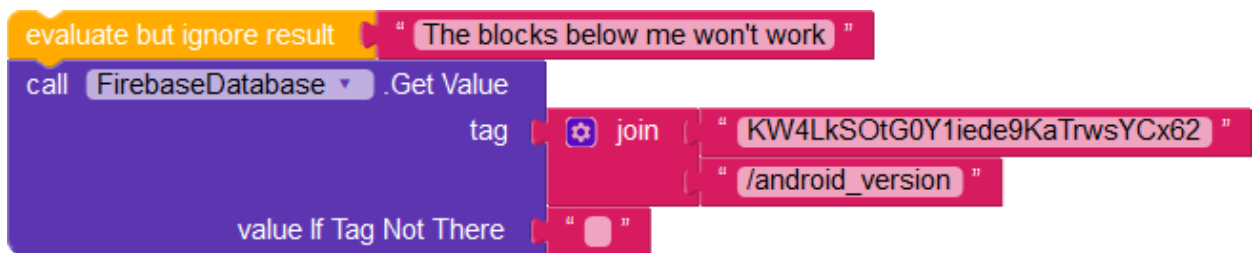
Pengguna juga dapat mengedit nilai nama, email, dan versi Android mereka.



Selain itu, pengguna dapat mengakses nama pengguna lain, katakanlah, satu dengan uid KW4LkSOtG0Y1iede9KaTrwsYCx62



Tetapi mereka tidak bisa mendapatkan email atau versi Android dari pengguna lain.



Mereka juga tidak dapat mengatur nama, email, atau versi Android dari pengguna lain.

Aturan untuk daftar uid

Setiap pengguna harus dapat mengakses daftar semua uid, tetapi tidak boleh mengirim spam ke daftar dengan nilai acak. Kita dapat memvalidasi data untuk memastikan pengguna menambahkan uid mereka ke daftar, dan bukan nilai lainnya.

Validasi dapat dilakukan dengan menggunakan aturan berikut:

```
"uids" : {
  ".read" : "auth != null",
  ".write" : "newData.child(auth.uid).exists()"
}
```

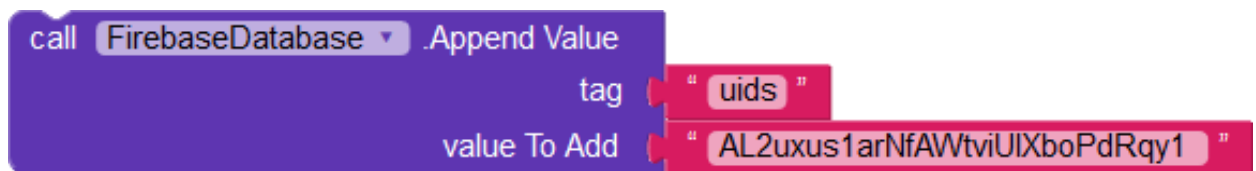
Dengan set aturan, semua pengguna yang diautentikasi akan dapat membaca nilai daftar uids. Pengguna yang diautentikasi juga dapat menambahkan uid mereka ke daftar. Namun, tidak ada nilai lain kecuali uid pengguna yang dapat dimasukkan. Intinya, variabel newData harus memiliki anak yang merupakan uid pengguna, yang menjamin bahwa daftar yang ditambahkan memiliki uid yang benar.

Implementasi saat ini, bagaimanapun, menambahkan id unik pengguna ke daftar setiap kali mereka masuk. Untuk menyimpan hanya nilai unik uid, kita mencegah pengguna menulis duplikat.

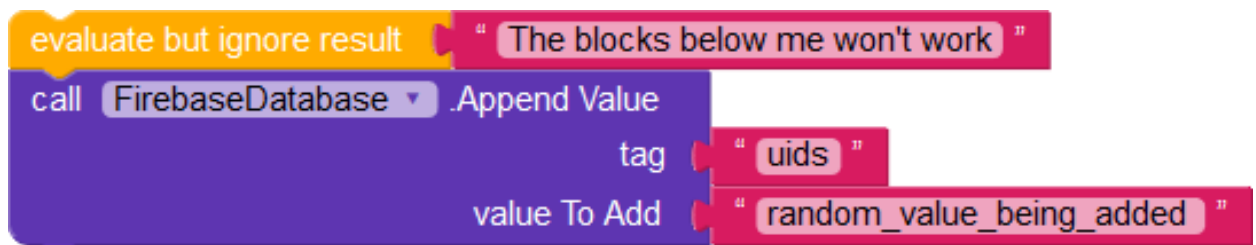
The .write rule can be changed from ".write" : "newData.child(auth.uid).exists()" to ".write" : "newData.child(auth.uid).exists() && !data.child(auth.uid).exists()"

Ini memastikan bahwa uid hanya ditulis jika data saat ini belum memuatnya. Mari kita ilustrasikan dengan contoh pengguna dengan uid AL2uxus1arNfAWtviUIXboPdRqy1.

Pengguna dapat menambahkan uid mereka ke daftar semua uid



Tetapi tidak dapat menambahkan apa pun selain uid mereka



Daftar aturan terakhir kita terlihat seperti ini:

```
{
  "rules": {
    "firebase_auth": {
      "$uid": {
```



```
"name": {
  ".read": "auth != null",
  ".write": "auth.uid == $uid"
},
"email": {
  ".read": "auth.uid == $uid",
  ".write": "auth.uid == $uid"
},
"android_version": {
  ".read": "auth.uid == $uid",
  ".write": "auth.uid == $uid"
}
},

"uids" : {
  ".read" : "auth != null",
  ".write" : "newData.child(auth.uid).exists() && !data.child(auth.uid).exists()"
}
}
}
}
```

Kesimpulan

```

when SignInButton Click
do
  call FirebaseAuthentication Google Sign In

when UpdateDetailsButton Click
do
  call FirebaseAuthentication Get Current User

when FirebaseAuthentication Current User Success
  user ID name email phone Number profile Picture
do
  call FirebaseDatabase Store Value
    tag join get user ID
    value To Store NameInput Text
  call FirebaseDatabase Store Value
    tag join get user ID
    value To Store get email
  call FirebaseDatabase Store Value
    tag join get user ID
    value To Store DeviceUtilities Android Version

when FirebaseAuthentication Current User Failed
do
  call MessageNotifier Show Alert
    notice Could not get user details. Are you signed in?

when GetDetailsButton Click
do
  set global uids to create empty list
  set global names to create empty list
  call FirebaseDatabase Get Value
    tag uids
    value If Tag Not There create empty list

when FirebaseDatabase Firebase Error
  message
do
  call MessageNotifier Show Alert
    notice get message

when FirebaseDatabase Got Value
  tag value
do
  if uids == get tag
  then
    set global uids to get value
    call FirebaseDatabase Get Value
      tag join select list item list get global uids
      index 1
      value If Tag Not There " "
  else
    add items to list list get global names
    item get value
    remove list item list get global uids
    index 1
    if length of list list get global uids > 0
    then
      call FirebaseDatabase Get Value
        tag join select list item list get global uids
        index 1
        value If Tag Not There " "
    else
      call MessageNotifier Show Message Dialog
        message list to csv row list get global names
        title All names
        button Text Close
  
```

Anda telah berhasil mengamankan database Firebase dari pengguna jahat!

Berikut adalah ringkasan dari apa yang telah Anda pelajari dalam panduan ini.

- Cara mengatur sistem login Google menggunakan Firebase Authentication.

- Cara menulis aturan yang hanya mengizinkan sebagian akses ke tag dan nilai.
- Cara mengambil detail semua pengguna secara rekursif.

MENARGETKAN PEMBERITAHUAN PUSH ANDA

Dalam panduan ini, Anda akan mempelajari cara menargetkan notifikasi ke audiens tertentu menggunakan tag OneSignal. Anda akan membuat aplikasi sampel dan secara acak menambahkan 50% dari semua pengguna ke grup "Pro", dan kemudian mendorong pemberitahuan hanya ke anggota grup ini.

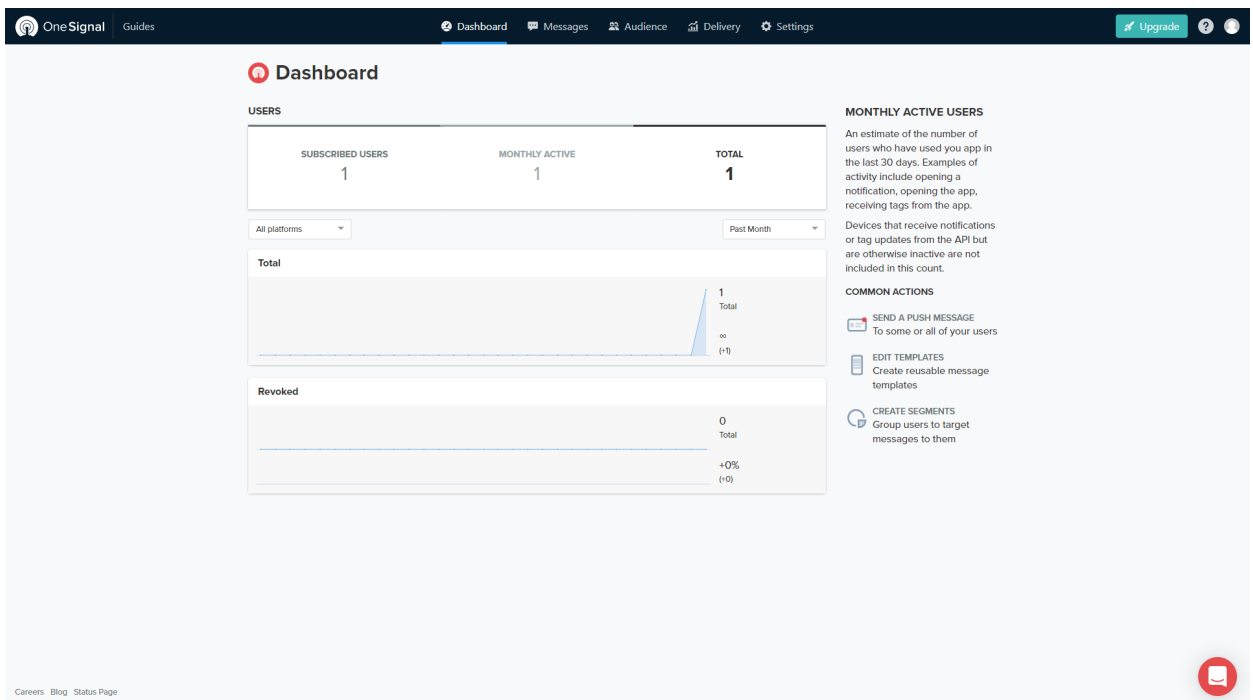
Prasyarat

Berikut adalah beberapa hal yang harus Anda ketahui sebelum Anda dapat memulai dengan panduan ini.

