

BAB 3

PERANCANGAN SISTEM

3.1 ALAT YANG DIGUNAKAN

Dalam proyek tugas akhir ini dibutuhkan beberapa alat dan bahan penunjang, berupa perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut beberapa perangkat yang digunakan yaitu:

3.3.1 Perangkat Keras (*Hardware*)

1) *NodeMCU* ESP8266

NodeMCU merupakan sebuah *opensource platform* IoT dan pengembangan *Kit* yang menggunakan bahasa pemrograman Lua untuk membantu programmer dalam membuat *prototype* produk IoT atau bisa dengan memakai *sketch* dengan Arduino IDE. Pengembangan *Kit* ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire dan ADC (*Analog to Digital Converter*) semua dalam satu *board*. *NodeMCU* berfungsi sebagai pusat pengendali utama dari sistem kerja alat. Untuk spesifikasi dari *NodeMCU* ESP8266 berikut ditampilkan pada table dibawah ini :

Tabel 3.1 Spesifikasi *NodeMCU*

| Spesifikasi | <i>NodeMCU</i> |
|-----------------------|-----------------------|
| Mikrokontroler | ESP8266 |
| Ukuran <i>Board</i> | 57 mm x 30 mm |
| Tegangan <i>Input</i> | 3.3 ~ 5V |
| GPIO | 13 pin |
| Kanal PWM | 10 Kanal |
| 10 bit ADC Pin | 1 Pin |
| <i>Flash Memory</i> | 1 MB |
| <i>Clock Speed</i> | 40/26/24 MHz |
| <i>Wifi</i> | IEEE 802.11 b/g/n |
| Frekuensi | 2.4 GHz – 22.5 GHz |
| USB <i>Port</i> | <i>Micro</i> USB |

| | |
|--------------------------------|-----------|
| <i>Card Reader</i> | Tidak ada |
| <i>USB to Serial Converter</i> | CH340G |

2) Laptop

Laptop digunakan sebagai media pembuatan kode program untuk sistem alat, dan untuk mengukur kualitas dari sistem alat yang telah dibuat. Berikut spesifikasi laptop yang digunakan.

Tabel 3.2 Spesifikasi Laptop

| Spesifikasi | Data |
|-------------------------|-----------------------|
| <i>Manufacture</i> | FUJITSU |
| Model | LIFEBOOK LH532 |
| <i>Processor</i> | Intel® Core™ i5 3230M |
| RAM | 8 GB |
| <i>Operating System</i> | Windows 10 Pro 64-bit |

3) Smartphone

Pada perancangan ini penulis juga menggunakan *smartphone android* yang berfungsi untuk *monitoring* alat dan melihat data yang dihasilkan dari alat yang dibuat. Pada *smartphone* nantinya akan diinstall aplikasi yang telah dibuat untuk bisa memonitoring alat dan melihat hasil data dari alat. Berikut spesifikasi *smartphone* yang digunakan.

Tabel 3.3 Spesifikasi Smartphone

| Spesifikasi | Data |
|-------------------------|------------------------------|
| <i>Chipset</i> | Exynos 7570 Quad |
| CPU | Quad-core 1,4 GHz Cortex-A53 |
| GPU | Mali-T720 MP2 |
| RAM | 2 GB |
| <i>Operating System</i> | Android 9 (Pie) |

4) Sensor Warna TCS3200

Pada perancangan ini menggunakan sensor warna sebagai masukan pada alat, dimana sensor warna akan mendeteksi warna yang digunakan sebagai kunci untuk

bisa membuka pintu lemari. Pada prosesnya sensor warna akan mendeteksi urutan warna pada kunci yang digunakan, jika warna yang terdeteksi benar maka kunci pada akan terbuka. Namun jika urutan warnanya tidak sesuai maka kunci tidak akan terbuka.

5) *Relay*

Pada perancangan tugas akhir ini menggunakan *Relay* sebagai saklar (Switch) yang dioperasikan secara listrik dan terdiri dari 2 bagian utama yaitu *electromagnet (coil)* dan mekanikal (*switch*). *Relay* berfungsi untuk menjalankan *logic function* atau fungsi logika. Pada implementasinya nanti *relay* akan memberikan daya ke *solenoid lock* untuk membuka kunci, jika urutan warna yang terbaca oleh sensor warna sudah benar. Dan juga *relay* akan menyalakan *buzzer* jika sensor magnet MC-38 terlepas secara paksa.

6) *Solenoid Lock*

Kemudian pada perancangan menggunakan *Solenoid Lock* yang berfungsi sebagai pengganti dari kunci konvensional. Untuk cara kerjanya, *solenoid lock* saat keadaan normal *solenoid lock* dalam keadaan memanjang atau terkunci, untuk dapat membukanya warna yang terbaca harus benar, jika pembacaan warna sudah benar maka *relay* akan memberikan daya ke *solenoid lock* dan kunci akan terbuka.

7) Sensor Magnet MC-38

Pada perancangan ini juga menggunakan sensor magnet MC-38 sebagai pengaman dobel dari *solenoid lock*. Pada dasarnya *magnetic switch* adalah dua batang magnet yang pada kondisi normalnya akan saling menempel, namun jika dialiri listrik kedua magnet tersebut akan saling melepaskan. Pada perancangan ini digunakan sebagai pengaman dobel dari *solenoid lock*, dikarenakan juga dapat digunakan sebagai indikator jika pintu dibuka secara paksa, jika 2 batang magnet terlepas secara paksa dan *solenoid lock* masih dalam keadaan terkunci, maka sistem akan menyalakan *buzzer* dan mengirimkan notifikasi ke *user* melalui aplikasi sebagai peringatan bahwasanya pintu telah terbuka dengan paksa.

