

**DATABASE DAN FILE
MIT APP INVENTOR
(CloudDB, File, TinyDB, TinyWebDB)**

**Ir. Agus Suharto, M.Kom.
Eko Suharyanto, S.T., M.Kom.**



eureka
media aksara

PENERBIT CV.EUREKA MEDIA AKSARA

**DATABASE DAN FILE
MIT APP INVENTOR
(CloudDB, File, TinyDB, TinyWebDB)**

Penulis : Ir. Agus Suharto, M.Kom.
Eko Suharyanto, S.T., M.Kom.

Desain Sampul : Ardyan Arya Hayuwaskita

Tata Letak : Ayu May Lisa

ISBN :

Diterbitkan oleh : **EUREKA MEDIA AKSARA, APRIL 2024**
ANGGOTA IKAPI JAWA TENGAH
NO. 225/JTE/2021

Redaksi:

Jalan Banjaran, Desa Banjaran RT 20 RW 10 Kecamatan Bojongsari
Kabupaten Purbalingga Telp. 0858-5343-1992

Surel : eurekamediaaksara@gmail.com

Cetakan Pertama : 2024

All right reserved

Hak Cipta dilindungi undang-undang

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun dan dengan cara apapun, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya tanpa seizin tertulis dari penerbit.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah Kami panjatkan puja dan puji syukur kehadirat Allah SWT yang senantiasa melimpahkan segala rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan buku dengan judul "*Database dan File MIT APP Inventor (CloudDB, File, TinyDB, TinyWebDB)*." Buku ini sebagai referensi bagaimana bekerja dengan Database storage (CloudDB, Files, TinyDB, TinyWeb DB) serta contoh sederhana.

Pokok dari penulisan ini adalah memenuhi kebutuhan kalangan masyarakat umum, mahasiswa/i dari berbagai program studi yang ingin belajar bagaimana membuat aplikasi berbasis mobile menggunakan MIT App Inventor berbasis block. Teknik penyajian yang diangkat dilakukan dengan tahapan tahapan mulai dari membuat proyek, desain setting properties, block sampai melakukan ujicoba ke perangkat gadget.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kasih kepada semua pihak yang telah membantu penyelesaian buku ini.

Penulis menyadari bahwa buku ini masih banyak kekurangannya, untuk itu penulis sangat membuka saran dan kritik yang sifatnya membangun.

Mudah-mudahan buku ini memberikan manfaat bagi kita semua. Aamiin Ya Rabbal Alamin.

Jakarta, April 2024

Penulis

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
BAB 1 PENGANTAR MIT APP INVENTOR 2.....	1
A. Pendahuluan.....	1
B. Konsep MIT App Inventor	2
C. Persyaratan Sistem Mit App Inventor	11
D. Istilah istilah pada MIT App Inventor	12
E. Memulai MIT App Inventor	14
F. Pembahasan Pada Buku ini	16
BAB 2 BEKERJA DENGAN CLOUDDB.....	18
A. Pendahuluan.....	18
B. <i>Properties</i> CloudDB.....	19
C. <i>Events</i> CloudDB	20
D. <i>Methods</i> pada <i>CloudDB</i>	21
E. Contoh Proyek menggunakan Komponen Clouddb	23
F. Contoh Proyek Aplikasi Login Clouddb.....	35
BAB 3 BEKERJA DENGAN FILE STORAGE.....	54
A. Pendahuluan.....	54
B. <i>Properties</i> Pada Komponen File.....	56
C. <i>Event</i> pada File.....	57
D. <i>Method</i> pada File.....	57
E. Contoh Proyek Aplikasi menggunakan File text.....	61
BAB 4 BEKERJA DENGAN TINYDB STORAGE	76
A. Pendahuluan.....	76
B. <i>Properties</i> pada Komponen TinyDB	77
C. <i>Event</i> pada Komponen TinyDB.....	77
D. <i>Method</i> pada Komponen TinyDB	78
E. Contoh Aplikasi TinyDB.....	78
BAB 5 BEKERJA DENGAN TINYWEBDB STORAGE	117
A. Pendahuluan.....	117
B. <i>Properties</i> pada komponen TinyWebDB	118
C. <i>Event</i> Pada Komponen TinyWebDB	118
D. <i>Method</i> pada TinyWebDB.....	119

E. Contoh Proyek Aplikasi Menggunakan	
TinyWebDB	120
DAFTAR PUSTAKA	134
TENTANG PENULIS	135



**DATABASE DAN FILE
MIT APP INVENTOR
(CloudDB, File, TinyDB, TinyWebDB)**

**Ir. Agus Suharto, M.Kom.
Eko Suharyanto, S.T., M.Kom.**



BAB 1 | PENGANTAR *MIT APP INVENTOR 2*

A. Pendahuluan

MIT App Inventor adalah lingkungan pemrograman visual dan intuitif yang memungkinkan semua orang, bahkan anak-anak dapat membuat aplikasi yang berfungsi penuh untuk ponsel Android, iPhone, dan tablet Android/iOS. Mereka yang baru mengenal MIT App Inventor dapat mengaktifkan dan menjalankan aplikasi pertama yang sederhana dalam waktu kurang dari 30 menit. Terlebih lagi, alat berbasis blok ini memfasilitasi pembuatan aplikasi yang kompleks dan berdampak tinggi dalam waktu yang jauh lebih singkat dibandingkan lingkungan pemrograman tradisional. Proyek MIT App Inventor berupaya mendemokratisasi pengembangan perangkat lunak dengan memberdayakan semua orang, terutama generasi muda, untuk beralih dari konsumsi teknologi menjadi pencipta teknologi.

MIT App Inventor dibangun oleh Sebuah tim kecil yang terdiri dari staf dan mahasiswa MIT CSAIL, dipimpin oleh Profesor Hal Abelson, membentuk inti gerakan penemu internasional. Selain memimpin penjangkauan pendidikan seputar MIT App Inventor dan melakukan penelitian mengenai dampaknya, tim inti ini mengelola lingkungan pengembangan aplikasi online gratis yang melayani lebih dari 6 juta pengguna terdaftar.

Program pengkodean berbasis blok menginspirasi pemberdayaan intelektual dan kreatif. MIT App Inventor melakukan lebih dari itu dengan memberikan pemberdayaan nyata bagi anak-anak untuk membuat perbedaan. ini sebuah cara untuk mencapai dampak sosial yang bernilai tak terukur bagi komunitas mereka.

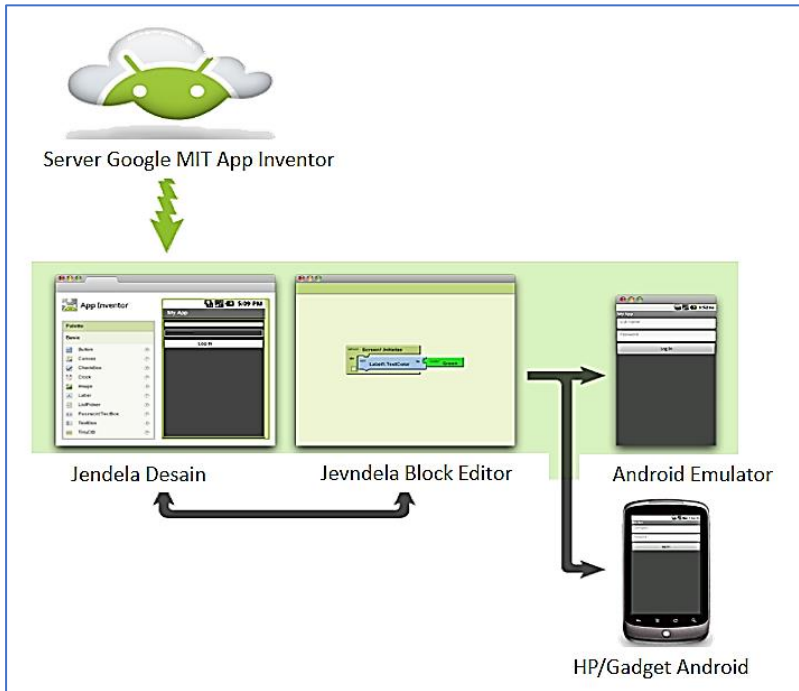
Dengan lebih dari satu juta pengunjung unik setiap bulannya dari 195 negara yang secara kolektif menciptakan hampir 30 juta aplikasi, MIT App Inventor mengubah cara dunia membuat aplikasi dan cara anak-anak belajar tentang komputasi.

MIT App Inventor Foundation adalah organisasi nirlaba 501c3 yang didirikan oleh pencipta App Inventor dari MIT dan Google (termasuk Hal Abelson, profesor MIT, Mark Friedman, mantan Googler, Jeff Schiller, MIT Enterprise Architect, dan Natalie Lao, Direktur Eksekutif). App Inventor Foundation memperluas inisiatif pendidikan proyek App Inventor dengan menawarkan sumber daya tambahan untuk guru dan siswa di seluruh dunia.

Untuk masuk ke MIT App Inventor, kita perlu menggunakan akun Google. Penggunaan atas akun google kita tersebut tunduk pada persyaratan layanan google dan informasi yang kita berikan kepada Google diatur oleh kebijakan privasi Google. MIT tidak memiliki akses ke akun Google kita atau informasi yang diberikan selain alamat email akun, yang mungkin digunakan untuk menghubungi kita. Di masa depan, pusat pembelajaran Seluler dapat memberikan cara alternatif untuk menggunakan MIT App Inventor tanpa akun Google.

B. Konsep MIT App Inventor

App Inventor memungkinkan kita mengembangkan aplikasi untuk ponsel Android / Iphone menggunakan browser web dan untuk pengujiannya menggunakan ponsel atau emulator yang terhubung. Server App Inventor menyimpan pekerjaan proyek kita dan membantu melacak proyek yang sudah tersimpan.



Beberapa konsep dalam App Inventor yang penting untuk diketahui pada daftar topik disertakan di bawah ini.

1. Variabel Global vs Variabel Lokal

Variabel dapat dianggap sebagai wadah yang menyimpan suatu nilai. Ada dua jenis variabel yang bisa kita gunakan yaitu global dan lokal.

a. Global Variable

Variabel global adalah variabel yang dapat diakses dalam berbagai cakupan. Artinya dimanapun variable berada dalam program, kita dapat menggunakan variabel tersebut untuk mendapatkan nilainya saat ini atau set nilainya ke nilai lain. Variabel global dibuat menggunakan nama *initialize global name to* yang ada pada *Drawer Variable*.

Contoh:



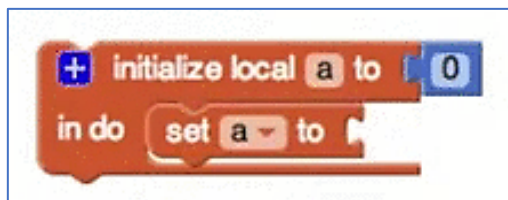
b. Local Variable

Variabel lokal adalah variabel yang dideklarasikan dalam suatu fungsi atau merupakan argumen yang diteruskan ke suatu fungsi. Ini berarti kita hanya dapat mengakses variabel-variabel ini dalam fungsi spesifik tempat variabel tersebut dideklarasikan atau diteruskan sebagai argumen.

Variabel lokal dibuat ketika:

- 1) Argumen diteruskan ke prosedur atau events
- 2) Menggunakan *Inizialize Local Name to* pada blok
- 3) Menggunakan *a for each in list* atau *for each from to* ke blok (perulangan for ini akan membuat variabel lokal).

Contoh:



Perhatikan bagaimana saat Anda menggunakan blok get atau set untuk variabel global. Blok tersebut akan menyebutkan nama global.



Saat menggunakan blok get atau set untuk variabel lokal, blok tersebut hanya akan menyebutkan nama. Ingatlah bahwa variabel lokal mencakup variabel yang dibuat dari argumen ke prosedur atau *events*, variabel yang dibuat untuk digunakan dalam perulangan for, atau menginisialisasi variabel lokal untuk ekspresi atau pernyataan pengembalian dengan menggunakan blok inisialisasi variabel lokal berwarna oranye.



Ringkasan

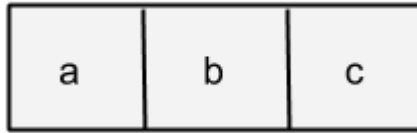
Jadi saat berada di dalam blok variabel lokal atau prosedur yang memiliki argumen, argumen atau variabel lokal yang kita berikan nama juga akan tersedia di dropdown. Namun saat kita tidak berada di dalam, variable tidak akan tersedia. Saat kita memilih nilai di dropdown, nilai internal yang disimpan ke variabel tersebut akan dikembalikan dan diteruskan ke blok mana pun yang terhubung dengannya.

2. Menggunakan Lists pada MIT App Inventor

Beberapa aplikasi menggunakan Lists data. Misalnya, sebuah game mungkin menyimpan daftar skor tertinggi dan aplikasi Facebook kita menyimpan daftar teman-teman kita.

Lists adalah jenis struktur data yang digunakan di setiap bahasa pemrograman, bukan hanya App Inventor. menggunakan *Lists* untuk membuat dan memanipulasi kumpulan nilai/elemen yang berbeda.

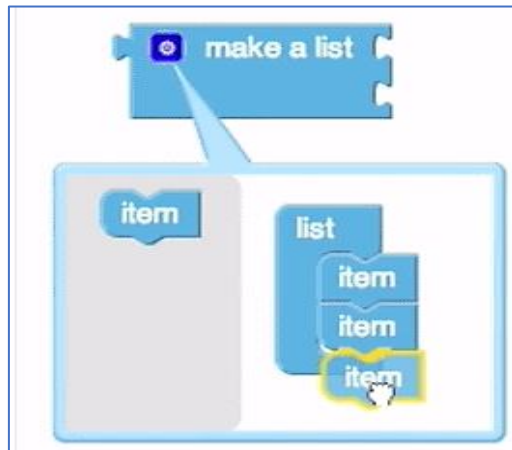
Posisi suatu elemen dalam *Lists* disebut indeks. Di App Inventor, elemen pertama dalam *Lists* adalah indeks 1. Pada gambar di bawah, a memiliki indeks 1, b memiliki indeks 2, dan c memiliki indeks 3.



Artinya adalah kita dapat mereferensikan elemen tertentu di dalam *Lists* kita jika kita mengetahui indeks mana yang dimilikinya dan apa nama *Lists* nya.

Di App Inventor, mungkin berguna jika memiliki *Lists* untuk menyimpan data seperti nomor di ponsel Anda, balasan untuk aplikasi Magic Eight Ball, atau nama orang yang Anda temui di konferensi.

App Inventor membuat *Lists* dengan menggunakan blok *make a list*. Blok ini adalah mutator dan dapat diperluas atau diperkecil dengan mengklik tanda tambah berwarna biru.



3. Perintah dan ekspresi

Perintah adalah sebuah blok yang menentukan tindakan yang akan dilakukan, ketika sebuah event handler diaktifkan, perintah akan mengeksekusi serangkaian perintah di tubuhnya. Misalnya pada kode blok dibawah ini, meng klik tombol **BtnHitung**.

Berikut beberapa contoh perintah yang tersedia pada sebuah aplikasi:



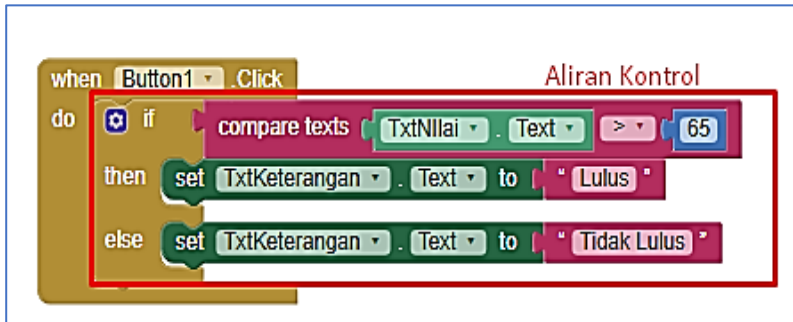
Blok perintah pada kotak berwarna orange didalamnya terdapat 4 perintah

- Memberikan nilai Global Alas dengan mengambil Nilai dari komponen TxtAlas.Text
- Memberikan nilai Global Tinggi dengan mengambil Nilai dari komponen TxtTinggi.Text
- Memberikan nilai Global Luas dengan proses kalkulasi $0.5 \times \text{Var. Global Alas} \times \text{Var. Global Tinggi}$
- Memberikan Nilai TxtLuas dengan hasil kalkulasi pada Var. Global Luas

4. Control Flow /Aliran Kontrol

Aliran kontrol adalah urutan di mana pernyataan, instruksi, atau pemanggilan fungsi individual dieksekusi atau dievaluasi. Aliran kontrol pada MIT App Inventor sama seperti bahasa pemrograman konvensional pada umumnya diatur oleh pernyataan kondisional, loop, dan pemanggilan fungsi.

Contoh Blok:

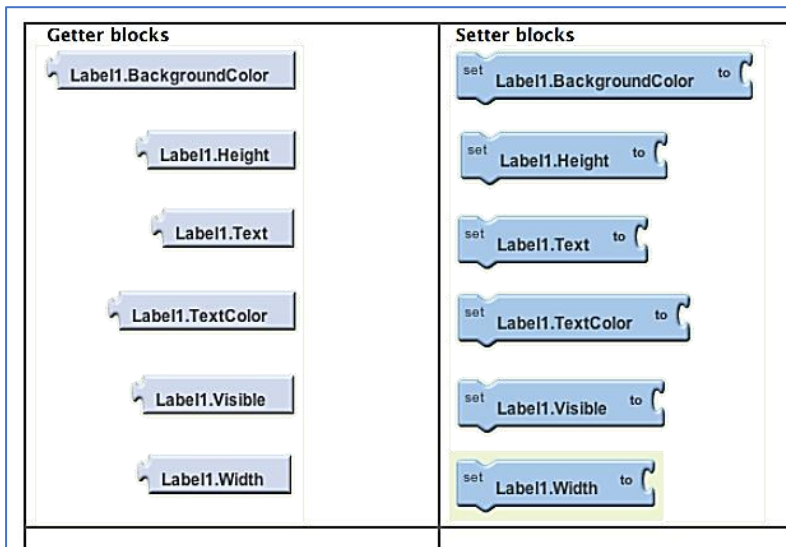


5. Penyusunan Komponen pada Layar

Komponen aplikasi disusun secara vertikal secara default. Di palet Designer, menggunakan Horizontal Arrangement dan Vertical Arrangement memungkinkan kita untuk mengubah organisasi komponen.

6. Memanipulasi Status Komponen: Menggunakan Getter & Setter

Setiap komponen mempunyai karakteristik yang berbeda-beda. Apa sajakah properti komponen Label? Nilai properti saat ini adalah status pada komponen. Kita dapat menentukan keadaan awal suatu komponen di panel Properties pada jendela Designer. Program App Inventor dapat memperoleh dan mengatur sebagian besar properti komponen melalui blok. Misalnya, berikut adalah blok untuk memanipulasi status Label1.



Blok pengambil (*Getter*) adalah ekspresi yang mendapatkan nilai properti saat ini. Blok penyetel (*Setter*) adalah perintah yang mengubah nilai yang terkait dengan properti. Beberapa properti Label tidak dapat dimanipulasi oleh blok.

7. Membuat Keputusan: Menggunakan Blok Bersyarat

Terkadang kita ingin aplikasi kita melakukan tindakan berbeda dalam kondisi berbeda. Jika Anda membuat aplikasi untuk menyimpan semua jam kerja dalam minggu ini, Anda perlu menguji hari apa dalam seminggu untuk mengetahui di mana menyimpan jam kerja tersebut.

Untuk menerapkan ini ke aplikasi, kita perlu menggunakan kondisional. Persyaratan mengacu pada ekspresi atau pernyataan yang bernilai benar atau salah.

8. Events dan Event Handlers

Events adalah pemberitahuan perubahan dalam suatu aplikasi yang diaktifkan. Halaman web, atau screen dapat bereaksi secara efektif terhadap perubahan berkat kode event

handler yang dapat dirancang dan dijalankan oleh pemrogram setiap kali suatu peristiwa terjadi.

Events dimulai oleh pengguna akhir aplikasi. Misalnya, jika pengguna mengklik tombol, aplikasi merespons dengan melakukan beberapa operasi (*Events Click*). Untuk ponsel dan perangkat layar sentuh, tindakan menyeret jari (drag) aplikasi mungkin merespons kejadian tersebut dengan menggambar garis dari titik sentuhan awal ke titik di mana kita mengangkat jari.

Events handler mirip dengan pengawal, selalu memperhatikan kejadian atau 'peristiwa' tertentu dan bereaksi terhadapnya. Dalam konteks MIT App Inventor, *Events* dapat berupa apa saja mulai dari klik mouse hingga penekanan tombol keyboard. Ketika peristiwa seperti itu terjadi, pengendali peristiwa akan bertindak, mengeksekusi blok kode yang dirancang untuk merespons peristiwa tersebut.

Contoh:

Events Click:



Events Item Select



9. Generator Bilangan Pseudorandom

Adalah algoritma untuk menghasilkan barisan bilangan yang mendekati sifat-sifat bilangan secara acak dimana untuk barisan bilangan selanjutnya tidak dapat diprediksi. Pada umumnya Generator bilangan

pseudorandom dimanfaatkan untuk digunakan dalam bidang krypto, simulasi, dan permainan elektronik. Algoritma ini disebut pseudorandom karena membutuhkan nilai pembangkit(*Seed*), dimana nilai seed ini sangat menentukan hasil dari bilangan acak. Jika nilai seed yang digunakan adalah bilangan benar benar acak, maka yang dihasilkan.

10. Data dan Database

Database atau basis data adalah tempat penyimpanan informasi atau data hingga dihapus atau diganti. Contoh nya Facebook menggunakan database untuk menyimpan nama pengguna dan kata sandi yang sesuai. Pada Perangkat Android memiliki database internal yang menyimpan informasi tentang kita atau ponsel kita. Dengan MIT App Inventor memungkinkan kita mengakses dan memudahkan penyimpanan data melalui komponen TinyDB dan CloudDB. Data selalu disimpan sebagai pasangan nilai tag dan tag untuk mengidentifikasi data untuk diambil nanti. Jika penyimpanan data langsung pada aplikasi maka dapat menggunakan TinyDB. Jika data digunakan dibagikan ke seluruh ponsel (misalnya, untuk game multipemain atau aplikasi pemungutan suara), kita dapat menggunakan CloudDB.

C. Persyaratan Sistem Mit App Inventor

1. Komputer dan sistem operasi

- a. Macintosh (dengan prosesor Intel): Mac OS X 10.5 atau lebih tinggi
- b. Windows: Windows 7, Windows 10 atau lebih tinggi
- c. GNU/Linux: Ubuntu 8 atau lebih tinggi, Debian 5 atau lebih tinggi (Catatan: Pengembangan langsung GNU/Linux hanya didukung untuk koneksi WiFi antara komputer dan ponsel cerdas/tablet.)

2. Browser

- a. Mozilla Firefox 3.6 atau lebih tinggi (Catatan: Jika Anda menggunakan Firefox dengan ekstensi NoScript, Anda harus mematikan ekstensi tersebut. Lihat catatan di halaman pemecahan masalah.)
- b. Apple Safari 5.0 atau lebih tinggi
- c. Google Chrome 4.0 atau lebih tinggi

Catatan: Microsoft Internet Explorer tidak didukung

3. Ponsel atau Tablet (atau emulator)

- a. iOS 9.0 atau lebih baru di iPhone, iPad, atau iPod Touch
- b. macOS 11 atau lebih tinggi di komputer dengan Apple M1 Silicon
- c. Sistem Operasi Android 2.1 ("Eclair") atau lebih tinggi untuk perangkat Android.

