

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori Jurnal**

Dalam pembuatan aplikasi ini penulis melakukan kajian dari penelitian-penelitian terdahulu, sehingga bisa dijadikan referensi dalam penelitian ini.

Pada penelitian sebelumnya yang dilakukan oleh (Yola dan Miftahur, 2018) dalam jurnal yang berjudul **“Alat Pembuka Pintu Kamar Menggunakan Sensor Sidik Jari”**. Permasalahannya adalah di Indonesia kebanyakan pintu masih menggunakan kunci konvensional dan keamanan pintu yang menggunakan *password* untuk saat ini masih banyak kelemahannya.

Metodologi penelitian dimulai dari perancangan sistem, blok diagram rangkaian, data flow diagram, perancangan alat dan perancangan program. Alat ini dapat mendeteksi sidik jari yang telah didaftarkan pada sensor, alat ini menggunakan sensor sidik jari fingerprint R305 sebagai sensor pendeteksi sidik jari dan arduino sebagai media pengontrol dan pemroses input data kontrol.

Seperti pada penelitian sebelumnya yang dilakukan oleh (Saputro, 2016) dalam tugas akhir yang berjudul **“Prototype Pengaman Gerbang Rumah dengan Pola Sidik Jari Berbasis Arduino Uno”**. Permasalahannya adalah bagaimana merancang pengaman rumah sehingga seminimal mungkin dapat mencegah tindak pencurian.

Metode penelitian yang digunakan dimulai dengan studi pustaka, perancangan sistem dan melakukan pengujian dan analisa. Penelitian yang dilakukan yaitu dengan membuat dan merancang simulasi. Pada penelitian ini menggunakan sensor sidik jari dengan modul ZFM 20 series yang terhubung ke mikrokontroler Arduino uno sebagai pengolah data serta ada relay yang mengendalikan solenoid untuk membuka kunci gerbang.

Pada penelitian sebelumnya yang dilakukan oleh (Yudhana, dkk 2017) dalam jurnal yang berjudul **“Perancangan Pengamanan Pintu Rumah Berbasis Sidik jari Menggunakan Metode UML”**. Permasalahannya adalah kunci pintu konvensional rentan terhadap resiko ketinggalan atau kehilangan. Penelitian ini

merancang pembuatan pengaman pintu rumah dengan menggunakan sidik jari. Proses yang dilakukan adalah input sidik jari, identifikasi sampai dengan verifikasi setiap sidik jari yang direkam dalam data base.

Metode yang digunakan adalah *Unified Modeling Language* (UML). Finger print C3 digunakan sebagai komponen input yang akan diolah oleh Arduino Uno ATmega 328. Hasil percobaan dari rancangan ini selenoid akan membuka dengan cara bekerja maju dan mundur etelah diberikan arus 5 volt.

Pada penelitian sebelumnya dapat disimpulkan bahwa peneliti menggunakan sensor sidik jari sebagai pengaman pintu dan pada dua penelitian yang telah dibahas di atas tidak ada yang menggunakan *database* jadi tidak dapat melihat *database* orang-orang yang telah membuka pintu dan tidak tahu waktu akses ketika seseorang membuka pintu.

## 2.2 Aplikasi

Menurut (Jogiyanto, 2004), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jadi aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Menurut (Daryanto, 2004), aplikasi adalah *software* atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus.

Menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari *user*.

Dari uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data.

## 2.3 Android

Menurut (Nazaruddin, 2012), android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi

para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Android umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti operasi *Symbian* di Nokia, *iOS* di Apple dan *Blackberry OS*. Android diciptakan oleh sebuah perusahaan kecil bernama *Android Inc* pada tahun 2000, yang kemudian perusahaan tersebut dibeli oleh *Google Inc*, untuk mengembangkan android lebih lanjut, dibentuklah *Open Handset Alliance*(OHA) yang terdiri dari 34 perusahaan *software*, *hardware* dan telekomunikasi diantaranya yaitu *Google*, *HTC*, *Motorola*, *T-Mobile* dan lain sebagainya.