Menyiapkan OneSignal di aplikasi Anda

Langkah Pertama

Buka Dasbor OneSignal Anda untuk menyiapkan "tag" baru. Tag dapat digunakan untuk mengidentifikasi pengguna tertentu berdasarkan nilainya. Dalam contoh ini, kita akan membuat tag bernama "pro".



The screenshot shows the OneSignal Dashboard interface. At the top, there is a navigation bar with tabs for Dashboard, Messages, Audience, Delivery, and Settings. The main content area is titled "Dashboard" and is divided into several sections:

- USERS**: A summary table showing 1 Subscribed User, 1 Monthly Active user, and 1 Total user. Below this is a line chart for "Total" users and a "Revoked" section showing 0 users.
- MONTHLY ACTIVE USERS**: A text-based section explaining that this is an estimate of users who have used the app in the last 30 days, with examples of activity like opening a notification or the app.
- COMMON ACTIONS**: A list of actions including "SEND A PUSH MESSAGE", "EDIT TEMPLATES", and "CREATE SEGMENTS".

At the bottom left, there are links for "Careers", "Blog", and "Status Page". At the bottom right, there is a red notification icon.

Navigasikan ke bagian "Audiens" menggunakan tab di bagian atas halaman.

The screenshot shows the OneSignal Segments dashboard. At the top, there are navigation links for Dashboard, Messages, Audience, Delivery, and Settings. The main content area is titled 'Segments' and includes a 'NEW SEGMENT' button. Below this, there is a table of existing segments:

Segment Name	Count
Subscribed Users	1
Active Users	1
Engaged Users	0
Inactive Users	0

On the right side, there is a 'SEGMENTS' section with explanatory text and a 'DATA TAGS' section. At the bottom right, there is a red notification icon.

Klik "New Segment" untuk membuat grup pengguna. Nama Segmen akan menjadi tag yang akan kita lewati dari blok aplikasi.

The screenshot shows the OneSignal Segment Editor interface. The segment name is 'Pro'. Below the name field, there is an 'ADD FILTER' button. On the right side, there is a section titled 'ADDING A FILTER' with instructions: 'To continue adding filters, press [Enter] or click 'Add Filter'. This will open the filter picker.' Below this, there is a 'FINISHED?' section with the instruction: 'Just click 'Create Segment' below to save your segment. Be sure to name it before though!'. At the bottom right, there is a 'CREATE SEGMENT' button.

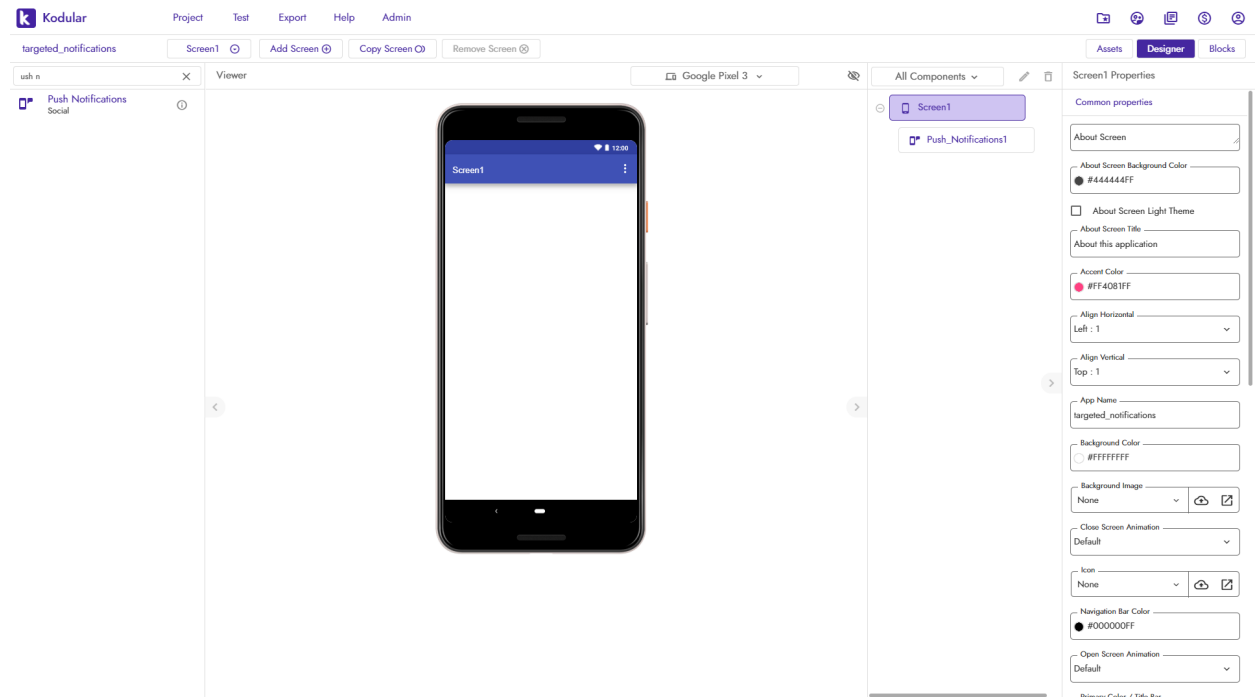
Ubah nama Segmen menjadi "Pro" dan klik "Buat Segmen".

Aplikasi OneSignal Anda sekarang siap untuk menangani pengguna dengan tag "Pro"!

Mendesain aplikasi

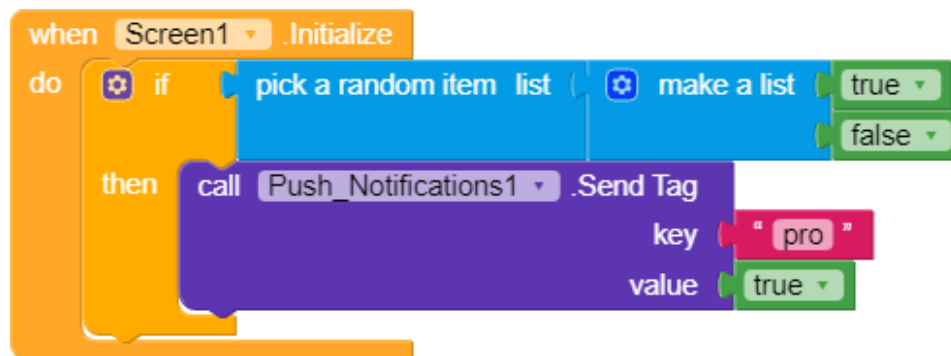
Seperti biasa, kita akan mulai dengan proyek baru. Yang kita perlukan untuk panduan ini adalah komponen Pemberitahuan Push dalam kategori Sosial. Pastikan Anda menggunakan ID Aplikasi OneSignal yang benar untuk aplikasi Anda.

Beginilah seharusnya tampilan desainer



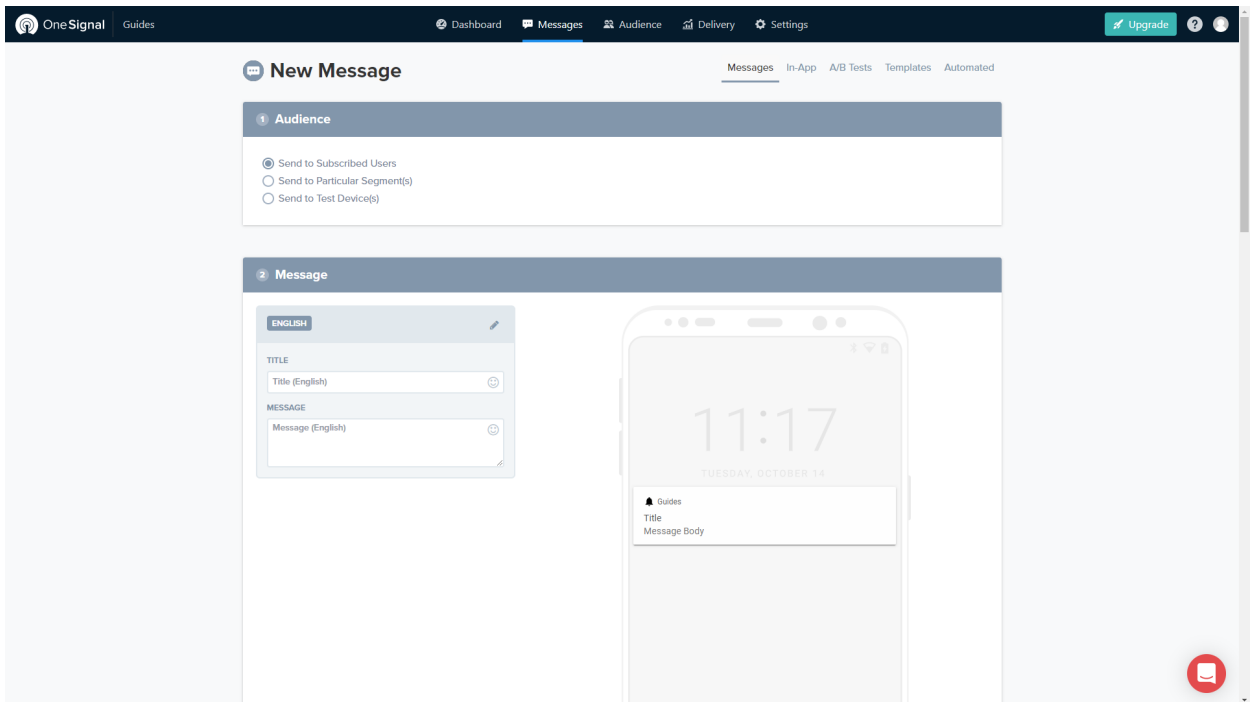
Coding blocks

Kita hanya akan menggunakan sembilan blok dalam panduan ini. Saat layar dibuka, kita memilih True atau False secara acak dan jika True, kita mengirim tag 'pro' ke OneSignal.