D. Istilah istilah pada MIT App Inventor

1. Argumen

Seringkali dalam Ilmu Komputer, Input terhadap prosedur atau *events* disebut argumen. Argumen ini adalah variabel lokal yang cakupannya berada di dalam prosedur atau *events* tersebut.

2. Behaviour

Suatu aplikasi dikatakan memiliki Behaviour. Behaviour adalah cara aplikasi merespons peristiwa eksternal dan yang dimulai oleh pengguna.

3. Block

App Inventor adalah bahasa pemrograman *block*. *Block* adalah bagian yang disusun untuk memberi tahu aplikasi apa yang dibuat. Penyusunan blok dilakukan pada Editor Blok. Block disini adalah kumpulan atau code block berbentuk graphic seperti puzzle yang berwarna warni, dimana didalamnya terdapat komponen komponen Logic, Control, Math, Text, Lists, Colors, Variables, dan Procedures.

4. *Block Editor*

Jendela Block ditemukan dengan mengklik tombol *Block* pada layar desain. Di sinilah aplikasi kita dibangun dengan menyusun komponen-komponen yang dibutuhkan.

5. *Condition*

Kondisi digunakan untuk membuat keputusan dalam program Anda. Ini membantu untuk membuat keputusan dalam program berdasarkan apakah kondisi tersebut dinilai benar atau salah. Itu seperti pernyataan jika-maka. Misalnya kalau hujan, maka saya akan membawa payung. Jika syaratnya terpenuhi, maka akan diambil tindakan tertentu. Di bawah ini adalah contoh kondisi di Scratch.

6. *Component*

Adalah bagian aplikasi yang melakukan tindakan untuk kita. Pada layar desain, komponen diseret/drag dari Palet Komponen dan ditempatkan di Layar Desain. Contoh komponennya adalah Label, Suara, atau Tombol.

7. *Design*

Layar tempat kita dapat menarik dan melepas bagian komponen dan mendesainnya menggunakan Antarmuka Pengguna.

8. *Drawer*

MIT App Inventor menggunakan metafora drawer/laci yang berisi potongan puzzle untuk pemrograman. Setiap item dalam palet *block* bawaan dianggap sebagai *drawer*/laci contoh *drawer User Interface, Layout, Media, Map* dll

9. *Drop Down*

Beberapa blok memiliki panah tarik-turun kecil di sebelah kanan nama blok. Anda dapat mengklik panah ini untuk mengubah nama dan fungsi blok. Blok *get* adalah contoh dropdown. Untuk bantuan lebih lanjut mengenai topik ini, lihat halaman dropdown.

10. *Emulator*

Nama ponsel palsu yang muncul di komputer Anda jika Anda tidak memiliki perangkat Android untuk digunakan adalah emulator.

11. *Event*

Event adalah sesuatu yang terjadi. Dalam konteks coding, peristiwa adalah sesuatu yang memicu suatu tindakan tertentu. Misalnya, ketika Anda mengklik sebuah tombol, itu adalah sebuah peristiwa. Saat Anda menggulir halaman web ke bawah, itu adalah peristiwa lain. Acara penting karena membantu membuat program Anda interaktif.

E. Memulai MIT App Inventor

MIT App Inventor adalah sebuah *tools* online untuk membangun proyek aplikasi seluler, artinya kita dapat membuat aplikasi untuk ponsel atau tablet langsung pada browser web kita. Situs web MIT App Inventor menawarkan semua dukungan yang kita perlukan untuk merancang aplikasi:

Petunjuk untuk Pengujian

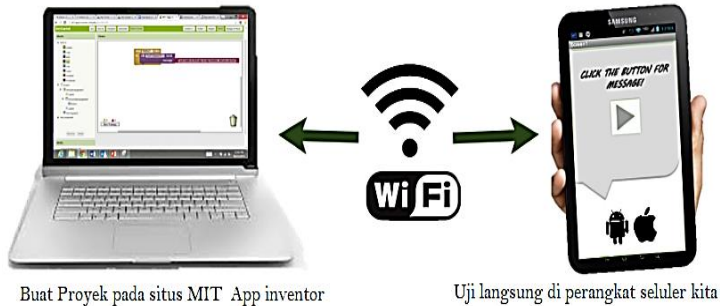
Siapkan ponsel atau tablet iOS atau Android Anda untuk pengujian langsung. (Atau, jalankan emulator sebagai opsi lainnya).

Ada empat opsi di untuk melakukan pengujian pada saat proyek kita selesai:

1. Opsi 1 (Rekomendasi):

Uji aplikasi kita dengan iPhone atau ponsel Android yang terkoneksi Wi-Fi.

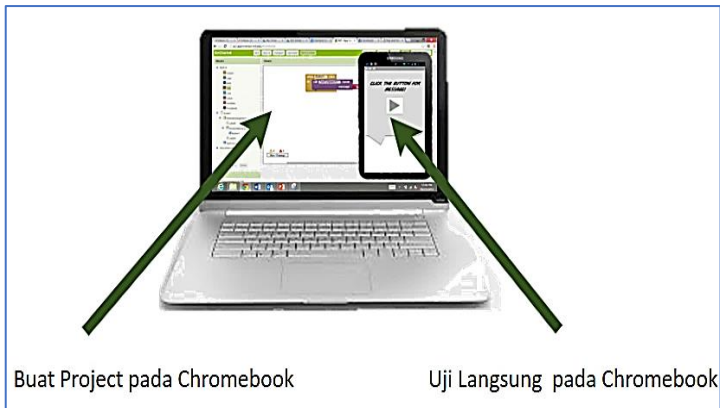
Menggunakan komputer/laptop, ponsel dan koneksi Wi-Fi, ini adalah cara termudah untuk membuat dan menguji aplikasi kita. Cukup instal aplikasi companion melalui playstore ataupun appstore di ponsel kita dan uji aplikasi melalui koneksi WiFi.



2. Opsi 2:

Uji aplikasi dengan Chromebook:

Banyak Chromebook yang mampu menjalankan aplikasi Android. Ini memungkinkan kita membuat, menguji, dan menjalankan aplikasi yang sudah selesai di perangkat yang sama.



3. Opsi 3:

Uji aplikasi dengan emulator:

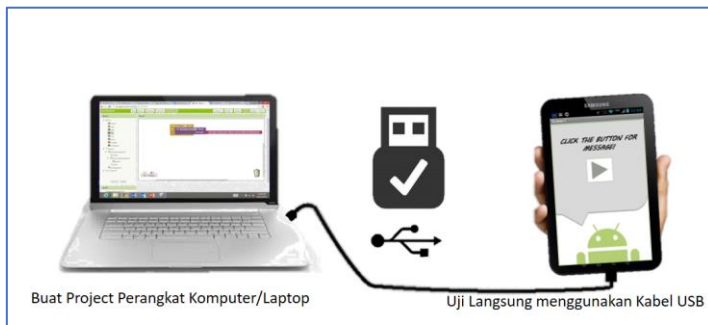
Jika tidak memiliki ponsel atau tablet, kita masih dapat menggunakan App Inventor dengan menginstal perangkat lunak emulator di perangkat komputer/laptop.



4. Opsi 4:

Uji aplikasi dengan perangkat Android dan Kabel USB: (Tanpa Wifi)

Beberapa firewall di sekolah, kampus atau suatu organisasi tidak mengizinkan jenis koneksi Wi-Fi yang diperlukan untuk App Inventor. Jika tanpa Wi- kita bisa menguji menggunakan USB (khusus Android).



F. Pembahasan Pada Buku ini

Pembelajaran memerlukan latihan jika ingin meningkatkan keterampilan sebagai profesi pengembang. Cara terbaik untuk berlatih adalah dengan membangun proyek proyek dasar.

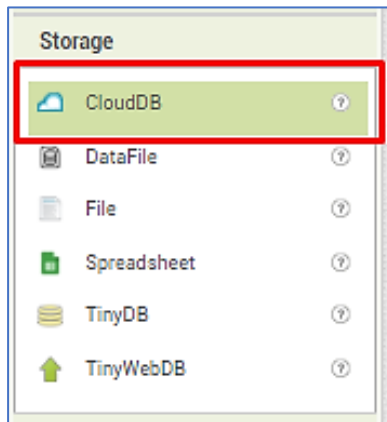
Buku ini membahas tentang bagaimana membuat proyek membangun aplikasi database dan file sederhana berbasis *mobile android* dengan komponen Clouddb, TinyDB, TinyWebDB, File

dan *Connetivity*, proyek dibahas tahap demi tahap, setiap bab berisi petunjuk tentang bagaimana membuat proyek baru, Analisa kebutuhan perancangan desain, *setting properties, block*, sampai menguji aplikasi langsung ke perangkat seluler baik HP ataupun tablet menggunakan companion sebagai *tool* untuk ujicobanya.

Buku ini juga dapat digunakan sebagai referensi belajar ataupun sebagai opsi bagi yang ingin menjadi pengembang aplikasi seluler android secara mudah berbasis block menggunakan MIT App Inventor 2.

BAB 2

BEKERJA DENGAN CLOUDDDB



A. Pendahuluan

Komponen CloudDB adalah komponen Tidak Terlihat (*Non- visible Component*) yang memungkinkan kita menyimpan data di server database yang terhubung ke Internet (menggunakan perangkat lunak Redis). Hal ini memungkinkan pengguna Aplikasi Kita untuk berbagi data satu sama lain. Secara default, data akan disimpan di server yang dikelola oleh MIT, namun Kita dapat mengatur dan menjalankan server Kita sendiri dengan mengatur properti RedisServer dan properti RedisPort untuk dapat mengaksesnya.

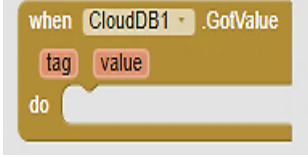

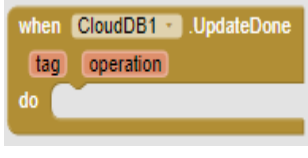
B. *Properties* CloudDB

Properties	Value	Keterangan
ProjectID	Text (read-only)	Memberikan Nama ID <i>CloudDB</i> untuk sebuah proyek
RedisPort	Number(Read-Only)	Port Server Redis yang akan digunakan. Defaultnya adalah 6381
RedisServer	Text (read-only)	Server Redis yang digunakan untuk penyimpanan data. Pengaturan "DEFAULT" berarti server MIT akan digunakan.
Token	Text (Designer-only)	Kolom ini berisi token autentikasi yang digunakan untuk masuk ke server Redis yang didukung. Untuk server "DEFAULT", jangan edit nilai ini, sistem akan mengisinya secara otomatis. Administrator sistem juga dapat memberikan nilai khusus kepada Anda yang dapat digunakan untuk berbagi data antara beberapa proyek dari banyak orang. Jika menggunakan server Redis yang diatur sendiri, atur kata sandi di konfigurasi server dan masukkan di kolom sini.
UseSSL	Boolean(<i>designer-only</i>)	Set nilai ke true untuk menggunakan SSL agar

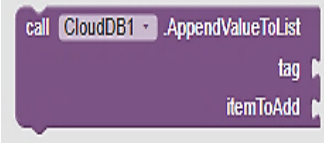
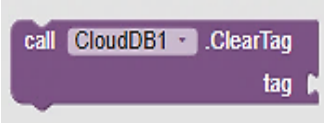

Properties	Value	Keterangan
		dapat terhubung dengan server. Nilai True di set jika untuk server "DEFAULT".

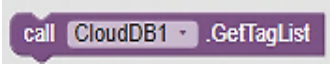
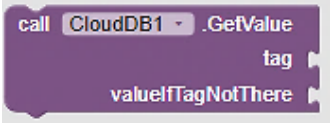
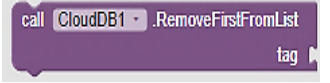
C. Events CloudDB


Events	Value	Keterangan
	(Text Message)	Event digunakan ketika mengindikasikan bahwa terjadi kesalahan saat berkomunikasi dengan server CloudDB Redis.
	(Text <i>tag, any value</i>)	Event digunakan untuk Menunjukkan bahwa data dalam proyek CloudDB telah berubah. dengan tag yang telah diperbarui dengan nilai data terkini.
	(<i>any value</i>)	Event digunakan yang dipicu oleh fungsi Remove FirstFromList. Nilai argumen adalah objek yang pertama dalam daftar, dan nilai datanya dihapus.

Events	Value	Keterangan
	(Text <i>tag, any value</i>)	Menunjukkan bahwa permintaan GetValue telah berhasil.
	(list, <i>value</i>)	Event dipicu ketika menerima daftar tag yang dikenal. Jalankan sebagai respons terhadap panggilan ke fungsi GetTagList.
	(Text <i>tag, text operation</i>)	Menunjukkan bahwa operasi yang menyimpan data ke CloudDB telah selesai.

D. Methods pada CloudDB

Method	Value	Keterangan
	(Text <i>tag, any ItemToAdd</i>)	Menambahkan nilai ke akhir daftar secara atomically. Jika dua perangkat menggunakan fungsi ini secara bersamaan, keduanya akan ditambahkan dan tidak ada data yang hilang.
	(Text <i>tag</i>)	Hapus tag dari CloudDB.
	(boolean)	Mengembalikan nilai /benar jika kita

Method	Value	Keterangan
		berada di jaringan dan kemungkinan besar dapat terhubung ke server CloudDB.
	(Text <i>tag</i> , any <i>value</i>)	Meminta CloudDB untuk mengambil semua tag didalam proyek. Daftar yang dihasilkan dikembalikan dalam <i>event</i> TagList.
	(text <i>tag</i> , any <i>valueIfTagNotThere</i>)	GetValue meminta CloudDB untuk menyimpan nilai di bawah tag yang diberikan. Ini akan meneruskan hasilnya ke GetValue yang akan diberikan.
	(Text <i>tag</i>)	Mendapatkan elemen pertama dari daftar dan hapus secara otomatis. Jika dua perangkat menggunakan fungsi ini secara bersamaan, yang satu akan mendapatkan elemen pertama dan yang lainnya akan mendapatkan elemen kedua, atau terjadi kesalahan jika tidak ada elemen yang tersedia. Ketika elemen tersedia, event FirstRemoved akan dipicu.

Method	Value	Keterangan
	(text tag, any valueToStore)	Meminta CloudDB untuk menyimpan nilai yang diberikan di bawah tag yang diberikan.

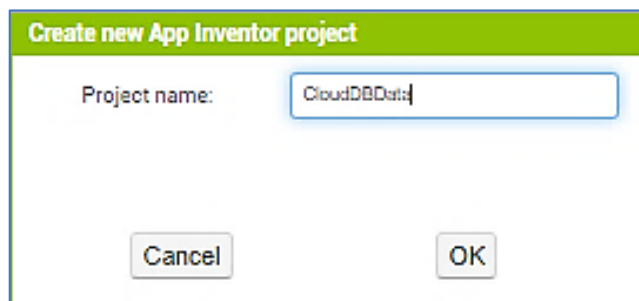
E. Contoh Proyek menggunakan Komponen Clouddb

1. Aplikasi Menyimpan data dan Menampilkan pada listview

Aplikasi ini menggunakan cloudDB untuk penyimpanan datanya, serta listview untuk menampilkan data yang sudah tersimpan.

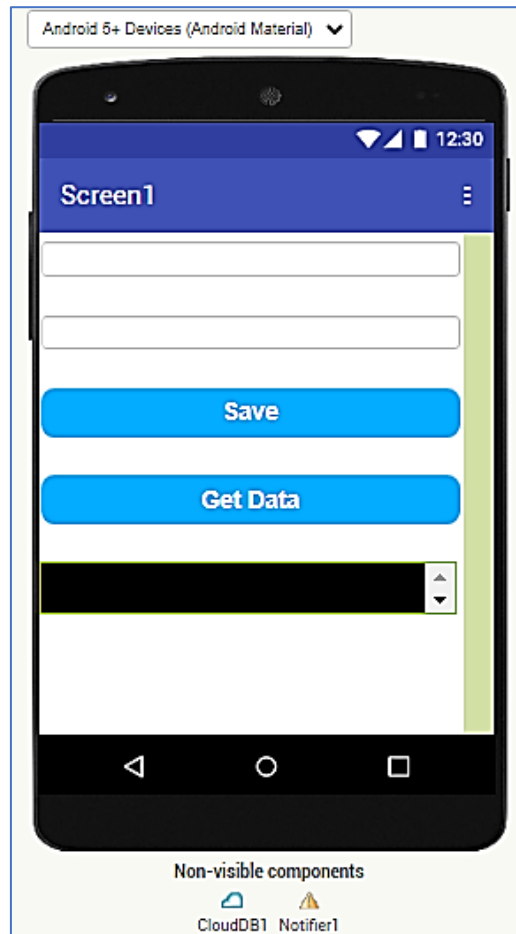
Langkah: 1 membuat Project baru

- Masuk ke web mit app inventor pada browser ketik: [https:// appinventor.mit.edu/](https://appinventor.mit.edu/)
- Setelah halaman app inventor muncul klik Create, selanjutnya masuk menggunakan akun google kita, atau klik new project jika halaman mit app sedang aktif.
- Namakan project seperti gambar dibawah ini dan selanjutnya Klik OK

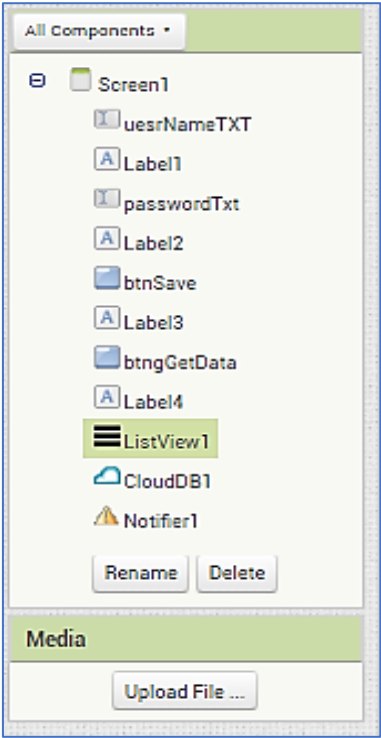


Langkah: 2 Perancangan Desain aplikasi CloudDBData

Perancangan desain aplikasi membutuhkan 2 komponen untuk input yaitu username dan password, dan 2 tombol yaitu *Save* untuk menyimpan ke cloudDB dan *GetData* untuk menampilkan data tersimpan ke 1 komponen listview.



Komponen keseluruhan pada jendela *Component* seperti gambar dibawah ini:



Langkah: 3 Set Properties semua komponen pada desain

No	Palllete	Component	Properties	Keterangan
1		Screen1	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Left: 1 AlignVertical: Top: 1 BackgroundColor: Default BackgroundImage: None BigDefaultText: None CloseScreenAnimation: Default HighContrast: None OpenScreenAnimation: Default ScreenOrientation: Unspecified Scrollable: None	Memberikan Nilai Properties untuk komponen Screen1

No	Pallette	Component	Properties	Keterangan
			ShowStatusBar: None Title: Screen1 TitleVisible: True	
2	User Interface	TextBox1 Name: UserNameTxt	userNameTXT (TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default... Height: Automatic... Width: Fill parent... Hint: UserName TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	Komponen Digunakan untuk input username
3	User Interface	Label Name: Label1	Label1 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Digunakan untuk Batas Spasi antara Komponen UsernameTxt dan PasswordTxt
4	User Interface	PasswordText Box1 Name: passwordTxt	passwordTxt (TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default	Komponen Digunakan untuk input Password

No	Pallette	Component	Properties	Keterangan
			Height: Automatic Width: Fill parent Hint: Password TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	
5	User Interface	Label2 Name: Label2	Label2 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Digunakan untuk Batas Spasi antara Komponen PasswordTxt dan Tombol <i>Save</i>
6	User Interface	Button1 Name: btnSave	btnSave (Button) Appearance BackgroundColor: Custom(blue) FontBold: True FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: rounded ShowFeedback: True Text: Save TextAlignment: center: 1 TextColor: White Visible: True	Digunakan untuk tombol simpan data ke clouddb setelah isian username dan password diberikan

No	Pallete	Component	Properties	Keterangan
			Behavior Enabled: True	
	User Interface	Label3 Name: Label3	Label3 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Digunakan untuk Batas Spasi antara Komponen tombol <i>save</i> dan Tombol <i>Get Data</i>
7	User Interface	Button2 Name: BtnGetData	btnGetData (Button) Appearance BackgroundColor: Custom(blue) FontBold: True FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: rounded ShowFeedback: True Text: Get Data TextAlignment: center: 1 TextColor: White Visible: True Behavior Enabled: True	Tombol Get Data Digunakan untuk menampilkan data yang tersimpan dari cloudDb ke komponen ListView
8	User Interface	Label4 Name: Label4	Label4 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None	Digunakan untuk Batas Spasi Komponen Tombol <i>Get Data</i> dan komponen List

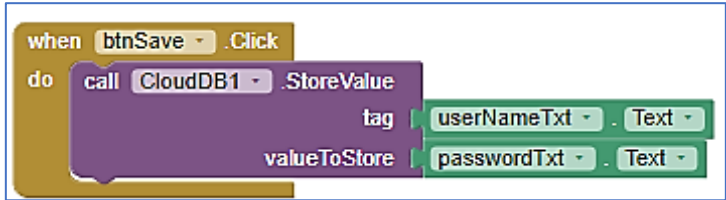
No	Pallette	Component	Properties	Keterangan
			HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	
9	User Interface	ListView1 Name: ListView1	ListView1 (ListView) Appearance BackgroundColor: Default FontSizeDetail: 14.0 FontTypeface: default FontTypefaceDetail: default Height: Automatic Width: Automatic ImageHeight: 200 ImageWidth: 200 ListViewLayout: MainText Orientation: vertical SelectionColor: Default TextColor: Default TextColorDetail: Default TextSize: 22 Visible: True Behavior ElementsFromString: None ListData: Click to Add/Delete Data Selection: None ShowFilterBar: None	List Digunakan untuk menampilkan daftar data (tag,value), setelah tombol <i>Get Data</i> di tekan/klik
10	User Interface	CloudDB1 Name: CloudDB1	CloudDB1 (CloudDB) Behavior ProjectID: CloudDBData RedisPort: Default 6381 RedisServer: DEFAULT Token: Default UseSSL: Default	CloudDB Digunakan untuk menampung/ menyimpan data Tag dan Value
11	User Interface	Notifier1 Name: Notifier1	Notifier1 (Notifier) Appearance BackgroundColor: Default NotifierLength: Long	Notifier digunakan untuk menampilkan

No	Palette	Component	Properties	Keterangan
			TextColor: Default	pesan password dan nilainya ketika salah satu item tag pada list di klik

Langkah: 4 Menyusun Block Perintah Aplikasi

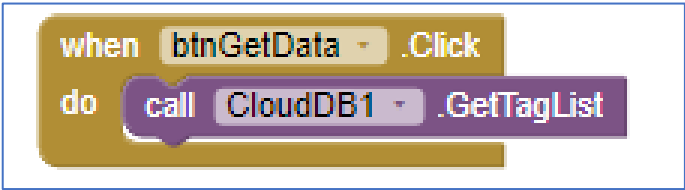
Menyusun Block untuk memberikan Perintah kode berupa *puzzle* dengan mengklik menu Block di sisi kanan pada halaman awal.

a. Event **Button.save click**



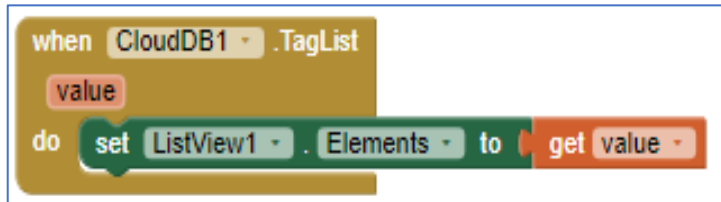
Penjelasan:
Perintah ini digunakan untuk menyimpan data dari kotak isian username dan password yang sudah diberikan dengan mengklik tombol *Save*

b. Event When **btnGetData.click**



Penjelasan:
Perintah ini digunakan untuk memanggil method *GetTagList* dengan mengklik tombol *Get data*

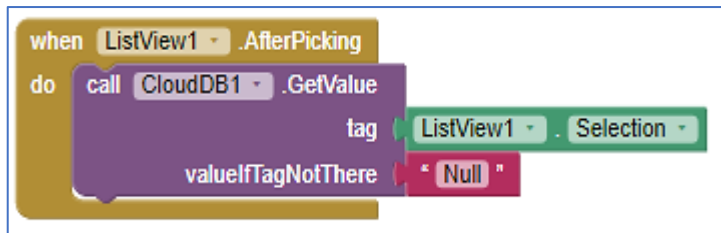
c. Event When **CloudDB1.Taglist**



Penjelasan:

Perintah ini digunakan untuk mengambil nilai data Tag dari CloudDB yang tersimpan dan ditampilkan pada ListView.

d. Event **ListView.After Picking**



Penjelasan:

Perintah ini digunakan untuk mengambil nilai data dengan meng klik daftar item yang diseleksi pada ListView.

