

5

Pengembangan Aplikasi Mobile dengan *Library* Kecerdasan Artifisial



Tujuan Pembelajaran

Setelah mempelajari bab ini kalian diharapkan mampu mengembangkan sebuah aplikasi berbasis mobile yang merupakan implementasi kecerdasan buatan dengan langkah-langkah: analisis, identifikasi persoalan, perancangan, implementasi, pengujian, dan penyempurnaan. Selanjutnya kalian juga diharapkan mampu untuk mengomunikasikan produk aplikasi, dan manfaatnya secara lisan maupun tertulis.



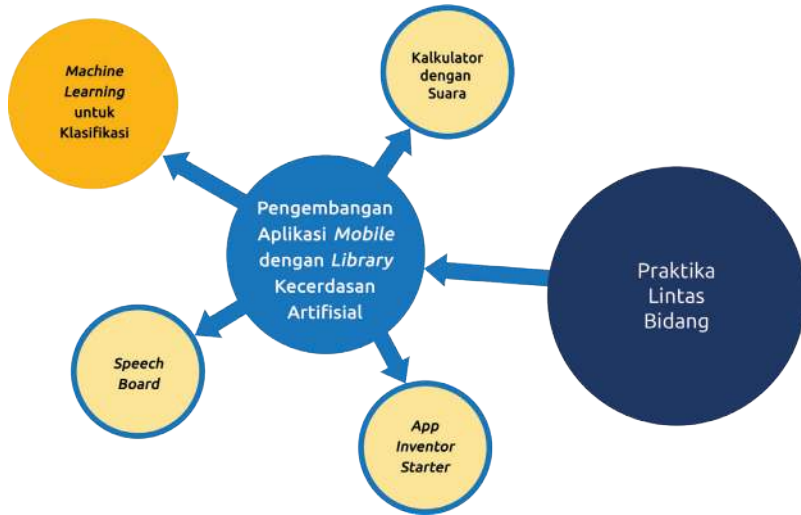
Pertanyaan Pemantik

Hakikatnya kita sebagai manusia memiliki akal untuk berpikir dalam menyelesaikan masalah dan dapat terus meningkatkan kecerdasan

kita, lalu bagaimana dengan sebuah mesin buatan manusia? Dapatkah mesin tersebut berpikir dan terus meningkatkan kecerdasan berpikirnya seperti manusia?



Peta Konsep



▲ Gambar 5.1 Peta Konsep Pengembangan Aplikasi Mobile dengan Library Kecerdasan Artifisial



Apersepsi

Saat ini mungkin sebagian dari kalian telah terbiasa menggunakan gawai (*mobile phone*) berupa ponsel, tablet, atau yang lain. Mungkin sebagian dari kalian juga telah terbiasa menggunakan aplikasi *mobile*. **Aplikasi *mobile*** adalah salah satu bentuk artefak komputasional yang bermanfaat bagi kehidupan masyarakat di era digital saat ini. Aplikasi ini sama dengan aplikasi lain yang telah kalian kembangkan pada jenjang sebelumnya namun dirancang untuk dapat berjalan pada ponsel (*mobile phone*). Pengembangan aplikasi *mobile* tidaklah sesulit yang dibayangkan. Pada bab ini kalian akan mempelajari bagaimana cara mengembangkan aplikasi *mobile*, dan dilanjutkan dengan penggunaan *library* atau komponen kecerdasan artifisial. **Library** adalah modul program dengan fungsi tertentu yang sudah dikemas sehingga siap dipakai

tanpa pemrogram pemakainya perlu mengimplementasi kodenya. Memakai *library* ini dapat diibaratkan kalian menggunakan ponsel atau komputer dengan mudah dan nyaman tanpa perlu tahu betapa rumit isi di dalamnya.



Kata Kunci

Aplikasi *mobile*, Kecerdasan Artifisial, *Machine Learning*, Klasifikasi.

A. Pengembangan Aplikasi Mobile dengan App Inventor

Saat ini kehidupan sehari-hari manusia banyak dibantu oleh aplikasi atau perangkat lunak yang terpasang pada ponsel pintar, komputer, atau diakses secara *online* lewat peramban. Aplikasi tersebut diantaranya adalah aplikasi perkantoran, aplikasi bertukar pesan, pemutar lagu, aplikasi desain, pengolah akuntansi, dll. Aplikasi dapat dibedakan berdasarkan platform pengembangan dan penggunaannya, yaitu aplikasi *desktop*, aplikasi *web*, dan aplikasi *mobile*. Penulisan aplikasi sering disingkat menjadi *apps*.

1. Desktop Apps

Aplikasi *desktop* adalah aplikasi yang dikembangkan dengan tujuan implementasi pada komputer *desktop* atau piranti lokal komputer. Aplikasi ini harus dipasang pada piranti lokal komputer, dan setelah terpasang aplikasi ini akan berada pada memori dari piranti lokal.

2. Web Apps

Aplikasi berbasis web adalah aplikasi yang dikembangkan dengan tujuan dapat diakses menggunakan koneksi jaringan komputer dan internet menggunakan protokol http. Aplikasi ini tidak terpasang pada piranti atau komputer *desktop* lokal, namun terpasang pada *server* tertentu. Aplikasi ini kebanyakan diakses menggunakan *browser*, namun ada juga

yang berbentuk *client side* dimana ada program kecil yang terpasang pada piranti lokal, tetapi proses komputasi utama dilakukan di *server*.

3. Mobile Apps

Aplikasi mobile juga disebut dengan mobile apps adalah aplikasi yang dirancang untuk dapat dieksekusi pada piranti *mobile* seperti ponsel, tablet, atau *smart watch*. *Mobile* memiliki arti mudah bergerak. Aplikasi *mobile* dapat dipasang pada ponsel, tablet, atau gadget lainnya. Aplikasi ini berkembang pesat karena kemudahan penggunaan piranti *mobile*, dan dapat diintegrasikan dengan sistem lain yang ada pada piranti mobile seperti GPS, kamera, sidik jari, dll. Saat ini tersedia jutaan aplikasi *mobile* yang ada di platform pendistribusian aplikasi digital yaitu *Play Store* ataupun *Apps Store*.

Pengembangan aplikasi banyak dibantu oleh perkakas pengembangan yang disebut dengan *Integrated Development Environment* (IDE). IDE membantu kemudahan dan efektifitas pengembangan perangkat lunak. Salah satu IDE yang digunakan untuk pengembangan aplikasi mobile adalah [App Inventor](#).

4. App Inventor

App Inventor adalah perangkat lunak IDE terintegrasi yang berbasis *web*. App Inventor pada awalnya dikembangkan oleh Google, yang saat ini dipelihara oleh Massachusetts Institute of Technology (MIT). App Inventor memungkinkan pemrogram komputer pemula dapat membuat *mobile apps* diatas OS Android maupun iOS. Aplikasi App Inventor bersifat *open source* dan *free*.