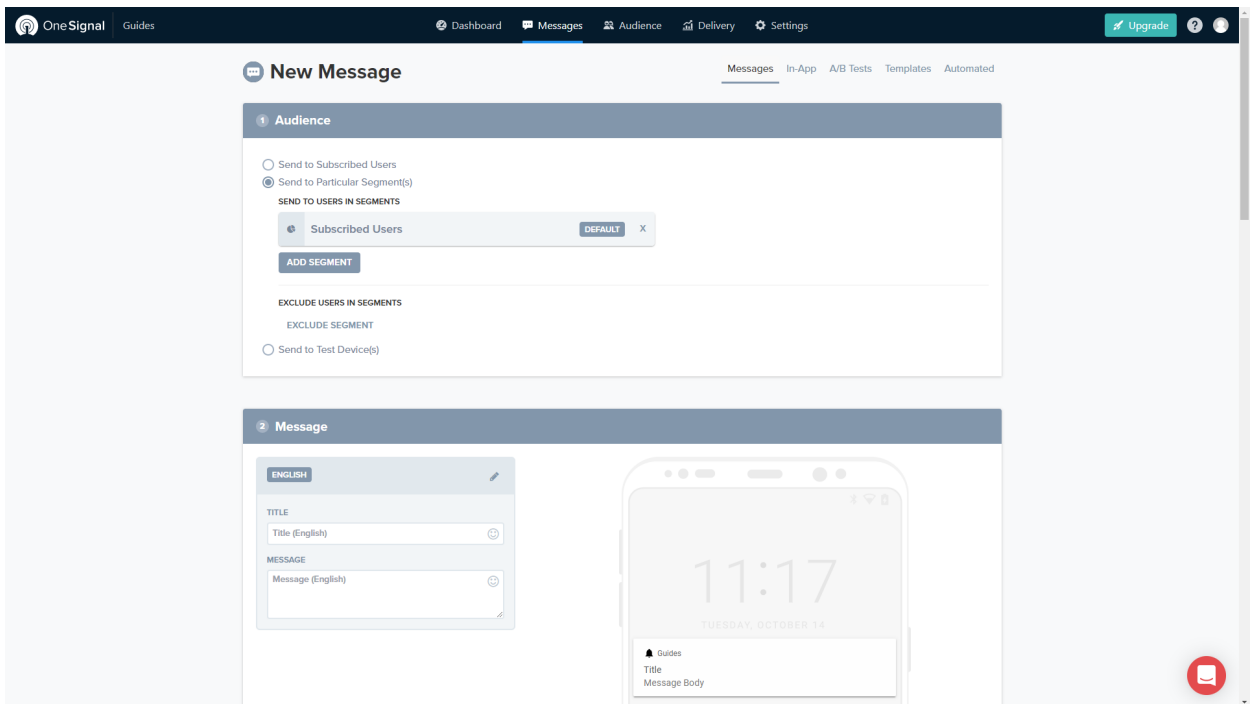


Mengirim pemberitahuan yang ditargetkan

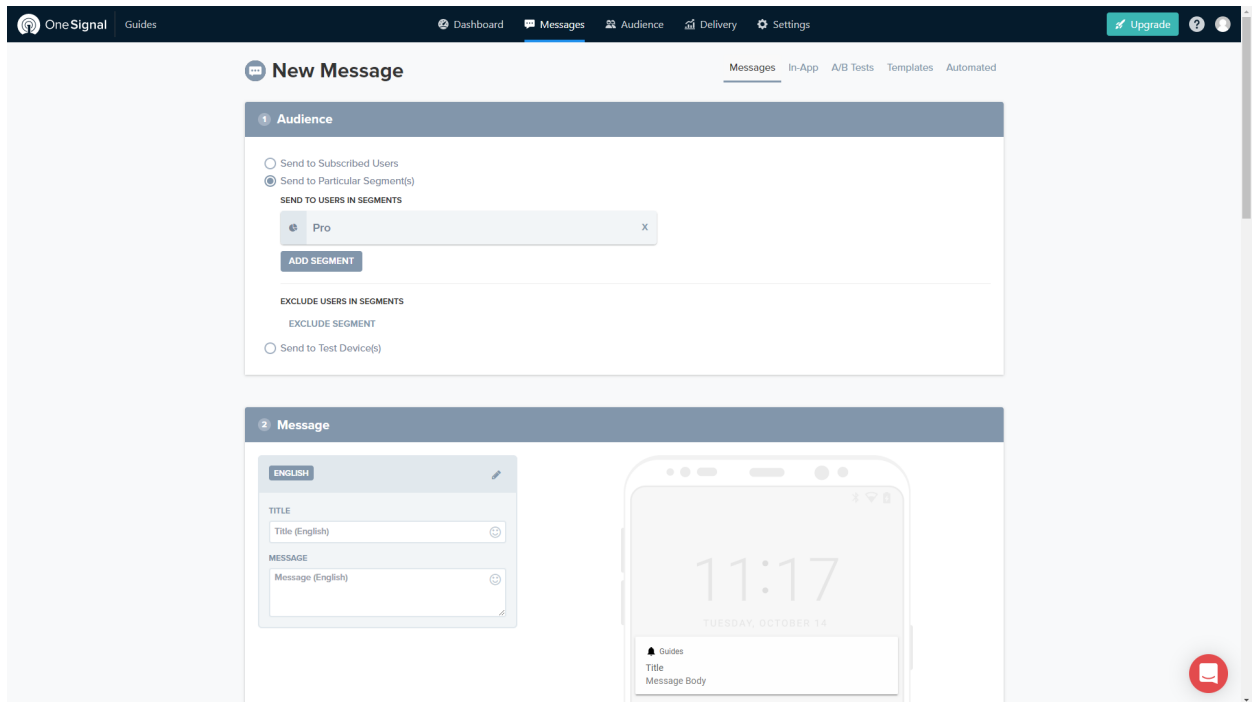
Di bagian "Pesan" di Dasbor OneSignal Anda, klik "Dorong Baru" untuk membuat pemberitahuan baru.



Ubah pengaturan "Audien" menjadi "Send to Particular Segments".



Hapus segmen "Subscribed Users" default dan gunakan tombol "Add Segment" untuk menambahkan segmen "Pro".



Isi detail notifikasi lainnya seperti biasa, dan "Confirm" notifikasi push Anda. Itu dia! Hanya pengguna "Pro" yang akan melihat notifikasi yang baru saja Anda kirim.

Kesimpulan

Anda telah berhasil menargetkan pemberitahuan push Anda ke pengguna tertentu!

Berikut adalah ringkasan dari apa yang telah Anda pelajari dalam panduan ini.

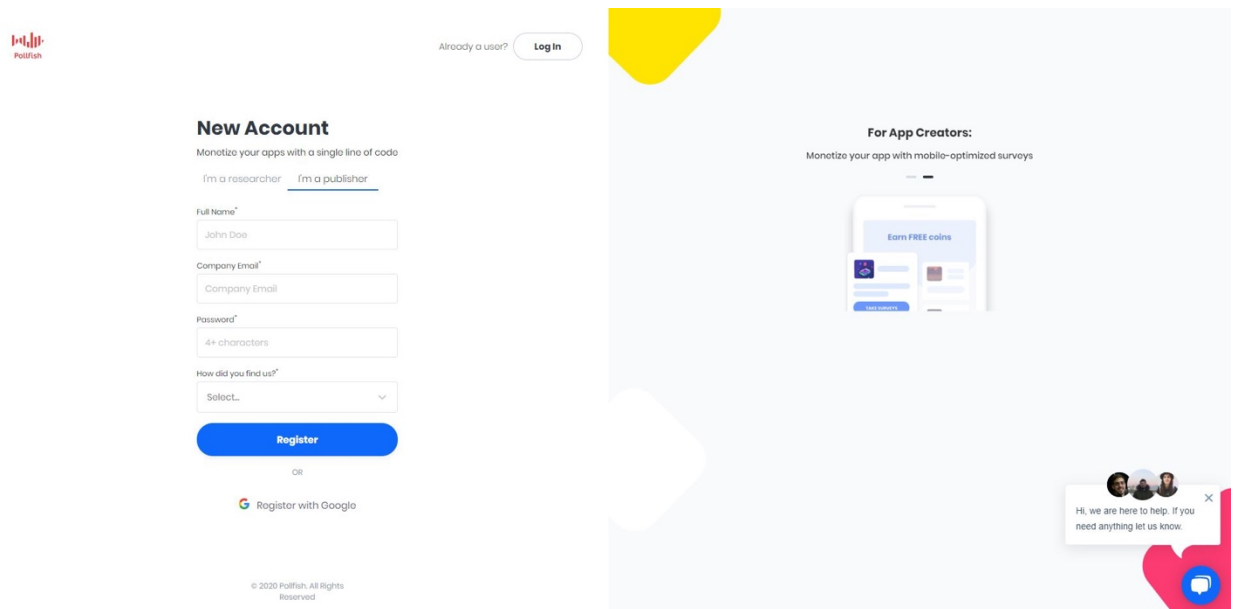
- Cara membuat Segmen OneSignal.
- Cara menambahkan pengguna ke Segmen tertentu menggunakan tag.
- Cara mengirim notifikasi ke pengguna Segmen tertentu.