8) *Buzzer*

Selanjutnya dalam pembuatan proyek tugas akhir ini penulis menggunakan *buzzer active* sebagai hasil *output* sistem rancang bangun. *Buzzer* disini sebagai *alarm* pengingat atau sebagai peringatan bahwa adanya percobaan untuk membuka

pintu secara paksa. *Buzzer* akan menyala jika *magnetic switch* terbuka secara paksa. Di saat *buzzer* berbunyi aplikasi juga akan memberikan notifikasi peringatan ke *user* untuk memberikan info mengenai kondisi keamanan lemari

9) *Smartphone*

Pada perancangan proyek tugas akhir ini membutuhkan sumber daya untuk dapat beroperasi, oleh karena itu penulis menggunakan *power supply* yang berfungsi sebagai pemasok daya. Untuk besaran daya yang dibutuhkan sebesar *volt*

3.3.2 Perangkat Lunak (*Software*)

1) *Software* Arduino IDE

IDE (Integrated Development Environment) adalah sebuah software yang sangat berperan untuk menulis program yang bersifat *open source* dan melakukan compile menjadi kode biner dan meng-upload ke dalam memory microcontroller. Bahasa yang digunakan Arduino IDE adalah bahasa pemrograman JAVA. Arduino IDE dilengkapi dengan *library C/C++* yang umumnya disebut *Wiring* bertujuan untuk membuat operasi input dan output menjadi lebih mudah. IDE dapat menulis program secara bertahap.

2) *Firebase*

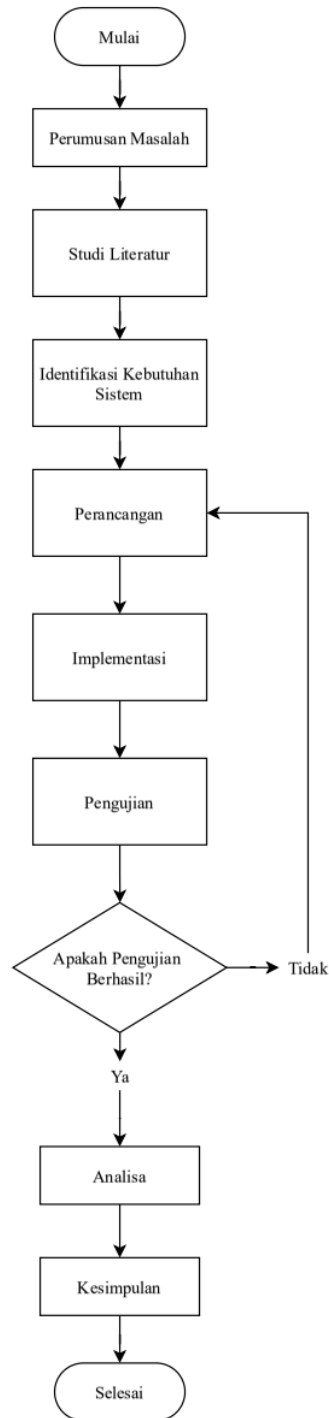
Firebase merupakan satu layanan yang disediakan oleh *Google* yang bertujuan untuk memudahkan para *app developer* untuk dapat mengembangkan aplikasi mereka. Penggunaan *firebase* pada perancangan tugas akhir ini sebagai *database* yang di-host di *cloud*. Pada *firebase* data akan disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang telah terhubung.

3) *Mit App Inventor*

Pada perancangan proyek tugas akhir ini penulis menggunakan *Mit App Inventor*. *Mit App Inventor* merupakan *platform* yang digunakan untuk pembuatan aplikasi *android*. Alasan menggunakan *Mit App Inventor* dikarenakan mudahnya untuk membuat aplikasi sederhana tanpa harus mempelajari dan menggunakan bahasa pemrograman yang terlalu banyak.

3.2 ALUR PENELITIAN

Pada perancangan proyek tugas akhir ini ada 8 tahapan yaitu perumusan masalah, studi literatur, identifikasi kebutuhan sistem, perancangan, implementasi, pengujian. Berikut *flowchart* dari alur penelitian ini.

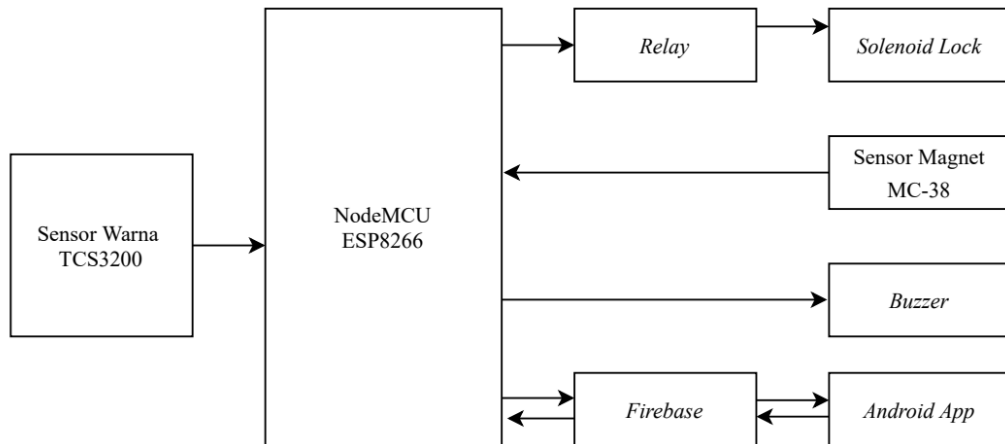


Gambar 3.1 *Flowchart* alur penelitian

Berikut penjelasan secara detail setiap alur sebagai berikut:

- 1) Perumusan masalah, merupakan dasar dari dibuatnya dan dilakukannya penelitian ini, dan sudah dibahas pada bab 1.
- 2) Studi literatur, sebelum lanjut ke tahap selanjutnya, terlebih dahulu melakukan studi dari berbagai sumber, di antaranya mengenai, mempelajari *Internet of Things*, mempelajari mengenai sensor warna dan beberapa sensor yang akan digunakan pada penelitian, mempelajari mengenai *software* Arduino IDE, dan lainnya.
- 3) Identifikasi kebutuhan sistem, pada tahap ini akan dilakukan identifikasi terhadap hal-hal yang dibutuhkan untuk sistem yang nantinya akan dibangun. Hasil identifikasi ini yang akan dijadikan acuan untuk mengembangkan alat dan aplikasi seperti apa yang akan diimplementasikan pada sistem ini. Pada tahap ini juga dilakukan Analisa terkait aplikasi/*software* apa saja yang diperlukan dan komponen-komponen elektronik apa saja yang dibutuhkan untuk proyek tugas akhir ini.
- 4) Perancangan, langkah selanjutnya setelah identifikasi kebutuhan sistem adalah perancangan. Perancangan terbagi ke dua proses yaitu perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*).
- 5) Implementasi, pada tahap ini perangkat yang sudah dirancang kemudian akan diimplementasikan menjadi sebuah sistem yang nantinya dapat digunakan dengan baik dan benar.
- 6) Pengujian, pada tahap ini perangkat yang sudah dirancang kemudian akan dilakukan testing, hasil yang didapatkan dari pengujian akan menentukan kelayakan apakah sudah dapat berkerja dengan baik dan benar atau belum, jika didapatkan kesalahan dan kekurangan maka akan balik tahap perancangan untuk memperbaiki terlebih dahulu.
- 7) Analisa, setelah dilakukan uji coba dan sistem berfungsi dengan baik, maka dilakukan analisa terhadap hasil pengujian yang didapatkan.
- 8) Kesimpulan, setelah dilakukan analisa, maka akan didapat sebuah kesimpulan bahwa sistem dapat bekerja dengan baik, kemudian dari hasil tersebut dibuat laporan penelitian

3.3 DIAGRAM BLOK SISTEM



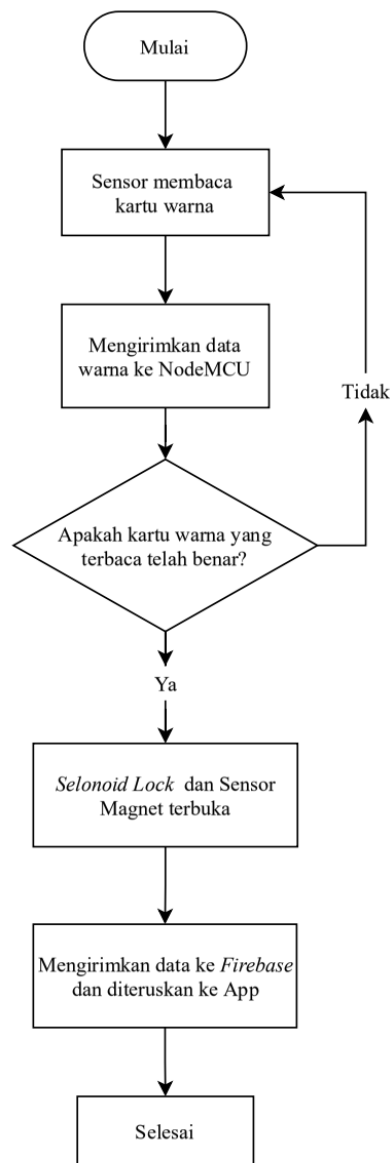
Gambar 3.2 Blok Diagram sistem

Berdasarkan gambar blok diagram sistem di atas terdapat 3 bagian yaitu *input*, proses dan *output*. Untuk bagian *input* terdapat sensor warna TCS3200 yang berfungsi untuk mendeteksi warna yang digunakan sebagai kunci untuk bisa membuka pintu lemari. Pada prosesnya sensor warna akan mendeteksi warna kartu yang berfungsi sebagai kunci, warna yang digunakan sebagai kunci adalah warna merah. Pada bagian pemrosesan terdapat pada *NodeMCU* ESP8266, yang berfungsi untuk mengendalikan seluruh sistem, dan di dalamnya terdapat program yang mengatur seluruh proses berjalannya sistem, juga menghubungkan sistem dengan jaringan internet.

