

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Pada Penelitian Asmah Akhriana, dan Irmawati pada tahun 2019 yang berjudul “Sistem Keamanan Pintu *Locker* dengan Memanfaatkan *Keypad* dan E-KTP Berbasis Arduino” meneliti tentang perancangan suatu sistem keamanan pintu lemari *locker* dengan memanfaatkan E-KTP dan *keypad* berbasis Arduino Uno R3. Pada sistem ini memanfaatkan RFID *Reader* dengan frekuensi 13,56 MHz dan Mikrokontroler ATmega328 yang berfungsi sebagai rangkaian pengontrol atau pengendali. Dan pada hasil dari penelitian RFID *Reader* mampu mendeteksi E-KTP dengan jarak maksimal 2,5 cm, dan solenoid akan mengunci kembali setelah 3 detik. Selain itu terdapat *keypad* yang dapat menjadi alternatif dari kunci manual karena E-KTP memiliki kode unik yang berbeda sehingga lebih praktis dan efisien untuk sistem yang membutuhkan verifikasi [2].

Pada Penelitian Ahmad Zaky Qolbuddin, Marlinda Ike Sari dan Rini Handayani pada tahun 2018 yang berjudul “Perancangan dan Implementasi Sensor Warna Untuk Kunci Elektrik” meneliti tentang perancangan kunci elektrik menggunakan sensor warna tcs3200, *solenoid door lock* dan Arduino Uno. Pada sistem ini memanfaatkan Arduino Uno sebagai mikrokontroler, sensor warna tcs3200 untuk mendeteksi nilai warna RGB (*Red, Green and blue*), LCD untuk menampilkan warna yang terbaca dan *solenoid door lock* sebagai pengunci pintu secara elektronik. Pada penelitian ini akan menggunakan urutan warna sebagai kunci dan jika urutan warna yang terbaca benar maka *solenoid door lock* akan terbuka dan LCD akan menampilkan tulisan “pintu terbuka” [5].

Pada penelitian Annisya, Lingga Hermanto, dan Robby Candra pada tahun 2017 yang berjudul “Sistem Keamanan Buka Tutup Kunci Brankas Menggunakan Sidik Jari Berbasis Arduino Mega” menjelaskan tentang sistem keamanan untuk brankas menggunakan sidik jari berbasis Arduino Mega. Sistem ini dibuat dengan sumber tegangan +5V DC untuk sensor sidik jari, *micro switch*, LCD, LED, *buzzer*,

dan +12V DC untuk *solenoid*. Pada pengaplikasiannya sidik jari dan penekanan pada *switch* berfungsi sebagai masukan, Arduino Mega sebagai Mikrokontroler, serta *solenoid*, LCD, LED, dan *buzzer* berfungsi sebagai media keluarannya. Untuk mengakses kunci diperlukan sidik jari yang telah terdaftar pada sistem, jika sidik jari telah terdaftar ke sistem maka *solenoid* akan terbuka, namun jika sidik jari tidak terdaftar pada sistem, maka LED merah akan menyala dan *buzzer* akan berbunyi [6].