App Inventor memiliki antarmuka berbasis grafis dan memiliki tampilan yang mirip dengan bahasa pemrograman block Scratch/Blockly yang telah kalian pelajari di kelas 7 dan 8. Dengan App Inventor kalian dapat membuat program dengan cara seret lepas (*drag and drop*) komponen-komponennya. App

Inventor sebagai perkakas, terus dikembangkan kecanggihannya melalui riset intensif di bidang *educational computing*. App Inventor mendukung penggunaan *cloud data* dengan *Firestore* dan *Firestore Realtime Database*.

App Inventor dapat diakses melalui peramban dari situs <https://ai2.appinventor.mit.edu>. Tampilan awal App Inventor tampak pada gambar 5.2 berikut:



▲ Gambar 5.2 Tampilan Awal App Inventor

App Inventor memiliki banyak komponen yang dapat digunakan dalam pembuatan aplikasi. Komponen tersebut dikelompokkan dalam *User Interface Components*, *Layout Components*, *Media Components*, *Drawing and Animation Components*, dll. *User Interface Components* memiliki komponen-komponen yang berhubungan dengan antarmuka pengguna, seperti: *Button* (Tombol), *CheckBox*, *DatePicker*, *Image*, dll. Masing-masing komponen memiliki *methods*, *events*, dan *properties* yang digunakan untuk memanipulasi komponen tersebut.

Properties pada komponen adalah atribut yang mendeskripsikan sifat dari komponen, misalnya lebar tombol, warna dari teks, dll. *Properties* biasanya dapat dibaca dan di set, namun ada juga *properties* yang hanya bisa dibaca.

Methods adalah fungsi yang dapat dikenakan pada komponen yang memilikinya, *methods* dapat digunakan untuk mengatur *properties*.

Events adalah kejadian yang terjadi karena pemanggilan *methods*, seperti aksi *mouse click* yang menghasilkan *mouse event* yang menyebabkan suatu fungsi/prosedur akan dieksekusi.

Sebagai contoh, komponen *Button* (Tombol) yang memiliki kemampuan untuk mendeteksi penekanan tombol, memiliki *properties*: warna *background* tombol yang dapat diubah sesuai keinginan, tombol dapat di set *enabled* (aktif) atau tidak aktif, dan *font* yang dapat diset *italic*, *bold*, dll. *Properties* tersebut dapat dimanipulasi pada *Designer Editor* atau pada *Blocks Editor*. Dalam **Tabel 5.1.** berikut memberikan berbagai contoh komponen, *methods*, *event*, dan *properties* dari komponen pada *App Inventor*.

▼ **Tabel 5.1.** Komponen, Properties, Events, dan Methods pada *App Inventor*

Komponen	Properties	Events	Methods
User Interface Components			
Button (Tombol): yang memiliki kemampuan mendeteksi penekanan (klik) dari user	BackgroundColor(Color)	Click()	(tidak ada)
	Enabled(Boolean)	GotFocus()	
	FontSize(Number)	LongClick()	
	FontBold(Boolean)	LostFocus()	
	FontItalic(Boolean)	TouchDown()	
	Image(Text)		
User Interface Components			
TextBox: Kotak tempat user mengisi teks	Text(Text)	GotFocus()	HideKeyboard()
	BackgroundColor(Color)	LostFocus()	RequestFocus()
	Enabled(Boolean)		
	FontSize(Number)		
	FontBold(Boolean)		
	FontItalic(Boolean)		
	Multiline(Boolean)		

Komponen	Properties	Events	Methods
Label: komponen untuk menampilkan teks	Text(Text)	(tidak ada)	(tidak ada)
	BackgroundColor(Color)		
	Enabled(Boolean)		
	FontSize(Number)		
	FontBold(Boolean)		
	FontItalic(Boolean)		
	TextColor(Color)		
Layout Components			
Horizontal Arrangement: Komponen ini memformat elemen yang diletakkan pada layar akan tertampil secara horizontal dari kiri ke kanan	AlignHorizontal(Number)	(tidak ada)	(tidak ada)
	AlignVertical (Number)		
	BackgroundColor(Color)		
	Image(Text)		
Media Components			
TextToSpeech: Komponen yang akan mengubah teks menjadi suara	Language(Text)		
	Country(Text)		
	SpeechRate(Number)		



Ayo Berlatih!

Aktivitas Individu

Aktivitas PLB-AI-K11-01: Starter App Inventor – Halo Dunia, dengan Text to Speech

Melalui aktivitas ini kalian akan mulai belajar cara pembuatan aplikasi mobile dengan App Inventor yang dapat dijalankan di ponsel Android. Aplikasi ini mampu mengubah teks yang dituliskan pada aplikasi ponsel menjadi suara yang terdengar melalui pengeras suara di ponsel.

Persiapan:

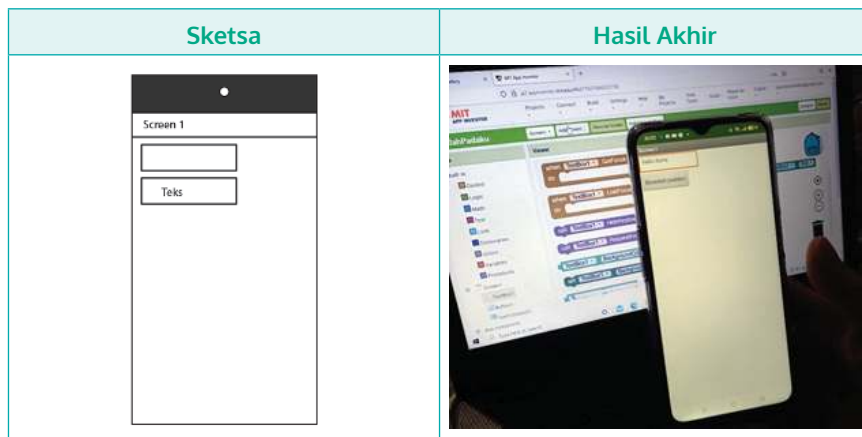
Pada aktivitas ini kalian memerlukan komputer yang terkoneksi dengan internet dan terpasang perangkat lunak **MIT AI2 Companion**, ponsel atau tablet dengan sistem operasi Android/iOS. Namun jika ponsel tidak tersedia maka kalian dapat menggunakan **emulator** ponsel yang akan muncul pada layar komputer kalian. Kalian harus melakukan pengaturan khusus untuk emulator ini.

Prasyarat:

Kalian sebagai peserta didik harus telah memahami pemrograman dengan Scratch/Blockly yang dipelajari di SMP.

Deskripsi Produk :

Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:



Spesifikasi Aplikasi:

- **Input:** Pengguna mengetikkan “Halo Dunia” pada *textbox* dan mengetuk tombol di bawah *textbox*
- **Proses:** Aplikasi mengubah teks yang ditulis pada program menjadi suara
- **Output:** Aplikasi akan memperdengarkan suara lewat speaker ponsel

Langkah-langkah:

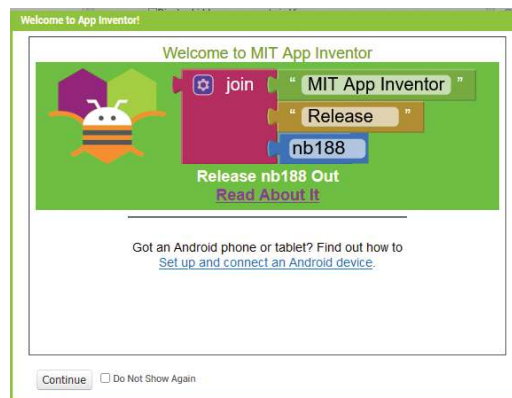
1. Persiapan:

- a. Masuk/*login* ke situs App Inventor (<https://appinventor.mit.edu/>) dengan menggunakan akun google kalian.