Menurut (Andry, 2011), tujuan pembuatan sistem operasi ini adalah untuk menyediakan *platform* yang terbuka, yang memudahkan orang mengakses internet menggunakan telepon seluler. Android juga dirancang untuk memudahkan pengembang membuat aplikasi dengan batasan yang minim sehingga kreativitas pengembang menjadi lebih berkembang.

Menurut (Lee, 2011), sebagai *Open Source* dan bebas dalam memodifikasi, di dalam Android tidak ada ketentuan yang tetap dalam konfigurasi *Software* dan *Hardware*. Fitur-fitur yang didapat dalam Android antara lain:

1. *Storage* : Menggunakan *SQLite*, *database* yang ringan, untuk sebuah penyimpanan data
2. *Connectivity* : Mendukung *GSM/EDGE*, *IDEN*, *CDMA*, *EV-DO*, *UMTS* dan sebagainya.
3. *Bluetooth*(termasuk *A2DP* dan *AVRCP*), *WiFi*, *LTE*, dan *WiMax*.
4. *Messaging* : Mendukung *SMS* dan *MMS*
5. *Web Browser* : Berbasiskan *open-source* *WebKit*, bersama mesin.
6. *Chrome's V8 JavaScript*
7. *Media support* : Termasuk mendukung untuk beberapa media berikut : *H.263*, *H.264* (dalam bentuk *3GP* or *MP4*), *MPEG-4 SP*, *AMR*, *AMRWB* (dalam bentuk *3GP*), *AAC*, *HE-AAC* (dalam bentuk *MP4* atau *3GP*), *MP3*, *MIDI*, *Ogg Vorbis*, *WAV*, *JPEG*, *GIF*, dan *BMP*.
8. *Hardware support* : Sensor akselerasi, Kamera, Kompas Digital, Sensor Kedekatan, *GPS*.

9. *Multi-touch* : Mendukung *multi-touch screens*.
10. *Multi-tasking* : Mendukung aplikasi *multi-tasking*
11. *Flash-support* : Android 2.3 mendukung *Flash10.1*.
12. *Tethering* : Mendukung pembagian dari koneksi Internet sebagai *wired/wireless hotspot*
13. *Play store* : katalog aplikasi yang dapat di-*download* dan di-*instal* pada *smartphone* secara *online*, tanpa menggunakan PC (*Personal Computer*).
14. Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk Eclipse IDE.

### 2.3.1 Aplikasi Android

Menurut (Lee, 2011), android memungkinkan penggunanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti *Google Play*, *Amazon Appstore*, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Di *Google Play*, pengguna bisa menjelajah, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh *Google* dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas *Google*.

*Google Play* akan menyaring daftar aplikasi yang tersedia berdasarkan kompatibilitasnya dengan perangkat pengguna, dan pengembang dapat membatasi aplikasi ciptaan mereka bagi operator atau negara tertentu untuk alasan bisnis. Pembelian aplikasi yang tidak sesuai dengan keinginan pengguna dapat dikembalikan dalam waktu 15 menit setelah pengunduhan. Beberapa operator seluler juga menawarkan tagihan langsung untuk pembelian aplikasi di *Google Play* dengan cara menambahkan harga pembelian aplikasi pada tagihan bulanan pengguna. Pada bulan September 2012, ada lebih dari 675.000 aplikasi yang tersedia untuk Android, dan perkiraan jumlah aplikasi yang diunduh dari *Play Store* adalah 25 miliar.

Menurut (Andry, 2011), Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk *debugger*, perpustakaan perangkat lunak, *emulator handset* yang

berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan *plugin Android Development Tools(ADT)*. Perangkat pengembangan lain yang tersedia di antaranya adalah *Native Development Kit* untuk aplikasi atau ekstensi dalam C atau C++, *Google App Inventor*, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas *platform*.

### 2.3.2 Arsitektur Android

Menurut (Andry, 2011), Android terdiri dari *linux kernel, libraries, android runtime, application framework* dan *applications*. Arsitektur Android dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Arsitektur Android

Berikut ini merupakan penjelasan dari Arsitektur Android tersebut.