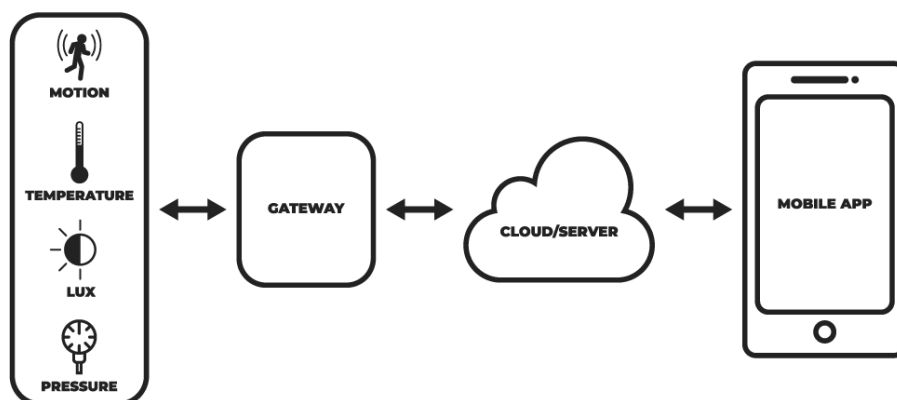
Pada penelitian Sumardi Sadi dan Muhammad Yoga Mulya Pratama pada tahun 2017 yang berjudul “Sistem Keamanan Buka Tutup Kunci Brankas Menggunakan *bluetooth* HC – 05 Berbasis Arduino Mega 2560” menjelaskan tentang perancangan sistem keamanan brankas yang *modern* dengan sistem keamanan buka tutup pintu menggunakan aplikasi *smartphone* berbasis Arduino Mega 2560. Untuk dapat terhubungnya antara Arduino dengan *smartphone* pada penelitian ini menggunakan *bluetooth module* HC-05. Dan untuk proses membuka brankas, pengguna harus menghubungkan terlebih dahulu antara *smartphone* dengan Arduino melalui *bluetooth*, setelah terhubung pengguna harus mengetahui *password* brankas untuk bisa membuka pintu brankas tersebut. Jika *password* yang dimasukkan benar maka Arduino akan memerintahkan motor *servo* untuk membuka pengunci pintu brankas. Hasil dari pengujian alat dapat berfungsi dengan baik [3].

Pada penelitian M.Agfar Dismawan pada tahun 2019 yang berjudul “Rancang Bangun Sistem Pengendalian dan Monitoring Keamanan Pintu Berbasis IoT (*Internet of Things*) dengan Menggunakan Aplikasi *Blynk* di Ponsel *Android*” menjelaskan mengenai perancangan sebuah sistem keamanan pintu rumah yang mampu dikendalikan melalui aplikasi *blynk* di ponsel *android* selain itu dapat diakses menggunakan RFid dan *led button*, serta bentuk akses keamanan pintu rumah dapat dimonitoring melalui aplikasi *blynk* dan dapat menerima pesan notifikasi pesan teks melalui *Gmail*. Pada hasil pengujian didapatkan bahwa alat bekerja dengan baik dan apabila sistem keamanan terputus dari jaringan internet maka alat akan beroperasi secara *offline* dan akan mengirimkan notifikasi ke pengguna melalui aplikasi *blynk*, meski demikian alat masih bisa diakses melalui RFid dan *led button* [4].

2.2 DASAR TEORI

2.2.1 *Internet of Things*

Internet of things atau biasa disingkat menjadi IoT adalah suatu jaringan yang menghubungkan berbagai objek yang memiliki identitas pengenal serta alamat IP, sehingga antar objek tersebut dapat saling berkomunikasi dan bertukar informasi mengenai dirinya maupun lingkungan yang diinderanya. Objek-objek dalam IoT dapat menggunakan maupun menghasilkan layanan-layanan dan saling bekerja sama untuk mencapai suatu tujuan bersama. *Internet of Things* (IoT) pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999. Meski telah diperkenalkan sejak 15 tahun yang lalu, hingga kini belum ada sebuah konsensus global mengenai definisi IoT. Namun secara umum konsep IoT juga dapat diartikan sebagai sebuah kemampuan yang dapat menghubungkan objek-objek cerdas dan memungkinkannya untuk berinteraksi dengan objek lain, lingkungan maupun dengan peralatan komputasi cerdas lainnya melalui jaringan internet. IoT dalam berbagai bentuknya telah mulai diaplikasikan pada banyak aspek kehidupan manusia. Dengan meluasnya pengaplikasian berbagai teknologi IoT, dapat membawa perubahan terhadap kehidupan manusia menjadi jauh lebih nyaman. Dari sisi pengguna perorangan, IoT sangat terasa pengaruhnya dalam bidang domestik seperti pada aplikasi mobil dan rumah cerdas. Dari sisi pengguna bisnis, IoT sangat membantu dalam meningkatkan jumlah produksi dan kualitas produksi, mengawasi distribusi barang, mencegah pemalsuan, mempersingkat waktu ketidaktersedian barang pada pasar retail, manajemen rantai pasok, dsb.