MENGINTEGRASIKAN POLLFISH DI APLIKASI ANDA

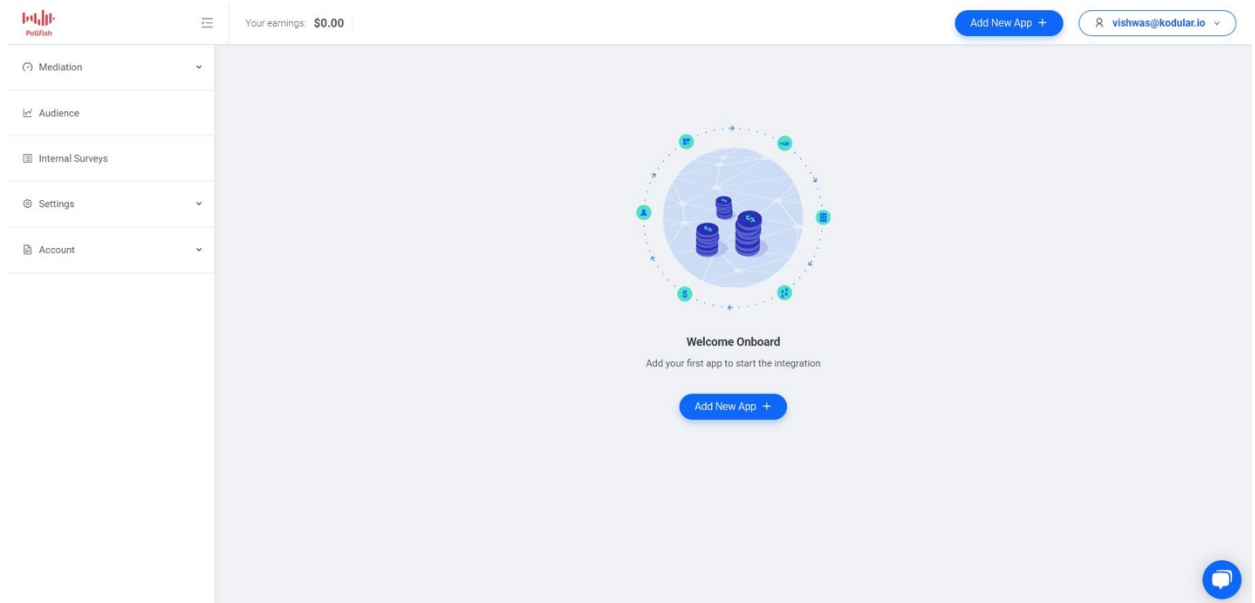
Dalam panduan ini, Anda akan mempelajari cara menggunakan survei Pollfish di aplikasi Anda. Komponen Pollfish digunakan untuk menampilkan survei asli dan bertarget yang mendorong audiens berkualitas tinggi ke aplikasi Anda. Menghargai pengguna karena menyelesaikan survei sangat bagus untuk Anda dan pengguna Anda karena meningkatkan keterlibatan dan retensi.

Langkah pertama

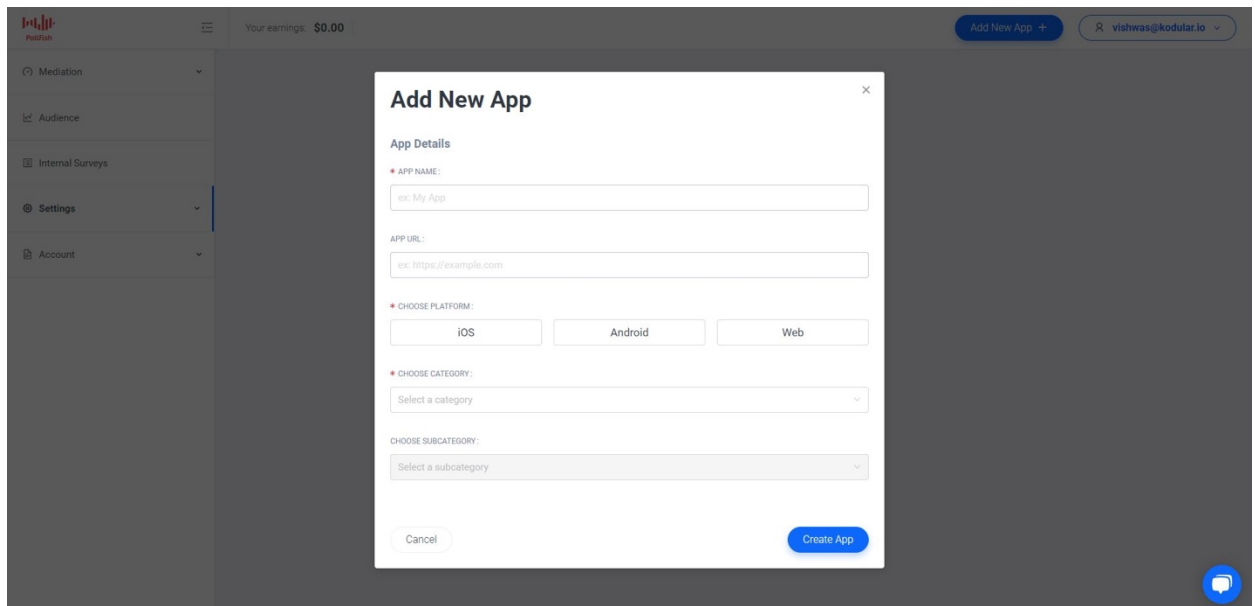
Pertama-tama Anda harus mendaftar untuk mendapatkan akun Pollfish gratis dan mendaftarkan aplikasi Anda. Anda akan diberikan API key yang nantinya akan kita gunakan untuk berkomunikasi antara Kodular dan Pollfish. Untuk memulai, pergilah ke Pollfish



Setelah Anda berhasil mendaftar, Anda akan diarahkan ke dasbor Pollfish Anda.



Klik "Create App" dan isi detail aplikasi Android Anda. Kirim formulir setelah Anda mengisi semua detail dengan akurat.

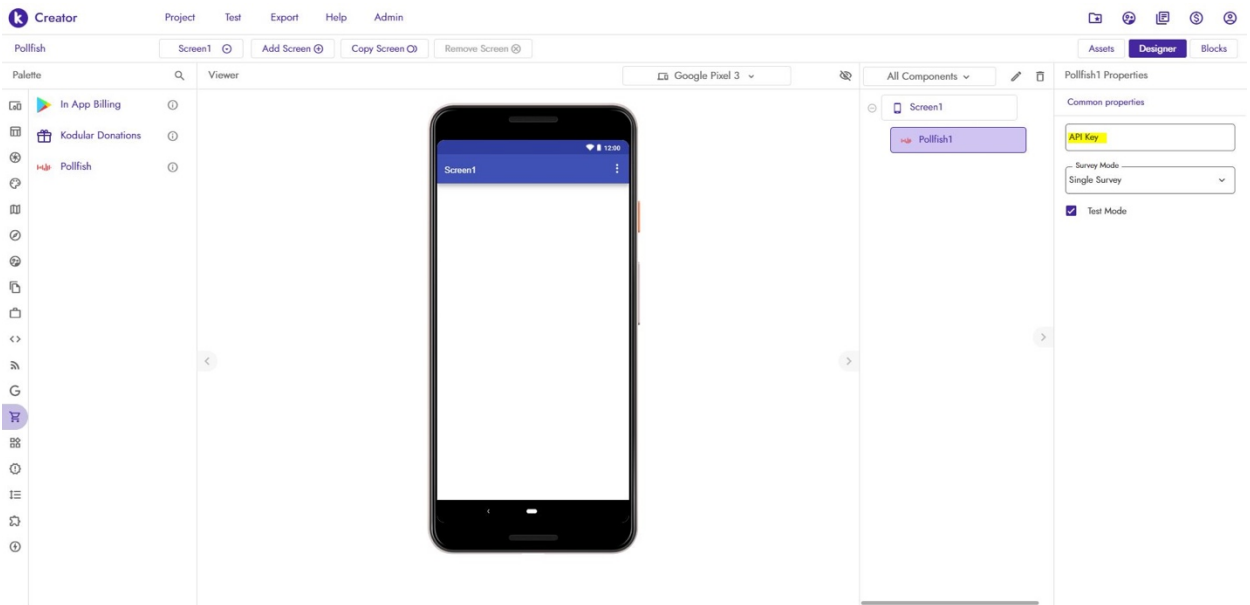


Anda akan menemukan kunci API Anda di bagian atas halaman. Salin kunci ini ke clipboard Anda atau simpan sementara. Anda akan menempelkannya nanti di proyek Kodular Anda.