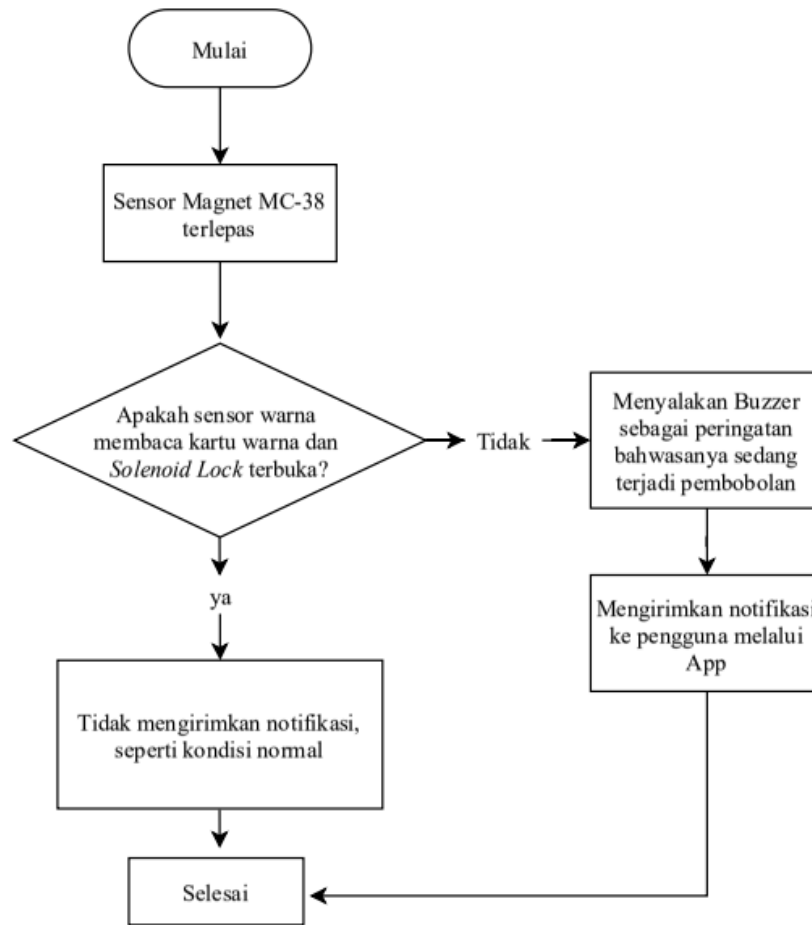
Pada bagian *output* terdapat *relay* yang berfungsi sebagai saklar otomatis yang terhubung dengan sensor magnet MC-38, *solenoid lock*, dan *buzzer*. *Solenoid lock* dan sensor magnet MC-38 akan terbuka jika warna yang terbaca benar, dan *buzzer* akan menyala jika sensor magnet mc-38 terlepas dengan paksa. Dan juga pada bagian *output* terdapat *firebase*, setelah data yang masuk diproses oleh *NodeMCU* ESP8266 maka data tersebut akan dikirimkan ke *firebase*, di *firebase* nantinya data akan disimpan dan data dapat diakses melalui aplikasi android, sehingga memudahkan pengguna untuk memonitoring kondisi dari kunci dan dapat memberikan notifikasi jika pintu terbuka secara paksa.

3.3.1 Flowchart Alur Sistem

Pada *flowchart* alur penelitian ini akan menggambarkan bagaimana alur dari sistem keseluruhan pada alat yang dirancang. Pada alat perancangan keseluruhan sistem akan dibuat dengan menggunakan sensor warna tcs3200, *NodeMCU* ESP8266, *relay*, *Solenoid lock*, sensor magnet MC-38, *buzzer*, dan beberapa komponen elektronik pembantu lainnya. Untuk *flowchart* dapat dilihat pada gambar di bawah ini.



Gambar 3.3 *Flowchart* alur sistem saat kondisi normal



Gambar 3.4 Flowchart alur sistem saat kondisi darurat

Pada alat terdapat 2 kondisi yaitu kondisi alat bekerja secara normal pemakaian dan kondisi darurat. Pada kondisi normal sensor warna akan membaca warna yang digunakan sebagai kunci, jika warna yang terbaca benar maka *NodeMCU* akan memberikan perintah pada sensor magnet dan *solenoid lock* untuk terbuka, setelah terbuka maka proses tersebut akan dilaporkan ke aplikasi android melalui perantara *firebase*. Namun, jika warna yang terbaca salah maka *solenoid lock* dan sensor magnet tidak akan terbuka sampai warna yang terbaca sudah benar.

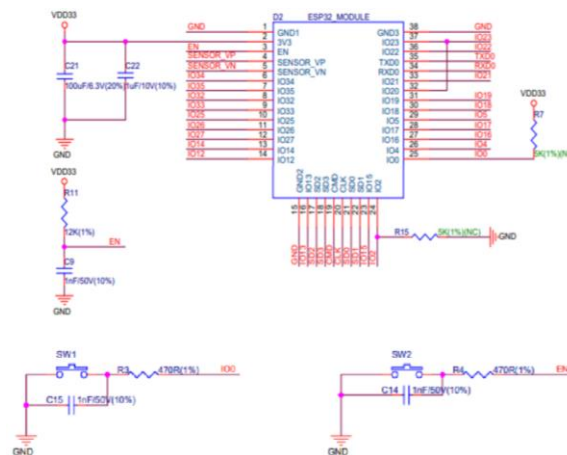
Sedangkan pada kondisi darurat di mana sensor magnet terbuka secara paksa maka *NodeMCU* akan memberikan perintah untuk menyalakan *buzzer* sebagai peringatan dan akan mengirimkan notifikasi ke pengguna melalui aplikasi bahwasanya alat dalam kondisi darurat.

3.3.2 Perancangan Perangkat Keras (*Hardware*)

Setelah membuat diagram blok, mengetahui fungsi serta komponen apa saja yang dibutuhkan dan mengetahui alur sistem, maka tahap selanjutnya adalah perancangan perangkat keras (*hardware*). Dalam perancangan *hardware* dilakukan beberapa proses, diantaranya adalah perancangan rangkaian masing-masing komponen, dan pengkabelan.

3.3.2.1 Rangkaian skematik *NodeMCU*

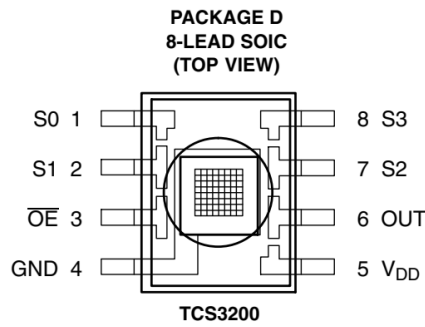
Berikut adalah gambar rangkaian skematik dari *NodeMCU* yang ditampilkan pada Gambar 3.4.