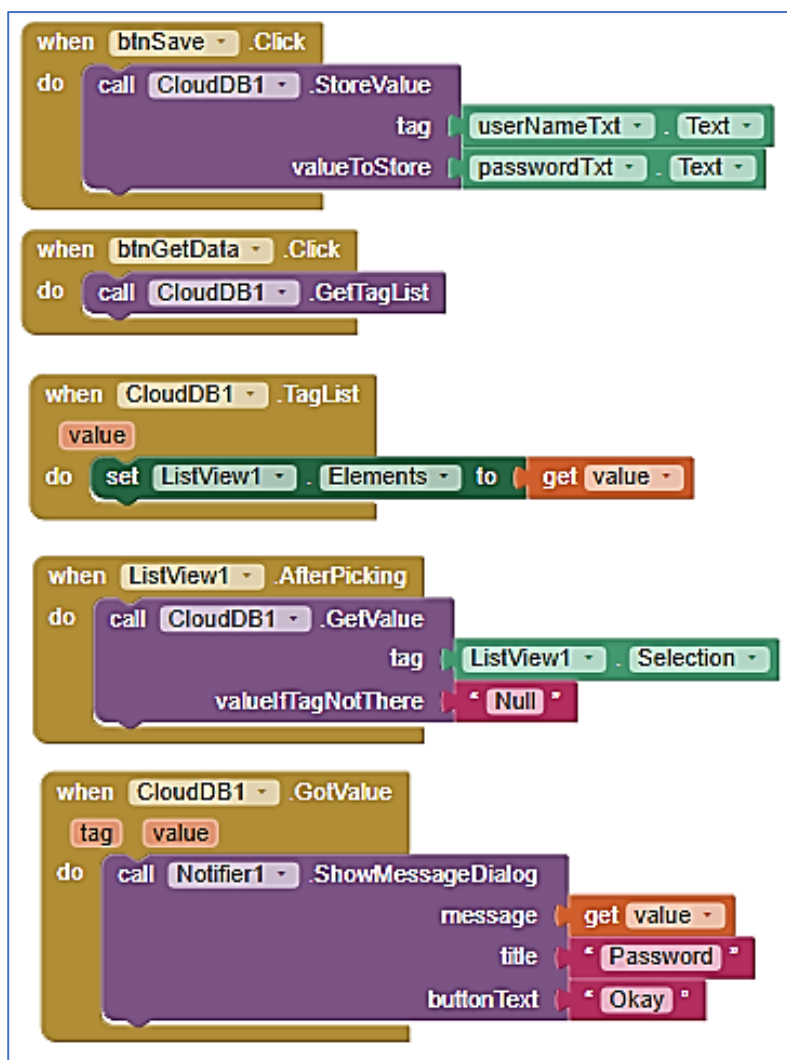
e. Event **CloudDB1.GotValue**



Penjelasan:

Perintah ini digunakan untuk mendapatkan Nilai Password yang didapat dari CloudDb1 dan ditampilkan pada kotak pesan.

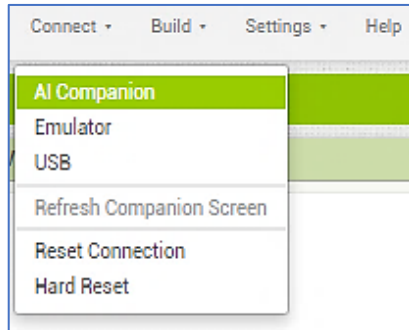
- f. Tampilan keseluruhan block aplikasi pada jendela editor Block seperti gambar dibawah ini:



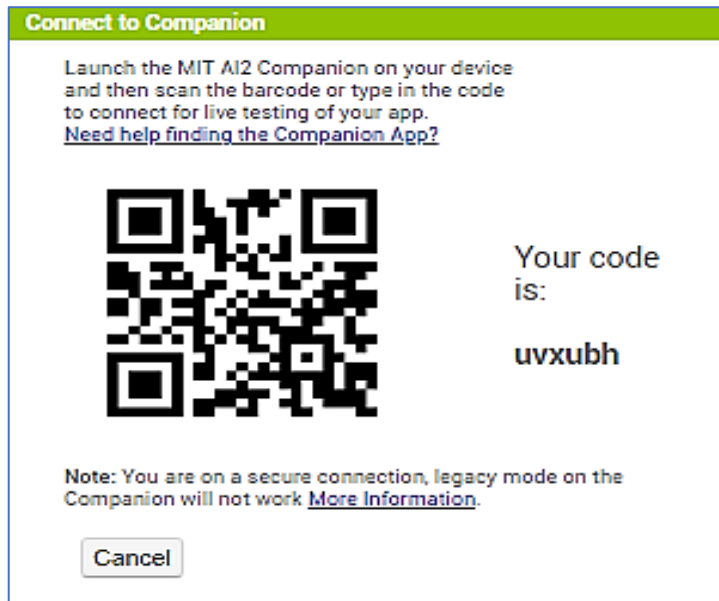
Langkah: 5 Menguji Aplikasi setelah Desain dan Block

Uji coba Aplikasi dengan menggunakan aplikasi companion, pastikan sudah ter install di perangkat gadget berbasis android sebagai target dan terhubung dengan internet. Selanjutnya pada Halaman MIT App inventor

a. klik menu Connect, AI Companion:

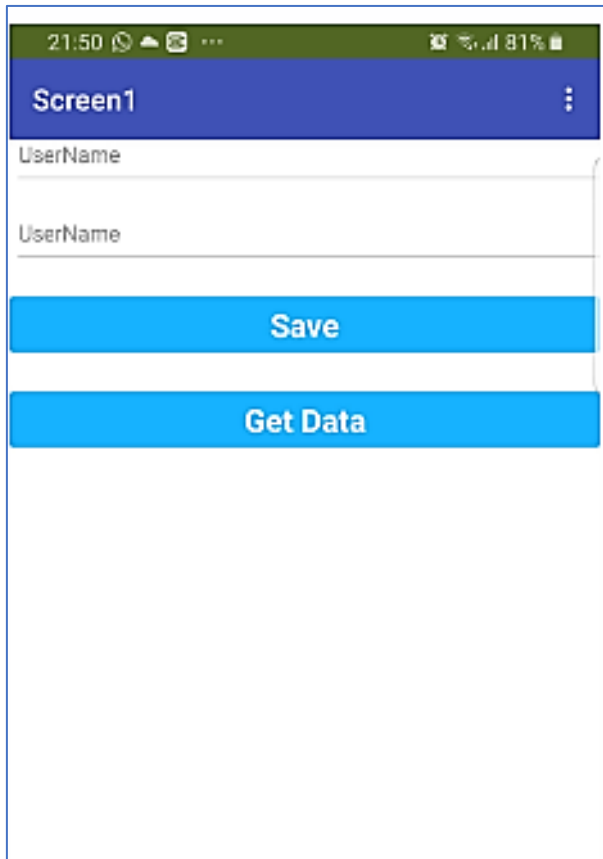


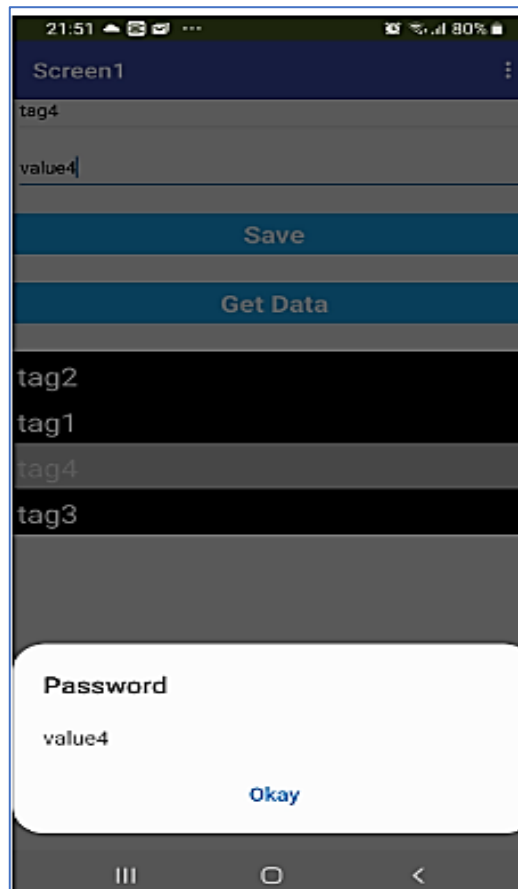
b. Scan barcode yang muncul setelah aplikasi companion terbuka pada perangkat gadget pengujian contoh barcode:



- c. Jika berjalan baik maka aplikasi akan ditampilkan pada perangkat gadget kita, lakukan pengujian dengan mengisi username, password, klik **Save**, selanjutnya klik tombol **Get Data** untuk ditampilkan pada listview.

Contoh hasil uji aplikasi pada perangkat target seperti gambar dibawah ini:



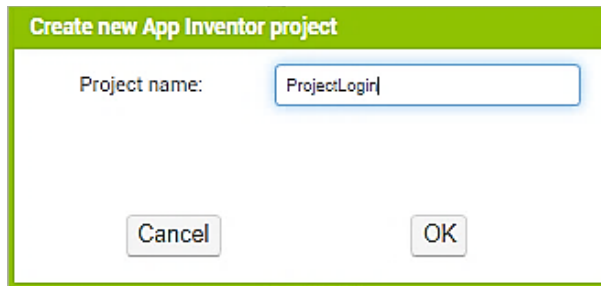


F. Contoh Proyek Aplikasi Login CloudDB

Aplikasi login ini untuk masuk ke halaman screen ke 2 atau *main screen*, menggunakan cloudDB untuk mencari dan menyimpan data username dan password, setelah mengisi kotak username dan password, jika datanya sudah ada maka akan masuk ke *screen* ke 2 atau *main screen*, Jika data username dan Password belum ada dan ingin disimpan pada CloudDb setelah pengisian klik tombol *Sign Up*.

Langkah: 1 membuat Project baru:

1. Masuk ke web mit app inventor pada browser ketik: <https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik Create, selanjutnya masuk menggunakan akun google kita, atau klik new project jika halaman mit app sedang aktif.
3. Namakan project seperti gambar dibawah ini



Create new App Inventor project

Project name:

Cancel OK

Langkah 2: Perancangan Desain

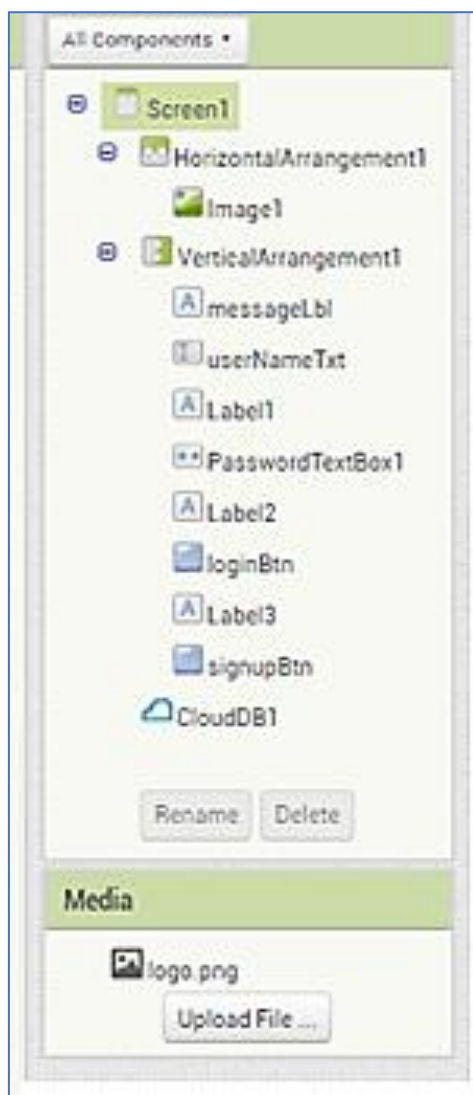
Terdapat 2 screen pada aplikasi, yakni Screen Login (Screen1) dan Main Screen:

1. Desain Screen Login

Screen login (Screen1) digunakan untuk masuk screen Main Screen aplikasi terdapat dua input yang akan divalidasi oleh aplikasi, yaitu username dan password, berikut ini tampilan desain nya



Dibawah ini adalah Komponen keseluruhan pada jendela *component*:



Langkah 3: Set Properties komponen keseluruhan

Setting properties semua komponen desain *screen Log in* mengacu pada tabel berikut ini:

No	Pallete	Component	Properties	Keterangan
1		Screen1	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Center: 3 AlignVertical: Top: 1 BackgroundColor: Pink BackgroundImage: None BigDefaultText: None CloseScreenAnimation: Default HighContrast: None OpenScreenAnimation: Default ScreenOrientation: Portrait Scrollable: None ShowStatusBar: True Title: Log In TitleVisible: True	Memberikan Nilai Properties untuk komponen Screen1
2	Layout	HorizontalArrangement1	HorizontalArrangement2 (HorizontalArrangement) Appearance AlignHorizontal: center: 3 AlignVertical: center: 2 Height: 30% Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun komponen dari kiri kekanan didalamnya terdapat Komponen Image
3	Layout	VerticalArrangement1	VerticalArrangement1 (VerticalArrangement) Appearance AlignHorizontal: Center: 3 AlignVertical: Top: 1 BackgroundColor: None Height: 67 percent Width: Fill parent Image: None	Digunakan untuk menyusun sekelompok komponen dari atas kebawah didalamnya terdapat MessageLbl

No	Pallete	Component	Properties	Keterangan
			Visible: True	textbox untuk input user, PasswordTe xtbox untuk input password, Label2 untuk spasi LoginBtn untuk Tombol Login, Label3 untuk spasi, SignUpBtn
4	User Interface	Label1 Name: MsgLabel	messageLbl (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 20 FontTypeface: default HTMLFormat: None HasMargins: True Height: 10 percent Width: Fill parent Text: None TextAlignment: center: 1 TextColor: Custom (#bf515bff) Visible: True	Digunakan untuk menampilkan pesan
5	User Interface	TextBox1 Name: userNameTxt	userNameTxt (TextBox) Appearance BackgroundColor: Custom FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Hint: UserName TextAlignment: left: 0	Digunakan untuk input username pada kotak isian

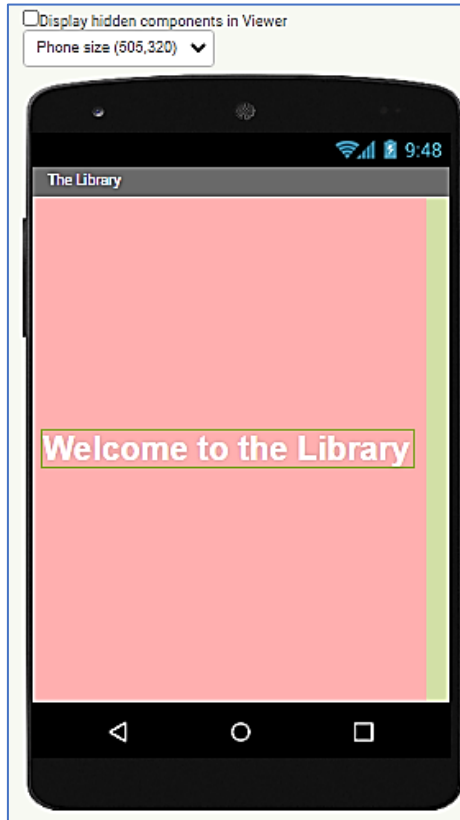
No	Pallette	Component	Properties	Keterangan
			TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	
6	User Interface	Label Name: Label1	Label1 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Digunakan untuk Batas Spasi Komponen
7	User Interface	PasswordTextBox Name: PasswordTxt	PasswordTxt (PasswordTextBox) Appearance BackgroundColor: Custom FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Hint: Password TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True NumbersOnly: None Text: None	Digunakan untuk input Password pada kotak isian
8	User Interface	Label Name:Label	Label2 (Label) Appearance	Digunakan untuk Batas

No	Pallete	Component	Properties	Keterangan
		2	BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Spasi Komponen
9	User Interface	Button1 Name: loginBtn	loginBtn (Button) Appearance BackgroundColor: Custom(#bf515bff) FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: 50 percent Image: None Shape: rounded ShowFeedback: True Text: LOG IN TextAlignment: center: 1 TextColor: Custom Visible: True Behavior Enabled: True	Digunakan untuk Menekan Tombol Login ketika isian username dan password diberikan
10	User Interface	Label Name:Label 3	Label3 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None	Digunakan untuk Batas Spasi Komponen

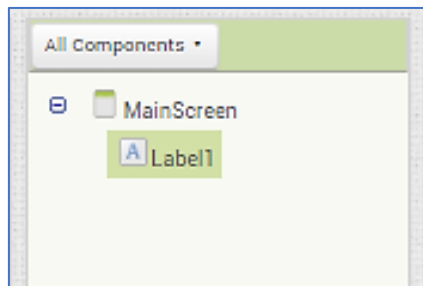
No	Pallete	Component	Properties	Keterangan
			TextAlignment: left: 0 TextColor: Default Visible: True	
11	User Interface	Button2 Name: signupBtn	signupBtn (Button) Appearance BackgroundColor: Custom(#bf515bff) FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: 50 percent Image: None Shape: rounded ShowFeedback: True Text: SIGN UP TextAlignment: center: 1 TextColor: Custom Visible: True Behavior Enabled: True	Digunakan untuk Menekan Tombol SignUp ketika isian username dan password baru diberikan
12	Storage	CloudDB Name: CloudDB1 (Non Visible component)	CloudDB1 (CloudDB) Behavior ProjectID: Notes RedisPort: Default 6381 RedisServer: DEFAULT Token: Default	Digunakan untuk penyimpanan data, pemanggilan data username dan password,
13		Media	Logo.png	Untuk gambar logo pada screen login (opsional)

2. Desain Screen Main Screen

Screen ini adalah screen yang dibuka ketika username dan password ditemukan pada penyimpanan CloudDB berikut tampilan Main Screen seperti gambar dibawah ini:



Komponen keseluruhan yang dibutuhkan pada jendela *component*:



Setting properties semua komponen pada desain
Mainscreen:

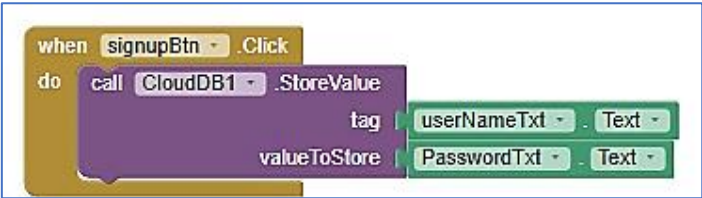
No	Pallete	Component	Properties	Keterangan
1		Main Screen	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Center: 3 AlignVertical: Top: 1 BackgroundColor: Pink BackgroundImage: None BigDefaultText: None CloseScreenAnimation: Default HighContrast: None OpenScreenAnimation: Default ScreenOrientation: Portrait Scrollable: None ShowStatusBar: True Title: Log In TitleVisible: True	Nilai Properties untuk komponen Main screen
2	User Interface	Label Name: Label1	Label1 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 30 FontTypeface: default HTMLFormat: None HasMargins: None Height: Automatic Width: Automatic Text: Welcome to the Library	Digunakan untuk memberikan judul pada Main Screen

No	Palette	Component	Properties	Keterangan
			TextAlignment: left: 0 TextColor: White Visible: True	

Langkah 3: Perintah Block Screen Login

Perintah Block untuk menyusun kode berupa *puzzle* pada screen login

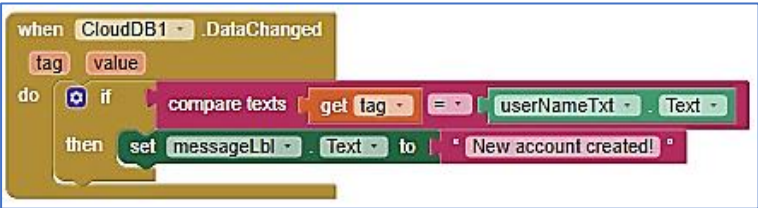
1. Event Klik Tombol SignUp



Penjelasan:

Ketika Tombol Sign Up di klik, panggil method FirebaseDatabase1.StoreValue untuk menyimpan data ke FirebaseDatabase mengambil nilai data dari kotak isian username dan Password.

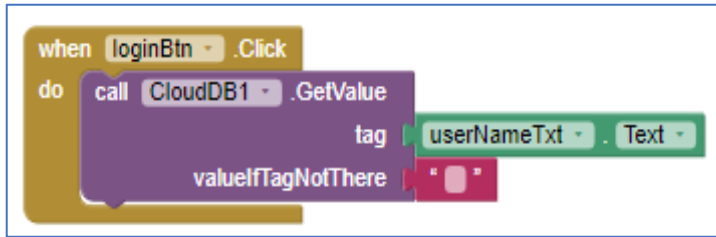
2. Event FirebaseDatabase.Datachanged



Penjelasan:

Ketika ada perubahan data, jika nilai tag sama dengan data pada kotak isian username, simpan tag mengambil dari username dan nilai mengambil dari kotak isian password.