Gambar 2.1 Blok sistem IoT [7]

Secara sederhana konsep IoT dapat digambarkan seperti pada Gambar 2.1. Pada tingkat pertama adalah *hardware* yang dapat mengenali dirinya dan mengindra lingkungannya, gerakan mesin, kondisi kesehatan, kondisi cuaca, membaca lokasi, dan sebagainya. Biasanya untuk perangkat yang digunakan pada tingkat pertama ini berupa berbagai jenis sensor, kontrol, RFID dan aktuator. Pada lapisan atau tingkat kedua merupakan *gateway*, yang berfungsi sebagai penghubung antara jaringan internal sensor yang mengumpulkan data, dengan jaringan luar internet melalui berbagai macam media komunikasi nirkabel seperti selular satelit, *Zigbee*, WiFi, *Bluetooth*, dan lain-lain. *Gateway* juga merupakan tempat pengolahan data tahap pertama, pengaturan *routing* dan pengalamatan. Data yang ditransmisikan melalui *gateway* kemudian disimpan dan diolah di *cloud server* dengan mesin analitik *Big Data*. Data yang telah diolah kemudian akan digunakan untuk melakukan hal-hal cerdas sesuai tujuan IoT. Sedangkan pada sisi pengguna layanan IoT dapat dimanfaatkan melalui aplikasi bergerak pada perangkat cerdas mereka. Aplikasi bergerak yang intuitif ini dapat membantu pengguna untuk mengatur dan memonitor perangkatnya dari jarak jauh. Tulang punggung dari semua ekosistem IoT adalah IPv6, yang merupakan alamat pengenalan dari setiap perangkat yang terhubung dengan internet. Dengan adanya IPv6 yang mampu menyediakan 2¹²⁸ alamat, setiap perangkat yang terhubung dengan internet bukan hanya dapat dikenali secara geografi seperti IPv4, namun juga secara individu [7].

2.2.2 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. *NodeMCU* memiliki keunikan diantaranya adalah memiliki ukuran yang kecil, dengan panjang 4.83 cm, lebar 2.56 cm, dan berat 7 gram, meskipun demikian *NodeMCU* sudah dilengkapi dengan fitur *wifi* dan *firmware* yang bersifat *opensource*.



Gambar 2.2 Board NodeMCU ESP8266 [8]

Dan untuk penggunaan *NodeMCU* memiliki beberapa keuntungan diantaranya dari segi biaya maupun efisiensi tempat, karena *NodeMCU* yang ukurannya kecil, lebih praktis dan harganya jauh lebih murah dibandingkan dengan Arduino Uno. Arduino Uno sendiri merupakan salah satu jenis mikrokontroler yang banyak diminati dan memiliki bahasa pemrograman C++ sama seperti *NodeMCU*, namun pada Arduino Uno tidak memiliki modul wifi, dan untuk bisa menyambungkan Arduino Uno dengan jaringan internet diperlukan perangkat tambahan berupa wifi *shield* [8].

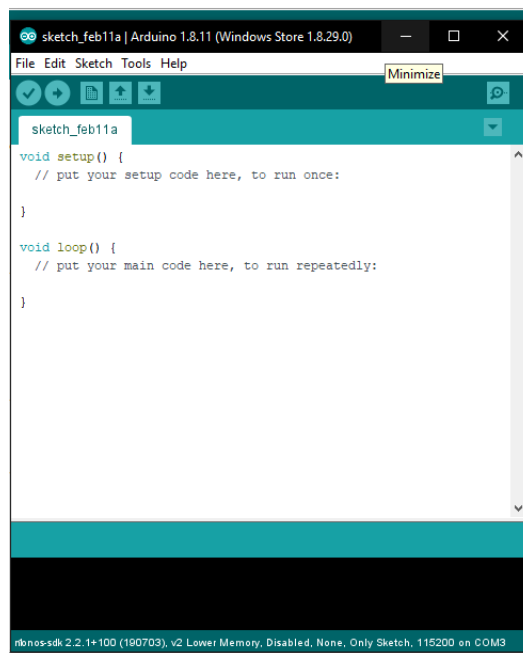
Di bawah ini merupakan spesifikasi dari *NodeMCU*:

Tabel 2.1 Spesifikasi *NodeMCU* [9]

Spesifikasi	<i>NodeMCU</i>
Mikrokontroler	ESP8266
Ukuran <i>Board</i>	57 mm x 30 mm
Tegangan <i>Input</i>	3.3 ~ 5V
Kanal PWM	10 kanal
GPIO	13 pin
10 bit ADC Pin	1 Pin
<i>Clock Speed</i>	40/26/24 MHz
<i>Flash Memory</i>	1 MB
<i>Wifi</i>	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 GHz
USB <i>Port</i>	<i>Micro</i> USB
USB to <i>Serial Converter</i>	CH340G

2.2.3 Arduino IDE

Arduino IDE berfungsi untuk membuat, membuka, dan mengedit program yang akan dimasukkan ke dalam *board* Arduino. Aplikasi ini dirancang agar memudahkan penggunaannya dalam membuat berbagai aplikasi. Arduino IDE memiliki struktur bahasa pemrograman yang sederhana dan fungsi yang lengkap, sehingga mudah untuk dipelajari oleh pemula sekalipun. Aplikasi Arduino Ide tersedia untuk beberapa sistem operasi computer diantaranya adalah Windows, Mac OS, Linux 32 *bits*, Linux 54 *bits*, dan Linux ARM [10].



Gambar 2.3 *Interface Software Arduino IDE*

Interface Arduino IDE tampak pada gambar 2.3 di atas. Dari kiri atas ke kanan dan bawah berikut bagian-bagian dari *software* Arduino IDE:

- a. *Verify*, pada versi sebelumnya dikenal dengan istilah *Compile*. Ini digunakan untuk memverifikasi *sketch* terlebih dahulu sebelum diupload ke *board*. Jika terdapat kesalahan pada *sketch* yang dibuat maka akan muncul *error*. Proses *verify* berfungsi mengubah *sketch* ke *binary code* untuk diupload ke mikrokontroler.
- b. *Upload*, untuk tombol *upload* berfungsi untuk mengupload *sketch* ke *board* Arduino. Walaupun sebelumnya tidak mengklik tombol *verify*, maka *sketch*

akan tetap *dcompile*, dan akan langsung diupload ke *board*. Berbeda dengan tombol *verify* yang hanya memverifikasi *source code* saja.

- c. *New Sketch*, berfungsi untuk membuka *window* dan membuat *sketch* baru.
- d. *Open Sketch*, berfungsi untuk membuka kembali *sketch* yang sudah pernah dibuat. *Sketch* yang sudah dibuat akan disimpan dengan ekstensi *file .ino*.
- e. *Save sketch*, berguna untuk menyimpan *sketch*, tapi tidak disertai dengan proses *compile sketch* terlebih dahulu.
- f. *Serial Monitor*, berfungsi untuk membuka *interface* untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya [10].

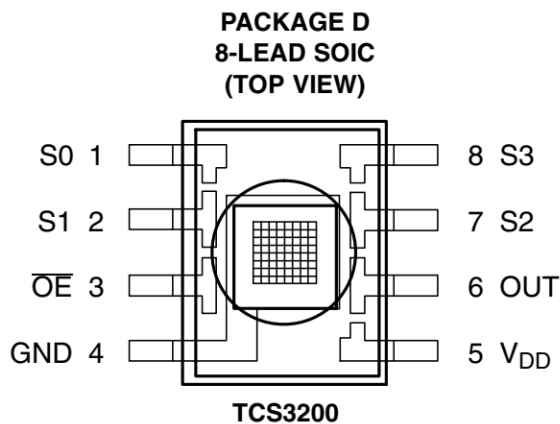
2.2.4 Sensor Warna

Sensor warna merupakan sensor yang digunakan pada aplikasi mikrokontroler untuk pendeteksian suatu objek benda atau warna dari objek yang dimonitor. TCS3200 pada dasarnya merupakan konverter yang diprogram untuk mengubah warna menjadi frekuensi yang tersusun atas konfigurasi *silicon photodiode* dan konverter arus ke frekuensi dalam IC CMOS *monolithic* yang tunggal. Keluaran dari sensor ini adalah gelombang kotak (*duty cycle 50%*) frekuensi yang berbanding lurus dengan intensitas cahaya (*irradiance*).