*jika belum memiliki akun google, kalian dapat menggunakan milik orang tua, kakak atau minta bantuan gurumu untuk mendaftarkannya.



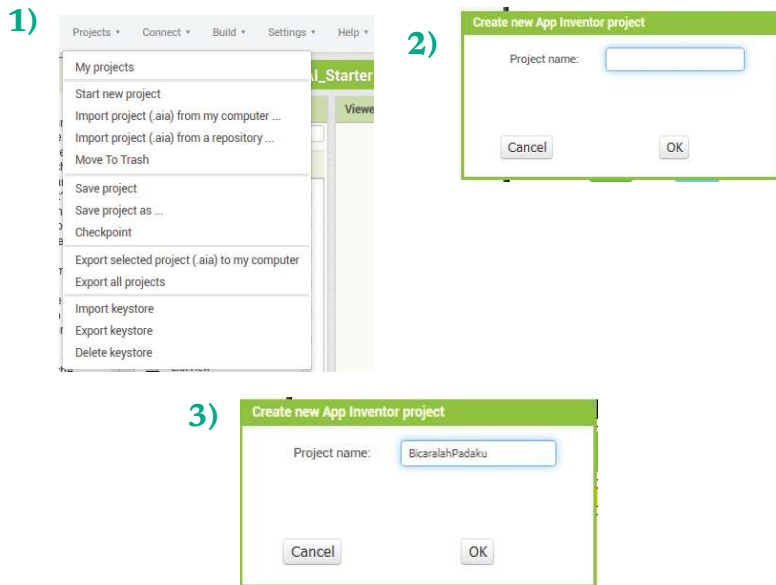
- b. Klik *Continue* saat layar pembuka (*splash screen*) muncul.



2. Pengkodean:

- a. Buat proyek baru dengan memilih menu **Projects** lalu klik **Start new project**. Lalu akan muncul *menu pop-up* untuk mengisi nama proyek, kalian beri nama proyek baru tersebut dengan “BicaralahPadaku” (tanpa spasi). Perlu diketahui setiap kalian membuat aplikasi di App Inventor, aplikasi tersebut disimpan dalam sebuah proyek yang berisi semua

file terkompilasi ke dalam sebuah *executable file*. *File* tersebut dapat berisi kode sumber, ikon, gambar, suara, *file data*, dsb.



b. Perancangan *User Interface* (UI):

Saat mengembangkan aplikasi mobile kalian, pengembangan dilakukan dengan merancang *User Interface* (UI) dan perancangan blok kode.

Perancangan UI dilakukan dengan menggunakan tampilan Designer, yang tampil dengan mengklik tombol *designer* pada bagian kanan atas. Tampilan Designer App Inventor memiliki empat kolom. Kolom *Palette* merupakan tempat komponen-komponen yang tersedia dari App Inventor, kolom *Viewer* merupakan kolom untuk perancangan UI aplikasi, kolom *Components* berisi komponen-komponen yang digunakan pada proyek, dan kolom *Properties* yang merupakan kolom untuk melakukan pengaturan terhadap komponen-komponen yang digunakan.

Perancangan blok kode dilakukan dari Tampilan Blocks yang tampil dengan mengklik tombol Blocks pada

bagian kanan atas disamping tombol Designer yang akan dijelaskan pada kemudian.

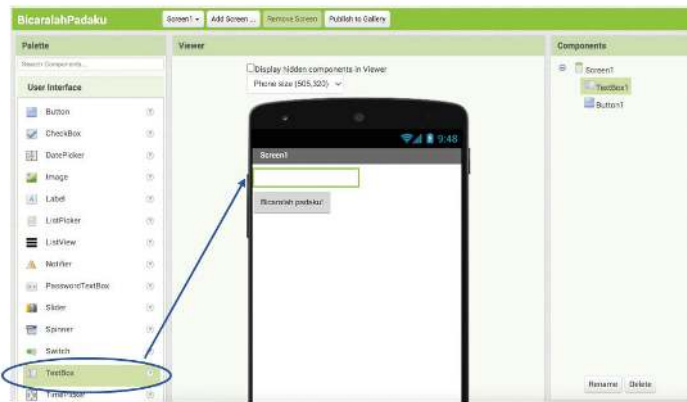


Langkah perancangan UI:

- 1) Tambahkan tombol Button pada Viewer, dengan seret dan lepaskan (*drag and drop*) dari kolom Palette ke kolom Viewer. Sebuah Button dengan nama Button1 akan tercipta. Nama Button1 dapat diganti dengan nama lain yang sesuai, namun pada latihan ini kalian masih menggunakan nama default.



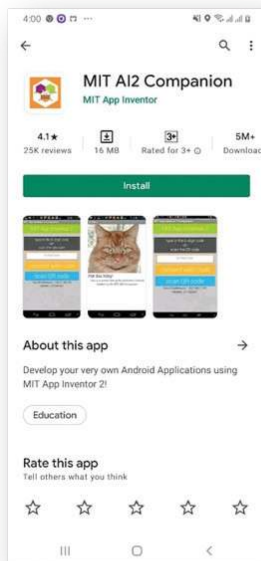
- 2) Tambahkan TextBox dengan *drag and drop* ke area Viewer. Secara otomatis TextBox dengan nama default TextBox1 akan muncul.



3. Persiapan Pengujian

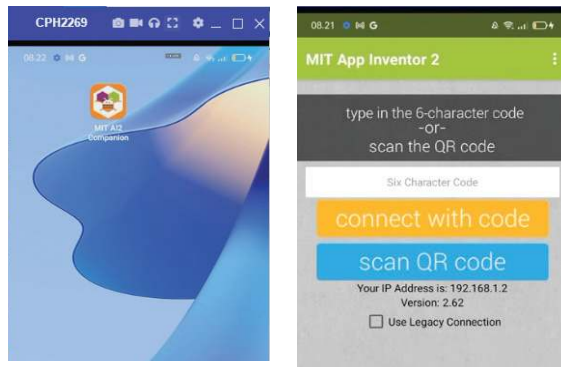
Aplikasi yang dikembangkan adalah aplikasi mobile yang berjalan pada piranti *mobile phone*, sehingga pengujian idealnya dilakukan dengan menguji dengan ponsel secara *live*, langkah persiapan pengujian dilakukan dengan langkah:

- a. Sambungkan App Inventor pada komputer dengan *smartphone* kalian dengan menyambungkannya melalui kabel USB atau menggunakan perangkat wifi.
- b. Unduh dan pasang/*install* MIT AI2 Companion di PlayStore/App Store pada ponsel kalian.