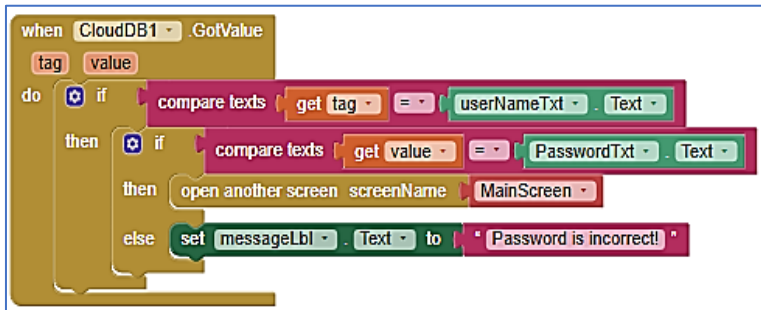
3. Event Klik Tombol



Penjelasan:

Ketika tombol Login di klik, ambil nilai tag dari usernameTxt, jika nilai tag tidak kosong

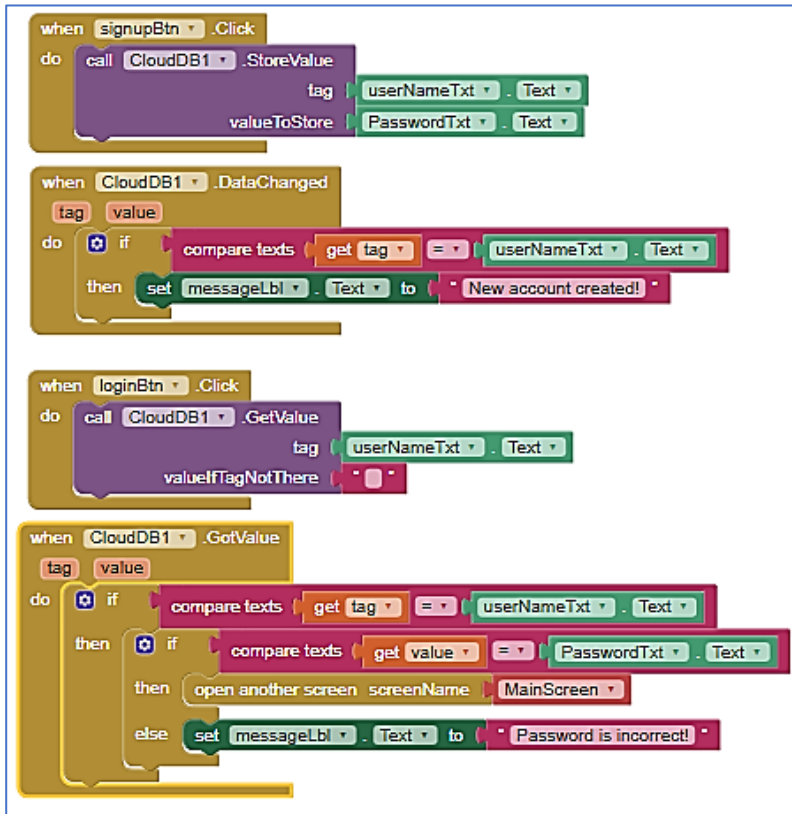
4. Event CloudDB.GotValue



Penjelasan:

Ketika data pada username dan password didapatkan, bandingkan dengan usernameTxt dan Password dengan PasswordTxt, jika sesuai maka buka screen MainScreen, jika tidak sesuai maka tampilkan pesan pada messageLbl *"Password Is Incorrect"*

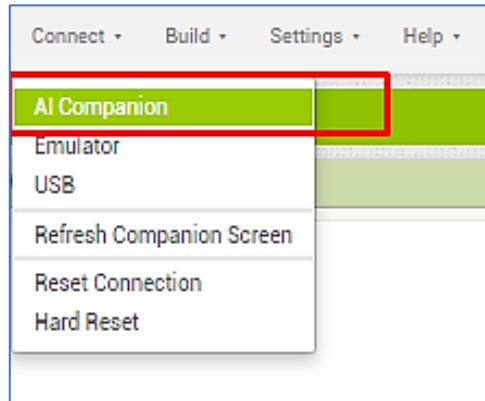
5. Berikut tampilan keseluruhan pada jendela block dibawah ini:



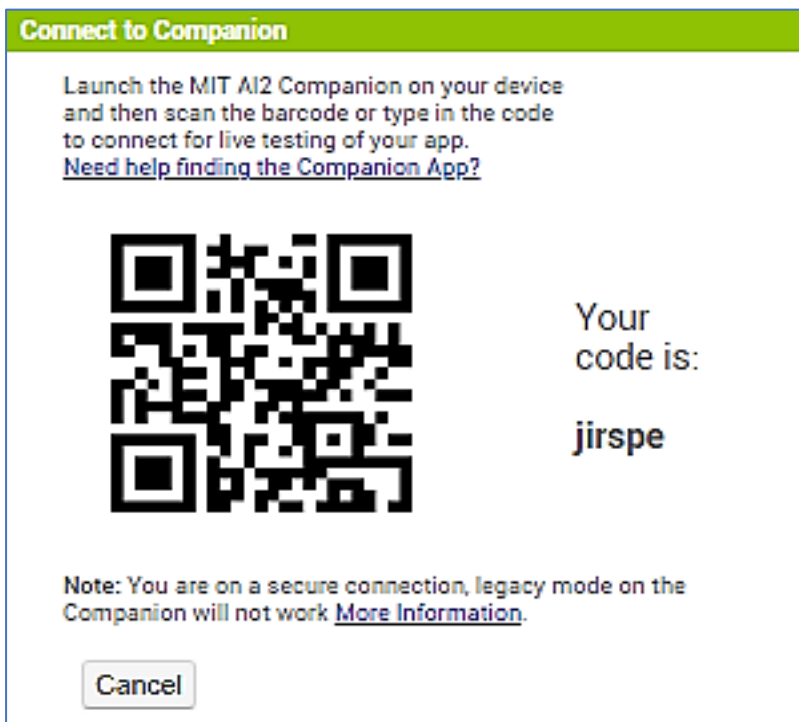
Langkah 4: Uji coba Aplikasi dengan Companion

Uji coba menggunakan aplikasi companion, pastikan sudah ter install di perangkat gadget berbasis android dan terhubung dengan internet. Selanjutnya pada Halaman MIT App inventor

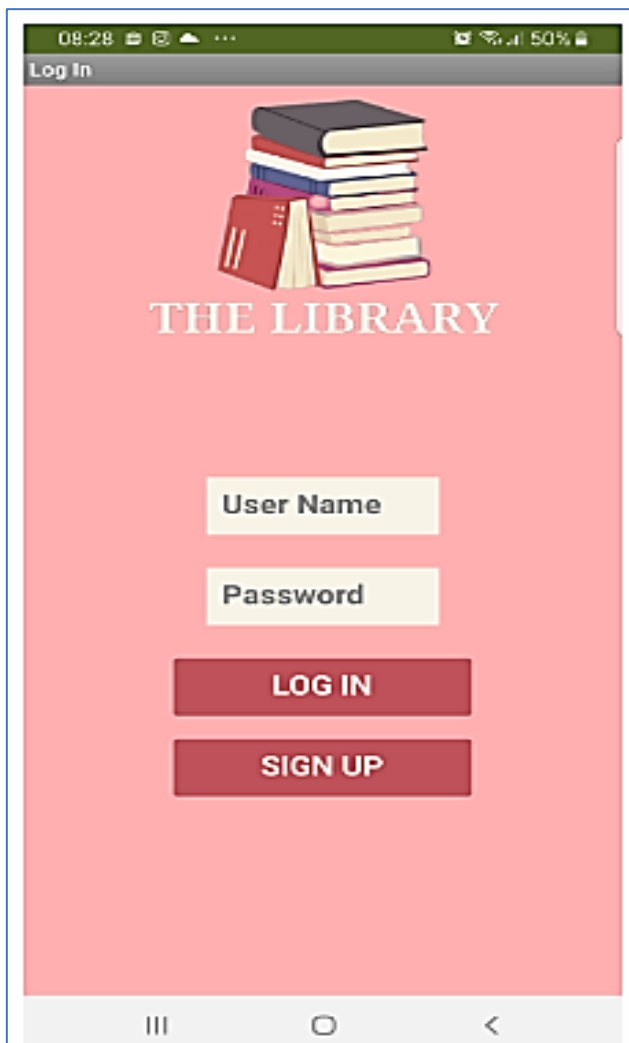
1. Klik menu *Connect, AI Companion*:



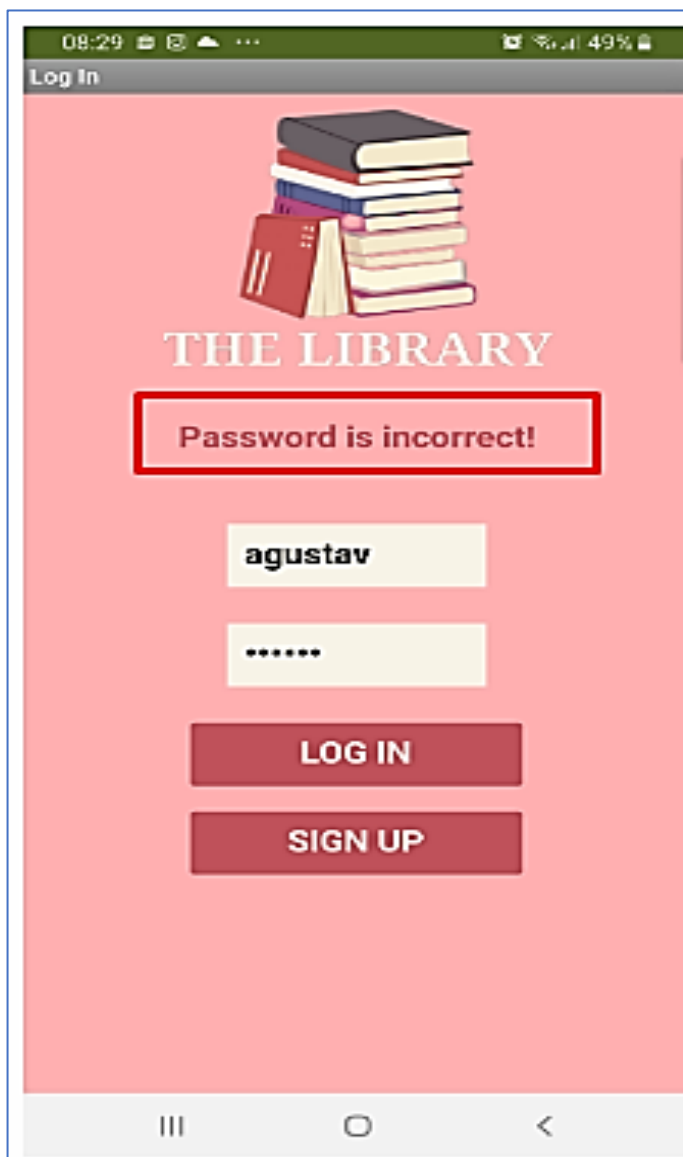
2. *Scan* barcode yang muncul setelah aplikasi companion terbuka pada perangkat gadget pengujian contoh barcode:



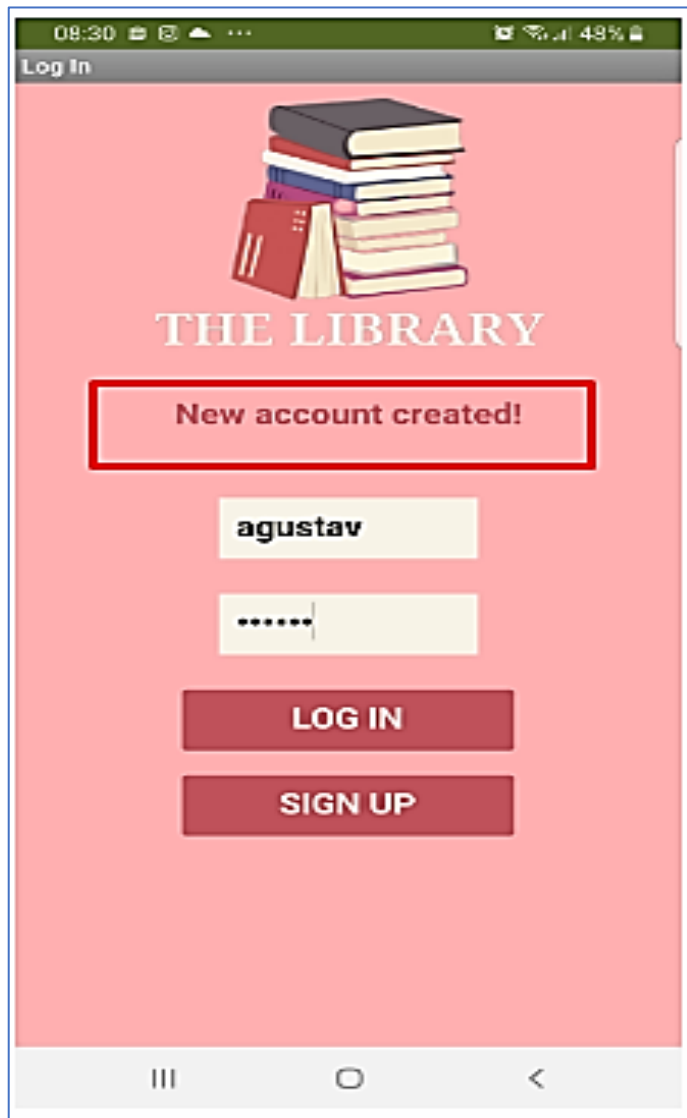
3. Jika berjalan baik maka aplikasi akan ditampilkan pada perangkat gadget, lakukan pengujian dengan mengisi username, password, klik *login* jika data sudah tersimpan dan dibuat, klik tombol *signup* jika data username dan password merupakan data baru yang belum tersimpan di CloudDB.
4. Contoh hasil uji yang ditampilkan pada perangkat gadget.
 - a. Tampilan awal screen aplikasi



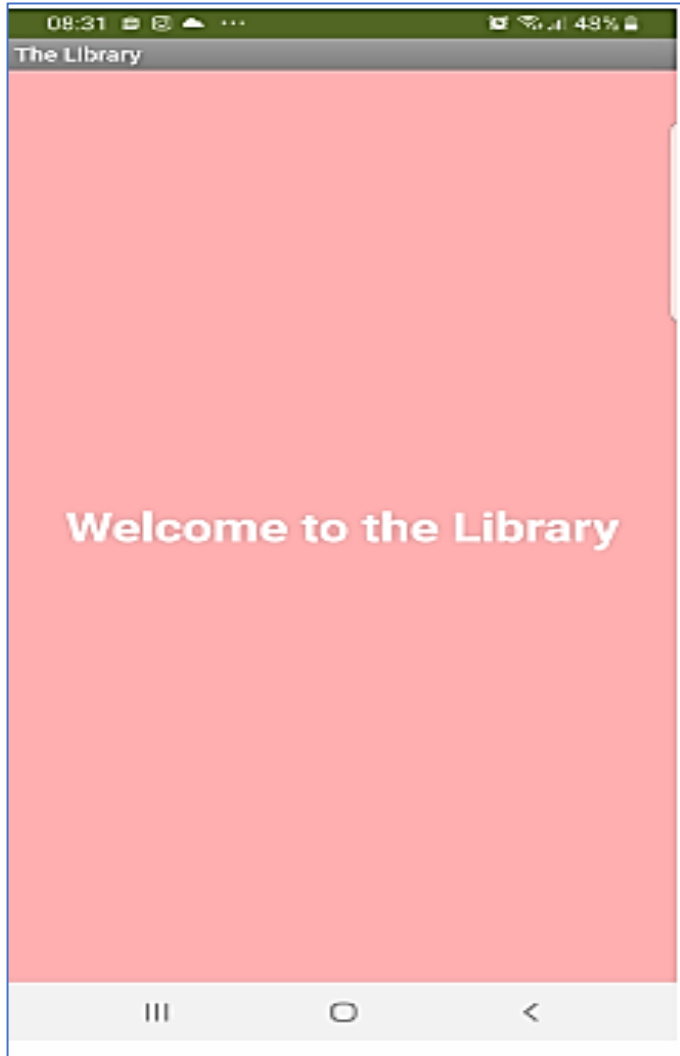
- b. Tampilan screen jika username, password tidak ada pada CloudDB setelah klik tombol **login**



- c. Tampilan screen jika username, password adalah data baru setelah klik tombol *Sign Up*

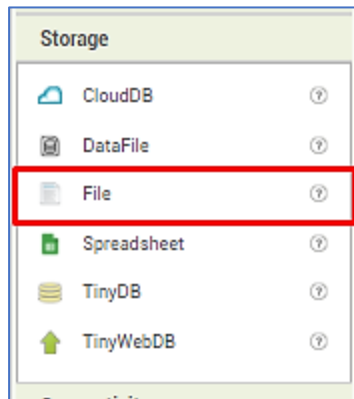


- d. Tampilan screen kedua / Main screen jika username, password ada pada cloudDB setelah klik tombol **Login**



BAB 3

BEKERJA DENGAN *FILE STORAGE*



A. Pendahuluan

Seperti Komponen CloudDB, komponen *File* adalah komponen tidak terlihat (*Non-Visible*) untuk menyimpan dan mengambil file. Komponen ini juga digunakan untuk menulis atau membaca file pada perangkat.

Lokasi pasti penempatan file eksternal merupakan fungsi dari nilai properti Scope, apakah aplikasi berjalan di Companion atau dikompilasi, dan di versi Android mana aplikasi tersebut dijalankan.

Karena versi Android yang lebih baru mengharuskan file disimpan di direktori khusus aplikasi, *DefaultScope* disetel ke App. Jika kita menggunakan Android versi lama dan memerlukan akses ke penyimpanan publik lama, ubah properti

DefaultScope menjadi *Legacy*. Kita juga dapat mengubah Cakupan menggunakan blok.

Di bawah ini adalah menjelaskan secara singkat setiap jenis cakupan:

1. **App:** File akan dibaca dan ditulis ke penyimpanan khusus aplikasi di Android 2.2 dan lebih tinggi. Pada versi Android sebelumnya, file akan ditulis ke penyimpanan lama.
2. **Asset:** File akan dibaca dari aset aplikasi. Merupakan kesalahan saat mencoba menulis ke aset aplikasi karena aset tersebut terdapat dalam penyimpanan hanya-baca saja.
3. **Cache:** File akan dibaca dan ditulis ke direktori cache aplikasi. Cache berguna untuk file-file sementara yang dapat dibuat ulang karena memungkinkan pengguna untuk menghapus file-file sementara untuk mendapatkan kembali ruang penyimpanan.
4. **Legacy:** File akan dibaca dan ditulis ke sistem file menggunakan aturan App Inventor sebelum rilis nb187. Artinya, nama file yang dimulai dengan satu / akan dibaca dan ditulis ke akar direktori penyimpanan eksternal, misalnya /sdcard/. Fungsionalitas lama tidak akan berfungsi di Android 11 atau lebih baru.
5. **Private:** File akan dibaca dan ditulis ke direktori pribadi aplikasi. Gunakan cakupan ini untuk menyimpan informasi yang tidak boleh terlihat oleh aplikasi lain, seperti aplikasi manajemen file.
6. **Shared:** File akan dibaca dan ditulis ke direktori media bersama perangkat, seperti file Gambar.


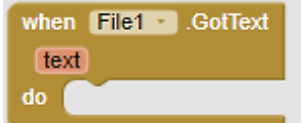
Catatan: Dalam mode *Legacy*, nama file dapat menggunakan salah satu dari tiga bentuk berikut:

1. File *private* tidak memiliki awalan / dan ditulis ke penyimpanan private aplikasi (misalnya, "file.txt")
2. File *external* memiliki satu awalan / dan ditulis ke penyimpanan publik (misalnya, "/file.txt")
3. *Bundled App* memiliki dua // di depan dan hanya dapat dibaca (misalnya, "//file.txt")

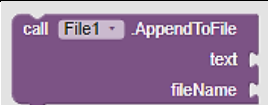
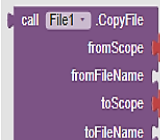
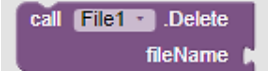
B. Properties Pada Komponen File


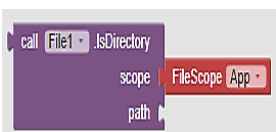
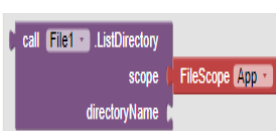
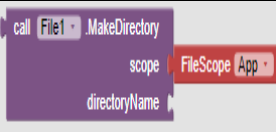
Properties	Value	Keterangan
DefaultScope	Filescope(write only, designer-only)	Menentukan cakupan default untuk file yang diakses menggunakan komponen File. Cakupan Aplikasi seharusnya berfungsi untuk sebagian besar aplikasi. Mode lawas dapat digunakan untuk aplikasi yang sudah ada sebelum batasan baru yang diakses Android pada akses file aplikasi.
ReadPermission	Boolean(write only, designer-only)	Properti khusus desainer yang dapat digunakan untuk mengaktifkan akses baca ke penyimpanan file di luar direktori khusus pada aplikasi.
Scope	filescope(block only)	Menunjukkan cakupan operasi saat ini seperti ReadFrom dan SaveFile.
WritePermission	Boolean(write only-designer only)	Properti khusus desainer yang dapat digunakan untuk mengaktifkan akses tulis ke penyimpanan file di luar direktori khusus aplikasi.

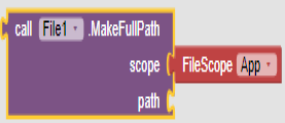
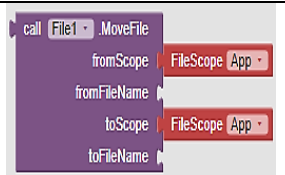
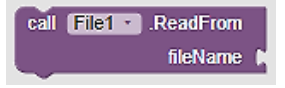
C. Event pada File

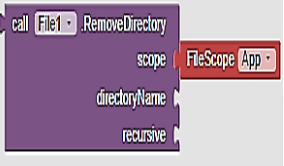
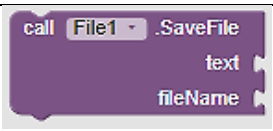
Events	Value	Keterangan
	(Text Filename)	Event yang menunjukkan bahwa isi file telah ditulis.
		Event yang menunjukkan bahwa konten dari file telah dibaca.

