**BAB II TAMBAHAN LANDASAN TEORI**

**2.1 Sensor LDR (*Light Dependent Reistor*)**

Modul Resistor peka cahaya atau fotoresistor adalah komponen elektronik

yang resistansinya akan menurun jika ada penambahan intensitas cahaya yang mengenainya. Fotoresistor dapat merujuk pula pada *light-dependent resistor* (LDR), atau fotokonduktor. Fotoresistor dibuat dari semikonduktor beresistansi tinggi yang tidak dilindungi dari cahaya. Jika cahaya yang mengenainya memiliki frekuensi yang cukup tinggi, foton yang diserap oleh semikonduktor akan menyebabkan elektron memiliki energi yang cukup untuk meloncat ke pita konduksi. Elektron bebas yang dihasilkan (dan pasangan lubangnya) akan mengalirkan listrik, sehingga menurunkan resistansinya.



**Gambar 2.1. Light Dependent Resistor (LDR)**

*Sumber (Sri Supatmi, 2011)*

4

Resistansi LDR akan berubah seiring dengan perubahan intensitas cahaya yang mengenainya atau yang ada disekitarnya. Dalam keadaan gelap resistansi LDR sekitar 10MΩ dan dalam keadaan terang sebesar 1KΩ atau kurang. LDR terbuat dari bahan semikonduktor seperti kadmium sulfida. Dengan bahan ini energi dari cahaya yang jatuh menyebabkan lebih banyak muatan yang dilepas atau arus listrik meningkat. Artinya resistansi bahan telah mengalami penurunan. (Sri Supatmi, 2011)

**2.2 Android**

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc. mengakuisi *Android* Inc. Yang mengembangkan *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, yaitu konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. HTC Dream yang pertama memakai sistem operasi Android.

Dari segi arsitektur system pada gambar 2.2, Android merupakan sekumpulan *framework* dan *virtual machine* yang berjalan di atas kernel *linux*. *Virtual machine Android* bernama *Dalvik Virtual Machine (DVM)*, *engine* ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan kernel linux. Sementara untuk *framework* aplikasi sebagian besar dikembangkan oleh google dan sebagian yang lain dikembangkan oleh pihak ketiga (*developer*). (Andrey David, 2015)



**Gambar 2.2 Lapisan Arsitektur Sistem Operasi *Android* Secara Umum**

*Sumber (Andrey David 2015)*

**2.2.1 Kelebihan Android**

1. *Switching* dan *multitasking* yang lebih baik Android sangat mendukung *multitasking* aplikasi, kini hal tersebut kembali ditingkatkan. Dalam *Honeycomb* penggguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah *icon* pada *system bar.*

2. Kapasitas yang lebih baik untuk beragam *widget* Kapabilitas terhadap beragam *widget* dijanjikan bakal makin memanjakan para penggunanya. Contohnya *widget* untuk email Gmail yang dipamerkan Google, pengguna tidak perlu membuka aplikasi Gmail untuk melihat isi di dalamnya.

3. Peningkatan kemampuan *copy-paste* Beberapa seri Android terdahulu memang sudah bisa melakukan *copypaste*, namun beberapa pengguna masalah pemilihan teks yang agak sulit. Kini hal tersebut coba diselesaikan, selain *copypaste* Google juga menambah *share it* pada teks yang diseleksi.

4. *Browser Crome* lebih cepat. Ada satu fitur yang hilang dalam *browser Chrome* yang diletakkan pada Android terdahulu, kemampuan *Tab*. *Chrome* yang ada di Honeycomb kini dapat melakukan hal tersebut. Selain itu pengguna juga bisa mensinkronisasi antara *browser* di ponsel dengan *Chrome* yang ada di komputer.

5. Notifikasi yang Mudah Terdengar. Dengan layar yang lebih besar, otomatis membuat Google lebih leluasa menempatkan notifikasi pada layar.

6. Peningkatan *Drag* and *Drop* serta *Multitouch* Ukuran layar yang lebih besar, menuntut Google untuk meningkatkan kemampuan *multitouch* di dalam Android, tak terkecuali fitur *drag* and *drop*. Pada demo yang ditayangkan, pengguna bisa melakukan *drag* and *drop* untuk memindahkan email di dalam aplikasi Gmail.

**2.2.2 Kekurangan Android**

1. Koneksi Internet yang terus menerus. Kebanyakan ponsel Android memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan dan baterai yang boros karena GPRS yang terus menyala.

2. Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu ada Iklan yang terpampang. (Apri Junaidi, 2015)

**2.3 Arduino Uno**

Arduino Uno adalah jenis suatu papan (*board*) yang berisi mikrokontroller. Dengan perkataan lain, Arduino dapat disebut sebagai sebuah papan mikrokontroller. ( Abdul Kadir, 2014 : 17)

Penjelasan singkat beberapa bagian penting di papan Arduino Uno adalah sebagai berikut:



**Gambar 2.3. *Board* Arduino Uno**

*Sumber (Neerparaj Rai, 2013)*

• Mikrokontroler Atmega 328 adalah “otak” papan Arduino. Komponen ini adalah sebuah IC (*Integrated Circuit*), yang dipasangkan ke *header socket* sehingga memungkinkan untuk dilepas.

• Konektor USB (*Universal Serial Bus*) berfungsi sebagai penghubung ke PC. Konektor ini sekaligus berfungsi sebagai pemasok tegangan bagi papan Arduino.

• Konektor catu daya berfungsi sebagai penghubung ke sumber tegangan eksternal diperlukan sekiranya konektor USB tidak dihubungkan ke PC. Adaptor AC-ke-DC atau baterai dapat dihubungkan ke konektor ini. Konektor ini dapat menerima tegangan dari +7 hingga 12V.

• Pin digital adalah pin yang digunakan untuk menerima atau mengirim isyarat digital. Isyarat 1 (sering dinyatakan dengan *HIGH)*

direpresentasikan dalam bentuk tegangan 5V dan isyarat 0 (kerap dinyatakan dengan *LOW*) diwujudkan dalam bentuk tegangan 0V. Nomor untuk pin digital berupa 0 hingga 13. Beberapa pin digital, yang dinamakan pin PWM dapat digunakan sebagai keluaran analog. Pin PWM ditandai dengan simbol~. Ada 6 pin PWM yaitu 2,5,6,9,10, dan 11.

• Pin analog adalah pin yang dipakai untuk menerima nilai analog. Jika dinyatakan dalam tegangan, nilai analog akan berkisar antara 0 hingga 5V. Di pin analog, nilai seperti 1,0 atau 2.5 dimungkinkan.

• Pin sumber tegangan adalah pin yang memberikan satu daya kepada pin-pin lain yang membutuhkannya.

1. Vin, berasal dari *voltage in*, adalah pin yang memberikan tegangan sama dengan tegangan luar yang diberikan ke papan Arduino.

2. GND, berasal dari *ground*. Total pin GND adalah 3. Satu di pin terletak di sebelah pin digital 13.

3. 5V berisi tegangan 5V.

4. 3.3V berisi tegangan 3,3V.

**•** LED yang tersedia berjumlah 4. Fungsi masing-masing adalah seperti berikut:

1. *ON* akan menyala kalau papan Arduino diberi sumber tegangan.

2. RX dan TX menyatakan data sedang dikirim dan diterima oleh papan Arduino.

3. L adalah LED yang terhubung ke pin 13.

4. Tombol *reset* akan membuat *sketch* dijalankan ulang. Kadangkala, instruksi yang diberikan di Arduino menimbulkan hal yang tidak normal. Pada keadaan seperti itu, tombol *reset* yang ditekan akan membuat sistem di-*reset* dan kemudian diaktifkan kembali. (Abdul Kadir, 2014 : 20)

**2.3.1 Komunikasi Arduino**

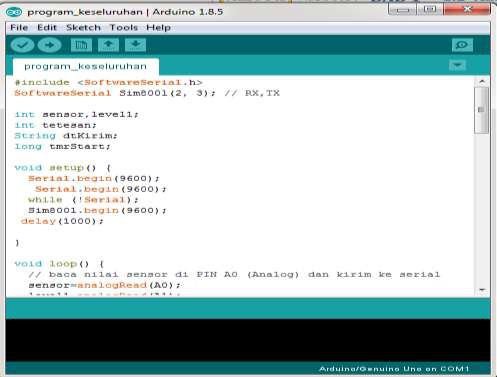
Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB

ke komputer. ([P S](http://www.arduino.cc/)ihombing, 2017)