Mendesain aplikasi

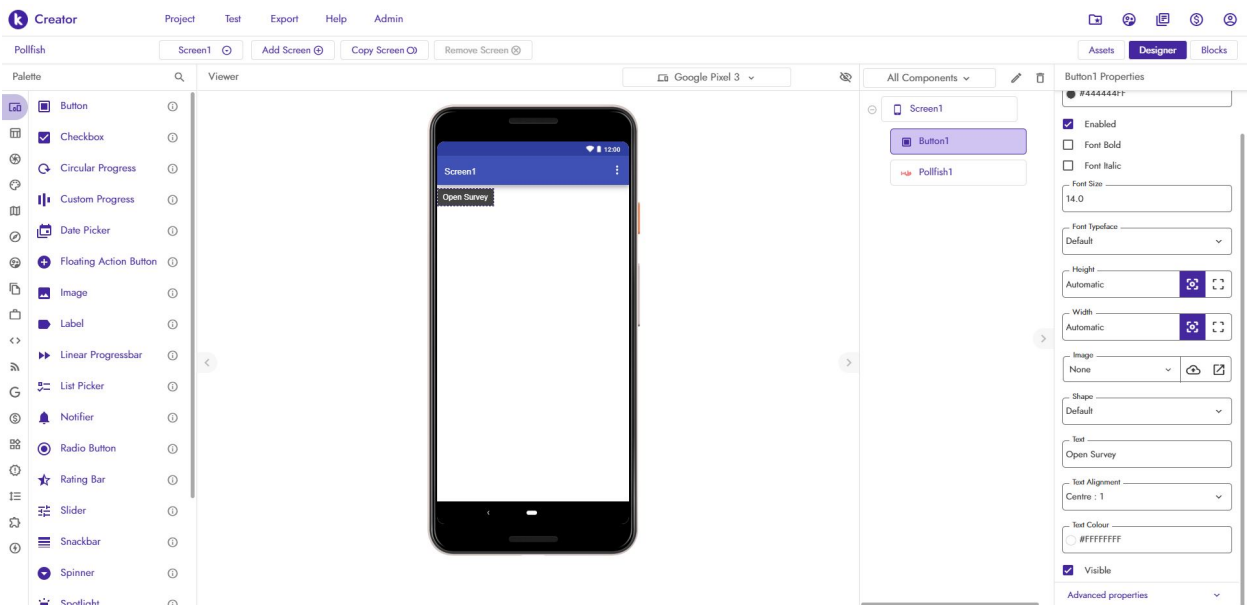
Kita akan memulai dengan proyek baru. Seret dan lepas komponen Pollfish dari kategori Monetisasi > Umum.

Pilih komponen Pollfish dan buka panel Properties. Setel properti Kunci API ke kunci yang Anda salin dari dasbor Pollfish Anda.



Kita akan menampilkan Survei Tunggal kepada pengguna dalam panduan ini. Buka Langkah berikutnya untuk informasi tentang penerapan survei Offerwall.

Juga tambahkan Tombol ke Layar. Kita akan menggunakan acara Klik Tombol untuk membuka survei. Desainer Anda akan terlihat seperti ini:



Coding Blocks

Sebelum kita dapat membiarkan pengguna membuka survei, kita harus memastikan bahwa survei telah dimuat dan siap. Gunakan blok Survei Diterima untuk memeriksa apakah survei telah dimuat. Kita akan menyimpan status survei dalam variabel global yang disebut `surveyReceived`.



```

when Pollfish1 .Survey Received
  survey Info
do set global surveyReceived to true

```

Kita kemudian mendengarkan klik pada Tombol. Kita menampilkan survei hanya jika telah dimuat.

```

when Button1 .Click
do if get global surveyReceived
  then call Pollfish1 .Show Survey

```

Terkadang mungkin tidak ada survei yang tersedia untuk ditampilkan kepada pengguna. Kita membuat prosedur yang disebut SurveyNotAvailable untuk menangani kasus saat memuat survei gagal.

```

to SurveyNotAvailable
do set Button1 .Text to "No survey is available right now. Please try lat..."
  set global surveyReceived to false

```

Prosedur akan dipanggil jika tidak ada survei yang tersedia atau jika pengguna tidak memenuhi syarat atau ditolak.

```

when Pollfish1 .Survey Not Available
do call SurveyNotAvailable

```

```

when Pollfish1 .User Rejected
do call SurveyNotAvailable

```

```

when Pollfish1 .User Not Eligible
do call SurveyNotAvailable

```

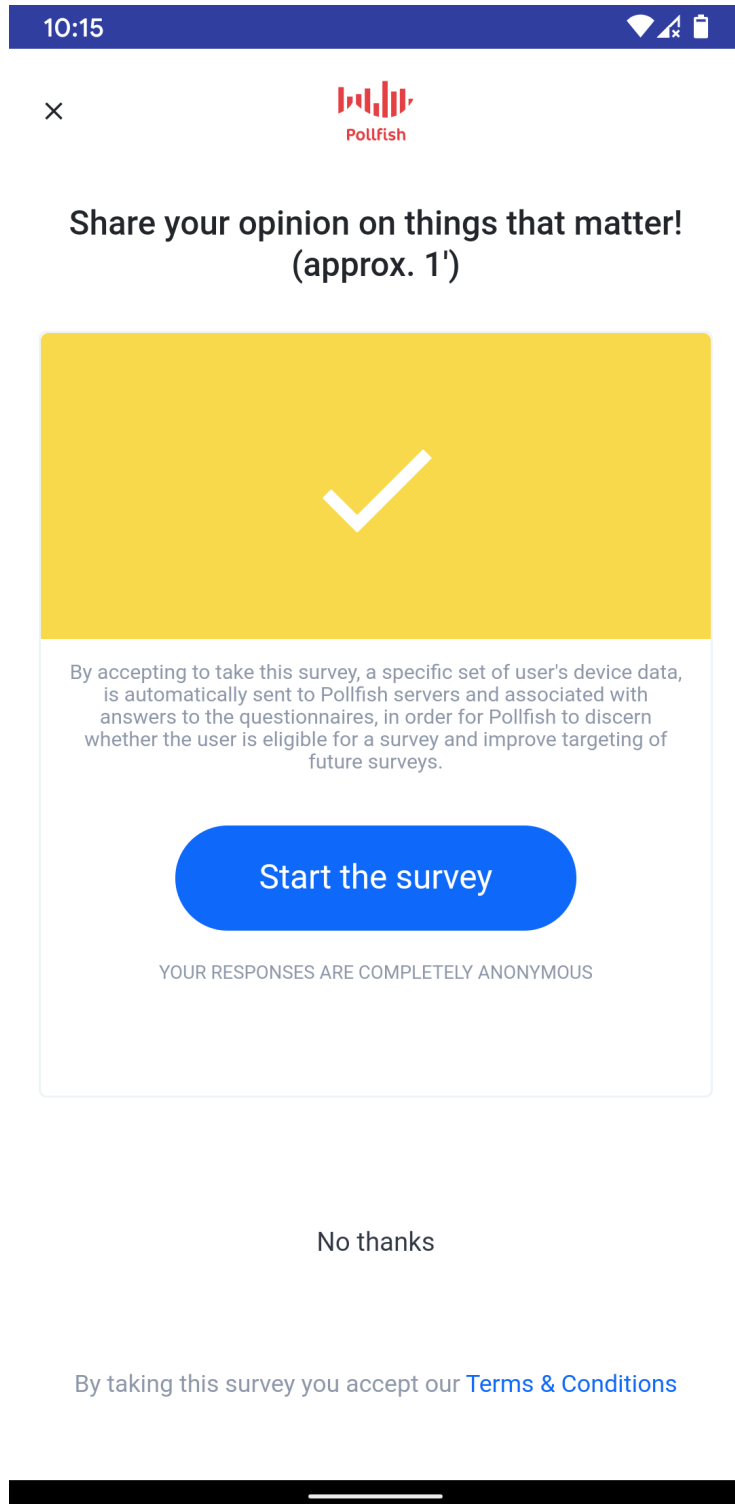
Terakhir, kita menangani hasil survei yang berhasil menggunakan blok Survey Completed. Di sinilah Anda dapat memberi penghargaan kepada pengguna Anda dengan item non-tunai (koin dalam game, kredit, dll.)

```

when Pollfish1 .Survey Completed
  survey Info
do set global surveyReceived to false
  set Button1 .Text to "Reward!"

```

Kesimpulan







```
initialise global surveyReceived to false
```

```
when Pollfish1 .Survey Received
  survey Info
do set global surveyReceived to true
```

```
when Pollfish1 .Survey Not Available
do call SurveyNotAvailable
```

```
when Pollfish1 .User Rejected
do call SurveyNotAvailable
```

```
when Pollfish1 .User Not Eligible
do call SurveyNotAvailable
```

```
to SurveyNotAvailable
do set Button1 .Text to "No survey is available right now. Please try lat..."
  set global surveyReceived to false
```

```
when Button1 .Click
do if get global surveyReceived
  then call Pollfish1 .Show Survey
```

```
when Pollfish1 .Survey Completed
  survey Info
do set global surveyReceived to false
  set Button1 .Text to "Reward!"
```

Berikut adalah ringkasan dari apa yang telah Anda pelajari dalam panduan ini.

- Cara mengatur akun penerbit Pollfish.
- Cara menemukan kunci API Pollfish Anda.

- Cara menggunakan komponen Pollfish untuk menampilkan Survei Tunggal kepada pengguna.