Gambar 3.5 Rangkaian Skematik *NodeMCU* [9]

3.3.2.2 Rangkaian skematik sensor warna TCS3200

Berikut adalah gambar rangkaian dari sensor warna TCS3200 yang ditampilkan pada Gambar 3.5. Sensor warna di sini akan berfungsi sebagai pendeteksi warna yang akan digunakan sebagai kunci. Pada prosesnya sensor akan mendeteksi warna, jika warna yang terdeteksi benar maka kunci akan terbuka, untuk warna yang digunakan sebagai kunci adalah warna merah.



Gambar 3.6 Rangkaian Skematik Sensor Warna TCS3200 [12]

Adapun konfigurasi pin sensor warna ke *NodeMCU* akan ditunjukkan pada tabel di bawah ini.

Tabel 3.4 Konfigurasi Pin Sensor Warna TCS3200

| <i>Sensor Warna</i> | <i>NodeMCU</i> |
|---------------------|----------------|
| S0 | D4 |
| S1 | D5 |
| S2 | D6 |
| S3 | D7 |
| GND | GND |
| VCC | 3V |
| OUT | D8 |

3.3.2.3 Rangkaian skematik *Relay*

Berikut adalah konfigurasi pin *relay* ke *NodeMCU* akan ditunjukkan pada tabel di bawah ini. Nantinya *relay* akan terhubung dengan *solenoid lock*, sensor magnet MC-38, dan *buzzer*. Sesuai penjelasan sebelumnya pada alur sistem, nantinya *relay* akan beroperasi dalam 2 kondisi, kondisi normal dan kondisi darurat. Pada kondisi normal *relay* hanya mengaktifkan *solenoid lock* dan sensor warna TCS3200, namun jika dalam kondisi darurat maka *relay* akan menyalakan *buzzer* sebagai peringatan, dikarekan sensor magnet MC-38 terlepas secara paksa.

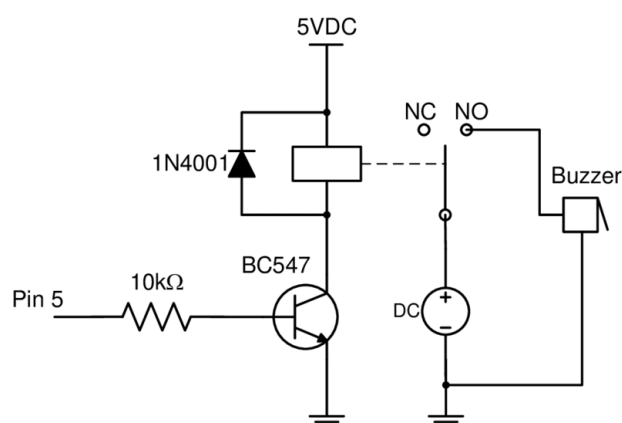
Tabel 3.5 Konfigurasi Pin *Relay*

| <i>Relay</i> | <i>NodeMCU</i> |
|--------------|----------------|
| <i>IN</i> | D3 |
| VCC | 5V |

| | |
|-----|-----|
| GND | GND |
|-----|-----|

3.3.2.4 Rangkaian *buzzer*

Berikut adalah gambar rangkaian skematik dari *buzzer* yang ditampilkan pada gambar di bawah ini. *Buzzer* pada rangkaian hanya aktif jikalau dalam kondisi darurat. Hal ini disebabkan *buzzer* difungsikan sebagai pengingat jika sensor magnet terlepas secara paksa tanpa adanya proses pembacaan warna.



Gambar 3.7 Rangkaian Skematik *Buzzer* [17]

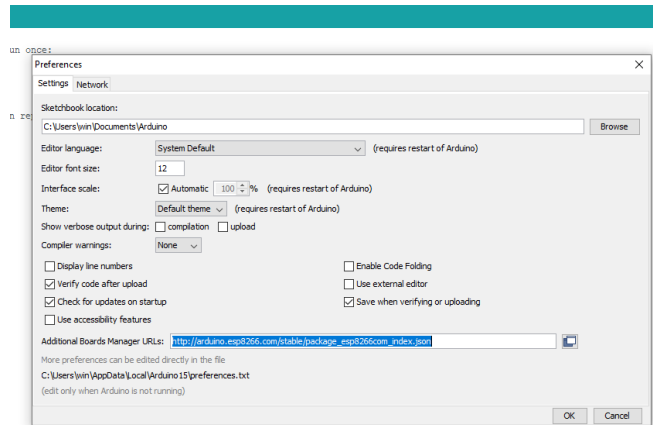
3.3.3 Perancangan Perangkat Lunak (*Software*)

Setelah melakukan proses perancangan perangkat keras selesai, maka tahapan selanjutnya adalah perncangan perangkat lunak (*software*). Perancangan ini mencakup pembuatan kode program yang nantinya akan di-*upload* ke *NodeMCU*, dan data yang didapatkan dari proses di *NodeMCU* akan disimpan pada *firebase* dan perancangan aplikasi android melalui *Mit App Inventor*.