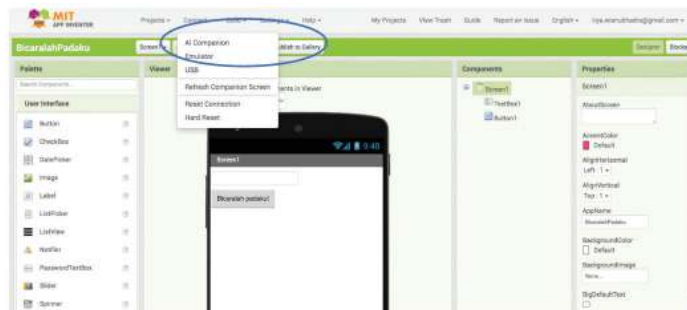


- c. Pengunduhan dan pemasangan AI Companion memerlukan pengaturan (*setting*) pada ponsel. Lakukan centang (*check*) untuk membolehkan “*Unknown Sources*” pada menu “*Security*” pada ponsel. Selanjutnya, Scan QR code untuk mengunduh langsung MIT AI2 Companion atau klik link “Need help finding the Companion App?”. Setelah selesai diunduh, pasang/*install* aplikasi MIT AI2 Companion.

Berikut tampilan App Inventor, jika sukses pemasangannya pada ponsel:



Berikut ini tampilan App Inventor pada komputer/laptop. Untuk menghubungkan aplikasi App Inventor pada ponsel dengan App Inventor yang digunakan pada komputer/laptop. Caranya dengan pilih menu Connect pada App Inventor komputer/laptop kalian, lalu klik menu AI Companion.

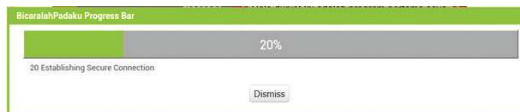


Berikut ini tampilan menu *pop-up* AI Companion pada komputer/laptop, akan muncul dalam *pop-up* kode 6 digit dan QR Code yang dapat digunakan untuk menghubungkan

kedua perangkat. Kalian dapat mengisikan kode 6 digit atau QR code yang dapat dipindai menggunakan ponsel untuk menghubungkan kedua perangkat tersebut.

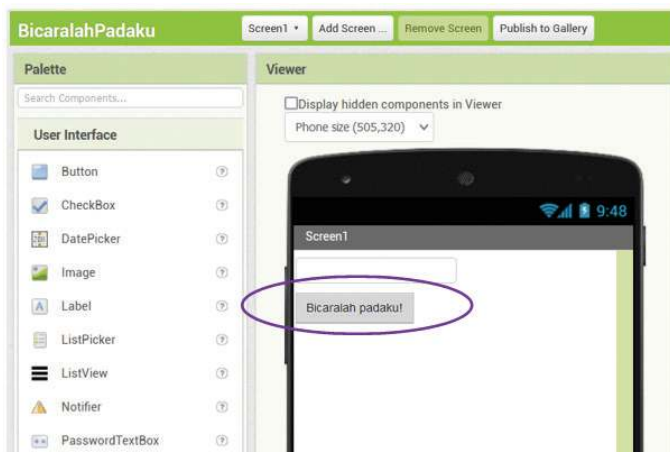


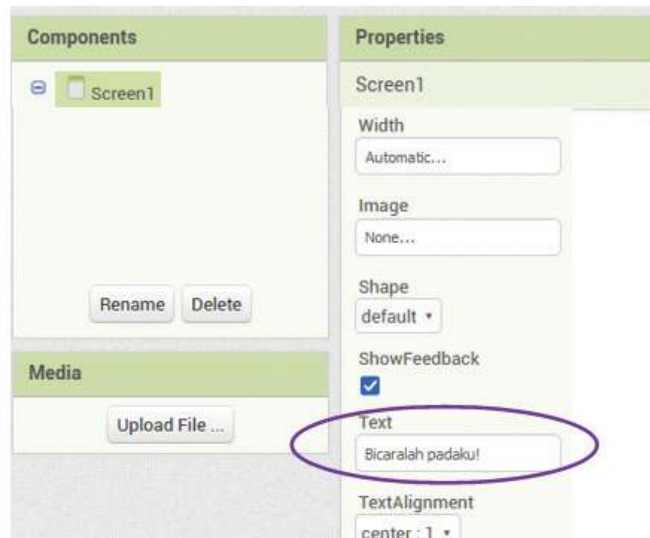
Selanjutnya akan muncul Progress Bar dalam komputer/ laptop kalian sebagai tanda proses menghubungkan kedua perangkat. Setelah itu, jika instalasi sukses maka kalian dapat melihat app kalian di ponsel. Jika kalian menambahkan komponen lain pada app kalian di komputer, maka perubahan akan terjadi juga di ponsel. Selamat, ponsel kalian siap digunakan untuk pengujian aplikasi.



4. Lanjutan Pengkodean:

- a. Masih pada Designer view, ubahlah teks pada TextBox1, menjadi “Bicaralah padaku!” pada kolom properties.





- b. Tambahkan komponen TextToSpeech pada Viewer, dengan cara drag and drop komponen tersebut ke kotak Viewer. Pilih komponen dari menu Pallette > Media > TextToSpeech.
- c. Pengkodean Blok: Setelah komponen TextToSpeech ditambahkan, selanjutnya kalian lanjutkan pengkodean blok dengan masuk ke mode editor Blocks. Beralih ke mode editor Blocks dilakukan dengan menekan tombol Blocks pada pojok kanan halaman.



Editor Blocks adalah tempat untuk menyusun program dari aplikasi. Pada editor ini terdapat blok Built-in yang telah tersedia dan dapat digunakan untuk menangani operasi Control, Logic, Math, dll. Blocks merupakan ruang kerja yang digunakan untuk menyusun program aplikasi.



Pengkodean Blok selanjutnya dilakukan dengan langkah berikut ini:

- 1) **Buatlah *event* ketika tombol Button1 ditekan, dengan cara:**
 - a) Klik tombol Button1 pada kolom Blocks
 - b) Pilih blok when Button1.Click pada Viewer
 - c) *Drag and drop* pada Viewer yang kosong, yang hasilnya tampak seperti pada gambar berikut:



- 2) **Tambahkan blok call TextToSpeech1.Speak dari komponen TextToSpeech1 ke blok when Button1.Click dengan cara:**

- a) Klik TextToSpeech1 pada kolom Blocks
- b) Pilih blok call TextToSpeech1.Speak di kolom Viewer
- c) *Drag and drop* blok call TextToSpeech1.Speak, pada kolom Viewer yang kosong, yang hasilnya tampak seperti pada gambar berikut:



3) Tambahkan blok TextBox1.Text dari komponen TextBox1 ke blok call TextToSpeech1.Speak dengan cara:

- a) Klik TextBox1 pada kolom Blocks
- b) Pilih blok call TextBox1.Text di kolom Viewer
- c) *Drag and drop* blok TextBox1.Text ke blok call TextToSpeech1.Speak pada kolom Viewer, yang hasilnya tampak seperti pada gambar berikut:



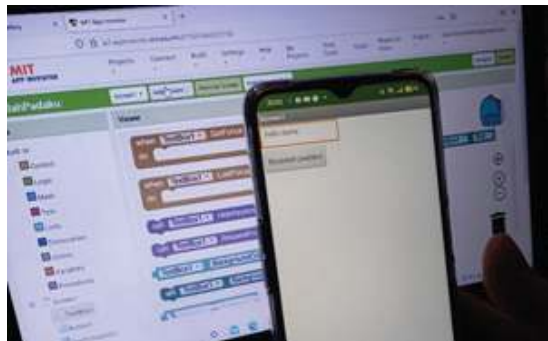
4) Simpan file proyek kalian dengan memilih menu Projects > Save Project

5. Pengujian Aplikasi:

Setelah selesai dengan langkah 4, langkah berikutnya adalah pengujian aplikasi. Pengujian aplikasi dilakukan dengan menuliskan teks “Halo Dunia” pada TextBox1, dan menyetuk tombol “Bicaralah padaku!” pada ponsel yang telah tersambung dengan komputer/laptop sebelumnya pada langkah 3. Jika speaker pada ponsel mengeluarkan suara Halo Dunia maka kalian telah berhasil membuat aplikasi mobile pertama kalian. Kalian saat ini menggunakan TextToSpeech, yang merupakan library App Inventor yang berfungsi mengubah teks menjadi suara, meniru kalian (manusia) membaca teks dan mengucapkannya! Kalian tinggal memakai, dan tidak perlu tahu betapa rumitnya program di dalamnya.

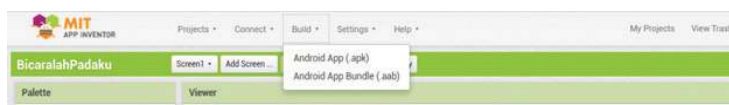
Setelah itu cobalah mengetikkan teks yang berbeda, atau dengan bahasa yang berbeda, misalnya bahasa Inggris atau Perancis, dan tekan tombol Bicaralah Padaku. Apa yang terjadi?

Berikut ini foto dari aplikasi BicaralahPadaku yang telah selesai dibuat.



Aplikasi yang telah kalian kembangkan dapat diunduh dan kalian bagikan ke teman dan orang tua, dengan cara:

- a. Pilih menu Build dan pilih Android App (.apk).





- b. Klik Download .apk now, dan file .apk akan terunduh. File .apk adalah file paket android (*Android Application Package*) yang digunakan untuk mendistribusikan aplikasi, file dapat digandakan dan dipasang pada piranti mobile kalian dan teman-teman kalian.



Ayo Kembangkan!

Aktivitas Kelompok

Aktivitas PLB-AI-K11-02: Speechboard

Pada aktivitas ini kalian akan belajar untuk mengembangkan aplikasi yang mampu memainkan sebuah rekaman pidato dengan menyentuh sebuah gambar.

Kebutuhan Alat dan Bahan:

1. Komputer yang terkoneksi dengan internet, ponsel/tablet dengan sistem operasi Android atau iOS. Komputer juga harus terpasang perangkat lunak MIT AI2 Companion. Jika ponsel tidak tersedia dapat digunakan emulator pada komputer.

2. Gambar proklamator pada saat mengucapkan proklamasi kemerdekaan Indonesia (*file type .jpg*) dan rekaman pidato pembacaan teks proklamasi pada tanggal 17 Agustus 1945 (*file type .mp3*). File dapat diunduh di https://static.buku.kemdikbud.go.id/content/media/rar/Informatika_XI.rar

Prasyarat:

1. Kalian harus telah memahami materi pemrograman menggunakan Scratch/Blockly yang dipelajari pada jenjang SMP.
2. Kalian memahami cara mengunduh file gambar dan suara dari internet, dan menyimpannya di komputer.

Deskripsi Produk :

Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:



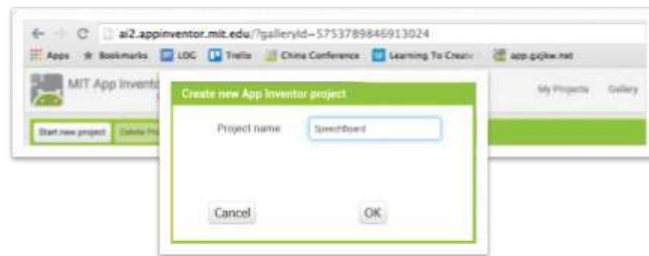
Spesifikasi Aplikasi:

- *Input*: Pengguna mengetuk gambar Sang Proklamator Indonesia di ponsel
- *Proses*: Aplikasi memainkan rekaman pidato proklamasi kemerdekaan Indonesia
- *Output*: Aplikasi memperdengarkan suara rekaman lewat speaker ponsel

Langkah langkah:

1. Persiapan:

- a. Buka aplikasi App Inventor dengan mengakses <https://ai2.appinventor.mit.edu>
- b. Mulailah membuat proyek baru, namailah proyek dengan nama “SpeechBoard”



2. Perancangan User Interface (UI):

- a. Berikut tampilan awal dari proyek,

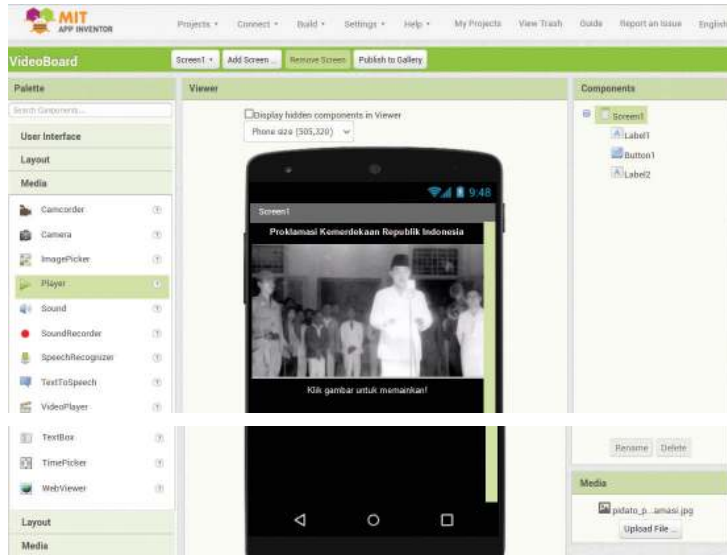


- b. Tambahkan Button pada layar dengan cara seret dan lepas (*drag and drop*)



- c. Ubahlah *background* Button pada *screen* dengan gambar Proklamasi Kemerdekaan RI, dengan sebelumnya mengunggah gambarnya (*file .png*).

- d. Tambahkan dua label dengan teks “Proklamasi Kemerdekaan Republik Indonesia” dan “Klik gambar untuk memainkan!” pada bagian atas dan bawah *button*. Seperti contoh berikut:



- e. Tambahkan komponen Player dari kolom Palette > Media, sehingga Player1 tercipta, dan upload file pidato kemerdekaan (*file .mp3*) sebagai Source dari Player1.