#### 1. Linux Kernel

Pada bagian paling bawah dari Android adalah Linux Kernel. Android menggunakan modifikasi dari Linux Kernel versi 2.6. Bagian ini bertanggung

jawab untuk mengelola dan berkomunikasi dengan perangkat keras dimana Android berjalan. Pemilihan Linux Kernel sebagai inti dari Android adalah karena dukungan dan kestabilan terhadap berbagai macam komponen perangkat keras. Pada bagian ini disediakan *driver* (program pengendali) perangkat keras, pengelolaan memori, pengelolaan proses, pengelolaan jaringan, dan keamanan.

## 2. *Libraries*

*Libraries* merupakan kumpulan kode yang dapat digunakan oleh komponen/program lain. Pada bagian ini kodenya ditulis menggunakan bahasa pemrograman C/C++. Fungsi masing-masing komponen di *libraries* ini adalah sebagai berikut :

- a. *Surface Manager* : mengelola penggambaran dan komposisi komponen di layar (jendela, tombol, teks input, objek tiga dimensi dan sebagainya).
- b. Open GL|ES : *engine grafis* tiga dimensi.
- c. SGL : *engine grafis* dua dimensi.
- d. *Media Framework* : *codec* untuk merekam dan memutar berbagai format video dan audio.
- e. *Free Type* : untuk *rendering font*.
- f. SSL : untuk keamanan komunikasi jaringan.
- g. SQLite : pengelolaan database untuk penyimpanan data.
- h. Webkit : *web browser engine* Android.
- i. Libc : *library C standar*.

## 3. *Android Runtime*

Pada bagian inilah aplikasi Android berjalan. Inilah bagian *middleware* yang disebutkan pada definisi Android di awal. Aplikasi Android ditulis menggunakan bahasa pemrograman Java dan untuk menjalankannya diperlukan sebuah *virtual machine* yang akan menerjemahkan instruksi-instruksi program Java tersebut ke instruksi yang dimengerti oleh sistem operasi. Namun berbeda dengan *platform* lainnya, Android tidak menggunakan Java *Virtual Machine* (JVM), melainkan menggunakan *Dalvik Virtual Machine* (DVM). *Dalvik Virtual Machine* adalah sebuah *virtual machine* yang dioptimasi untuk perangkat yang

memiliki memori kecil, sumber tenaga terbatas, dan kemampuan prosesor yang kecil

#### **4. *Application Framework***

Bagian ini adalah *framework* atau perangkat yang dapat digunakan oleh pengembang aplikasi dalam membuat aplikasi Android. Bagian ini menyertakan program-program yang mengelola fungsi-fungsi dasar seperti informasi lokasi, fungsi telepon, pengelolaan perpindahan antar program atau proses, dan sebagainya. Pengembang aplikasi dapat memiliki akses *framework* yang sama dengan yang dapat diakses oleh aplikasi-aplikasi utama dari Android (*phone dialer*, buku telepon, SMS, e-mail, kalender, dan sebagainya). Disini terlihat keinginan dari Android untuk menjadi *platform* yang terbuka, sehingga memungkinkan pengembang aplikasi untuk lebih kreatif menciptakan aplikasi yang menarik dan inovatif.

#### **5. *Applications***

Pada bagian ini terdapat aplikasi-aplikasi yang dapat digunakan oleh pengguna perangkat Android. Selain aplikasi inti seperti yang ada di diagram arsitektur, aplikasi-aplikasi tambahan yang dipasang sendiri oleh pengguna akan menempati bagian ini dan memiliki hak akses yang sama terhadap *Application Framework*. Apabila diperlukan, pengguna ataupun vendor dapat menggantikan aplikasi inti yang ada dengan aplikasi buatan sendiri yang memiliki fungsionalitas yang sama. Hal ini membuat kustomisasi sistem operasi Android jauh lebih bebas dibandingkan sistem operasi perangkat mobile lainnya.