3.3.3.1 Perancangan pada *Arduino IDE*

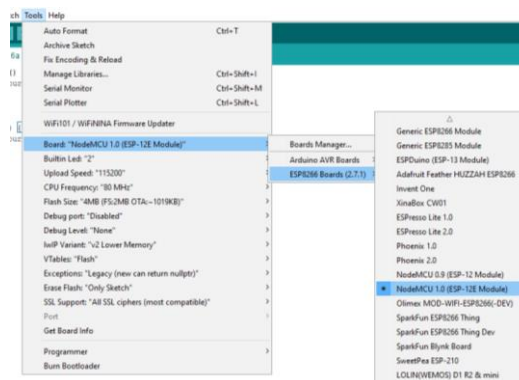
Untuk pembuatan kode program dilakukan di *software* *Arduino IDE*. Pada *Arduino IDE* dapat membuat, membuka dan mengedit *source code Arduino* atau biasa disebut dengan *sketches*. Untuk Langkah awalnya adalah mengatur *Arduino IDE*nya terlebih dahulu, karna pada saat kondisi *default*, *Arduino IDE* belum memiliki pilhan untuk *board NodeMCU*, maka dari itu perlu ditambahkan secara manual. Berikut untuk mengatur *Arduino IDE*, pertama-tama buka program

Arduino IDE, setelah itu lanjut ke menu *file* untuk membuka *preference*. Pada jendela *preference* di bagian bawah nanti akan ada *link* pada kolom *Additional Boards Manager URLs* kemudian pada kolom *link* tersebut paste *link* berikut: http://arduino.esp8266.com/stable/package_esp8266com_index.json seperti pada gambar di bawah ini. Kemudian klik *ok*, tahap selanjutnya adalah meng-*upgrade boardnya*.



Gambar 3.8 Tampilan pada jendela *preference* di *software* Arduino IDE

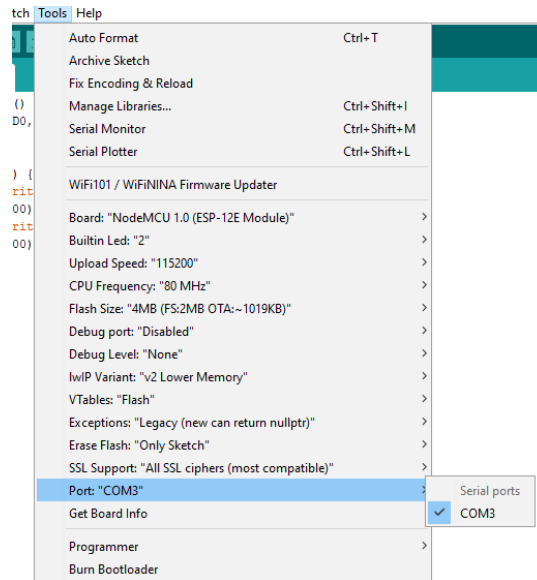
Untuk *upgrade board* buka *tools – board – boards manager*, pada *boards manager* cari “ESP8266” kemudian *install*. Setelah proses tersebut selesai maka pada *submenu board* akan terdapat *board NodeMCU* yang digunakan, seperti pada gambar di bawah ini.



Gambar 3.9 Tampilan pada *Tools* di *software* Arduino

Setelah itu adalah menambahkan *library* dari komponen-komponen yang akan digunakan. Dan untuk pengaturan *port* yang digunakan terlebih dahulu

menyambungkan laptop dan *NodeMCU* dengan kabel usb, setelah terhubung dan terbaca maka tampilan pada *port* di *menu tools* akan seperti gambar dibawah ini.

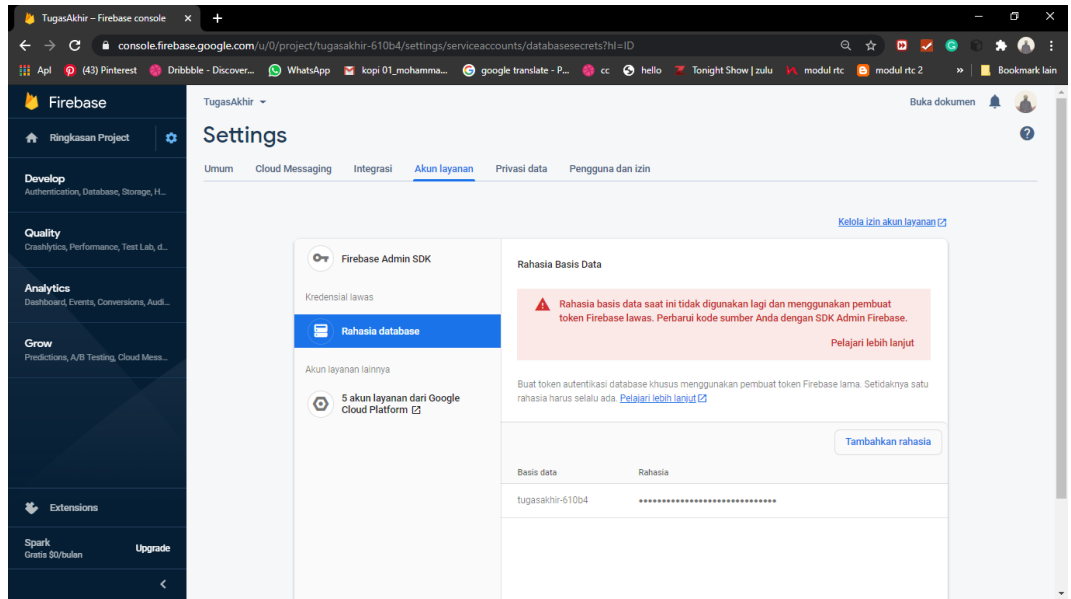


Gambar 3.10 Tampilan pada *Port* di *Menu Tools*

Setelah semua yang diatas telah dilakukan maka tinggal membuat program perancangan sesuai dengan cara kerja yang sudah dijelaskan pada diagram alur sistem.