3. Pengkodean Blok:

- a. Berikutnya, kalian harus menambahkan blok kode dengan beralih ke mode Blocks.
- b. Tambahkan kode program untuk memainkan *file* pidato dengan blok when Button1.Click.
- c. Isi blok call Player1.Start ke dalam blok when Button1.Click.



- d. Dan terakhir jangan lupa untuk menyimpan proyek (save project) kalian.

4. Pengujian:

Ujilah kode program dengan mengetuk tombol pada ponsel untuk memainkan rekaman pidato, seperti pada langkah pengujian aktivitas PLB-AI-K11-01. Jika program telah berhasil memperdengarkan suara proklamasi, maka program kalian telah sesuai dengan spesifikasi, dan lanjutkan dengan aktivitas pengembangan.



Ayo Kembangkan!

Aktivitas Kelompok

Deskripsi Proyek:

Kalian diharapkan mengembangkan proyek perangkat lunak berbasis mobile yang dinamakan speechboard, dengan melakukan modifikasi aplikasi tersebut dengan menambahkan beberapa pidato dari para pahlawan Indonesia, misalnya Ki Hajar Dewantara, Sutomo, dll. Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:



Spesifikasi Aplikasi:

- **Input:** Pengguna mengetuk salah satu gambar pahlawan Indonesia di ponsel
- **Proses:** Aplikasi memainkan rekaman pidato yang terkenal dari para pahlawan tersebut
- **Output:** Aplikasi akan memperdengarkan rekaman pidato dari pahlawan yang dipilih lewat speaker ponsel

Gunakan Lembar Kegiatan Peserta Didik saat kalian mengembangkan aplikasi untuk berbagi peran dan tugas, berikut:

LKPD-01 Format Lembar Kegiatan Peserta Didik Pengembangan Aplikasi

Peran	Penanggung Jawab
Analisis Program: a. Deskripsi Produk b. Spesifikasi Aplikasi c. Kebutuhan resource: file, alat, dll	
Perancang User Interface (UI)	
Pemrogram Kode	
Penguji Program	
Pemapar Presentasi	

Spesifikasi (Deskripsi Produk, Fungsionalitas Aplikasi, Kebutuhan Resource)

Rancangan User Interface (UI)

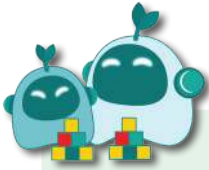
Kode Program

Pengujian

Diisi saat perencanaan			Diisi setelah Pengujian		
No.	Fitur	Dikerjakan Oleh	Sesuai dengan spesifikasi		Keterangan Hasil Pengujian
			YA	TIDAK	

▲ **Tabel 5.2** Format LKPD-01 Pengembangan Aplikasi Aktivitas PLB-AI-K11-02

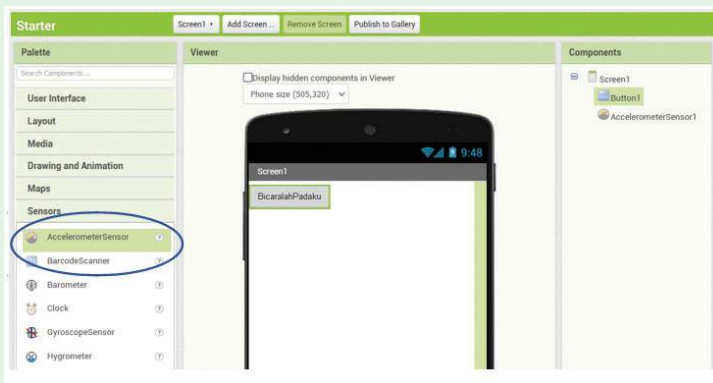
Selanjutnya, kembangkanlah aplikasi untuk dapat menghentikan suara (pause) ketika gambar diketuk, dan memperdengarkan suara kembali ketika diketuk lagi (toggle). Setelah itu, kembangkan juga dengan hanya boleh satu player yang hidup pada suatu saat tertentu. Jangan lupa setelah ditambah fungsionalitasnya ujilah kembali aplikasi kalian tersebut.



Pengayaan

Untuk meningkatkan pemahaman dan kemampuan kalian dalam materi Pengembangan Aplikasi Mobile ini, kalian dapat mengikuti aktivitas pengayaan berikut. Aplikasi yang telah kalian buat pada aktivitas PLB-AI-K11-01 dan PLB-AI-K11-02 dapat dikembangkan menjadi aplikasi lain, misalnya:

Aplikasi pada aktivitas PLB-AI-K11-01 dapat dikembangkan dengan memperdengarkan suara teks saat ponsel digerakkan naik turun. Aplikasi ini akan mengakses sensor accelerometer, sehingga teks akan disuarakan ketika ponsel digerakkan naik turun.



Mengembangkan aplikasi pemanggil nama peserta didik, sehingga guru dapat memanggil peserta didik dengan menekan tombol pada aplikasi di ponsel.



Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Setelah mengikuti pembelajaran ini, bagaimana pendapat kalian tentang pengembangan aplikasi mobile? Mudah bukan? Apakah kemudian kalian ingin belajar lebih banyak tentang pengembangan aplikasi mobile ini?
2. Kesulitan apa yang kalian temukan pada materi ini?
3. Dengan pemahaman saat ini, ide pengembangan apa yang kalian sarankan untuk membuat aplikasi yang lebih menarik?
4. Setelah kalian berhasil membuat aplikasi, apakah kalian merasa lebih familiar dengan teknologi? Apakah kalian ingin mengirimkan aplikasi kalian yang kalian buat ke orang tua atau teman kalian?

B. Pengembangan Aplikasi Kecerdasan Artifisial dengan App Inventor

Tahukah kalian saat ini teknologi kecerdasan artifisial telah banyak diimplementasikan pada kehidupan kita sehari-hari. Kalian mungkin pernah mendengar Google Assistant, Apple Siri, Amazon Alexia yang merupakan aplikasi asisten pribadi yang dapat melakukan pekerjaan tertentu dengan perintah menggunakan suara. Saat ini banyak perusahaan di Indonesia yang menggunakan *chatbot* untuk berinteraksi dengan konsumen secara otomatis, atau ketika kalian menggunakan youtube maka akan muncul video rekomendasi yang sesuai dengan kesukaan kalian.

Nama-nama produk diatas adalah contoh-contoh produk hasil dari kecerdasan artifisial, dan masih banyak contoh lain yang digunakan di industri dalam bentuk robot otomasi industri, robot penjelajah ruang angkasa, dll.

Kecerdasan Artifisial

Kecerdasan artifisial atau *Artificial Intelligence* (AI) adalah kecerdasan yang dimiliki oleh sistem atau mesin atau komputer. AI mampu untuk melakukan tugas yang umumnya terkait dengan kemampuan makhluk cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang memiliki sifat intelektualitas manusia, seperti kemampuan untuk menalar, menemukan makna, melakukan generalisasi, atau belajar dari pengalaman masa lalu. Sejak perkembangan komputer digital pada tahun 1940-an, AI telah banyak diimplementasikan untuk melakukan tugas yang kompleks seperti, misalnya, menemukan bukti untuk teorema matematika atau bermain catur dengan sangat mahir.