**2.3.2 *Software* Arduino IDE**

Arduino IDE adalah *software* yang disediakan di situs arduino.cc yang ditujukan sebagai perangkat pengembangan *sketch* yang digunakan sebagai program di papan Arduino. IDE (*Integrated Development Environment*) berarti bentuk alat pengembangan program yang terintegrasi sehingga berbagai keperluan disediakan dan dinyatakan dalam bentuk antarmuka berbasi menu. Dengan menggunakan Arduino IDE, anda bisa menulis *sketch*, memeriksa ada kesalahan

atau tidak di *sketch* , dan kemudian mengunggah *sketch* yang sudah terkompilasi ke papan Arduino. Arduino IDE tersedia untuk *platform* Windows, Mac OS X, maupun Linux. (Abdul Kadir, 2014 : 26)

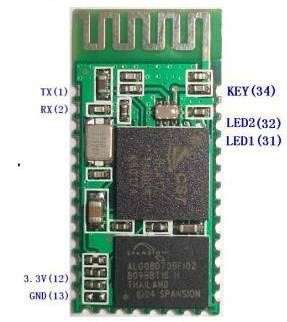


**Gambar 2.4. Tampilan IDE Arduino dengan Sebuah *Sketch***

*Sumber* [*(*http://www.arduino.c](http://www.arduino.cc)c)

**2.4 *Module Bluetooth* HC-05**

Modul *Bluetooth* HC-05 Bluetooth adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain1. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. Untuk gambar *module bluetooth* dapat dilihat pada gambar 2.5. dibawah ini:

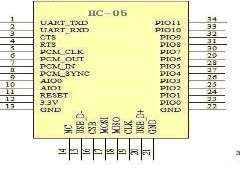


**Gambar 2.5. Modul *Bluetooth* HC-05**

*Sumber (Linarti, 2014)*

Modul *Bluetooth* HC-05 dengan *supply* tegangan sebesar 3,3 V ke pin 12 modul *Bluetooth* sebagai VCC. Pin 1 pada modul *Bluetooth* sebagai transmitter. kemudian pin 2 pada *Bluetooth* sebagai receiver. Berikut merupakan konfigurasi pin *bluetoooth* HC-05 ditunjukkan pada gambar 2.6. dibawah ini:

1

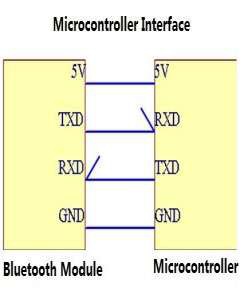


**Gambar 2.6. Konfigurasi Pin HC-05**

*Sumber (Linarti, 2014)*

Berikut merupakan *Bluetooth-to-Serial-Module* HC-05 dapat dilihat pada gambar

2.7. dibawah ini:

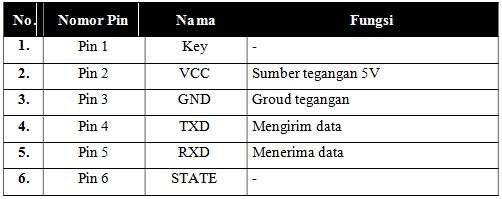


**Gambar 2.7. *Bluetooth-to-Serial-Module* HC-05**

*Sumber ( Linarti, 2014)*

Konfigurasi pin modul *Bluetooth* HC-05 dapat dilihat pada table 2.1. berikut ini :

**Tabel 2.1. Konfigurasi pin *Module Bluetooth* CH-05**



*Sumber (Linarti, 2014)*

*Module Bluetooth* HC-05 merupakan *module Bluetooth* yang bisa menjadi *slave* ataupun *master* hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan *pairing* keperangkat lain, maupun perangkat lain tersebut yang melakukan *pairing* ke *module Bluetooth* CH-05. Untuk mengeset perangkat *Bluetooth* dibutuhkan perintah-perintah AT *Command* yang mana perintah AT *Command* tersebut akan di respon oleh perangkat *Bluetooth* jika modul *Bluetooth* tidak dalam keadaan terkoneksi dengan perangkat lain. Table 2.2. dibawah adalah *table AT Command Module Bluetooth* CH-05. Keterangan *AT Command Module Bluetooth* CH-05 dapat dilihat pada table 2.2. berikut:

**Tabel 2.2 *AT Command Module Bluetooth* CH-05**



*Sumber (Linarti, 2014)*

**2.5. UML (*Unified Modeling Language*)**

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan *system* berorientasi objek dan juga merupakan alat untuk mendukung pengembangan *system* (Ade Hendini, 2016). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