D. Method pada File

Method	Value	Keterangan
	(text,fileName))	Menambahkan teks ke akhir file. Membuat file jika belum ada. Jika berhasil, event AfterFileSaved akan dijalankan.
	(fromScope, fromFileName, toScope, toFileName)	Salin konten dari file pertama ke file kedua.
	(text filename)	Menghapus file dari penyimpanan. Awali Nama file dengan / untuk menghapus file tertentu di kartu SD (misalnya, /myFile.txt akan menghapus file /sdcard/myFile.txt) . Jika Nama file

Method	Value	Keterangan
		tidak dimulai dengan /, maka file yang terletak di penyimpanan pribadi program akan dihapus. Memulai Nama file dengan // merupakan kesalahan karena file aset tidak dapat dihapus.
	(file scope, text path)	Menguji apakah jalur path ada dalam cakupan tertentu.
	(file scope, text path)	Menguji apakah jalur yang disebutkan dalam cakupan tertentu adalah direktori/folder.
	(file scope, text directoryName)	Mendapatkan daftar file dan direktori/folder pada direktori tertentu.
	(file scope, text directoryName)	Membuat folder baru untuk menyimpan file. Semantik dari metode ini sedemikian rupa sehingga akan mengembalikan

Method	Value	Keterangan
		nilai true jika folder ada pada saat penyelesaiannya. Ini bisa berarti bahwa folder tersebut sudah ada sebelum dilakukan pemanggilan.
	(file scope, text path)	Mengkonversi cakupan dan jalur path menjadi string tunggal untuk komponen lainnya.
	(fromScope, fromFileName, toScope, toFileName)	Memindahkan file dari satu lokasi ke lokasi lain.
		Membaca teks dari file di penyimpanan. Awali Nama file dengan / untuk membaca dari file tertentu di SD Card (misalnya, /myFile.txt akan membaca file /sdcard/myFile.txt. Untuk membaca aset yang dikemas dengan aplikasi (juga berfungsi untuk Pendamping) mulai Nama file

Method	Value	Keterangan
		dengan // (dua garis miring). Jika Nama file tidak dimulai dengan garis miring, maka akan dibaca dari penyimpanan pribadi aplikasi (untuk aplikasi terpaket) dan dari /sdcard/AppInventor/data untuk Pendamping.
	(file scope, directory Name,recursive)	Menghapus direktori / folder dari sistem file. Jika rekursif benar, maka semuanya akan dihapus. Jika rekursif salah, hanya direktori yang dihapus dan hanya jika direktori tersebut kosong.
	(text, text filename)	Menyimpan teks ke file. Jika Nama file dimulai dengan garis miring (/), file tersebut ditulis ke sdcard (misalnya, menulis ke /myFile.txt akan menulis file ke /sdcard/myFile.txt. Jika Nama file tidak

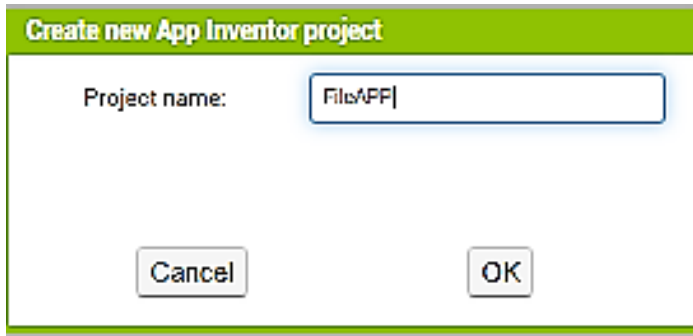
Method	Value	Keterangan
		dimulai dengan garis miring, maka akan ditulis dalam direktori data pribadi program yang tidak dapat diakses oleh program lain di telepon. Ada pengecualian khusus untuk AI Companion di mana file-file ini ditulis ke /sdcard/AppInventor/data untuk memfasilitasi proses debug. Perhatikan bahwa blok ini akan menimpa file jika sudah ada. Jika Anda ingin menambahkan konten ke file yang sudah ada gunakan metode AppendToFile.

E. Contoh Proyek Aplikasi menggunakan File text

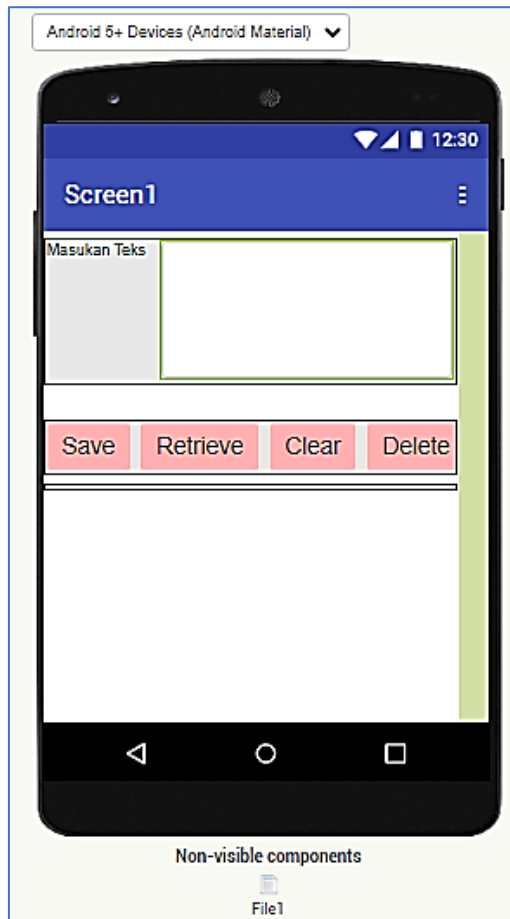
Contoh aplikasi ini akan menunjukkan bagaimana cara menulis teks dan menyimpannya ke file txt ke perangkat gadget berbasis android. Aplikasi ini juga akan menunjukkan bagaimana cara mengambil data dari file teks dan menampilkannya pada perangkat gadget kita.

Langkah: 1 membuat Project baru:

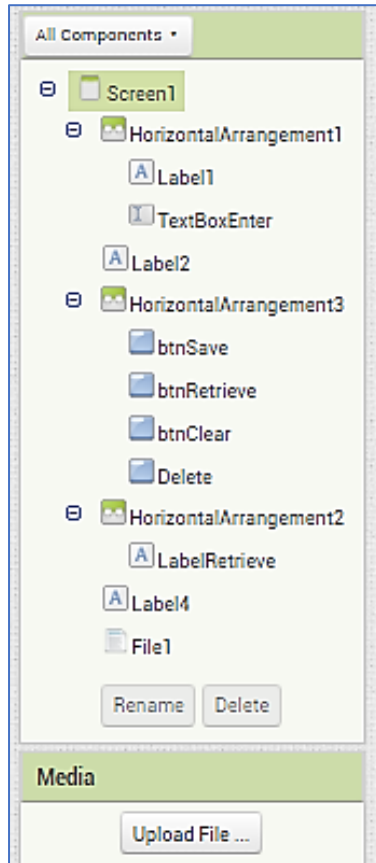
1. Masuk ke web mit app inventor pada browser ketik: <https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik Create, selanjutnya masuk menggunakan akun google kita, atau klik new project jika halaman mit app sedang aktif.
3. Namakan project seperti gambar dibawah ini

**Langkah 2: Perancangan Desain**

Desain Screen terdapat satu input text yang akan disimpan pada File teks, 4 tombol *save*, *retrieve*, *Clear*, *delete*, satu komponen file untuk penyimpanan dan pemanggilan file teks berikut ini tampilan desain nya:



Berikut ini komponen keseluruhan pada jendela *component*:



Langkah 3: *Setting properties* semua komponennya

No	Pallette	Component	Properties	Keterangan
1		Screen1	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Left: 1 AlignVertical: Top: 1 BackgroundColor: Default BackgroundImag: None BigDefaultText: None CloseScreenAnimati	Memberikan Nilai Properties untuk komponen Screen1

No	Pallette	Component	Properties	Keterangan
			on: Default HighContrast: None OpenScreenAnimati on: Default ScreenOrientatio: Portrait Scrollable: None ShowStatusBar: True Title: Screen1 TitleVisible: True	
2	Layout	HorizontalArrangement1	HorizontalArrangement1 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/ mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Label dan Textbox
3	User Interface	Label1 Name:Label1	Label1 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: Automatic Width: Automatic Text: Masukan Teks TextAlignment: left: 0 TextColor: Default Visible: True	Komponen Label1 digunakan untuk memberikan keterangan disebelah kotak textbox input

No	Pallete	Component	Properties	Keterangan
4	User Interface	TextBox1 Name: TextBoxEnter	TextBoxEnter (TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: 105 pixels Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: True NumbersOnly: None ReadOnly: None Text: None	Komponen TextBoxEnter digunakan untuk input data yang akan di simpan pada file teks
5	User Interface	Label2 Name: Label2	Label2 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai pembatas spasi
6	Layout	HorizontalArrangement3	HorizontalArrange ment3	Digunakan untuk menyusun/ menge

No	Pallete	Component	Properties	Keterangan
		Name: HorizontalArrangement3	(HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: automatic Width: Fill parent Image: None... Visible: True	lompokan komponen dari kiri kekanan didalamnya terdapat Komponen Label dan Textbox
7	User Interface	Button1 Name: btnSave	btnSave (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Image: None Shape: default ShowFeedback: Text: Save TextAlignment: center: 1 TextColor: Default Visible: Behavior Enabled: True	Komponen Button ini digunakan untuk menyimpan teks yang sudah di input pada kotak TextBox Enter
8	User Interface	Button2 Name: btnRetrieve	btnRetrieve(Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Image: None	Komponen button ini digunakan untuk menampilkan semua teks yang sudah di simpan di file teks

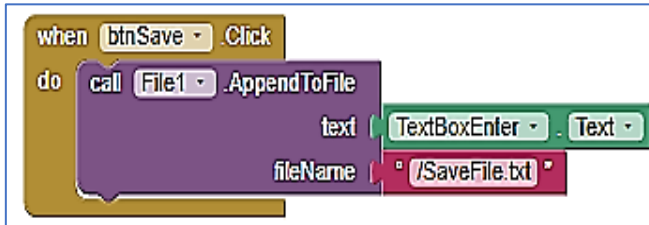
No	Pallette	Component	Properties	Keterangan
			Shape: default ShowFeedback: Text: Retrive TextAlignment: center : 1 TextColor: Default Visible: Behavior Enabled: True	
9	User Interface	Button3 Name: btnClear	btnclear (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Image: None Shape: default ShowFeedback: Text: Retrive TextAlignment: center : 1 TextColor: Default Visible: Behavior Enabled: True	Komponen button ini digunakan untuk menghapus / membersihkan teks yang ada kotak TextBoxEnter
10	User Interface	Button3 Name: btnDelete	btnDelete(Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 20 FontTypeface: default Height: Automatic Width: Automatic Image: None Shape: default ShowFeedback: Text: Retrive TextAlignment:	Komponen button ini digunakan untuk menghapus / File Teks yang tersimpan

No	Palette	Component	Properties	Keterangan
			center: 1 TextColor: Default Visible: Behavior Enabled: True	
11	Layout	HorizontalArrangement2 Name: HorizontalArrangement2	HorizontalArrangement2 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: automatic Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/ mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Label untuk menampilkan isi dari file teks pada saat di klik tombol retrieve
12	User Interface	Label3 Name: LabelRetrive	LabelRetrieve (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: True Height: automatic Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai wadah teks yang tersimpan pada file teks

Langkah 4: Perintah Block Screen Aplikasi

Perintah Block untuk menyusun kode berupa *puzzle* pada screen Aplikasi

1. Event when **btnSave.click**



Penjelasan Block:

Perintah Event ini digunakan untuk menyimpan teks setelah di input ke sebuah file bernama SaveFile.txt dengan memanggil *method* call file1.appendToFile

2. Event When **btnRetrieve.click**



Penjelasan Block:

Perintah Event ini digunakan untuk memanggil dan membaca teks yang tersimpan dari file SaveFile.txt dan ditampilkan ke sebuah komponen labelRetrieve.

3. Event When File1.GotText



Penjelasan *Block*:

Perintah Event ini digunakan untuk menampilkan teks yang tersimpan dari file SaveFile.txt ketika data dipanggil dan ditemukan.

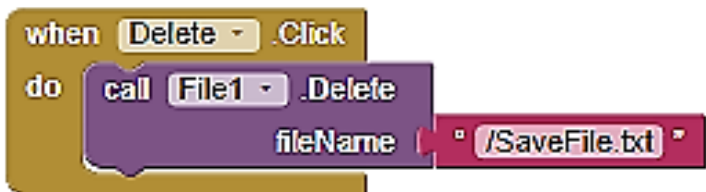
4. Event When btnClear.Click



Penjelasan *Block*:

Perintah Event ini digunakan untuk membersihkan teks dari komponen TextBoxEnter, dan siap untuk di inputkan kembali.

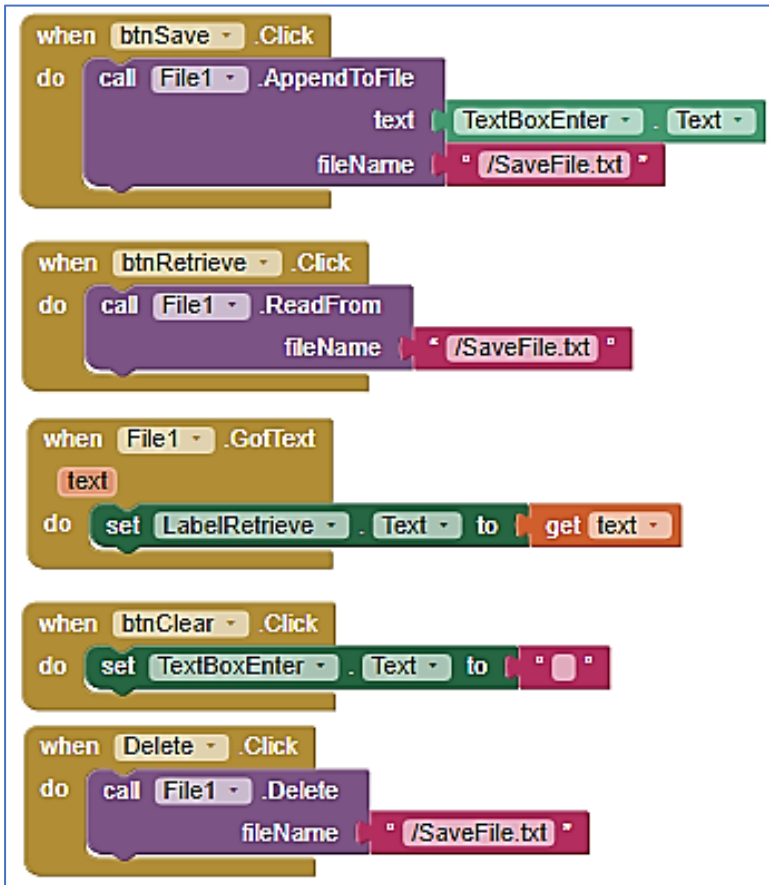
5. Event When Delete.Click



Perintah *Event* ini digunakan untuk menghapus nama File SaveFile.txt

6. *Block keseluruhan aplikasi*

Block keseluruhan dari semua event yang sudah di tampilkan dijelaskan di atas adalah sebagai berikut:

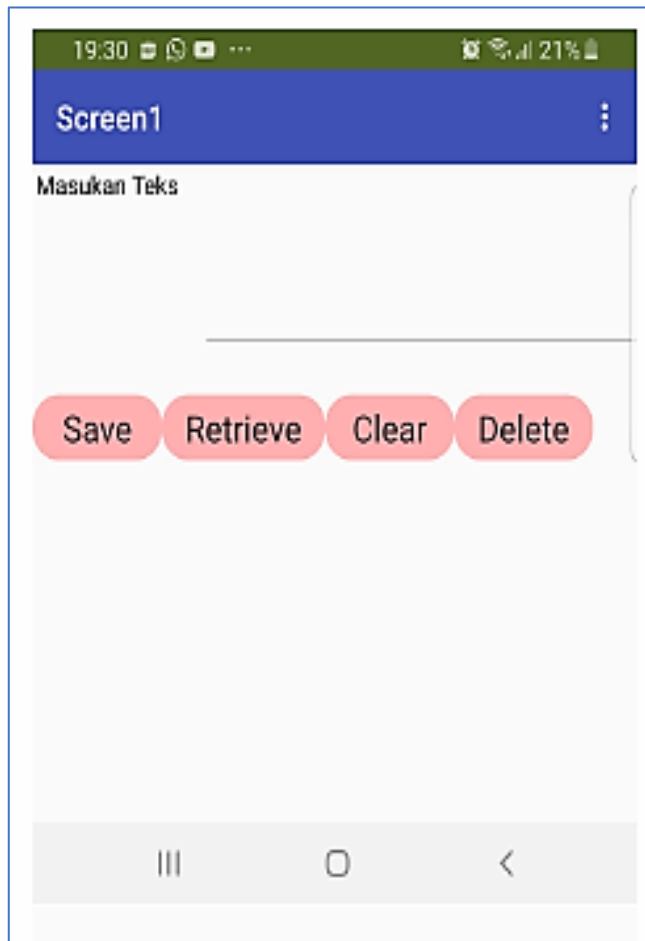


Langkah 5: Uji coba Aplikasi dengan Companion

Uji coba menggunakan aplikasi companion, langkahnya sama seperti pada aplikasi pada halaman sebelumnya.

Berikut ini gambar hasil ujicoba ke perangkat gadget android.

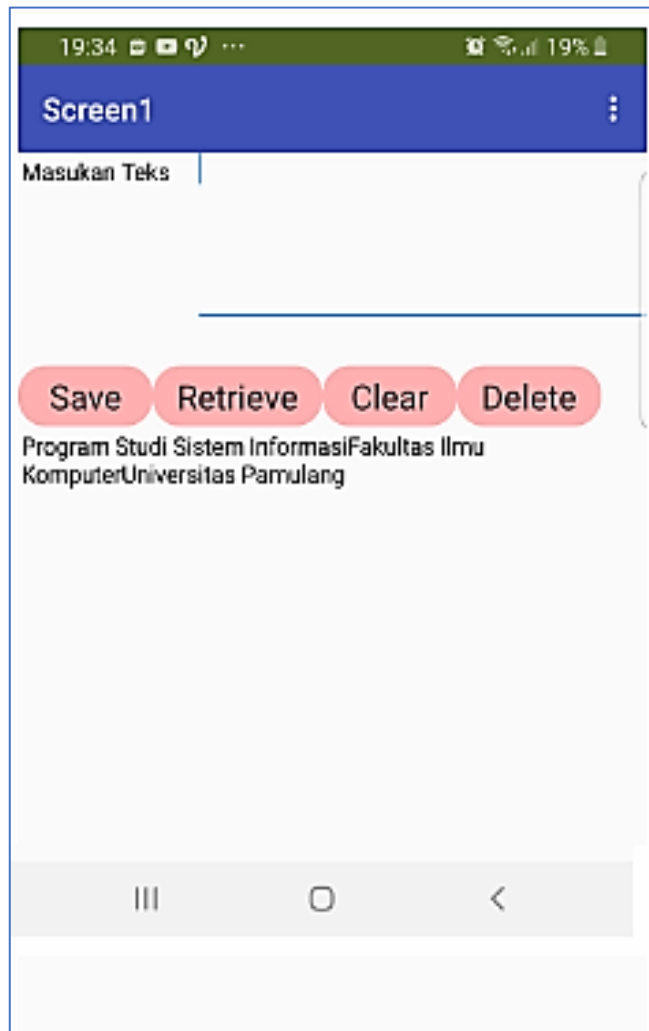
1. Tampilan Screen Awal hasil ujicoba aplikasi dijalankan pada perangkat gadget android



2. Tampilan Screen hasil ujicoba dengan mengisi teks pada kotak textboxEnter

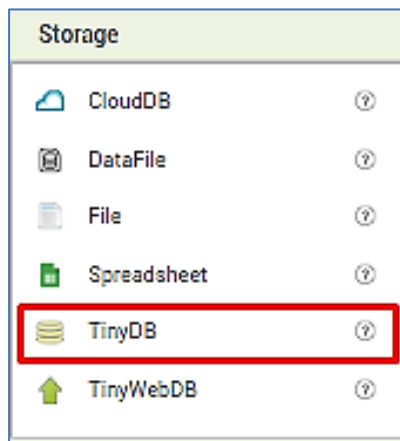


3. Tampilan Screen hasil ujicoba dengan mengisi teks, mengklik tombol *save*, mengklik tombol *retrieve* dan ditampilkan pada *labelretrieve*



BAB 4

BEKERJA DENGAN *TINYDB STORAGE*



A. Pendahuluan

TinyDB adalah komponen database lokal sederhana yang tidak terlihat (*Non Visible component*) dimana dapat menyimpan data untuk suatu aplikasi.

Ponsel atau tablet biasanya memiliki dua jenis memori utama yaitu RAM dan FLASH RAM. RAM adalah singkatan dari “Random Access Memory, RAM mempertahankan nilai selama daya dialirkan ke sirkuit RAM. Begitu kita mematikan daya, nilai yang tersimpan di RAM akan hilang.

Aplikasi yang dibuat dengan App Inventor diinisialisasi setiap kali dijalankan tersimpan dalam RAM. Artinya jika aplikasi menetapkan nilai variabel dan pengguna kemudian

keluar dari aplikasi, nilai variabel tersebut tidak akan tersimpan saat aplikasi dijalankan lagi. Sebaliknya, TinyDB adalah penyimpanan data persisten (nilai data tetap ada) tersimpan dalam Flash RAM pada aplikasi tersebut. Data yang disimpan di TinyDB akan tersedia setiap kali aplikasi dijalankan. Contohnya adalah permainan yang menyimpan skor tinggi dan mengambilnya kembali setiap kali permainan dimainkan.

Item data dari TinyDB terdiri dari tag dan nilai. Untuk menyimpan item data, kita harus menentukan tag tempat penyimpanannya. Tag harus berupa blok teks, yang memberi nama pada data. Selanjutnya, kita dapat mengambil data yang disimpan di bawah tag tertentu.

TinyDB tidak dapat digunakan untuk meneruskan data antara dua aplikasi berbeda di ponsel, meskipun Kita dapat menggunakan TinyDB untuk berbagi data antara layar berbeda pada aplikasi multilayar.

Saat kita mengembangkan aplikasi menggunakan AI Companion, semua aplikasi yang menggunakan Companion tersebut akan berbagi TinyDB yang sama. Berbagi itu akan hilang setelah aplikasi dikemas dan di install di perangkat gadget HP atau table android. Selama pengembangan, kita harus berhati-hati dalam menghapus data aplikasi Companion setiap kali kita mulai mengerjakan aplikasi baru.

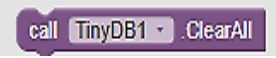
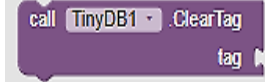

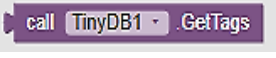
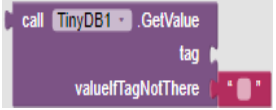

B. *Properties* pada Komponen TinyDB

Properties	Value	Keterangan
Namespace	Text	Namespace untuk menyimpan data.

C. *Event* pada Komponen TinyDB

Event tidak ada pada komponen TinyDB

D. Method pada Komponen TinyDB

Method	Value	Keterangan
		Menghapus seluruh penyimpanan data.
	(Tag text)	Menghapus entri dengan tag yang diberikan.
	Dictionary	Metode untuk mendapatkan semua data dalam bentuk Kamus
	any	Mengembalikan daftar semua tag di penyimpanan data.
	(text tag, any valueIfTagNotThere)	Mengambil nilai yang disimpan di bawah tag yang diberikan. Jika tidak ada tag seperti itu, mengembalikan valueIfTagNotThere
	(text tag, any valueToStore)	Simpan valueToStore yang diberikan di bawah nilai tag. Penyimpanan tetap ada di ponsel saat aplikasi dimulai ulang.