### **2.3.3 Versi Android**

Versi dari Android dapat dilihat pada Gambar 2.2. Pada gambar 2.2 android dimulai dai versi 1.0 atau alpha, kemudian disusul versi beta. Versi pertama Android yang laku dipasaran dimulai dari versi 1.5 yang diberi nama Cupcake, setelahnya nama-nama dari versi Android menggunakan nama makanan yang manis atau dessert seperti Donut, Eclair, Froyo dan selanjutnya.



**Gambar 2.2** Versi Android

### 2.3.4 Kelebihan Android

Menurut (Nazaruddin, 2012), kelebihan dari pemrograman berbasis android yaitu bersifat *opensource*, sehingga kita dapat mengkustomisasi aplikasi yang berbasis android, bahkan membuat sendiri aplikasinya tanpa harus membayar sejumlah uang tertentu.

1. Implementasinya yang lintas *platform*, karena ditulis dengan pemrograman java, maka dapat dijalankan pada berbagai macam *platform*, dengan menyertakan *java virtual machine* yang disebut *dalvik virtual machine*.
2. Android juga menyediakan SDK dan IDE yang semuanya gratis, sehingga makin memudahkan kita ketika akan men-devel aplikasi.

### 2.4 Basis Data

Menurut (Indrajani, 2015), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

Menurut (Connolly dan Begg, 2010), basis data adalah sebuah kumpulan data yang secara logis terkait dan dirancang untuk memenuhi suatu kebutuhan informasi dari sebuah organisasi.

Menurut (Pakereng dan Wahyono 2004), basis data merupakan kumpulan data yang dipakai ada dalam suatu lingkup, misalkan instansi, perusahaan, dan lain-lain atau kasus tertentu. Menurut (Pakereng dan Wahyono, 2004), sebuah konsep database memiliki hal sebagai berikut :

### 1. **Entitas**

Entitas merupakan tempat informasi direkam, dapat berupa orang, tempat, kejadian dan lain-lain. Sebagai contoh dalam kasus Adminitrasi Siswa misalnya, maka terdapat entitas siswa, mata kuliah, guru, pembayaran.

### 2. **Atribut**

Atribut dapat juga disebut sebagai data elemen, data *field*, atau data item yang digunakan untuk yang menerangkan suatu entitas dan mempunyai harga tertentu, misalnya atribut dari entitas siswa diterangkan oleh nama, tanggal lahir, alamat.

### 3. **Data Value**

Data *value* merupakan suatu informasi atau data aktual yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

### 4. **File/Table**

Merupakan kumpulan *record* sejenis yang merupakan panjang elemen yang sama, atribut yang sama, namun berbeda nilainya. Misalnya tabel mahasiswa berisi *field* NIM, nama, kelas dan sebagainya. Pada *field* NIM panjang elemennya tiga puluh tapi tiap nilainya berbeda.

### 5. **Record/Tuple**

Merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu datu atau informasi.

## 2.5 **NodeMCU ESP8266**

Menurut (Widiyaman, 2016), NodeMCU merupakan papan pengembangan produk *Internet of Things* (IoT) yang berbasiskan *Firmware eLua* dan *System on a*