**2.5.1 *Use Case Diagram***

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

**Tabel 2.3. Simbol Use Case Diagram**

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Use Case* menggambarkan fungsi yang |
|  | disediakan sistem sebagai unit-unit yang |
|  | bertukar pesan antar unit dengan aktor, |
|  | yang dinyatakan dengan menggunakan |
|  | kata kerja. |
|  | *Actor* atau aktor adalah *abstraction* dari |
|  | orang atau sistem yang lain yang |
|  | mengaktifkan fungsi dari target sistem. |
|  | Untuk mengidentifikasikan aktor, harus |
|  | ditentukan pembagian tenaga kerja dan |
|  | tugas-tugas yang berkaitan dengan peran |
|  | pada pada konteks target sistem. Orang |
|  | atau sistem bisa muncul dalam beberapa |
|  | peran. Perlu dicatat bahwa aktor |
|  | berinteraksi dengan *use case*, tetapi |
|  | tidak memiliki kontrol terhadap *use* |

|  |  |
| --- | --- |
|  | *case.* |
|  | Asosiasi antara aktor dan *use case*, |
|  | digambarkan dengan garis tanpa panah |
|  | yang mengindikasikan siapa atau apa |
|  | yang meminta interaksi secara langsung |
|  | dan bukannya mengindikasikan data. |
|  | Asosiasi antara aktor dan *use case* yang |
|  | menggunakan panah terbuka untuk |
|  | mengindikasikan bila aktor berinteraksi |
|  | secara pasif dengan sistem. |
| <<include>> | *Include*, merupakan di dalam *use case*  lain (*required*) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program. |
|
|
|
|
|  | *Extend*, merupakan perluasan dari *use* |
| < | *case* lain jika kondisi atau syarat |
|  | terpenuhi. |
| <<extends>> |  |

*Sumber (Ade Hendini, 2016)*

**2.5.2 Diagram Aktivitas (*Activity Diagram*)**

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

**Tabel 2.4. Simbol Diagram *Activity***

|  |  |
| --- | --- |
|  | *Start Point*, diletakkan pada pojok kiri |
|  | atas dan merupakan awal aktivitas. |
|  | *End Point*, akhir aktivitas. |
|  | *Activities*, menggambarkan suatu proses. |
|  | *Fork*/Percabangan, digunakan untuk |
|  | menunjukkan kegiatan yang dilakukan |
|  | secara paralel atau untuk menggabung |
|  | kan dua kegiatan paralel menjadi satu. |
|  | *Join* (Penggabungan) atau *rake*, |
|  | digunakan untuk menunjukkan adanya |
|  | dekomposisi. |
|  | *Decision Point*, menggambarkan pilihan |
|  | untuk pengambilan keputusan, *true of* |

|  |  |
| --- | --- |
|  | *false*. |
|  | *Swimlane,* pembagian *activity* diagram  untuk menunjukkan siapa melakukan apa. |

*Sumber ( Ade Hendini, 2016)*

***2.6. Flow Chart***

*Flowchart* merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. (Eka Iswandi, 2015)

Menurut Adelia, Jimmy (2011:116) Flowchart adalah penggambaran secara grafik dari langkah-langkah dan uruturutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa mempersentasikan komponen – komponen dalam bahasa pemprograman.

**Tabel 2.5. Simbol-simbol *Flowchart***



|  |  |  |
| --- | --- | --- |
| No. | Simbol | Keterangan |
| 1. |  | Permulaan sub program |
| 2. |  | Perbandingan, pernyataan, penyeleksian data |
|  | yang memberikan pilihan untuk lengkah |
|  | selanjutnya |
| 3. |  | Penghubung bagian-bagian *flowchart* yang |
|  | berada pada satu halaman |
| 4. |  | Penghubung bagian-bagian *flowchart* yang |
|  | berada pada halaman berbeda |
| 5. |  | Permulaan/akhir program |
| 6. |  | Arah aliran program |
| 7. |  | Proses inisialisasi/pemberian harga awal |
| 8. |  | Proses penghitung/proses pengolahan data |
| 9. |  | Proses *input*/*output* data |

*Sumber (Ade Hendini, 2016)*

**2.7. Program *App Inventor* Pada Pemrograman Android**

Mulyadi (2013: 1) menjelaskan *App Inventor* adalah sebuah *tool* untuk membuat aplikasi android yang berbasis *visual block programming*, sehingga pengguna bisa membuat aplikasi tanpa melakukan *codding*. *Visual block programming* maksudnya adalah dalam penggunaannya *user* akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi *–event handler* tertentu dalam membuat aplikasi, dan secara sederhana bisa disebut tanpa menuliskan kode program.