Gambar 2.4 Sensor Warna [11]

Seperti pada gambar 2.4, konverter cahaya ke frekuensi membaca sebuah *array 8x8* dari *photodiode*, 16 *photodiode* mempunyai penyaring warna merah, 16 *photodiode* mempunyai penyaring warna biru, 16 *photodiode* mempunyai penyaring warna hijau, dan 16 *photodiode* untuk warna terang tanpa penyaring.



Gambar 2.5 Skema Pin Sensor Warna [12]

Tabel 2.2 Fungsi Pin Sensor Warna 3200

Nama	No Pin IC	I/O	Fungsi Pin
GND	4	-	Sebagai <i>Ground</i> pada <i>power supply</i>
OE	3	I	<i>Output enable</i> , sebagai <i>input</i> untuk frekuensi skala rendah
<i>OUT</i>	6	O	Sebagai <i>output</i> frekuensi
S0, S1	1 dan 2	I	Sebagai saklar pemilih pada frekuensi <i>output</i> skala tinggi
S2, S3	7 dan 8	I	Sebagai saklar pemilih 4 kelompok dioda
Vdd	5	-	<i>Supply</i> tegangan

Pada sensor warna TCS3200 memiliki konfigurasi pin dengan fungsi yang berbeda setiap *pin* yang ada. 4 tipe warna dari *photodiode* telah diintergrasikan untuk meminimalkan efek ketidak seragaman dari insiden *irradiance*. Semua *photodiode* dari warna yang sama telah terhubung secara paralel. *Pin* S2 da S3 digunakan untuk memilih grup dari *pin* (merah, hijau, biru, dan jernih) yang telah aktif. Prinsipnya sensor warna melakukan proses pembacaan warna secara bertahap yaitu membaca frekuensi warna dasar, untuk itu diperlukan sebuah pengaturan atau pemrograman untuk memfilter tiap-tiap warna tersebut. Untuk prinsip kerja sensor warna tcs3200 adalah dengan cara membaca nilai intensitas cahaya yang

dipancarkan oleh led *super bright* terhadap objek, pembacaan nilai intensitas ini dilakukan melalui matrik 8x8 *photodiode*, dimana 64 *photodiode* dibagi menjadi 4 kelompok pembaca warna, setiap warna yang disinari led akan memantulkan sinar led menuju *photodiode*, pantulan sinar tersebut memiliki panjang gelombang yang berbeda-beda tergantung pada warna objek yang terdeteksi, hal ini yang menyebabkan sensor warna dapat membaca beberapa macam warna [11].

2.2.5 Solenoid Lock

Solenoid door lock merupakan salah satu *solenoid* yang difungsikan khusus sebagai pengunci pintu secara elektronik. *Solenoid door lock* mempunyai dua sistem kerja, yaitu *Normaly Close* (NC) dan *Normaly Open* (NO). Untuk perbedaan antar kedua sistem tersebut adalah jika cara kerja *solenoid* NC apabila diberi tegangan, maka *solenoid door lock* akan memanjang (tertutup). Dan untuk cara kerja dari *solenoid* NO adalah kebalikannya [13].

Solenoid lock berfungsi sebagai aktuator, biasanya alat ini dibuat khusus untuk pengunci pintu otomatis. *Solenoid* akan bergerak atau bekerja jika diberi tegangan. Umumnya *solenoid* yang dijual pasaran memerlukan tegangan sebesar 12 Volt, meskipun ada pula yang 6 Volt dan 24 Volt.