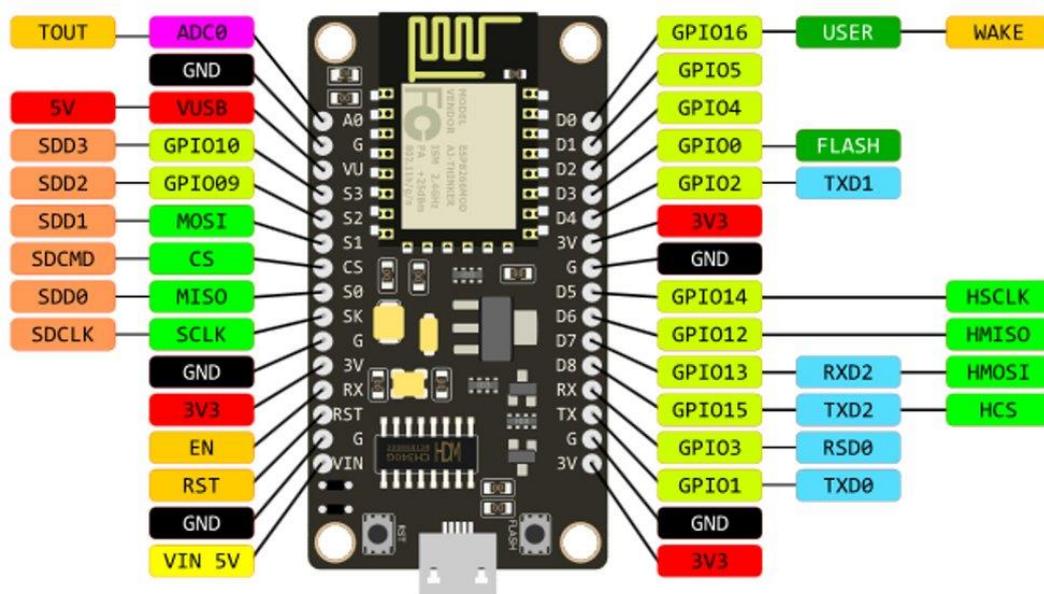
Chip (SoC) ESP8266-12E. ESP8266 sendiri merupakan chip *WiFi* dengan *protocol stack* TCP/IP yang lengkap.

NodeMCU dapat dianalogikan sebagai *board* arduinonya ESP8266. Program ESP8266 sedikit merepotkan karena diperlukan beberapa teknik *wiring* serta tambahan modul USB to serial untuk mengunduh program. Namun NodeMCU telah mempackage ESP8266 ke dalam sebuah *board* yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap *Wifi* juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan charging *smartphone*.

Alasan penulis memilih NodeMCU ESP8266 ialah karena mudah diprogram dan memiliki pin I/O yang memadai dan dapat mengakses jaringan Internet untuk mengirim atau mengambil data melalui koneksi *WiFi*.

Spesifikasi dari NodeMCU sebagai berikut :

1. 10 port pin GPIO
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 *Wire*
5. ADC



Gambar 2.3 NodeMCU ESP826 Skema Pin

Gambar 2.3 merupakan kaki pin yang ada pada NodeMCU. Berikut penjelasan dari pin-pin NodeMCU tersebut.

1. ADC: Analog Digital Converter. Rentang tegangan masukan 0-1v, dengan skop nilai digital 0-1024
2. RST : berfungsi mereset modul
3. EN: Chip Enable, Active High
4. IO16 :GPIO16, dapat digunakan untuk membangunkan chipset dari mode deep sleep
5. IO14 : GPIO14; HSPI\_CLK
6. IO12 : GPIO12: HSPI\_MISO
7. IO13: GPIO13; HSPI\_MOSI; UART0\_CTS
8. VCC: Catu daya 3.3V (VDD)
9. CS0 :Chip selection
10. MISO : Slave output, Main input.
11. IO9 : GPIO9
12. IO10 GBIO10
13. MOSI: Main output slave input
14. SCLK: Clock
15. GND: Ground
16. IO15: GPIO15; MTDO; HSPICS; UART0\_RTS
17. IO2 : GPIO2;UART1\_TXD
18. IO0 : GPIO0
19. IO4 : GPIO4
20. IO5 : GPIO5
21. RXD : UART0\_RXD; GPIO3
22. TXD : UART0\_TXD; GPIO1

Untuk tegangan kerja ESP8266 menggunakan standar tegangan JEDEC (tegangan 3.3V) untuk bisa berfungsi. Tidak seperti mikrokontroler AVR dan sebagian besar board Arduino yang memiliki tegangan TTL 5 volt. Meskipun begitu, NodeMCU masih bisa terhubung dengan 5V namun melalui *port micro* USB atau pin Vin yang disediakan oleh *boardnya*. Namun karena semua pin pada

ESP8266 tidak toleran terhadap masukan 5V. Maka jangan sekali-kali langsung mencatunya dengan tegangan TTL jika tidak ingin merusak *board* anda. Anda bisa menggunakan *Level Logic Converter* untuk mengubah tegangan ke nilai aman 3.3v.