Namun, meskipun kemajuan terus-menerus dalam kecepatan pemrosesan komputer dan kapasitas memori, belum ada program yang dapat menandingi fleksibilitas manusia dalam domain yang lebih luas atau dalam tugas-tugas yang membutuhkan banyak pengetahuan sehari-hari. Di sisi lain, beberapa program telah mencapai tingkat kinerja yang sangat impresif yang dapat menggantikan para ahli dan profesional manusia dalam melakukan tugas-tugas tertentu tertentu, seperti diagnosis medis, mesin pencari komputer, dan pengenalan suara atau tulisan tangan.

Kecerdasan Artifisial kemudian berkembang dengan memunculkan berbagai subbidang yaitu:

- a. ***Machine Learning***: *Machine Learning* adalah mesin pembelajar yang mampu melakukan pembuatan model analitik secara otomatis. Mesin ini menggunakan beberapa metode berbasis statistik, jaringan saraf, fisika, dll untuk menemukan *insight* (wawasan) tersembunyi dari data tanpa diprogram secara eksplisit. Mesin ini mampu mengambil kesimpulan secara otomatis

- b. **Deep Learning:** *Deep Learning* adalah mesin pembelajar dengan pembelajaran mendalam menggunakan jaringan syaraf tiruan dengan ukuran dan lapisan unit pemrosesan yang besar. *Deep Learning* memanfaatkan kemajuan dalam kemampuan komputasi dari perangkat komputer yang semakin cepat dan *algoritma training* (pelatihan) yang terus meningkat kinerjanya untuk mengenali pola kompleks dalam data besar. Aplikasi umum *Deep Learning* yang banyak digunakan adalah pengenalan gambar dan suara. Pada beberapa literatur disebutkan bahwa *Deep Learning* adalah subset dari *Machine Learning*, dan *Machine Learning* adalah subset dari Kecerdasan Artifisial. **Gambar 5.3** berikut menunjukkan ilustrasi keterkaitan Kecerdasan Artifisial, *Machine Learning*, dan *Deep Learning*.



▲ Gambar 5.3. Kecerdasan Buatan (Artificial Intelligence)

Machine Learning adalah bagian dari bidang ilmu kecerdasan artifisial yang mempelajari cara membuat mesin atau sistem yang memiliki kecerdasan dan menyerupai manusia yang mampu belajar dan memecahkan masalah. Beberapa contoh dari *machine learning* adalah *search engine* pada peramban, (contoh: Google, Bing), sosial media yang memiliki kemampuan memberi saran kepada pengguna yang biasa disebut *recommendation system* (contoh: Youtube, Netflix, E-Commerce), mobil otonom

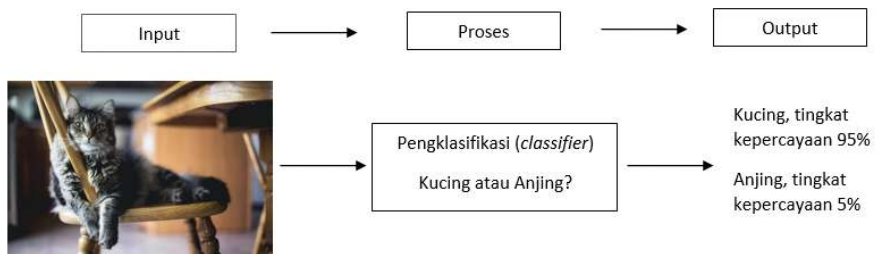
(contoh: Tesla), game dengan pengambilan keputusan otomatis (contoh: *strategic game*).

Machine learning diharapkan menjadi sistem yang mampu belajar terus menerus. Dengan semakin banyak data yang dipelajari, sistem akan menjadi semakin pintar.

Bagaimana cara kerja machine learning untuk klasifikasi gambar (*image*)?

Klasifikasi gambar adalah salah satu fitur penting pada *machine learning*, sebagai contoh pada mobil otonom yang dapat bergerak tanpa sopir, mobil ini harus mampu dengan cepat untuk menginterpretasikan dan mengklasifikasi sebuah objek yang dilihat dari kamera. Mobil harus menentukan apakah objek tersebut kendaraan lain, pejalan kaki (*pedestrian*) atau rambu lalu lintas. Hal ini sangat penting bagi mobil otonom karena sangat berpengaruh pada gerak jalan mobil.

Salah satu *library/extension* perkakas machine learning di MIT App Inventor yang mampu mengklasifikasikan gambar adalah LookExtension. Library ini dapat menerima gambar yang diambil dari kamera sebagai **input** dan mampu mengklasifikasi gambar input tersebut menjadi **output** yang disajikan dalam bentuk teks / tulisan. Contoh pada gambar 5.4 adalah klasifikasi sebuah gambar apakah kelas/kategorinya adalah kucing atau anjing.

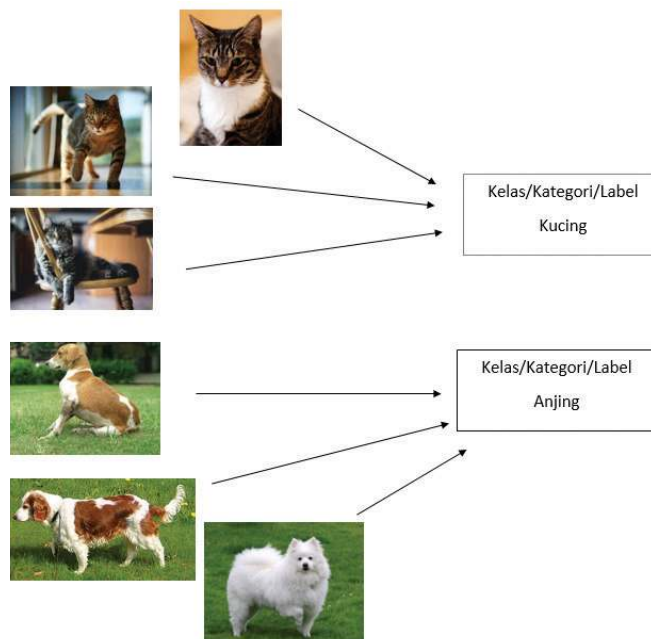


▲ Gambar 5.4. Ilustrasi Proses Klasifikasi Gambar

Untuk mendapatkan output klasifikasi berupa teks “Kucing (tingkat kepercayaan 95%)” atau “Anjing (tingkat kepercayaan 5%)” terjadi proses penghitungan/komputasi yang cukup

tinggi dan kompleks. Proses komputasi akan menghasilkan kesimpulan sebuah gambar adalah kucing atau anjing.