E. Contoh Aplikasi TinyDB

1. Aplikasi Penyimpanan dan Pencarian data TinyDB

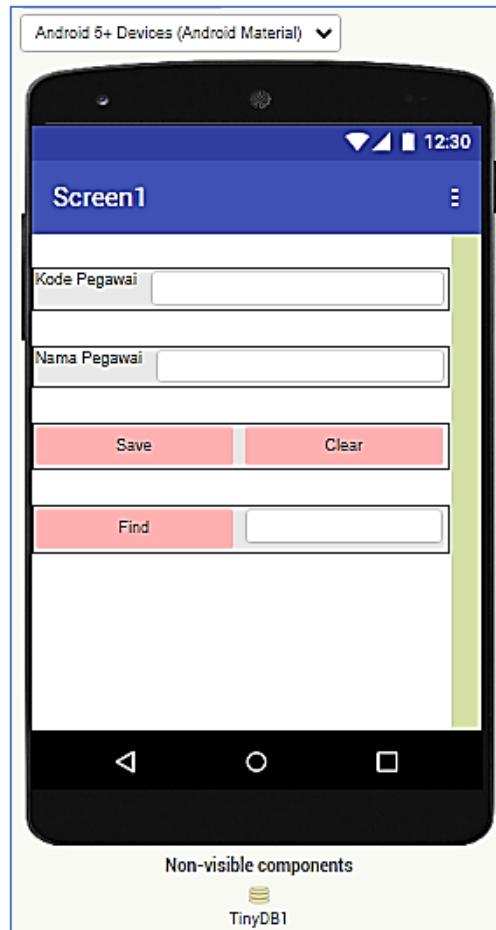
Aplikasi sederhana penyimpanan dan Pencarian data menggunakan Komponen tinyDB, membutuhkan 2 input yaitu kode pegawai dan Nama Pegawai, terdapat 3 tombol yaitu *Save* untuk penyimpanan ke tinyDB, *Clear* untuk mengosongkan komponen kode pegawai, nama pegawai dan *Find* untuk pencarian data tersimpan.

Langkah 1: Buat Project baru

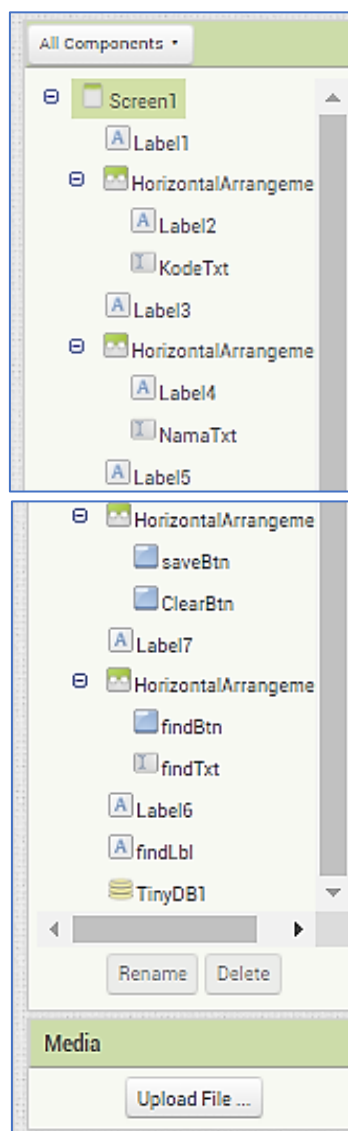
Langkah nya sama seperti contoh membuat project aplikasi sebelumnya.

Namakan Project contoh : tinyDBApp1

Langkah 2 Desain Aplikasi seperti contoh pada gambar dibawah ini:



Tampilan keseluruhan pada jendela *All Component*



Langkah 3: *Setting Properties* semua komponen

Berikut ini table acuan untuk men set property pada semua komponen desain:

No	Pallete	Component	Properties	Keterangan
1		Screen1	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Left: 1 AlignVertical: Top: 1 BackgroundColor: Default BackgroundImage: None BigDefaultText: None CloseScreenAnimati on: Default HighContrast: None OpenScreenAnimati on: Default ScreenOrientation: Portrait Scrollable: None ShowStatusBar: True Title: Screen1 TitleVisible: True	Memberikan Nilai Properties untuk komponen Screen1
2	User Interface	Label1 Name: Label1	Label1 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent	Komponen ini digunakan sebagai pembatas spasi diatas komponen horizontalArrangement1

No	Pallette	Component	Properties	Keterangan
			Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	
3	Layout	HorizontalArrangement1	HorizontalArrangement1 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/ mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Label2 dan KodeTxt
4	User Interface	Label2 Name: Label2	Label2 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: Automatic Width: Automatic Text: Kode Pegawai TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini Digunakan untuk memberikan judul Kode Pegawai disebelah kotak isian TextBox Kode Pegawai
5	User Interface	Textbox1 Name: Kodetxt	KodeTxt (TextBox) Appearance BackgroundColor:	Komponen ini digunakan sebagai kotak isian input kode pegawai

No	Pallete	Component	Properties	Keterangan
			Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	
6	User Interface	Label3 Name: Label1	Label3 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai pembatas spasi antara kode pegawai dengan nama pegawai
7	Layout	HorizontalArrangement2 Name: HorizontalArrangement2	HorizontalArrangement2(HorizontalArrangement)	Digunakan untuk menyusun/mengelompokkan komponen dari

No	Pallete	Component	Properties	Keterangan
			Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	kiri kekanan didalamnya terdapat Komponen Label4 (Label Nama pegawai dan isian textbox Nama Pegawai
8	User Interface	Label4 Name: Label4	Label4 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: Automatic Width: Automatic Text: Nama Pegawai TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini Digunakan untuk memberikan judul Nama Pegawai disebelah kotak isian TextBox Nama Pegawai
9	User Interface	Textbox1 Name: NamaTxt	NamaTxt (TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default	Komponen ini digunakan sebagai kotak isian input Nama pegawai

No	Pallete	Component	Properties	Keterangan
			Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	
10	User Interface	Label5 Name: Label5	Label5(Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai pembatas spasi
11	Layout	HorizontalArrangement3 Name: HorizontalArrangement3	HorizontalArrangement3 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/menge-lompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Button saveBtn dan clearBtn

No	Pallete	Component	Properties	Keterangan
12	User Interface	Button1 Name: saveBtn	saveBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Save TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	Komponen tombol ini digunakan untuk menyimpan data kodepegawai, nama pegawai dengan mengklik tombol save
13	User Interface	Button2 Name: clearBtn	clearBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Clear TextAlignment: center: 1 TextColor: Default Visible: True	Komponen tombol ini digunakan untuk membersihkan isian kodepegawai, nama pegawai, jika ingin menambah data baru.

No	Pallete	Component	Properties	Keterangan
			Behavior Enabled	
14	User Interface	Label7 Name: Label7	Label7(Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai pembatas spasi antara horizontalArrangement3 dan horizontalArrangement4
15	Layout	HorizontalArrangement4 Name: HorizontalArrangement4	HorizontalArrangement4(HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Button sfindBtn dan FindTxt Textbox
16	User Interface	Button3 Name: findBtn	findBtn(Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface:	Komponen tombol ini digunakan untuk mencari data kodepegawai, nama pegawai yang tersimpan pada tinyDB

No	Pallette	Component	Properties	Keterangan
			default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Find TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	
17	User Interface	Textbox1 Name: FindTxt	findTxt(TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	Komponen ini digunakan sebagai kotak isian untuk pencarian dengan meng input kodepegawai dan akan ditampilkan nama pegawai jika ditemukan
18	User Interface	Label6 Name: Label6	Label6(Label) Appearance BackgroundColor: None	Komponen ini digunakan sebagai pembatas spasi

No	Pallete	Component	Properties	Keterangan
			FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	
19	User Interface	Label8 Name: findLbl	Label6(Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: None TextAlignment: left: 0 TextColor: Default Visible: True	Komponen ini digunakan sebagai penampil data Nama pegawai jika ketika pencarian ditemukan
20	Storage	TinyDB1 Name: TinyDB1	TinyDB1 (TinyDB) Behavior Namespace: TinyDB1	Digunakan sebagai penampung data tag kodepegawai, value nama pegawai

Langkah 4: Perintah Block Screen Aplikasi

Berikut ini perintah *block* untuk menyusun kode berupa *puzzle* pada screen Aplikasi

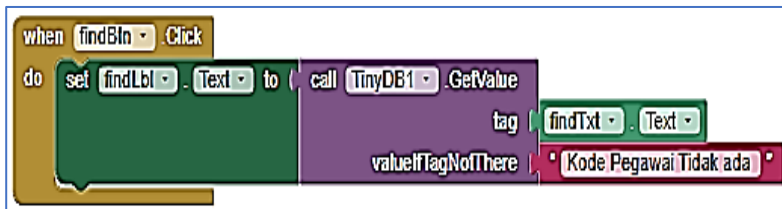
a. *Event when saveBtn.click*



Penjelasan:

Event *when saveBtn.click* digunakan untuk menyimpan data kode pegawai, nama pegawai ke tinyDB1 setelah pengisian kode pegawai, nama pegawai dilakukan ketika tombol Save di klik.

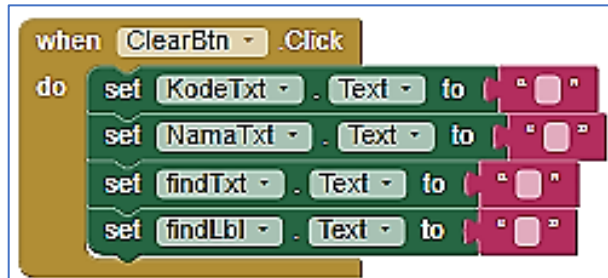
b. *Event when findBtn.click*



Penjelasan:

Event *when findBtn.click* digunakan untuk menampilkan data nama pegawai dari database tinyDB1 dan men set nilai findLbl dengan tag yang di input dari kotak isian findTxt ketika tombol find di klik

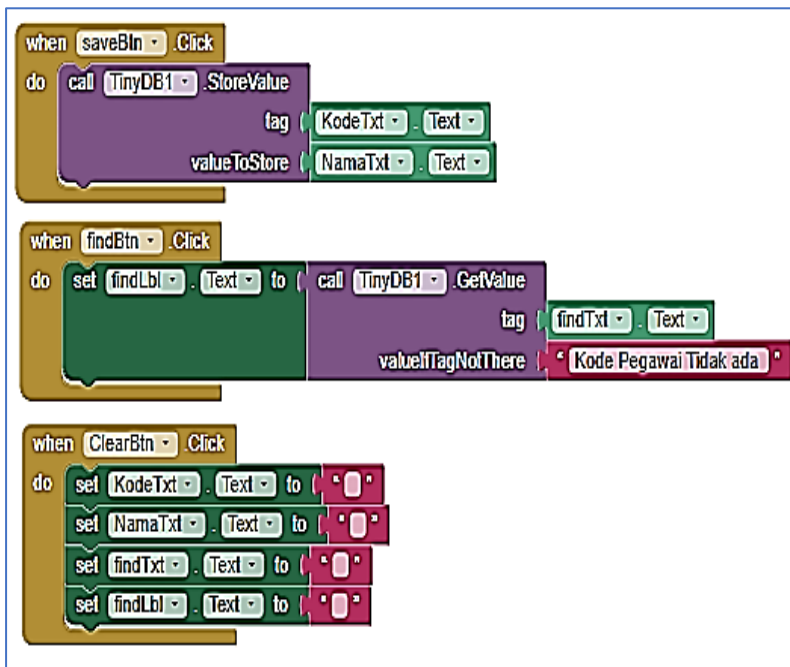
c. Event *when clearBtn.Click*



Penjelasan:

Event *when ClearBtn.click* digunakan untuk mengosongkan semua komponen textbox yaitu kodepegawai, Nama Pegawai, kotak pencarian dan label tampilan hasil pencarian ditemukan ketika tombol find di klik

d. Tampilan *block* keseluruhan seperti gambar dibawah ini:

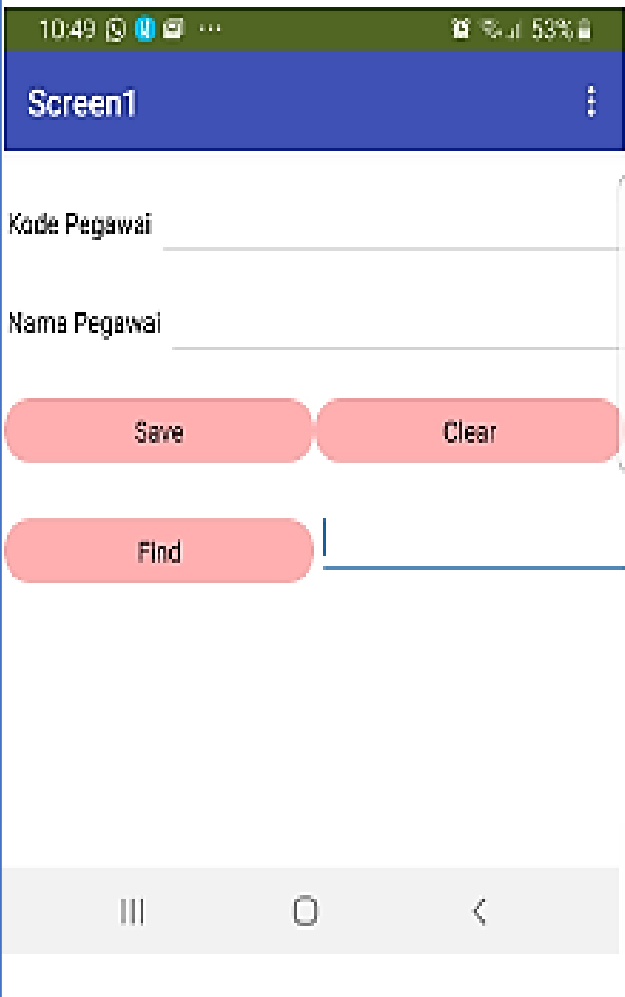


Langkah 5: Uji coba Aplikasi dengan Companion

Uji coba menggunakan aplikasi companion, langkahnya sama seperti pada ujicoba aplikasi pada halaman sebelumnya.

Berikut ini gambar hasil ujicoba ke perangkat gadget android.

a. Tampilan Screen Awal hasil ujicoba



The screenshot shows the initial screen of an Android application. At the top, the status bar displays the time 10:49, signal strength, Wi-Fi, and battery level at 53%. Below the status bar is a blue header bar with the text "Screen1" and a vertical ellipsis menu icon on the right. The main content area is white and contains two text input fields labeled "Kode Pegawai" and "Nama Pegawai". Below these fields are two red buttons labeled "Save" and "Clear". Further down is a red button labeled "Find" next to a text input field. At the bottom of the screen is a grey navigation bar with three icons: a vertical ellipsis, a circle, and a left-pointing arrow.

- b. Tampilan Screen pengisian kode pegawai, nama pegawai kemudian menekan tombol *save*

11:01 46%

Screen1

Kode Pegawai 002

Nama Pegawai Restu

Save Clear

Find

Ratu Resto Eres tú ...

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l ñ

↑ z x c v b n m ↵

!#1 , < Español (ES) > . Sisi

- c. Tampilan Screen hasil pencarian setelah kode pegawai di input selanjutnya meng klik tombol *find*

11:01 46%

Screen1

Kode Pegawai

Nama Pegawai

Save Clear

Find 002

Restu

Virtual Keyboard (Spanish):

años de minutos

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l ñ

↑ z x c v b n m ↵

!#1 , < Español (ES) > . Sisi

2. Aplikasi Penyimpanan, Pencarian data dan Menampilkan data dengan ListView TinyDB

Aplikasi sederhana penyimpanan dan Pencarian data serta menampilkan menggunakan ListView tinyDB, sama seperti contoh aplikasi tinyDB sebelumnya, perbedaan adalah semua yang sudah tersimpan pada database tinyDB ditampilkan pada *ListView*.

Aplikasi membutuhkan 2 input yaitu

- a. NPM,
- b. Nama Mahasiswa,

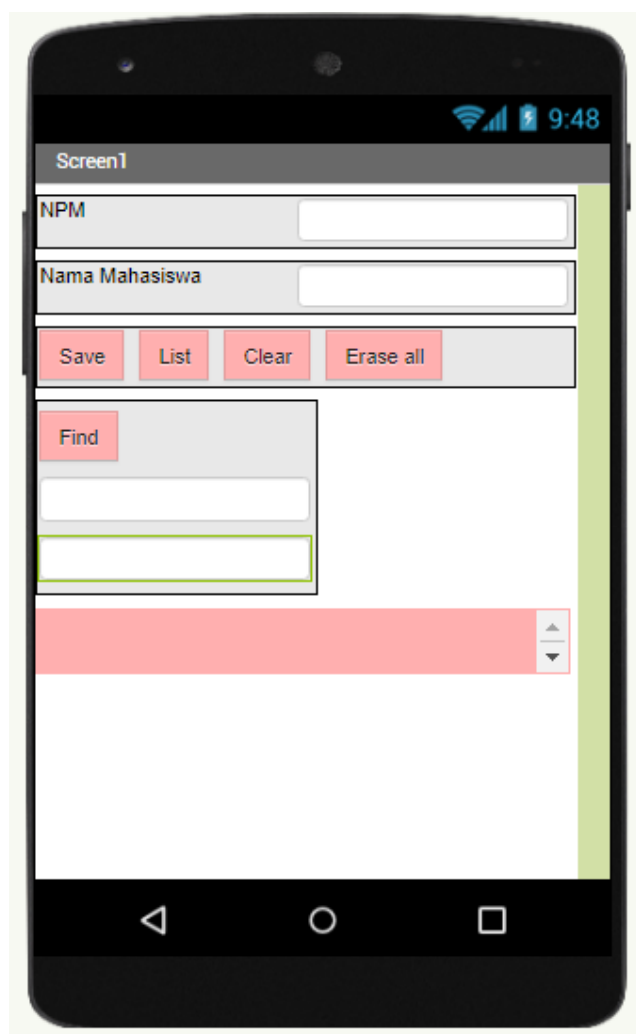
Terdapat 5 tombol yaitu

- a. Save,
- b. List,
- c. Clear,
- d. Erase All,
- e. Find.

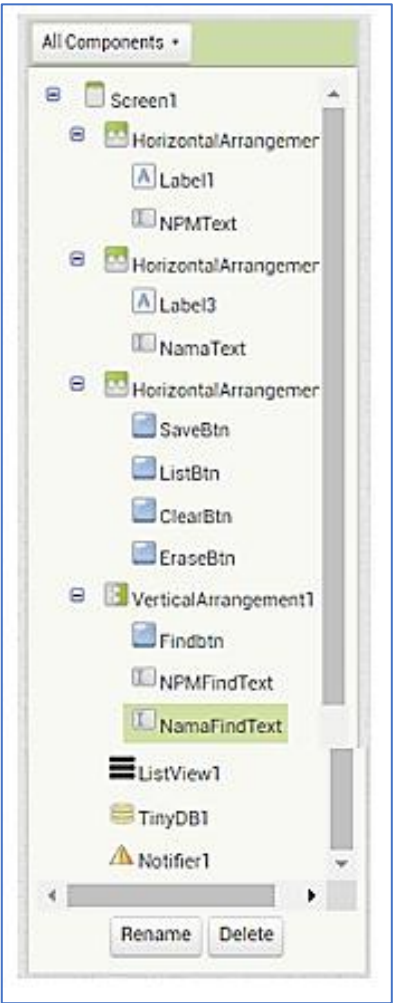
Tombol *Save* digunakan untuk penyimpanan ke tinyDB, Tombol *List* digunakan untuk menampilkan data tersimpan dan ditampilkan pada ListView, Tombol *Clear* untuk mengosongkan komponen NPM pada Find, Nama pada Find. Tombol *Erase All* digunakan untuk menghapus semua data tersimpan pada *tinyDB*, Tombol Find digunakan untuk pencarian data tersimpan

Langkah 1: Buat Project Baru Namakan contoh : tinyDBApp2

Langkah 2. Desain Aplikasi Seperti pada gambar dibawah ini:



Tampilan keseluruhan pada jendela *All Component*



Langkah 3: Setting Properties semua komponen

Berikut ini table acuan untuk men set property pada semua komponen desain:

No	Pallette	Component	Properties	Keterangan
1		Screen1	tinyDBApp2 (Form) Appearance AboutScreen: None AlignHorizontal: Left: 1	Memberikan Nilai Properties untuk komponen Screen1

No	Pallete	Component	Properties	Keterangan
			AlignVertical: Top: 1 BackgroundColor: Default BackgroundImage: None BigDefaultText: None CloseScreenAnimatio: Default HighContrast: None OpenScreenAnimatio: Default ScreenOrientation: Portrait Scrollable: None ShowStatusBar: True Title: tinyDBApp1 TitleVisible: True	
2	Layout	HorizontalArrangement1	HorizontalArrangement1 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: 110pixels Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Label1 dan NPMTText
3	User Interface	Label2 Name: Label1	Label2 (Label) Appearance BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: Automatic	Komponen ini Digunakan untuk memberikan judul NPM disebelah kotak isian TextBox NPMTText

No	Pallette	Component	Properties	Keterangan
			Width: Automatic Text: NPM TextAlignment: left: 0 TextColor: Default Visible: True	
4	User Interface	TextBox Name: NPMText	NPMText(TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	Komponen ini digunakan sebagai kotak isian input NPM
5	Layout	HorizontalArrangement2	HorizontalArrangement2 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: Automatic Width: Fill parent Image: None... Visible: True	Digunakan untuk menyusun/mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen Label3 dan NamaText
6	User Interface	Label3 Name: Label1	Label3 (Label) Appearance	Komponen ini digunakan sebagai

No	Pallette	Component	Properties	Keterangan
			BackgroundColor: None FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default HTMLFormat: None HasMargins: None Height: 3 percent Width: Automatic Text: Nama Mahasiswa TextAlignment: left: 0 TextColor: Default Visible: True	pemberian judul Nama Mahasiswa
7	User Interface	TextBox Name: NPMText	NPMText(TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	Komponen ini digunakan sebagai kotak isian input NPM
8	Layout	HorizontalArrangement3	HorizontalArrangement3 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1	Digunakan untuk menyusun/mengelompokkan komponen dari kiri kekanan didalamnya terdapat Komponen

No	Pallette	Component	Properties	Keterangan
			BackgroundColor: Default Height: Automatic Width: Fill parent Image: None... Visible: True	SaveBtn, ListBtn,ClearBtn ,EraseBtn,
9	User Interface	Button1 Name: saveBtn	saveBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Save TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	Komponen tombol ini digunakan untuk menyimpan data NPM, Nama Mahasiswa dengan mengklik tombol <i>SaveBtn</i>
10	User Interface	Button2 Name: ListBtn	ListBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback:	Komponen tombol ini digunakan untuk menampilkan data NPM, Nama Mahasiswa pada List View dengan mengklik tombol <i>ListBtn</i>

No	Pallette	Component	Properties	Keterangan
			Text: List TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	
11	User Interface	Button3 Name: clearBtn	clearBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Clear TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	Komponen tombol ini digunakan untuk membersihkan isian NPM, Nama Mahasiswa, jika ingin menambah data baru.
13	User Interface	Button4 Name: EraseBtn	EraseBtn (Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback:	Komponen tombol ini digunakan untuk menghapus data NPM, Nama Mahasiswa yang tersimpan pada tinyDB

No	Pallette	Component	Properties	Keterangan
			Text: Erase All TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	
14	Layout	VerticalArrangement Name: VerticalArrangement1	VerticalArrangement1 (VerticalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: top: 1 BackgroundColor: Default Height: Automatic Width: Automatic Image: None... Visible: True	Digunakan untuk menyusun/men kelompokkan komponen dari atas ke bawah didalamnya terdapat Komponen Button FindBtn ,NPMFindText, NamaFindText
15	User Interface	Button Name: FindBtn	findBtn(Button) Appearance BackgroundColor: Pink FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Image: None Shape: default ShowFeedback: Text: Find TextAlignment: center: 1 TextColor: Default Visible: True Behavior Enabled	Komponen tombol ini digunakan untuk mencari data NPM, Nama Mahasiswa yang tesimpan pada tinyDB
16	User Interface	Textbox Name:	NPMfindText(TextBox)	Komponen ini digunakan