## 2.6 MySQL

Menurut (Saputra, 2013), MySQL adalah database yang menghubungkan script PHP menggunakan perintah query dan escapes character yang sama dengan PHP. MySQL mempunyai tampilan client yang mempermudah dalam mengakses database dengan kata sandi untuk mengizinkan proses yang bisa dilakukan. phpMyAdmin adalah sebuah software yang berbentuk seperti halaman situs yang terdapat pada web server. Fungsi dari halaman ini adalah sebagai pengendali database MySQL sehingga pengguna MySQL tidak perlu repot untuk menggunakan perintah-perintah SQL. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman phpMyAdmin.

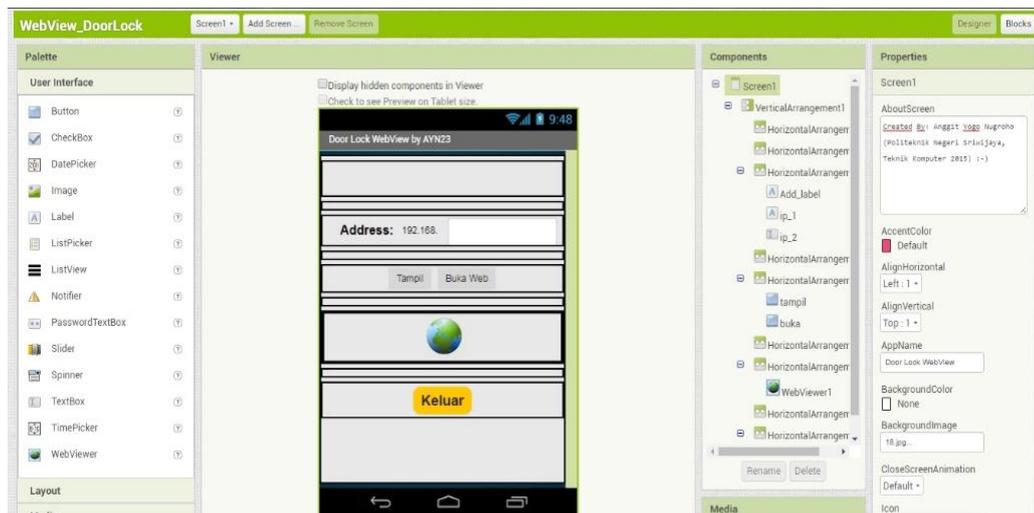
MySQL adalah sebuah implementasi dari system manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (GeneralPublic License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis

## 2.7 Android Software Development Kit (SDK)

Menurut (Nazaruddin, 2012), android *Software Development Kit* (SDK) adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan bagian perangkat lunak untuk ponsel yang

meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Sebagai *platform* aplikasi yang netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *smartphone*.