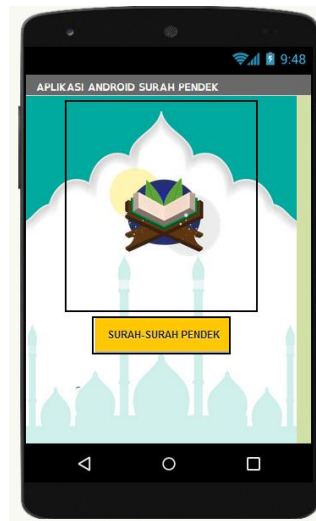
BAB 14
APP SURAH-SURAH PENDEK AL-QURAN

MERANCANG DESAIN APP SURAH-SURAH PENDEK AL-QURAN

1. Merancang Desain Interface (Tampilan Screen)

Dalam perancangan interface ini penulis memetakan menjadi 5 Screen dimana dapat dipetakan sebagai berikut,

A. Tampilan Screen 1



Berikut metode perancangan desain

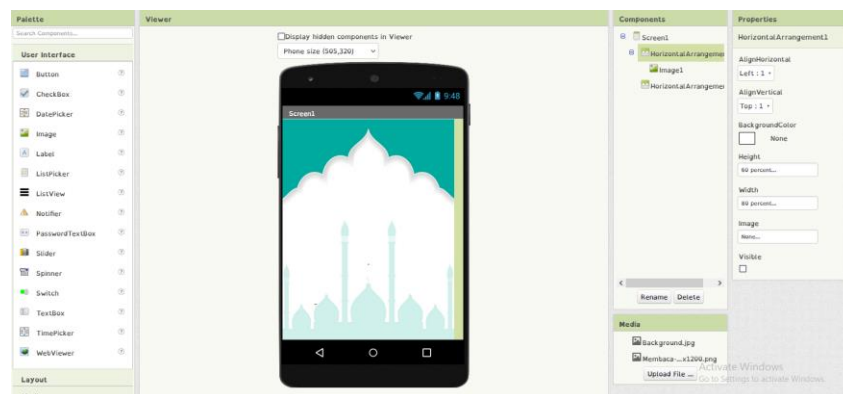
1. Pilih menu screen1 pada laman components

Selanjutnya lakukan pengaturan tataletak pada ;

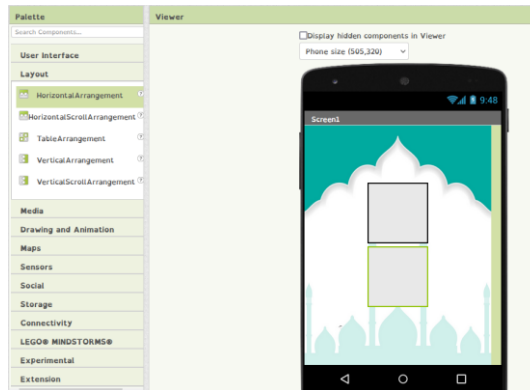
→ AligHorizontal [Center : 3]

→ AligVertical [Top : 1]

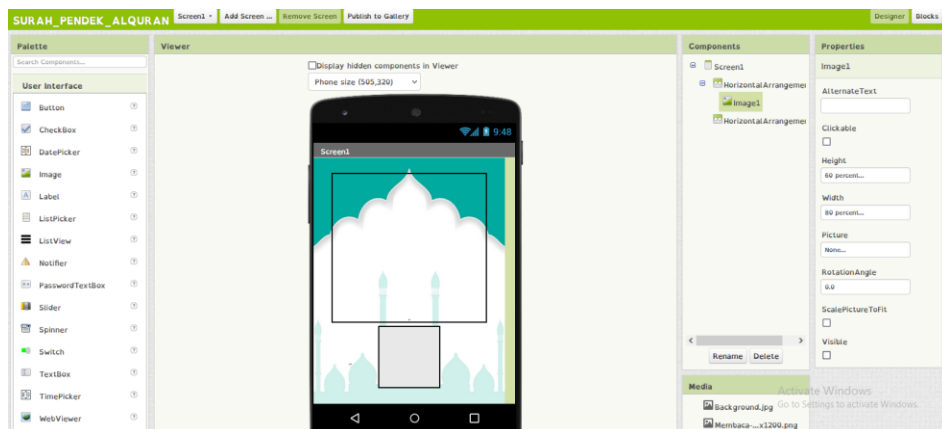
Kemudian pilih menu BackgroundImage untuk pilih gambar background seperti gambar berikut,



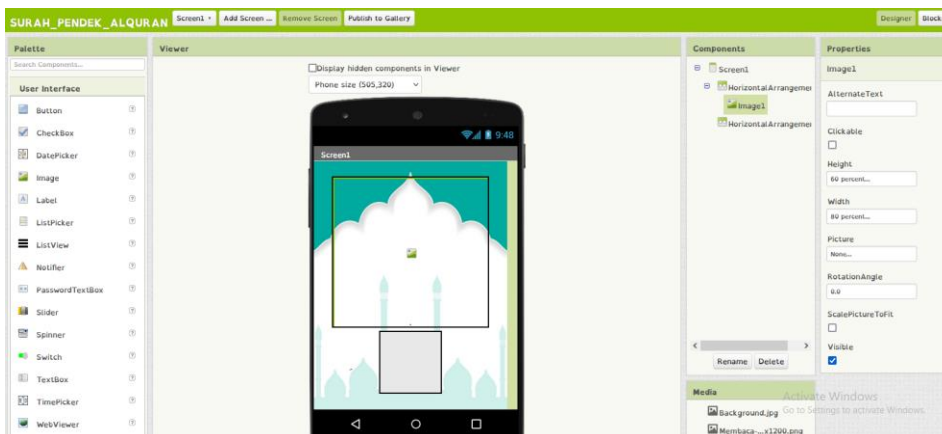
- Selanjutnya pilih menu HorizontalArrangement di layout pada laman Palette untuk selanjutnya drag menuju laman viewer seperti gambar berikut ini,



- Lakukan pengaturan tataletak pada menu HorizontalArrangement bagian atas pada laman properties,
 - Height [60 percents]
 - Width [80 percents]Atur menu BackgroundColor dengan none

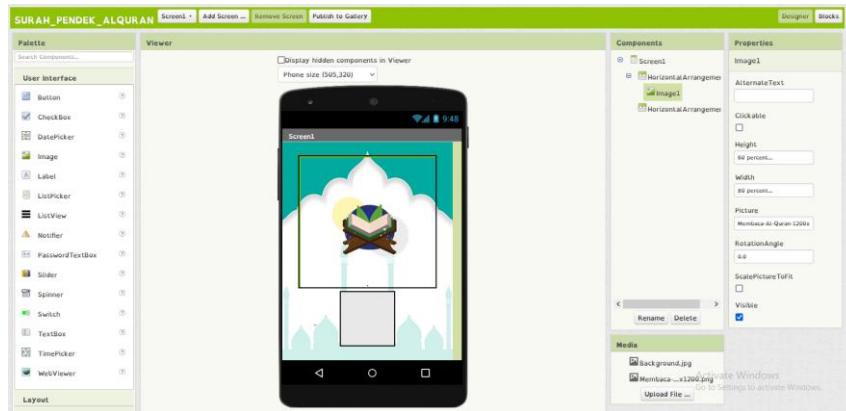


- Selanjutnya pilih menu image di User Interface pada laman palette kemudian drag menuju HorizontalArrangement bagian atas seperti gambar berikut ini,



- Lakukan pengaturan tataletak pada menu image pada laman properties,
 - Height [60 percents]
 - Width [80 percents]

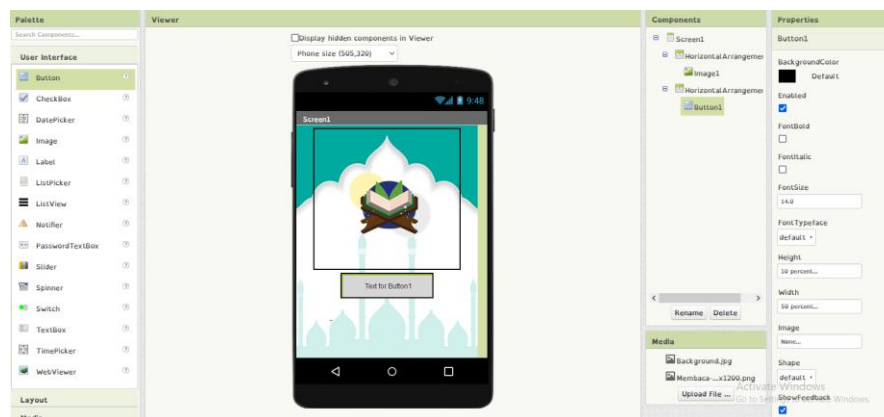
Selanjutnya upload gambar pada menu picture sebagaimana gambar berikut,



- Selanjutnya Lakukan pengaturan tataletak pada menu HorizontalArrangement bagian bawah pada laman properties,
 - Height [10 percents]
 - Width [50 percents]

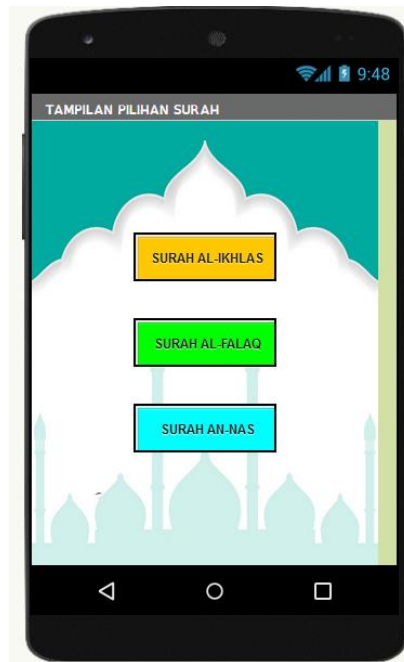
Untuk selanjutnya pilih menu Button di user interface pada laman palette untuk selanjutnya drag menuju HorizontalArrangement bagian bawah seperti gambar dibawah ini dan atur tataletak sebagaimana berikut,

- Height [10 percents]
- Width [50 percents]



Selanjutnya setting menu button dengan kalimat “Surah-Surah Pendek” dan ubah warna menu button dengan warna kuning