Aplikasi *App Inventor* ini pada dasarnya adalah aplikasi yang disediakan oleh google dan sekarang di-*maintenance* oleh *Massachusetts Institute of Technology* (MIT). Aplikasi ini selesai dibuat pada 12 juli 2010 dan dirilis untuk public pada 31 Desember 2011. *App Inventor* sekarang dipegang oleh MIT *Centre for Mobile Learning* dengan nama *MIT App* Inventor. Gambar berikut adalah gambar tampilan aplikasi *App Inventor*:

Dengan menggunakan *App Inventor* ini, ada beberapa aplikasi yang dapat dibuat diantaranya yaitu:

1. Aplikasi *game*

2. Aplikasi berbasis *tracking* lokasi

3. Aplikasi SMS

4. Aplikasi berbasis *web*

5. Aplikasi kompleks

Aplikasi *App* Inventor memiliki tiga opsi dalam proses pengujian sistem, diantaranya adalah:

A. Membuat aplikasi dengan perangkat Android dengan koneksi Wi-Fi.



**Gambar 2.8. Koneksi Android dengan *Wi-Fi***

*Sumber* [*(http://appinventor.mit.edu*](http://appinventor.mit.edu)*)*

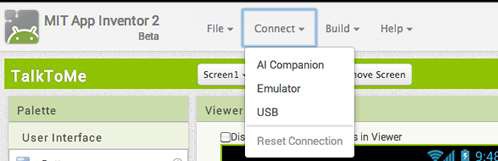
Langkah – langkah yang harus dilakukan dalam pembuatan aplikasi android dengan koneksi *Wi-Fi* adalah sebagai berikut:

1. Unduh dan *instal* aplikasi MIT AI2 *Companion* di Ponsel

2. Sambungkan komputer dan perangkat dengan jaringan *WiFi* yang sama

3. Buka proyek *App* Inventor dan hubungkan ke perangkat

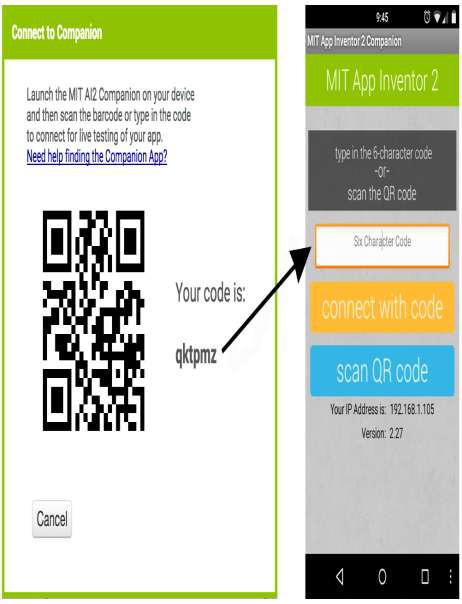
Setelah ketiga langkah diatas selesai, selannjutnya buka Aplikasi Inventor kemudian pilih "*Connect*" dan "*AI Companion*" dari menu teratas di *browser* AI2:



**Gambar 2.9. *Connect Al Companion***

*Sumber (*[http://appinventor.mit.edu)](http://appinventor.mit.edu/)

Setelah proses diatas selesai selanjutnya secara otomatis akan muncul tampilan kode QR pada halaman MIT Inventor lalu *scan* kode QR dengan aplikasi yang telah di unduh sebelumnya dari google *playstore*. Setelah semua proses selesai secara otomatis aplikasi yang telah dibuat akan terhubung ke perangkat android.



**Gambar 2.10. *Scan QR code***

*Sumber* [*(http://appinventor.mit.edu*](http://appinventor.mit.edu)*)*

B. Membuat aplikasi dengan Emulator pada PC (*Personal Computer*)

Langkah – langkah yang harus dilakukan dalam pembuatan aplikasi dengan emulator pada PC yaitu dengan meng*instal software App* Inventor 2 pada PC. Aplikasi ini dapat berjalan disemua sistem operasi windows dan menggunakan *search engine google chrome* dan *mozila firefox*. Jalankan *software* yang sudah di *install* sebelumnya, lalu buka *search engine* dan masuk ke halaman *project App* Inventor dan buka emulator.

C. Membuat aplikasi dengan Android dan USB

Membuat aplikasi dengan Android dan USB dapat dilakukan dengan menghubungkan perangkat dengan kabel USB lalu menjalankan koneksi pada halaman kerja *App* Inventor agar mentransfer data ke perangkat.