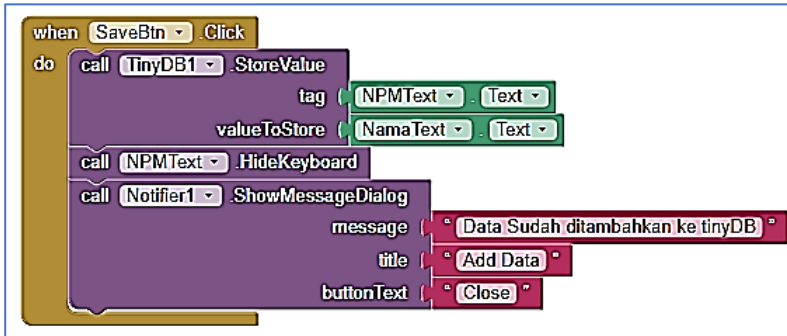
No	Pallette	Component	Properties	Keterangan
		NPMFindText	Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	sebagai kotak isian untuk pencarian dengan meng input NPM dan akan ditampilkan Nama Mahasiswa jika ditemukan
17	User Interface	Textbox Name: Nama FindText	NamafindText(TextBox) Appearance BackgroundColor: Default FontBold: None FontItalic: None FontSize: 14.0 FontTypeface: default Height: Automatic Width: Fill parent Hint: None TextAlignment: left: 0 TextColor: Default Visible: True Behavior Enabled: True MultiLine: None NumbersOnly: None ReadOnly: None Text: None	Komponen ini digunakan sebagai kotak isian untuk Menampilkan Nama Mahasiwa jika ditemukan ketika pencarian dilakukan dengan meng input NPM
18	User Interface	ListView Name:	ListView1(ListView)	Komponen ini digunakan

No	Pallette	Component	Properties	Keterangan
		ListView1	<p>Appearance BackgroundColor: pink FontSizeDetail: 14.0 FontTypeface: default Height: Automatic Width: Automatic Image Height:200 Image Width:200 ListViewLayout: MainText Orientation: Vertical Selection Color: Default TextColor: Default TextColorDetail: Default TextSize: 22</p> <p>Behavior ElementFromString: None ListData: ClickToAdd/DeleteData</p>	sebagai penampil data Nama pegawai jika ketika pencarian ditemukan
19	Storage	TinyDB1 Name: TinyDB1	<p>TinyDB1 (TinyDB) Behavior Namespace: TinyDB1</p>	Digunakan sebagai penampung data tag NPM, value Nama Mahasiswa
20	User Interface	Notifier Name: Notifier1	<p>Notifier1 (Notifier) Appearance BackgroundColo: Default NotifierLength: Long ▼ TextColor:Default</p>	Digunakan untuk menampilkan Pesan

Langkah 4: Perintah Block Screen Aplikasi

Berikut ini perintah *block* untuk menyusun kode berupa *puzzle* pada screen Aplikasi

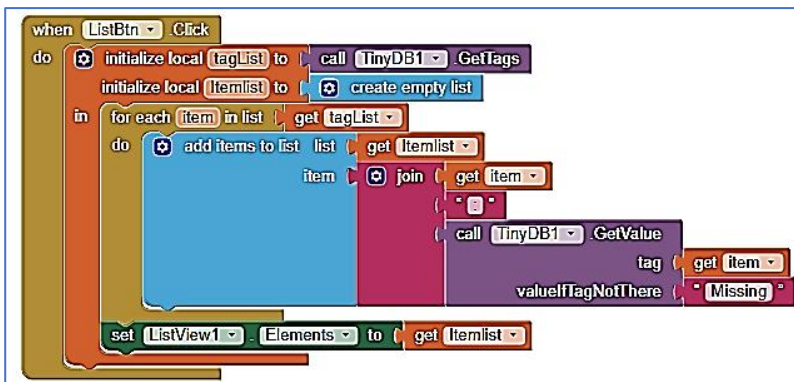
a. *Event When SaveBtn.Click*



Penjelasan:

Event *when saveBtn.click* digunakan untuk menyimpan data NPM, Nama Mahasiswa ke database tinyDB1 setelah pengisian NPM, Nama Mahasiswa dilakukan lalu meng klik tombol Save, selanjutnya menampilkan notifikasi "Data Sudah ditambahkan ke tinyDB".

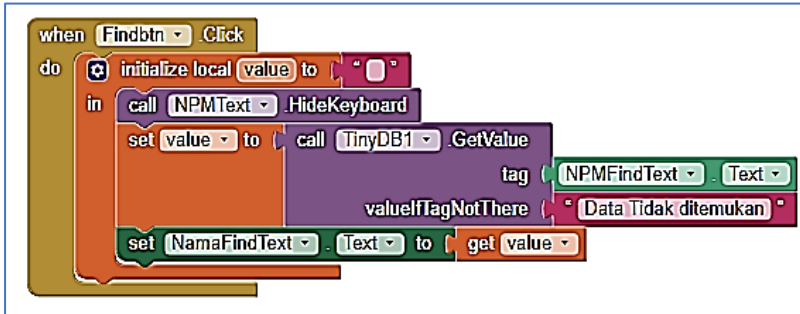
b. *Event When ListBtn.Click*



Penjelasan:

Event *when ListBtn.click* digunakan untuk menampilkan data NPM, Nama Mahasiswa yang sudah tersimpan pada database tinyDB1 dengan meng klik tombol List.

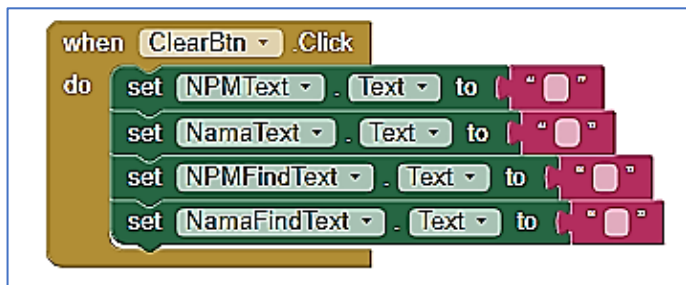
c. *Event When FindBtn.Click*



Penjelasan:

Event *when Findbtn.click* digunakan untuk mencari data tersimpan dengan memasukkan NPM pada NPMTextFind sebagai Tag, jika data tidak ada akan menampilkan "Data Tidak ditemukan", jika NPM ditemukan tampilkan Nama Mahasiswa pada komponen NamaFindText.

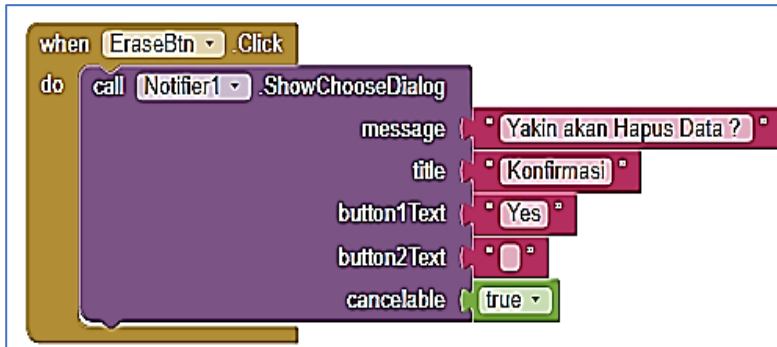
d. *Event When ClearBtn.Click*



Penjelasan:

Event *when ClearBtn.click* digunakan untuk membersihkan semua kotak isian textbox NPMText, NamaText, NPMFindtext, NamaFindText setelah pengisian data baru atau pencarian data dilakukan.

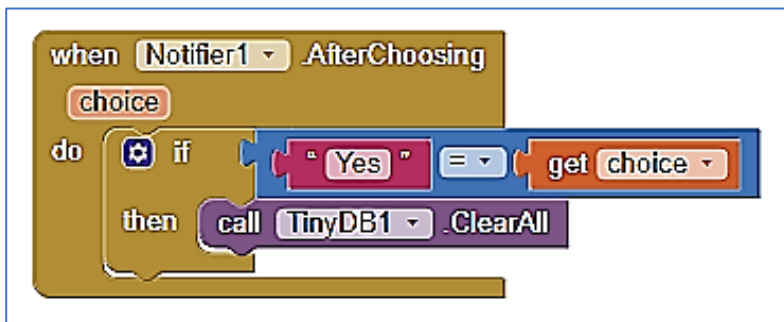
e. Event When EraseBtn.Click



Penjelasan:

Event *when EraseBtn.click* digunakan untuk menghapus data NPM, Nama Mahasiswa yang sudah tersimpan pada tinyDB, sebelum menghapus akan dimunculkan pesan notifikasi untuk memastikan dengan tampilan pesan “Yakin akan Menghapus Data? , jika menekan tombol Yes akan menuju ke block notifier untuk penghapusan data.

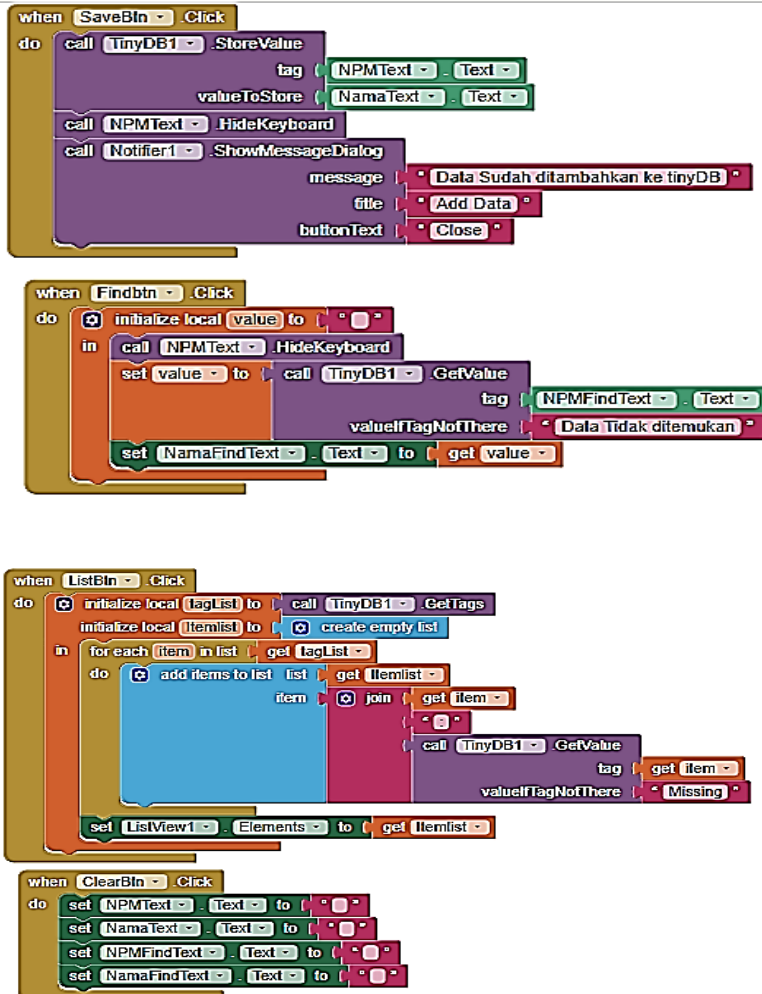
f. Event When Notifier.AfterChoosing

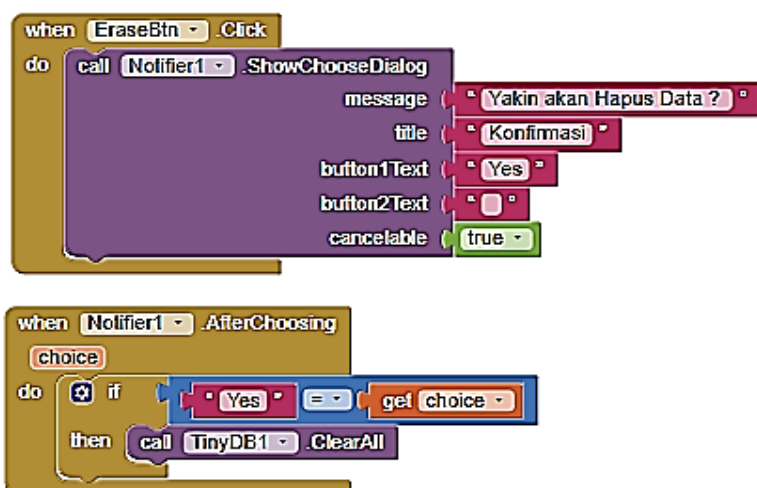


Penjelasan:

Event *when Notifier.AfterChoosing* adalah lanjutan dari Event *when EraseBtn.click* dimana menampilkan notifikasi ketika tombol Yes di tekan/dipilih akan menghapus semua data yang tersimpan pada tinyDB.

g. Tampilan pada jendela *Block Keseluruhan*



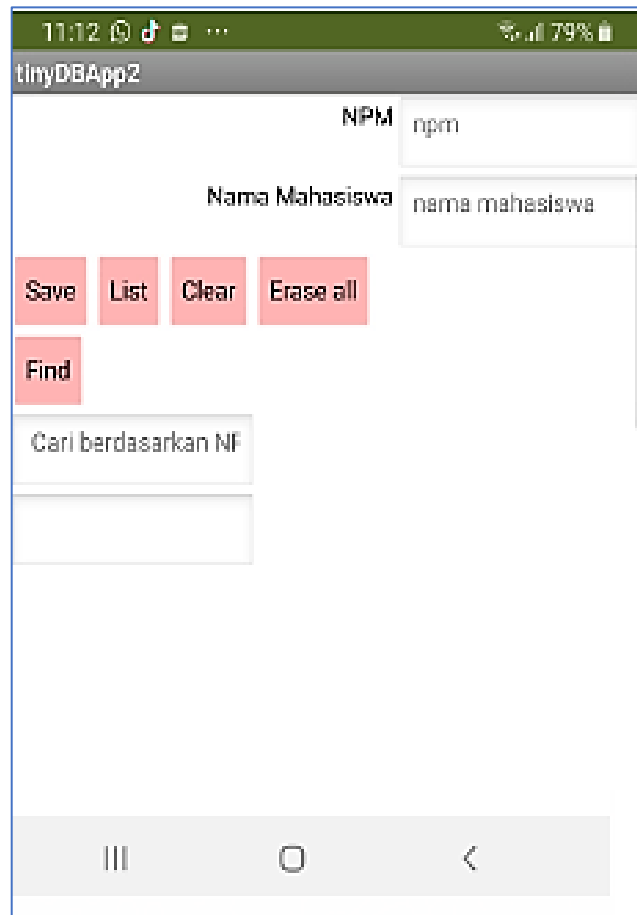


Langkah 5: Uji coba Aplikasi dengan Companion

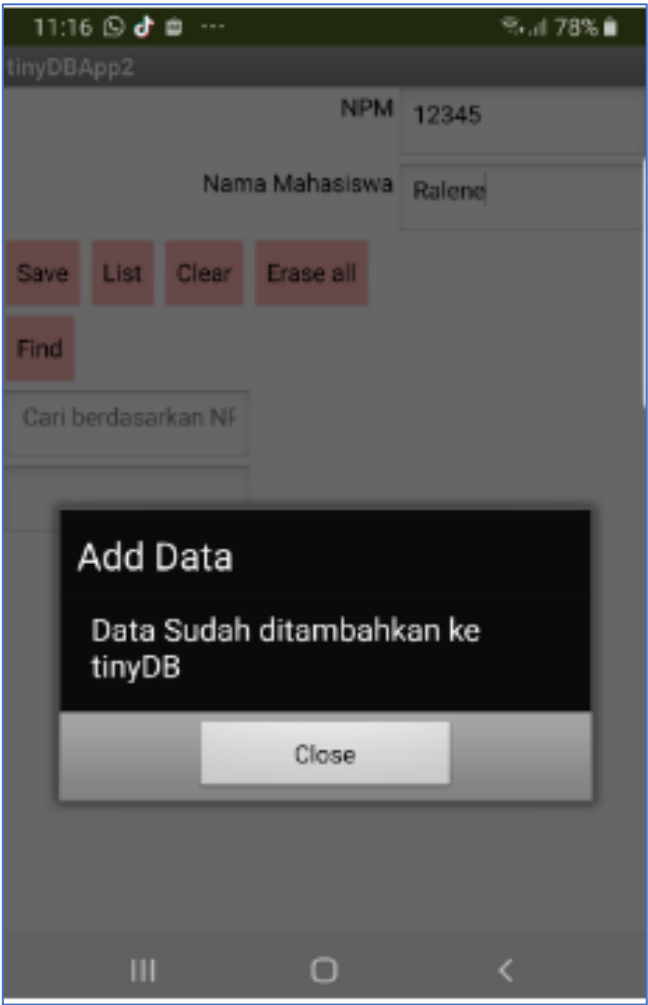
Uji coba menggunakan aplikasi companion, langkahnya sama seperti pada ujicoba aplikasi pada halaman sebelumnya.

Berikut ini gambar hasil ujicoba ke perangkat gadget android.

a. Tampilan Screen Awal hasil ujicoba



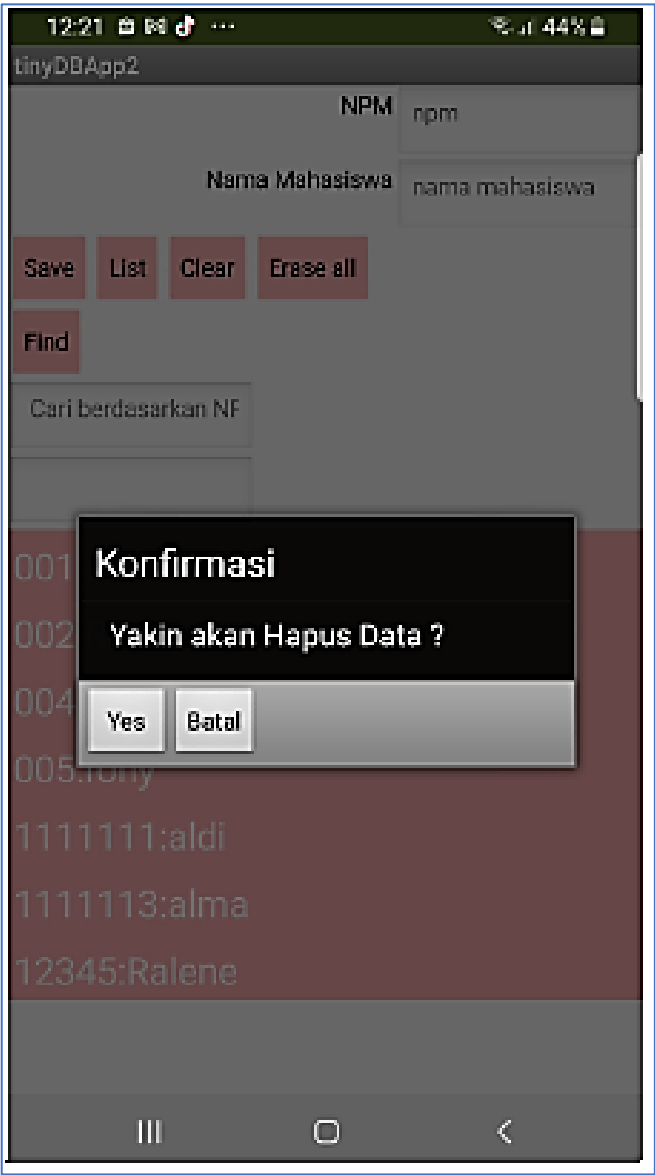
b. Tampilan Screen Menambah Data Baru



c. Tampilan Screen Menampilkan pada List

The screenshot shows the 'tinyDBApp2' application interface on a mobile device. At the top, the status bar displays the time 12:19, signal strength, and 45% battery. The app title 'tinyDBApp2' is in the top bar. Below it, there are two input fields: 'NPM' with the value 'npm' and 'Nama Mahasiswa' with the value 'nama mahasiswa'. A row of four red buttons follows: 'Save', 'List', 'Clear', and 'Erase all'. Below these is a 'Find' button and a search input field containing the text 'Cari berdasarkan NF'. The main area of the app is a red list box containing the following text: '001:Joko', '002:Restu', '004:dude', and '005:rony'. At the bottom, there is a navigation bar with three icons: a list icon (three vertical bars), a home icon (a circle), and a back icon (a left-pointing arrow).

d. Tampilan Screen Menghapus semua Data(Erase all)

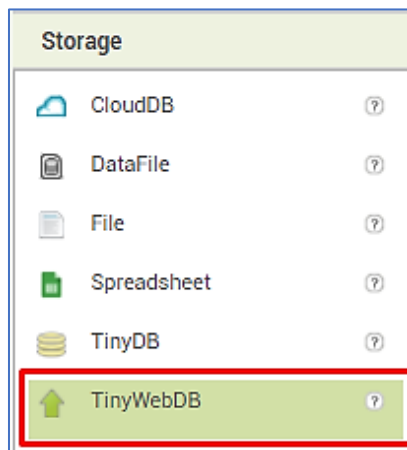


e. Tampilan Screen Menampilkan Hasil Pencarian (Find)





BAB 5 | BEKERJA DENGAN *TINYWEBDB STORAGE*



A. Pendahuluan

TinyWebDB adalah komponen App Inventor yang memungkinkan kita menyimpan data secara persisten dalam database di web. Karena data disimpan di web dan bukan di ponsel tertentu, TinyWebDB dapat digunakan untuk memfasilitasi komunikasi antara ponsel dan aplikasi (misalnya, game dengan multi-pemain).

Secara default, komponen TinyWebDB menyimpan data pada layanan pengujian yang disediakan oleh App Inventor, yaitu di <http://tinywebdb.appinventor.mit.edu/>. Layanan ini berguna untuk pengujian, tetapi digunakan bersama oleh semua

pengguna App Inventor, dan memiliki batas 1000 entri. Jika kita menggunakannya, data kita pada akhirnya akan tertimpa.



Untuk sebagian besar aplikasi yang kita tulis, dimana jika ingin membuat layanan web *custom* yang tidak dibagikan dengan aplikasi dan pemrogram App Inventor lainnya. Kita bisa membuat layanan web sendiri tahapan nya ada pada link berikut ini


<https://ai2.appinventor.mit.edu/reference/other/tinywebdb.html>. Pada Bab ini kita menggunakan layanan web secara default.

B. Properties pada komponen TinyWebDB

Properties	Value	Keterangan
ServiceURL	Text	Menentukan URL layanan Web. Nilai defaultnya adalah layanan demo yang berjalan di App Engine.

C. Event Pada Komponen TinyWebDB

Events	Value	Keterangan
	(Text) tagFromWebDB, (any) valueFromWebDB	Menunjukkan bahwa permintaan server GetValue telah berhasil.
		Menunjukkan bahwa permintaan server StoreValue telah berhasil.