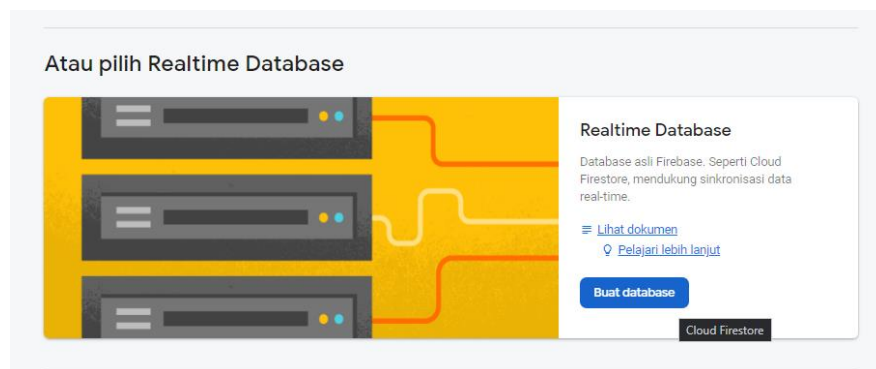
3.3.3.2 Pengaturan pada *Firebase*

Setelah kode program sudah dibuat dan tidak terdapat *error* di dalamnya Langkah selanjutnya adalah menyambungkan dengan *firebase*. Tujuan disambungkannya antara *NodeMCU* dengan *firebase* adalah agar data yang diperoleh dapat dimonitoring melalui aplikasi android. Untuk pertama-tama harus memiliki akun *firebase* terlebih dahulu, kemudian membuat *database* yang bersifat *real time*. Ada 2 informasi yang dibutuhkan dari *database* ini yang nantinya akan digunakan untuk pemrograman *board*, yaitu URL dari *database host* dan kunci autentifikasi keamanan. Untuk mendapatkan kedua hal tersebut pada *firebase* pilih *overview* – stelkan proyek, kemudian pilih akun layanan dan klik pada rahasia *database*. Pada tampilannya akan ada *firebase auth* verifikasi kode. Seperti pada gambar di bawah ini.

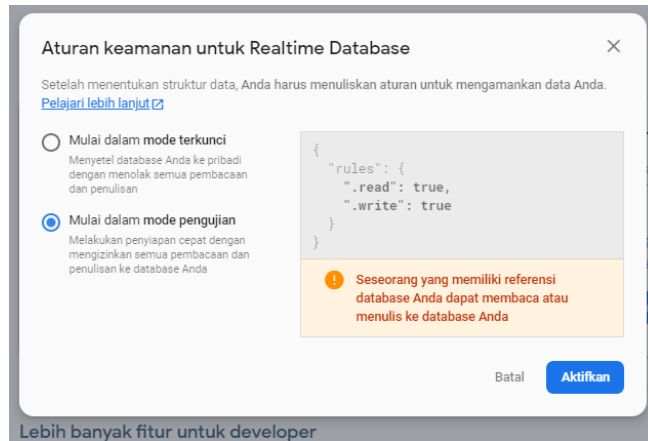


Gambar 3.11 Tampilan rahasia database pada *firebase*

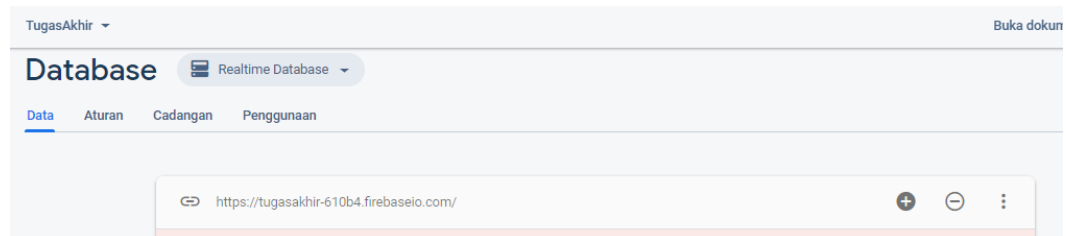
Lanjut kelangkah selanjutnya adalah klik *Develop* dan pilih *Database*, nanti pada laman *database* pilih *realtime database* dan klik buat *database*. Kemudian akan muncul *pop-up* yang memberikan dua pilihan, pilih mode dalam pengujian test. Dan nantinya akan terdapat *firebase host link* yang dibutuhkan. Setelah semuanya sudah siap nantinya pada saat pembuatan program di Arduino IDE perlu ditambahkan *firebase host link* dan *firebase auth* verifikasi. Berikut gambar dari prosesnya.



Gambar 3.12 Membuat *realtime database* pada *firebase*



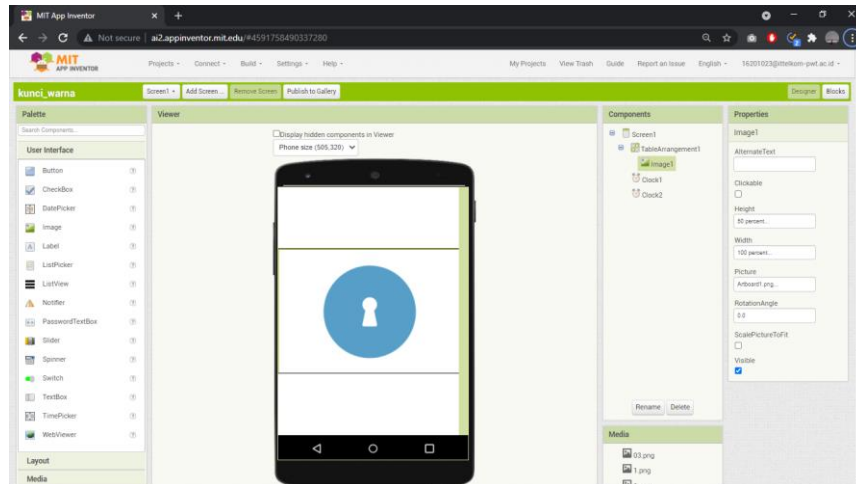
Gambar 3.13 Memilih *mode* dalam pengujian *test*