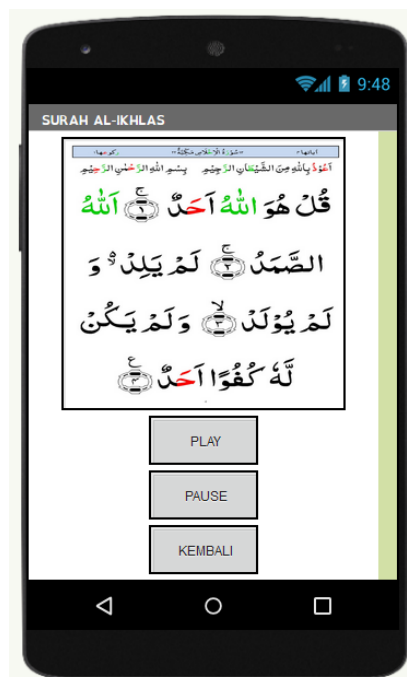
B. Tampilan Screen 2



Pada screen 2 ini dilakukan setting tataletak dengan menggunakan layout HorizontalArragement, dengan dilakukan pembagian 3 menu button yang diatur sesuai dengan pengaturan pada sreech 1

“Note : ikuti saat tutor diwebiner”

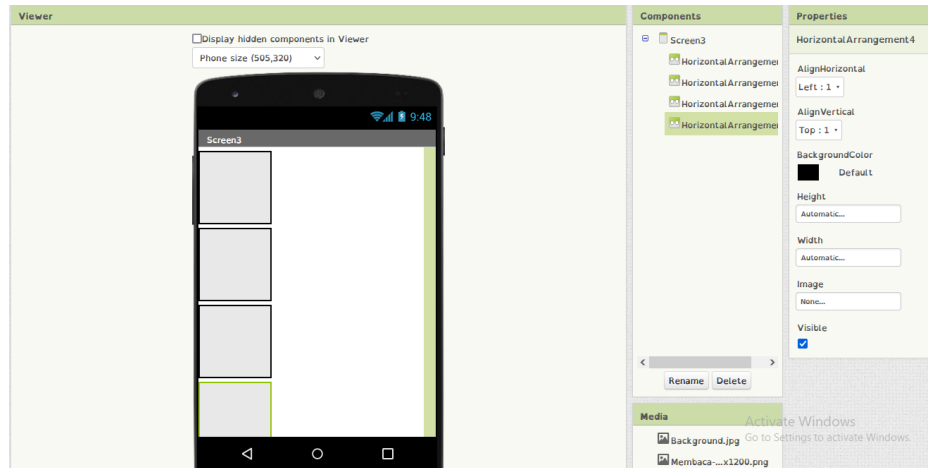
C. Tampilan Screen 3



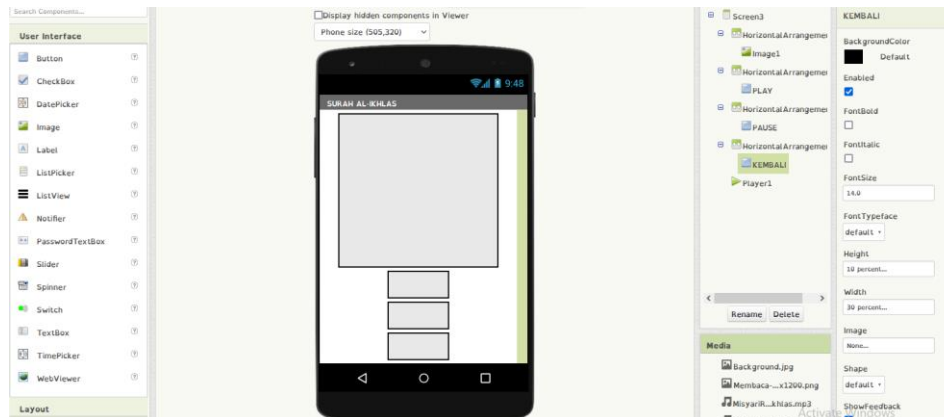
Berikut metode perancangan desain

Pada perancangan screen 3 ini kita akan membuat tampilan surah Al-Ikhlas dengan tambahan 3 menu button yang masing masing memiliki fungsi masing-masing. Adapun perancangan desain dapat di jabarkan sebagai berikut

1. pilih menu HorizontalArrangement sebanyak 4 buah



2. atur tata letak masing-masing HorizontalArrangement



HorizontalArrangement bagian atas (no.1) pada laman properties,

→ Height [60 percents]

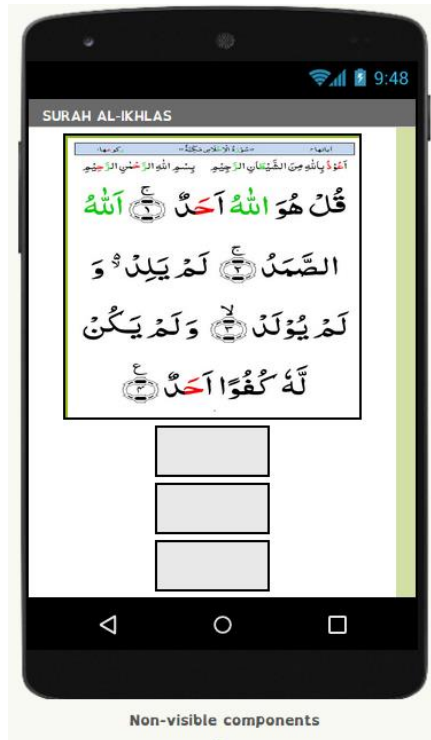
→ Width [80 percents]

HorizontalArrangement bagian bawah (no.2,3,4) pada laman properties,

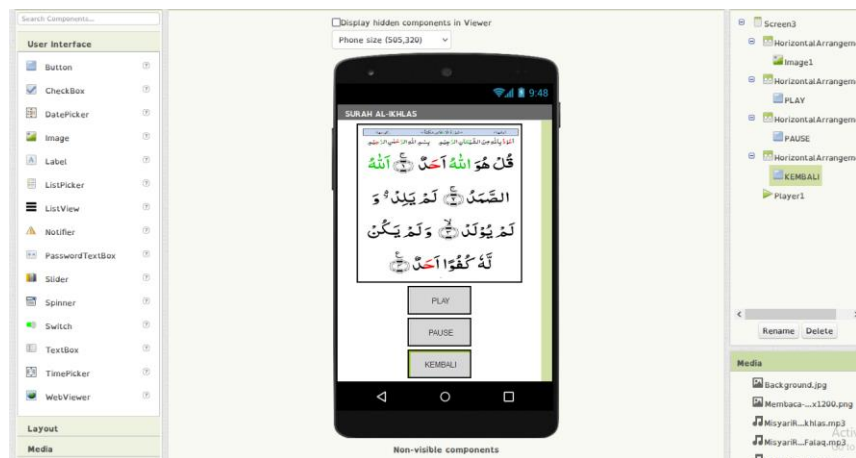
→ Height [10 percents]

→ Width [30 percents]

3. selanjutnya pilih menu image untuk didrag ke HorizontalArrangement bagian atas selanjutnya atur setting tata letak dan gambar image seperti pada gambar dibawah ini,



4. selanjutnya pilih menu button untuk didrag ke HorizontalArrangement bagian bawah (no.2,3,4) selanjutnya atur setting tata letak dan setting dengan isian nama (Play ; Pause ; Kembali) sebagaimana gambar berikut ini.



5. langkah selanjutnya pilih menu palyer di media pada laman pallete untuk di drag ke viewer sebagai mana berikut



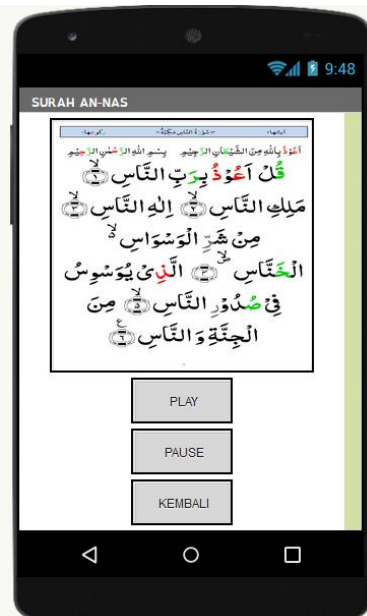
D. Tampilan Screen 4



setting screen 4 hampir sama dengan setting pada screen 3 namun tampilan pada surahnya yang dirubah

“Note : ikuti saat tutor diwebiner”

E. Tampilan Screen 5



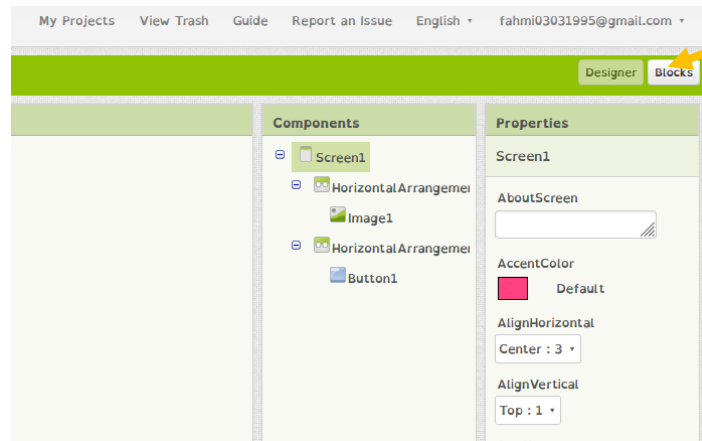
setting screen 5 hampir sama dengan setting pada screen 3 namun tampilan pada surahnya yang dirubah

“Note : ikuti saat tutor diwebiner”