Events	Value	Keterangan
	(Text) Message	Menunjukkan bahwa komunikasi dengan layanan Web menandakan adanya kesalahan

D. Method pada TinyWebDB

Method	Value	Keterangan
	(text tag)	GetValue meminta layanan Web untuk menyimpan nilai di bawah tag yang diberikan. Terserah pada layanan Web apa yang harus dikembalikan jika tidak ada nilai yang disimpan di bawah tag. Komponen ini hanya menerima apapun yang dikembalikan. Event GotValue akan dijalankan setelah selesai.
	Text tag, any valueToStore	Mengirimkan permintaan ke layanan Web untuk

Method	Value	Keterangan
		menyimpan valueToStore yang diberikan di bawah tag yang diberikan. Event ValueStored akan dijalankan setelah selesai.

E. Contoh Proyek Aplikasi Menggunakan TinyWebDB

Contoh aplikasi ini akan menunjukkan bagaimana cara menyimpan data ke web service default menggunakan tinyWebDB.

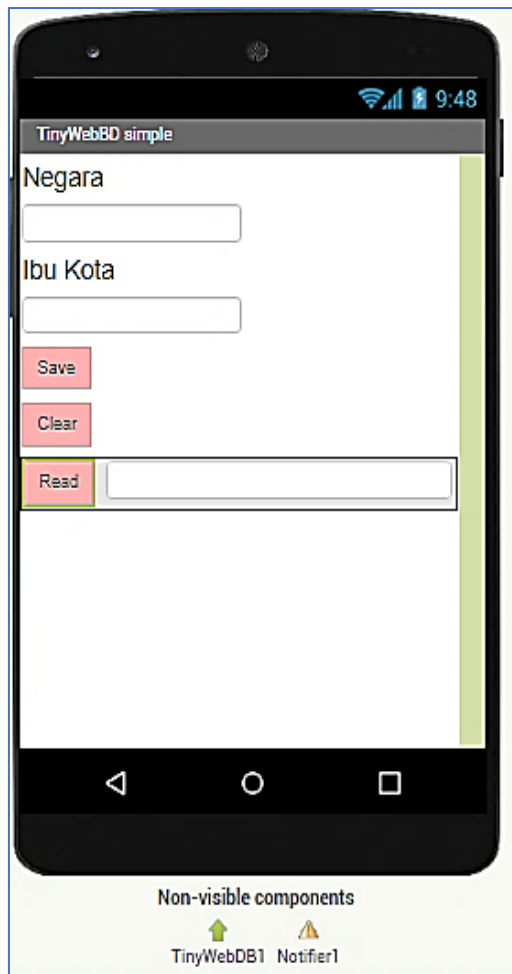
Aplikasi akan diminta untuk input negara dan ibukota, lalu akan menunjukkan bagaimana cara mengambil data yang tersimpan dan menampilkannya pada perangkat gadget kita.

Langkah: 1 membuat Project baru:

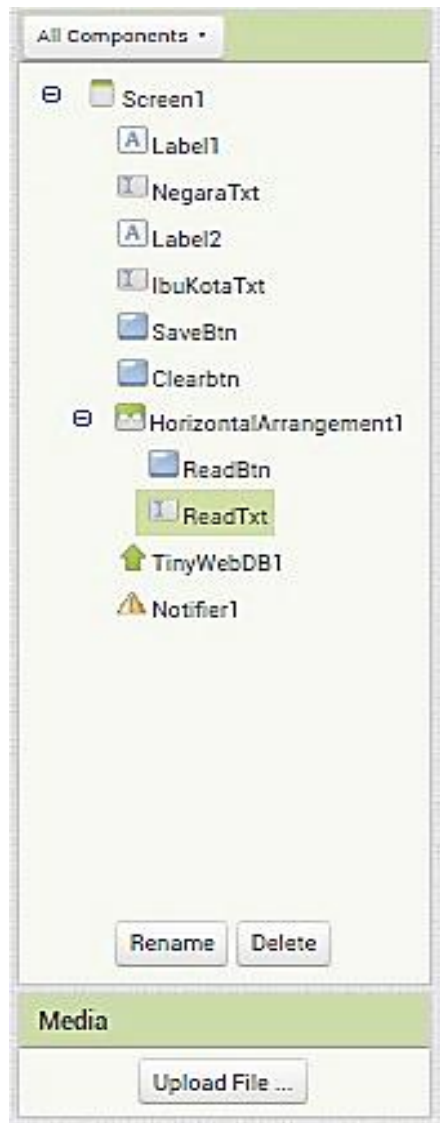
1. Masuk ke web mit app inventor pada browser ketik: <https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik Create, selanjutnya masuk menggunakan akun google kita, atau klik new project jika halaman mit app sedang aktif.
3. Lalu beri nama project contoh seperti gambar dibawah ini:

Langkah: 2 Perancangan Desain

Desain Screen terdapat dua Komponen input textbox yaitu input Negara dan input Ibukota nya, terdapat tiga *Button* tombol terdiri dari *Save* yang digunakan untuk menyimpan data Negara dan Ibukota, *Clear* untuk membersihkan 3 kotak textbox, *Read* digunakan untuk membaca data dengan meng input Negara dan akan menampilkan notifikasi Negara serta Ibukota nya yang sudah tersimpan, berikut ini tampilan desain nya:



Berikut ini gambar Komponen keseluruhan pada jendela *All component*:



Langkah 3: Set Properties komponen keseluruhan

Setting properties semua komponen desain *screen Log in* mengacu pada tabel berikut ini:

No	Pallete	Component	Properties	Keterangan
1		Screen1	Screen1 (Form) Appearance AboutScreen: None AlignHorizontal: Left: 1 AlignVertical: Top: 1 BackgroundColor: None BackgroundImage: None BigDefaultText: None CloseScreenAnimation: Default HighContrast: None OpenScreenAnimation: Default ScreenOrientation: Portrait Scrollable: None ShowStatusBar: True Title: Log In TitleVisible: True	Memberikan Nilai Properties untuk komponen Screen1
2	User Interface	Label Name: Label1	Label1 (Label) Appearance BackgroundColor: <i>None</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 20 FontTypeface: <i>default</i> HTMLFormat: <i>None</i> HasMargins: <i>None</i> Height: <i>Automatic</i> Width: <i>Automatic</i> Text: Negara TextAlignment: <i>left: 0</i> ▾ TextColor: <i>Default</i> Visible: <i>True</i>	Label Digunakan untuk memberikan judul disisi kiri kotak isian Negara
3	User Interface	TextBox Name: NegaraTxt	NegaraTxt (TextBox) Appearance BackgroundColor: <i>Default</i> FontBold: <i>None</i>	Digunakan sebagai input Negara pada kotak isian/TextBox

No	Pallete	Component	Properties	Keterangan
			FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default</i> Height: <i>Automatic</i> Width: <i>Automatic</i> Hint: <i>None</i> HintColor: <i>Default</i> TextAlignment: <i>left: 0</i> TextColor: <i>Default</i> Visible: <i>True</i> Behavior Enabled: <i>True</i> MultiLine: <i>None</i> NumbersOnly: <i>None</i> ReadOnly: <i>None</i> Text: <i>None</i>	
4	<i>User Interface</i>	<i>Label Name: Label2</i>	Label2 (Label) Appearance BackgroundColor: <i>None</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 20 FontTypeface: <i>default</i> HTMLFormat: <i>None</i> HasMargins: <i>None</i> Height: <i>Automatic</i> Width: <i>Automatic</i> Text: <i>Ibu Kota</i> TextAlignment: <i>left : 0 ▾</i> TextColor: <i>Default</i> Visible: <i>True</i>	Label Digunakan untuk memberikan judul disisi kiri kotak isian Ibu Kota
5	<i>User Interface</i>	<i>TextBox2 Name: IbuKotaTxt</i>	IbuKotaTxt (TextBox) Appearance BackgroundColor: <i>Default</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default...</i> Height: <i>Automatic...</i> Width: <i>Automatic...</i> Hint: <i>None</i> HintColor: <i>Default</i>	Digunakan untuk input Ibu Kota pada kotak isian

No	Pallete	Component	Properties	Keterangan
			TextAlignment: <i>left : 0</i> ▼ TextColor: <i>Default</i> Visible: <i>True</i> Behavior Enabled: <i>True</i> MultiLine: <i>None</i> NumbersOnly: <i>None</i> ReadOnly: None Text: <i>None</i>	
6	<i>User Interface</i>	Button Name: <i>SaveBtn</i>	SaveBtn (Button) Appearance BackgroundColor: <i>Pink</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default...</i> Height: Automatic... Width: Automatic... Image: <i>None...</i> Shape: <i>default</i> ▼ ShowFeedback: <i>None</i> Text: <i>Save</i> TextAlignment: <i>center : 1</i> TextColor: <i>Default</i> Visible: <i>True</i> Behavior Enabled: <i>True</i>	Tombol ini digunakan untuk menyimpan data Negara, Ibukota ke TinyWebDB
7	<i>User Interface</i>	Button Name: <i>ClearBtn</i>	SaveBtn (Button) Appearance BackgroundColor: <i>Pink</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default...</i> Height: Automatic... Width: Automatic... Image: <i>None...</i> Shape: <i>default</i> ▼ ShowFeedback: <i>None</i>	Tombol ini digunakan untuk membersihkan text yang ada kotak textbox Negara, Ibukota,

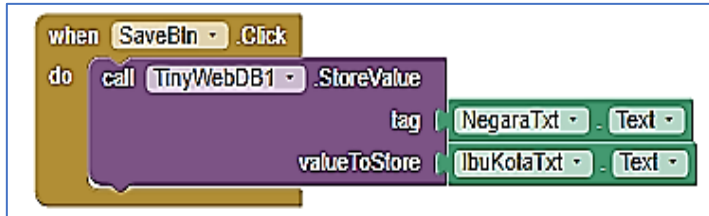
No	Pallette	Component	Properties	Keterangan
			Text: <i>Clear</i> TextAlignment: <i>center : 1</i> TextColor: <i>Default</i> Visible: <i>True</i> Behavior Enabled: <i>True</i>	
8	<i>Layout</i>	HorizontalArrangement Name: HorizontalArrangement1	HorizontalArrangement 1 (HorizontalArrangement) Appearance AlignHorizontal: Left: 1 AlignVertical: Top: 1 BackgroundColor: <i>Default</i> Height: <i>Automatic...</i> Width: <i>Fill parent...</i> Image: <i>None</i> Visible: <i>True</i>	Digunakan untuk mengelompokkan komponen dari kiri kekanan , didalamnya terdapat komponen Button ReadBtn, TextBox Readtxt
9	<i>User Interface</i>	Button Name: <i>ReadBtn</i>	ReadBtn (Button) Appearance BackgroundColor: <i>Pink</i> FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default...</i> Height: <i>Automatic...</i> Width: <i>Automatic...</i> Image: <i>None...</i> Shape: <i>default ▼</i> ShowFeedback: <i>None</i> Text: <i>Read</i> TextAlignment: <i>center : 1</i> TextColor: <i>Default</i> Visible: <i>True</i> Behavior Enabled: <i>True</i>	Tombol ini digunakan untuk mencari data berdasarkan Negara, akan ditampilkan Notifikasi Negara dan Ibukota jika data ditemukan
10	<i>User Interface</i>	TextBox3 Name:	ReadTxt (TextBox)	Digunakan untuk input

No	Pallete	Component	Properties	Keterangan
		<i>ReadTxt</i>	<p>Appearance BackgroundColor: Default FontBold: <i>None</i> FontItalic: <i>None</i> FontSize: 14.0 FontTypeface: <i>default...</i> Height: <i>Automatic...</i> Width: <i>Automatic...</i> Hint: Ketik Nama Negara HintColor: <i>Default</i> TextAlignment: <i>left : 0</i> ▾ TextColor: <i>Default</i> Visible: <i>True</i></p> <p>Behavior Enabled: <i>True</i> MultiLine: <i>None</i> NumbersOnly: <i>None</i> ReadOnly: <i>None</i> Text: <i>None</i></p>	pencarian berdasarkan Negara,
11	<i>Storage</i>	TinyWebDB Name: <i>TiniWebDB1</i>	<p>TinyWebDB1 (TinyWebDB)</p> <p>Behavior ServiceURL: http://tinywebdb.appinventor.mit.edu/</p>	Digunakan untuk mengkoneksikan ke web service default
12	<i>User Interface</i>	Notifier Name: <i>Notifier1</i>	<p>Notifier1 (Notifier)</p> <p>Appearance BackgroundColor: Default NotifierLength: <i>Long</i> ▾ TextColor: <i>Default</i></p>	Digunakan untuk menampilkan Notifikasi Pesan ketika data berhasil di simpan dan notifikasi Data ditemukan ketika pencarian /read

Langkah 4: Perintah Block Screen Aplikasi

Berikut ini perintah block untuk menyusun kode berupa puzzle pada screen Aplikasi

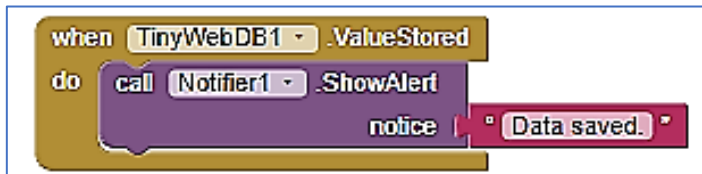
1. Event When SaveBtn.Click



Penjelasan:

Event when saveBtn.click digunakan untuk menyimpan data Negara, Ibu Kota ke database web service tinywebDB1 setelah pengisian Negara, Ibu kota dilakukan lalu meng klik tombol Save, selanjutnya menampilkan notifikasi "Data Sudah ditambahkan ke tinyDB.

2. Event When TinyWebDB1.ValueStored



Penjelasan:

Event when TinyWebDB1.ValueStored digunakan untuk menampilkan notifikasi "Data saved" ketika data berhasil di simpan ke tinyWebDB setelah mengklik tombol *Save*.

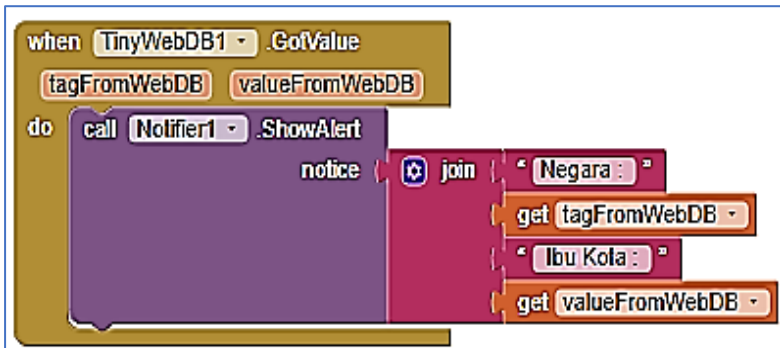
3. Event When ReadBtn.click



Penjelasan:

Event when ReadBtn.Click digunakan untuk membaca data negara yang tersimpan pada tinyWebDB setelah mengisi data negara yang diinginkan.

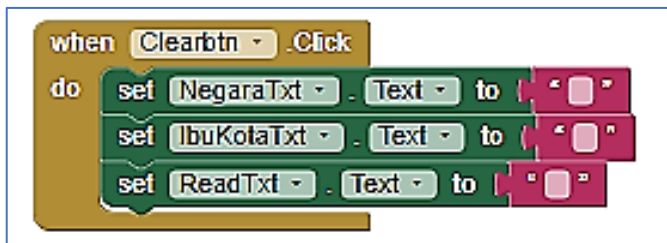
4. Event TinyWebDB1.GotValue



Penjelasan:

Event When TinyWebDB1.GotValue digunakan untuk mengambil data Negara, Ibu Kota dari database TinyWebDBdata1, yang ditemukan setelah pencarian/Read dan ditampilkan pada Notifikasi.

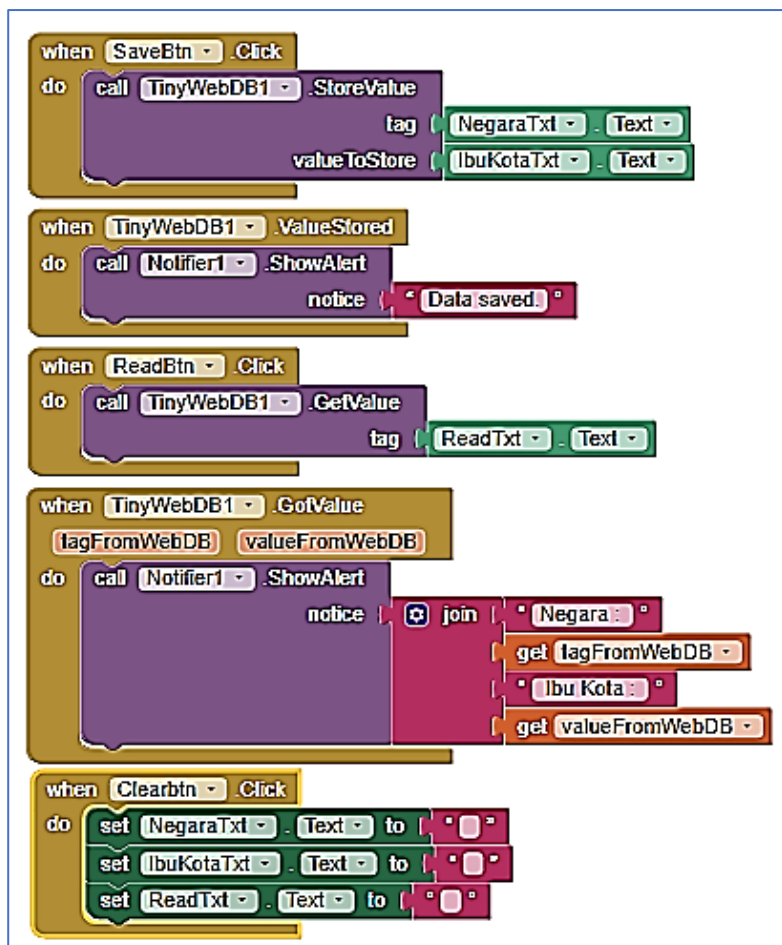
5. Event When Clearbtn.Click



Penjelasan:

Event When Clearbtn.click digunakan untuk membersihkan kotak textbox NegaraTxt, IbuKotaTxt, ReadTxt, dan siap untuk pengisian data baru

6. Tampilan pada jendela *Block* Keseluruhan



Langkah 5: Uji coba Aplikasi dengan Companion

Uji coba menggunakan aplikasi companion, langkahnya sama seperti pada ujicoba aplikasi pada halaman sebelumnya.

Berikut ini gambar hasil ujicoba ke perangkat gadget android.

1. Tampilan Screen Awal hasil ujicoba

08:43 80%

TinyWebBD simple

Negara

Ibu Kota

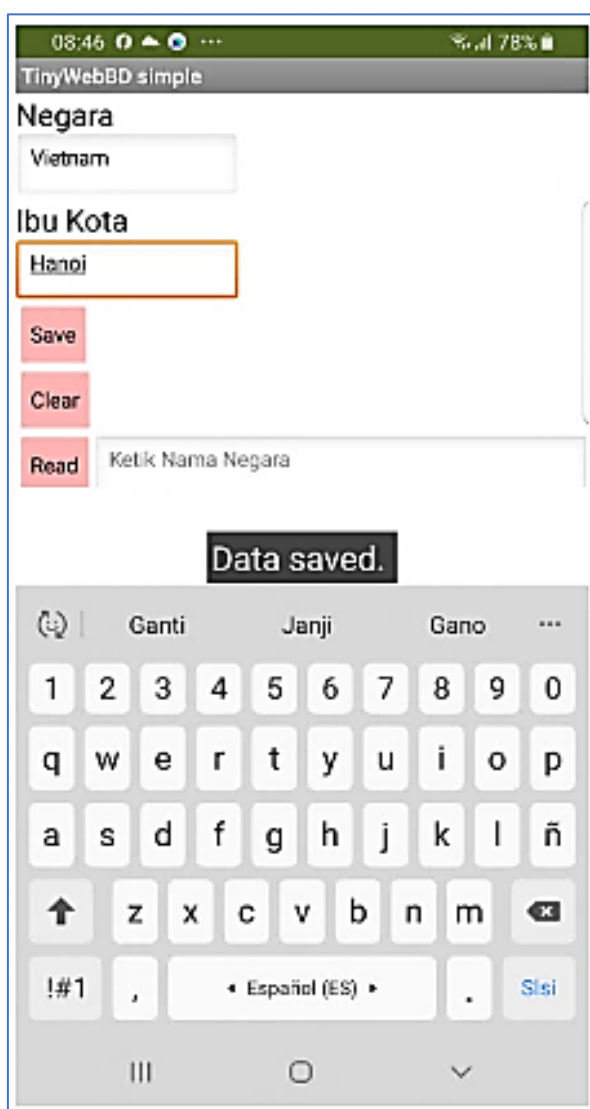
Save

Clear

Read

Ketik Nama Negara

2. Tampilan Screen Menambah data dan klik tombol *Save*



3. Tampilan Screen membaca data dan klik tombol *Read*



DAFTAR PUSTAKA

- Abeywardena, Ishan Sudeera, (2015). EDUCATIONAL APP DEVELOPMENT TOOLKIT FOR TEACHERS AND LEARNERS, Commonwealth of Learning, Burnaby, British Columbia Canada V5H 4M2.
- Agus Suharto. Mudah Membuat Aplikasi android (2021). Penerbit Adab CV. Adanu Abimata. Indramayu Jawabarat.
- Derek Walter, Mark Sherman. *Learning MIT App Inventor: A Hands-On Guide to Building Your Own Android Apps* (2015). Addison Wesley. United States Crawfordsville Indiana.
- Evan W. Patton, Michael Tissenbaum and Farzeen Harunani. *MIT App Inventor: Objectives, Design, and Development* (2019). Massachusetts Institute of Technology, Cambridge, MA, USA.
- <http://appinventor.mit.edu/explore/ai2/tutorials>, diakses pada 1 Juli 2020
- <http://appinventor.mit.edu/explore/library>, diakses pada 20 agustus 2020
- Juan Antonio Villalpando, Programming mobile phones with Android by App Inventor 2, diakses pada 28 Februari 2024. http://kio4.com/appinventori/8file_database.htm

TENTANG PENULIS



Ir. Agus Suharto, M.Kom.

Saat ini aktif mengajar sebagai Dosen Universitas Indonesia program vokasi sejak tahun 2004, dan Dosen tetap Universitas Pamulang program studi Sistem Informasi sejak tahun 2016, Aktif membuat buku rumpun ilmu Komputer, kompetensi mata kuliah mengampu Database, Pemrograman, Rekayasa Perangkat Lunak, Selain itu sebagai praktisi tenaga ahli di Organisasi dan Konsultan IT.



Eko Suharyanto, S.T., M.Kom.

Dilahirkan di Jakarta, 18 April 1964. Merupakan anak pertama dari 12 bersaudara dari bapak Suchaemi (alm) dan ibu Harmani Djoharijah (almh), Lulus S1 Teknik Informatika Fakultas Teknik Universitas Pancasila Jakarta Tahun 2009, lulus S2 Teknik Informatika STMIK Eresha Jakarta tahun 2016, Saat ini adalah Dosen Tetap Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Pamulang Tangerang Selatan, Mengampu Matakuliah Kalkulus I, Kalkulus II, Aljabar Linier, Matematika Diskrit. Selain itu penulis juga pernah bekerja sebagai tenaga kependidikan di Fakultas Ekonomi dan Bisnis Universitas Pancasila dari tahun 1992 sampai 2022 dengan Jabatan terakhir Kepala Bagian Akademik serta pernah menjabat sebagai Ketua Koperasi Dosen dan Karyawan Fakultas Ekonomi dan Bisnis Universitas Pancasila.