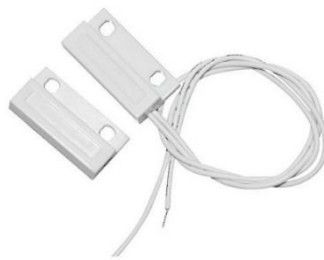
Gambar 2.6 Soleniod Lock

Untuk prinsip kerjanya *solenoid* sendiri adalah kondisi pada kondisi normal *solenoid* dalam posisi memanjang atau mengunci, dan jika solenoid diberi tegangan, maka tuas akan memendek atau terbuka. Di dalam *solenoid* terdapat kawat yang melingkar pada inti besi. Ketika arus listrik mengalir melalui kawat ini, maka yang terjadi medan magnet menghasilkan energi untuk menarik inti besi ke dalam [14].

2.2.6 Sensor Magnet MC-38

Sensor Magnet MC-38 adalah sensor yang digunakan untuk mendeteksi bukaan/tutupan pada pintu yang bekerja berdasarkan prinsip elektromagnetik. Dalam kondisi normal sensor dan magnet tidak dalam keadaan berdekatan yaitu saklar berada dalam kondisi terbuka (*open circuit*), sedangkan untuk kondisi aktif sensor dan magnet saling berdekatan atau saling menempel dan saklar berada dalam kondisi tertutup (*close circuit*) dengan nilai hambatan sebesar $\pm 4\Omega$. Pada komponen sensor terdapat kabel yang dihubungkan ke mikrokontroler, atau juga dapat digunakan sebagai saklar pada rangkaian elektronik lainnya [15].

Sensor magnet MC-38 merupakan saklar yang dapat merespon medan magnet yang berada disekitarnya. Sensor ini seperti halnya sensor *limit switch* yang diberikan tambahan plat logam yang dapat merespon adanya magnet. Sensor ini biasa digunakan untuk pengamanan pada pintu dan jendela [16].



Gambar 2.7 Magnetic Switch

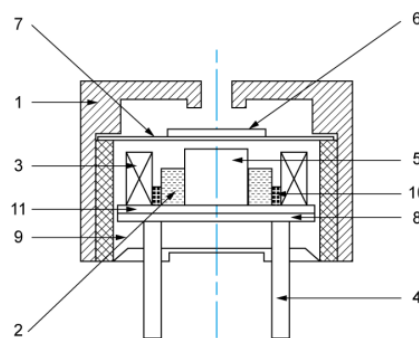
2.2.7 Buzzer

Buzzer merupakan komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Prinsip kerja *buzzer* yakni terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah

selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*). Adapun bentuk fisik dari *buzzer* [17].



Gambar 2.8 Buzzer



Gambar 2.9 Struktur buzzer elektromagnetik [18]

2.2.8 Adaptor

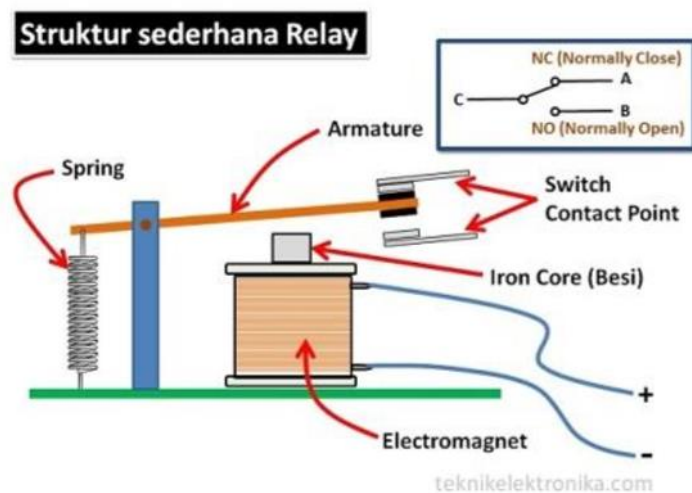
Adaptor merupakan sebuah perangkat yang berfungsi untuk mengubah tegangan listrik yang besar menjadi tegangan yang lebih kecil, atau rangkaian yang berfungsi untuk mengubah arus bolak-balik (arus AC) menjadi arus searah (arus DC). Adaptor biasanya digunakan untuk menurunkan tegangan AC 22 Volt menjadi kecil antara 3 Volt sampai 12 Volt sesuai kebutuhan alat elektronik yang digunakan. Untuk jenisnya terdapat 2 jenis berdasarkan system kerjanya, adaptor sistem *switching* dan sistem trafo *step down*. Pada sistem *switching* menggunakan teknik transistor maupun IC *switching*, untuk adaptor ini lebih baik dibandingkan adaptor teknik induksi, dikarenakan tegangan yang dikeluarkan lebih stabil sehingga suhu pada komponen tidak terlalu panas dan mengurangi resiko kerusakan alat.