## 2.8 MIT App Inventor



Gambar 2.4 Interface MIT

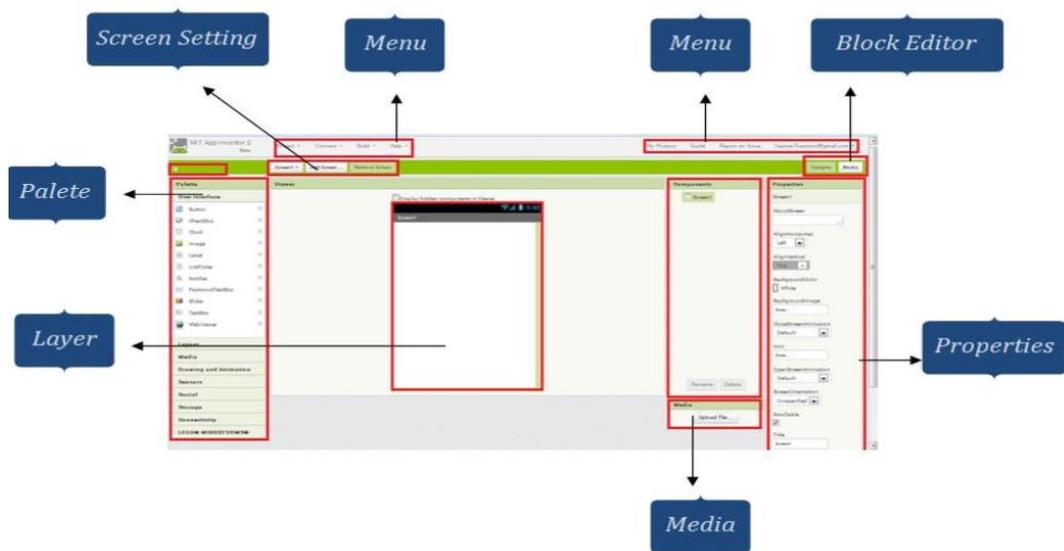
Menurut (Mulyana,2012), *App Inventor for Android* adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-*maintenance* oleh *Massachusetts Institute of Technology* (MIT). *App Inventor* menggunakan bahasa *Kawa Language Framework* dan *Kawa's dialect* yang dikembangkan oleh Per Bothner. Kedua aplikasi tersebut dijadikan sebagai *compiler* dan menerjemahkan *Visual Block Programming*. Tampilan antar muka *MIT App Inventor* dapat dilihat pada gambar 2.4 di atas.

Pembuatan Aplikasi pada Android menggunakan *graphical user interface* (GUI), yang memudahkan pengguna untuk membuat aplikasi dengan tampilan *code block* sehingga aplikasi *App Inventor* memungkinkan para pemula untuk menciptakan suatu program aplikasi sendiri dengan berbasis sistem operasi Android.

*App inventor* memanfaatkan tampilan yang mudah digunakan bagi pemula. Penggunaan aplikasi pemrograman ini dilengkapi fitur *drag and drop* untuk mendesain *user interface* aplikasi Android yang akan dibuat. Sekilas, cara menggunakan *App Inventor* mirip dengan pemrograman *Visual Basic* dengan fitur *drag and drop* nya. Namun yang membedakan antara *App Inventor* dengan *Visual Basic* adalah dari segi *coding*. Pada *App Inventor*, proses *coding* dilakukan dengan lebih sederhana karena hanya membutuhkan logika saja, tanpa harus mengingat bahasa pemrograman secara langsung. Dengan begitu aplikasi ini sangat cocok digunakan untuk proyek-proyek yang memanfaatkan Android namun tidak rumit dalam hal *coding*.

### 2.8.1 Area kerja MIT App Inventor

Menurut (Mulyana,2012), untuk dapat menggunakan *App Inventor* diperlukan pengenalan tentang area kerja dari *App Inventor* tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 2.5.



**Gambar 2.5** Area Kerja MIT *App Inventor*

Berikut ini penjelasan dari masing-masing elemen yang terdapat pada area kerja *App Inventor*:

1. *Screen Setting* merupakan sebuah kelompok yang berguna untuk mengatur layar, menambah layar, dan menghapus layar.

2. *Palete* adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain.
3. Menu merupakan sekelompok menu yang berguna dalam membuat *project* baru, proses *debugging*, konversi *file* apk, dll.
4. *Block Editor* adalah suatu tombol untuk masuk ke halaman kode blok untuk proses pengkode-an.
5. *Properties*: untuk mengatur komponen yang telah di buat menjadi desain di layer.
6. *Media* : Tempat dimana untuk meng-*upload* file.
7. *Layer* : Area untuk men-desain

## 2.9 Java

Menurut (Hariyanto, 2003), java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer, termasuk telepon genggam. Bahasa Java memiliki fitur keamanan *built-in* yang membuat tugas pemrograman lanjut seperti pemrograman jaringan, pemrograman tersebar, koneksifitas *database*. Dan juga memiliki fitur *multithreading* sehingga mempermudah dalam pemrograman aplikasi *multithreading*, pemrograman menjadi lebih aman. Sebagai sebuah bahasa pemrograman, java dapat membuat seluruh bentuk aplikasi, dekstop, website, dan lainnya sebagaimana yang dibuat bahasa pemrograman yang lain.