Gambar 3.14 Tampilan *firebase host link*

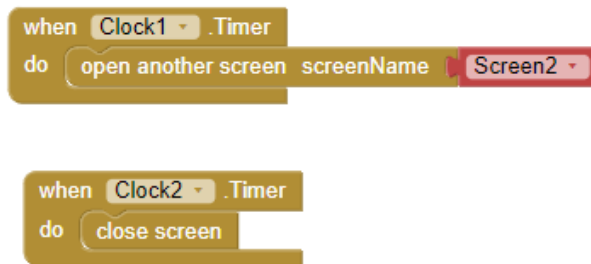
3.3.3.3 MIT APP *Inventor*

MIT APP *Inventor* merupakan aplikasi web *open source* yang pada awalnya dikembangkan oleh Google, dan untuk saat ini dikelola oleh Massachusetts Institute of Technology (MIT). Pada Mit App *Inventor* dapat memudahkan pengguna yang baru memulai untuk membangun aplikasi yang dapat beroperasi di Android. Hal ini dikarenakan pada Mit App *Inventor* untuk membangun aplikasinya menggunakan sitem *block* yang serupa dengan *puzzle*, sehingga sangat memudahkan penggunaanya. . Berikut tampilan awal pada Mit App *Inventor*.



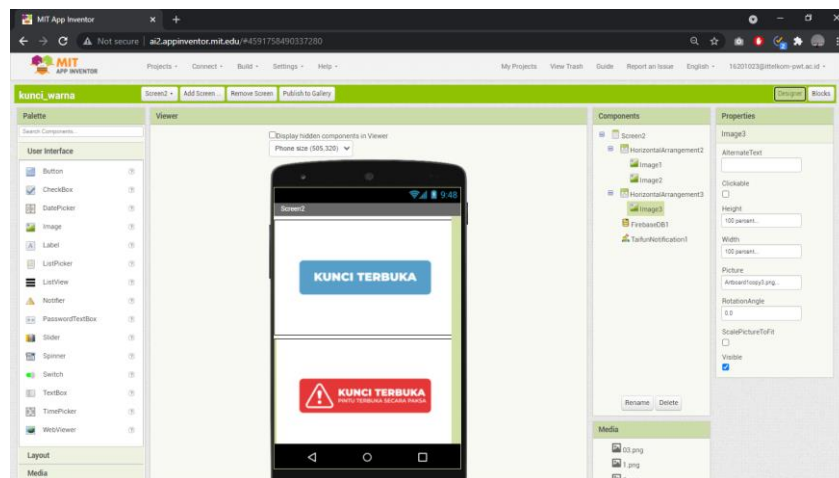
Gambar 3.15 Tampilan pada MIT APP Inventor

Pada gambar 3.10 adalah tampilan awal yang hanya terdapat tampilan logo aplikasi.



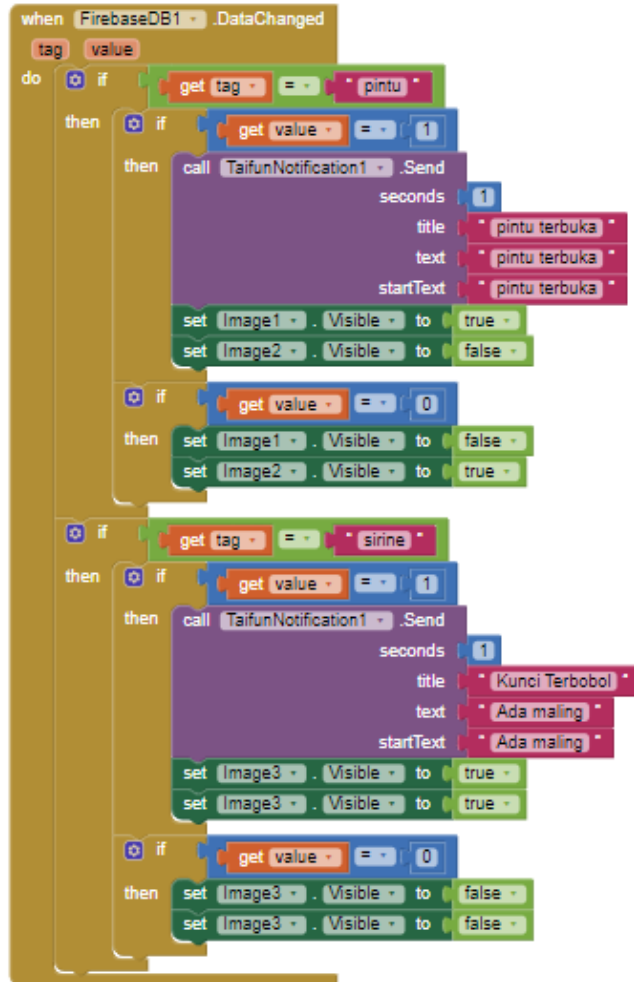
Gambar 3.16 Block pada screen 1

Pada gambar 3.11 merupakan *Blocks* atau program yang digunakan untuk mengalihkan tampilan dari *Screen 1* menuju *screen 2*.



Gambar 3.17 Tampilan pada screen 2

Pada gambar 3.12 merupakan tampilan pada *screen 2* sebagai halaman monitoring kondisi kunci apakah sedang terbuka atau terkunci, juga dapat menampilkan peringatan jika pintu dibuka dengan paksa.



Gambar 3.18 *Block* pada *screen 2*

Pada gambar 3.13 adalah *blocks* program yang dibuat untuk sistem monitoring kondisi kunci. Dengan nilai mengambil dari *database firebase realtime*. Dan akan memunculkan informasi kondisi kunci apakah terbuka atau tertutup dan akan memunculkan notifikasi jika pintu dibuka dengan paksa.