Agar dapat menghasilkan kesimpulan yang mampu mengklasifikasikan gambar, sistem pada awalnya **dilatih** untuk dapat mengklasifikasikan gambar (*image*) tertentu. Sebagai contoh, jika kalian ingin mengembangkan sistem untuk dapat mengenali dan mengklasifikasi gambar kucing atau anjing, maka kalian harus **melatih** sistem dengan memberikan banyak gambar kucing dan memberi nama kategori (label) kucing dan memberikan banyak gambar anjing serta memberi label atau kategori anjing juga agar sistem dapat mengingat dan mengenalinya. Memberikan kategori atau label sangat penting bagi sistem, sehingga ketika sistem diberikan input gambar baru, sistem akan dapat menentukan apakah gambar tersebut lebih mirip gambar kucing atau anjing.

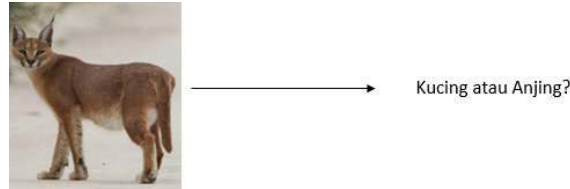


▲ Gambar 5.5. Gambar kucing dan anjing dan kelasnya

Dengan contoh gambar kucing dan anjing yang cukup, program akan terlatih untuk menentukan/mengklasifikasi suatu gambar adalah anjing atau kucing. Secara umum, semakin banyak gambar

yang dilatihkan untuk tiap tiap kelas, sistem akan menjadi semakin baik dan andal ketika mengklasifikasikan gambar baru.

Ketika sistem telah dilatih dengan gambar yang cukup, selanjutnya sistem dapat diuji dengan memberikan gambar baru yang tidak diberikan pada saat pelatihan (*training*).



▲ Gambar 5.6. Pengujian dengan gambar baru

Menurut kalian apa hasil klasifikasi dari gambar diatas? Jika hasil klasifikasinya keliru maka sistem *machine learning* kita dapat dilatih dan diuji kembali dengan gambar baru tadi, sama seperti manusia yang terus menerus belajar. Pelatihan/ pembelajaran yang terus menerus membuat sistem kita semakin pintar.

Tapi kita harus berhati-hati karena sistem kita hanya dirancang untuk hanya mampu mengklasifikasikan gambar yang telah kita latihkan, dalam hal ini adalah kucing atau anjing. Sebuah gambar yang sangat berbeda bisa jadi akan diklasifikasi sebagai kucing atau anjing. Sebagai contoh gambar kuda berikut, bisa saja terklasifikasi sebagai kucing atau anjing, yang merupakan klasifikasi yang salah. Jadi karena sistem pengklasifikasi kita hanya bisa membedakan dua kelas/kategori/label maka gambar berbeda akan diklasifikasi pada kedua kelas tersebut. Kelas dapat dikembangkan/ditambah untuk kelas/kategori yang lain dengan proses pelatihan dan pengujian kembali.

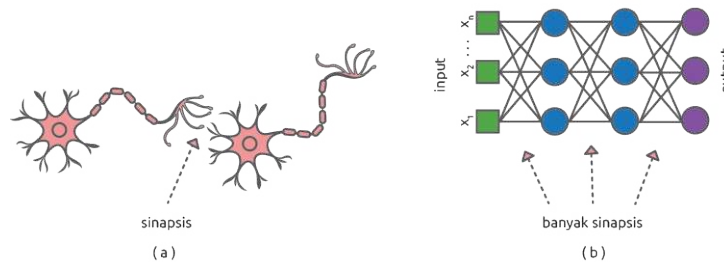


▲ Gambar 5.7. Pengujian dengan gambar yang sangat berbeda dari kelas

Salah satu algoritma penting untuk klasifikasi gambar pada *Machine Learning* adalah **Artificial Neural Network** (Jaringan Saraf Buatan). *Artificial Neural Network* (ANN) diinspirasi dari cara kerja otak manusia yang terdiri dari kumpulan neuron. Pada sub berikut kalian akan belajar dasar dari algoritma ANN.

Artificial Neural Network

ANN adalah algoritma machine learning yang digunakan pada *library/extension* LookExtension MIT di IDE App Inventor. LookExtension akan kalian eksplorasi pada aktivitas selanjutnya. ANN memiliki cara untuk merepresentasikan pengetahuan dalam kumpulan *node* yang diilhami dari kumpulan *neuron* pada otak manusia yang saling tersambung. Hubungan antar *node* digambarkan dalam bentuk garis yang diilhami oleh sinapsis pada otak manusia. Pengetahuan pada *node* dan sinapsis tercipta melalui runtunan proses komputasi random, spesifik, dan saling terkait dengan *node* lain. Ilustrasi ANN sederhana tampak pada gambar berikut, *node* digambarkan dalam bentuk bulatan dan garis (hubungan antar *node*) digambarkan dengan garis panah.



▲ Gambar 5.8. (a) Jaringan otak manusia, (b) Artificial Neural Network

Node input pada lapisan input menerima input, maka gambar akan direpresentasikan dalam bentuk data di *node* input, *node* berikutnya akan melakukan komputasi untuk menentukan kelas apa dari gambar pada input tersebut. Garis panah memiliki bobot/*weight* berbeda yang tampak dengan tebal tipisnya garis panah.

Proses komputasi yang terjadi dengan melibatkan bobot garis panah dan *node* terus berlanjut lapisan berikutnya yang akan dibandingkan dengan gambar sesuai label. Ada proses umpan balik (*feedback*) yang terjadi yang terus menerus, perbedaan perbandingan akan memicu perubahan bobot pada garis panah sehingga membentuk konfigurasi yang optimal. Umpan balik akan berhenti saat konfigurasi telah dianggap optimal.

Proses pelatihan adalah proses untuk memperbaharui bobot pada garis panah menuju kondisi optimal yang terbaik untuk pengklasifikasian pola. Pelatihan dianggap cukup, jika pengklasifikasian telah mampu mengklasifikasinya gambar sesuai dengan labelnya, bahkan juga jika gambar adalah gambar baru.

Proses pelatihan adalah komputasi yang cukup kompleks seperti bagaimana menghitung error pada saat perbandingan, pembaharuan bobot, dan termasuk bagaimana gambar untuk pelatihan telah mewakili keseluruhan dari gambar yang akan diklasifikasi. Hal ini terus menjadi topik riset di *Machine Learning* dan ANN sampai saat ini.

Pada ANN dikenal dua jenis pelatihan yang disebut *supervised* dan *unsupervised learning*. *Supervised learning* adalah cara umum yang digunakan untuk klasifikasi gambar, dimana gambar input dan label telah diketahui. Contoh pada klasifikasi gambar dengan kelas/kategori/label “kucing” dan “anjing”, adalah salah satu contoh *supervised learning*.

Jenis pelatihan *unsupervised learning* menggunakan cara dimana input tersedia, namun kelas/kategori/label belum diketahui. Pelatihan jenis ini biasanya digunakan untuk mencari pola baru, misalnya dari data perjalanan para turis, data medis yang akan dicari pola baru yang belum diketahui sebelumnya.