## 2.10 Smartphone

(Wikipedia) Ponsel pintar atau yang lebih populer dengan *smartphone* merupakan sebuah ponsel yang memiliki OS (*Operating System*) yang terdiri dari atas kombinasi dari fitur sebuah PC (*Personal Computer*) dan fitur yang terdapat pada ponsel juga tentunya. Sebagian besar *smartphone* dapat digunakan untuk mengakses *internet*, dengan layar sentuh, dan dilengkapi dengan kamera. Mobile Operating System yang digunakan untuk *smartphone* terdiri dari beberapa jenis, yaitu:

1. Android
2. IOS

3. Windows Phone
4. Blackberry
5. Firefox OS
6. Sailfish OS
7. Ubuntu Touch

### **2.11 Telegram**

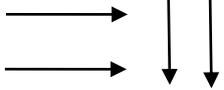
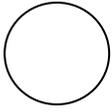
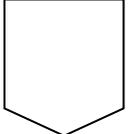
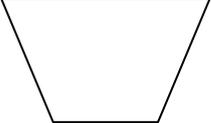
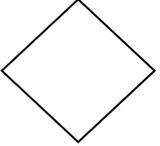
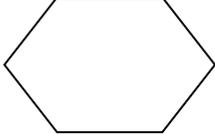
(Nugroho, 2018) Telegram adalah sebuah aplikasi layanan pengirim pesan instan multiplatform berbasis *cloud* yang bersifat gratis dan nirlaba. Klien Telegram tersedia untuk perangkat telepon seluler (Android, iOS, Windows Phone, Ubuntu Touch) dan sistem perangkat komputer (Windows, OS X, Linux). Para pengguna dapat mengirim pesan dan bertukar foto, video, stiker, audio, dan semua tipe fail atau berkas. Telegram juga menyediakan pengiriman pesan ujung ke ujung terenkripsi opsional.

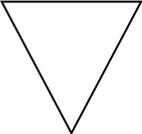
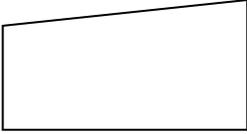
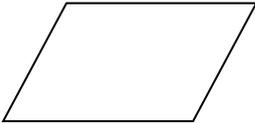
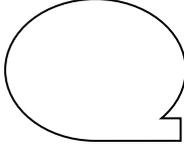
Telegram dikembangkan oleh Telegram *Messenger* LLP dan didukung oleh wirausahawan Rusia Pavel Durov. Kode pihak kliennya berupa perangkat lunak sistem terbuka namun mengandung blob binari, dan kode sumber untuk versi terbaru tidak selalu segera dipublikasikan, sedangkan kode sisi servernya bersumber tertutup dan berpaten. Layanan ini juga menyediakan API kepada pengembang independen.

### **2.12 Flowchart**

Menurut (Saputra, 2014), *flowchart* merupakan suatu diagram yang menggambarkan alur kerja suatu sistem. Tujuan utama penggunaan *flowchart* adalah untuk menggambarkan suatu tahap penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Dalam penulisan *flowchart* dikenal dua model yaitu *flowchart* sistem dan *flowchart* program. *Flowchart* sistem merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antara peralatan tersebut. Simbol-simbol *flowchart* dapat dilihat pada Tabel 2.1.

**Tabel 2.1.** Simbol Diagram *Flowchart*

No	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol <i>manual</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak
7		Simbol <i>terminal</i> , berfungsi untuk menyatakan permulaan atau akhir suatu program
8		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>

10		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11		Simbol <i>manual input</i> , berfungsi untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
12		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis
14		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
15		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i> )
16		Simbol <i>punched card</i> , berfungsi untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu

(Sumber: Yolanda, 2018)