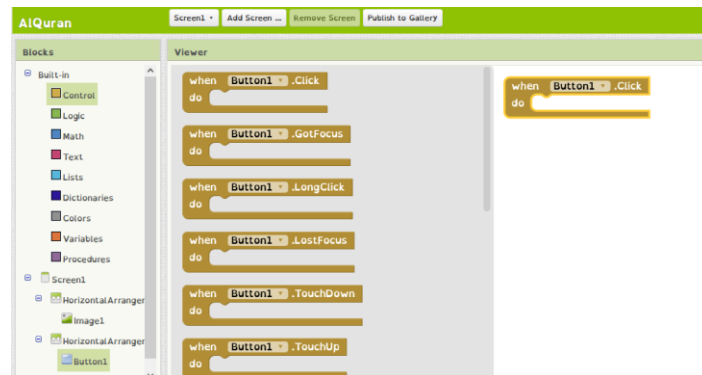
B. Memprogram Hasil Desain Hasil Rancangan

Pada perancangan program ini kita akan membuat aplikasi surah-surah pendek dalam alquran sebagaimana berikut

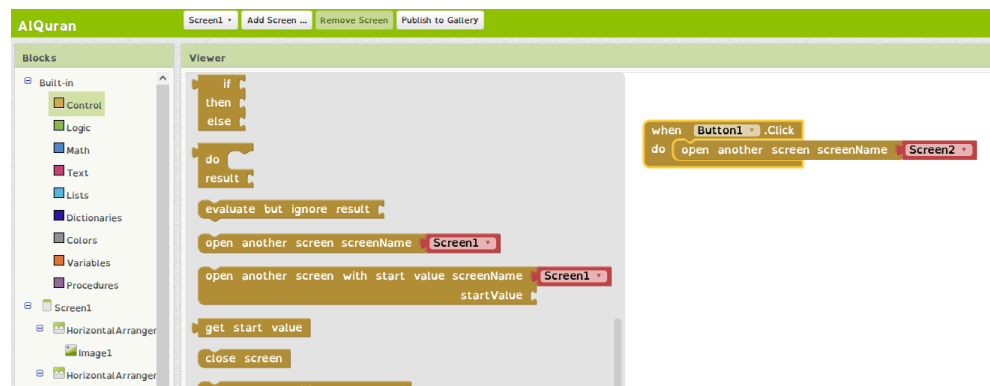
1. pemrograman pada screen 1
pertama klik menu bloks, seperti gambar berikut ini



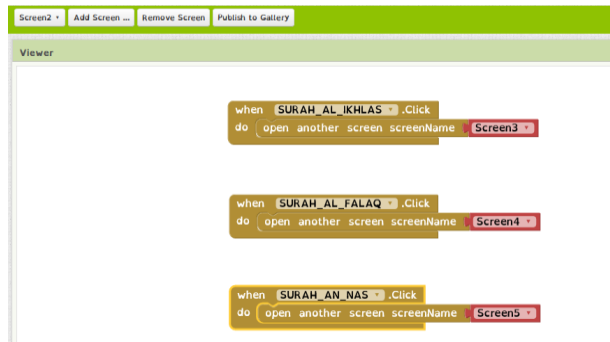
Klik menu button selanjutnya drag ke viewer



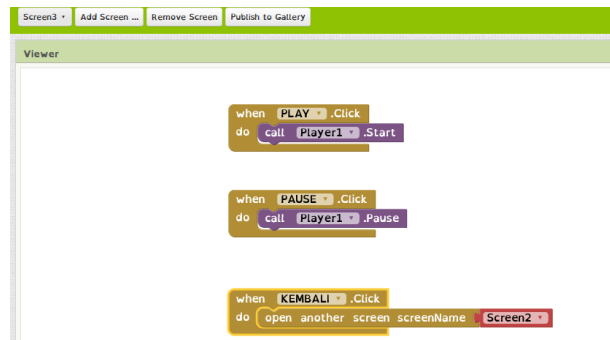
Klik menu control pilih open another screen screenName



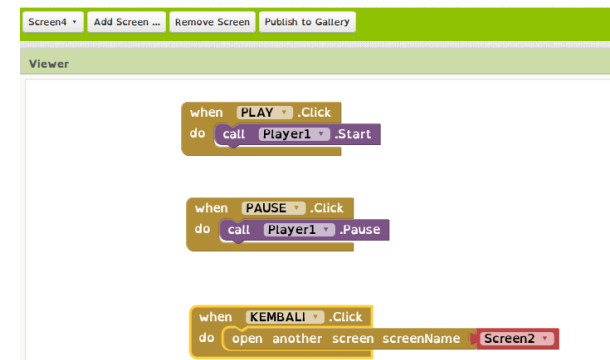
2. pemrograman pada screen 2



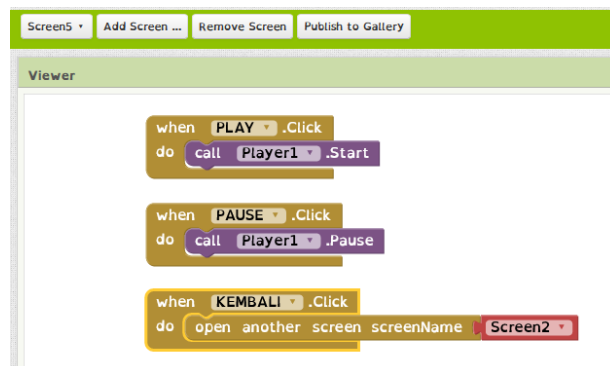
3. pemrograman pada screen 3



4. pemrograman pada screen 4



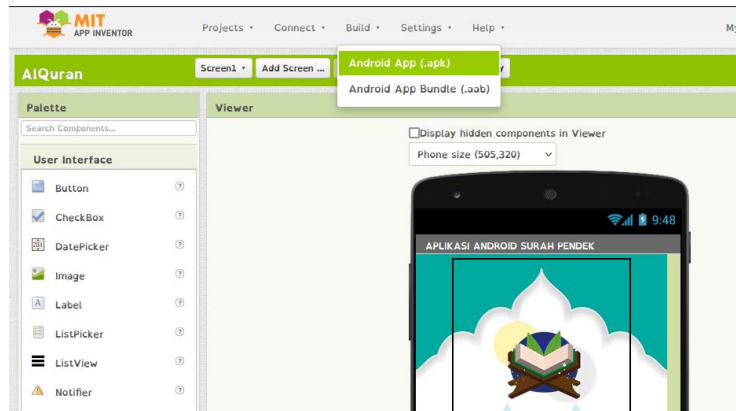
5. pemrograman pada screen 5



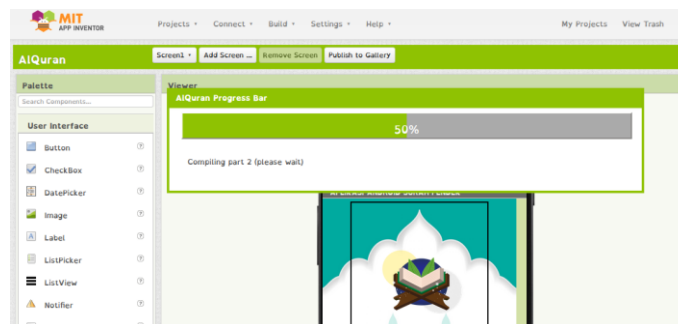
2. proses building Aplikasi (.APK)

Pada proses building aplikasi ini kita akan membuat .apk sebagai ekstensi untuk kita install di hp kita agar aplikasi dapat berjalan dengan baik.

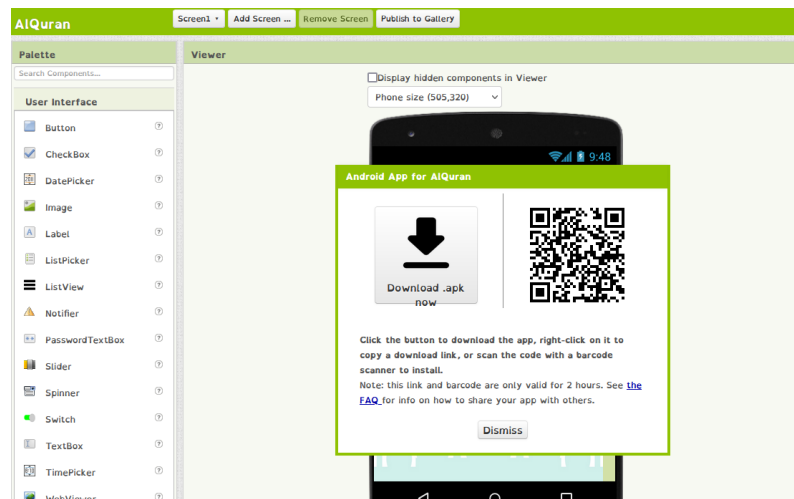
1. pilih menu Build selanjutnya klik Andorid App (.apk)



2. Tunggu proses sampai selesai



3. Kemudian scan dengan aplikasi scanner MIT inventor



4. Selanjutnya instaal di hp andorid masing-masing.

DAFTAR PUSTAKA

- Classroom. In Proceedings of the 18th Panhellenic conference on informatics (pp. 1-6).
- Inventor. In Proceedings of the 50th Annual Southeast Regional Conference (pp. 405-406).
- Gray, J., Abelson, H., Wolber, D., & Friend, M. (2012, March). Teaching CS principles with app
- Kong, S.-C., & Abelson, H. (2019). *Computational Thinking Education*. Hong Kong: This Springer.
- Kodular. (2022, August 5). *Kodular*. Retrieved from Kodular: <https://www.kodular.io>
- MIT. (2022, August 5). *MIT App Inventor*. Retrieved from MIT App Inventor: <https://appinventor.mit.edu>
- Perdikuri, K. (2014, October). Students' Experiences from the use of MIT App Inventor in