Sedangkan pada adaptor *step-down* menggunakan teknik medan magnet, untuk komponen utamanya berupa kawat *email* yang dililitkan pada teras besi,

terdapat 2 lilitan yaitu lilitan skunder dan lilitan primer. Ketika listrik masuk ke lilitan primer akan terjadi induksi pada kawat *email* sehingga akan terjadi gaya medan magnet pada teras besi kemudian akan menginduksi lilitan skunder. Pada perancangan in menggunakan adaptor *power supply*, yang mana adaptor berfungsi untuk mengubah tegangan listrik AC yang besar menjadi tegangan DC yang kecil [19].

2.2.9 Relay

Relay dapat juga disebut dengan saklar (*Switch*) yang untuk pengoperasiannya secara listrik dan merupakan komponen *electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar/*Switch*). *Relay* menggunakan prinsip elektromagnetik untuk dapat menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan *armature relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.



Gambar 2.10 Prinsip Kerja Relay [11]

Berdasarkan gambar diatas, sebuah besi (*Iron Core*) yang dililit oleh kumparan *Coil* yang berfungsi untuk mengendalikan Besi tersebut. Apabila

kumparan *Coil* dialirkan arus listrik, maka akan timbul gaya elektromagnet yang kemudian akan menarik *armature* untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana *armature* tersebut berada sebelumnya (NC) akan menjadi tidak terhubung atau *open*. Pada kondisi tidak dialiri arus listrik, *armature* akan kembali lagi ke posisi awal (NC). Untuk arus listrik yang dibutuhkan untuk menarik *contact poin* ke posisi *close* biasanya relatif kecil [11].

2.2.10 Firebase

Firebase adalah *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. *Firebase* merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi *mobile* [1]. Dua fitur yang menarik dari *Firebase* adalah *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu juga terdapat fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu *Firebase Notification Console*. *Firebase Database* merupakan penyimpanan basis data *nonSQL* yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String*, *Boolean*, dan *Long*. Data pada *Firebase Database* disimpan sebagai objek *JSON tree*. Tidak seperti basis data *SQL*, tidak ada tabel dan baris pada basis data *non-SQL*. Ketika ada penambahan data, data tersebut akan menjadi *node* pada struktur *JSON*. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke *Firebase Database* dikenal dengan istilah *push*. Pada pengembangan aplikasi, layanan lainnya yang digunakan pada pengembangan aplikasi adalah *Firebase Storage*. Layaknya sebuah penyimpanan awan, *Firebase Storage* memungkinkan pengembang untuk mengunggah atau mengunduh sebuah berkas [20].

2.2.11 MIT App Inventor

App Inventor for Android (version v.134) adalah aplikasi yang awalnya disediakan oleh Google dan sekarang di *maintenance* oleh *Massachusetts Institute of Technology* (MIT)[10]. *App Inventor* memungkinkan semua orang untuk membuat *software* aplikasi untuk sistem operasi *Android*. Pengguna dapat

menggunakan tampilan grafis GUI dan tampilan *drag and drop visual object* untuk membuat aplikasi yang akan dijalankan pada sistem operasi Android. Dalam penggunaannya *App Inventor* dimulai melalui *web-based service* pada *browser* secara *online* (<http://ai2.appinventor.mit.edu>) atau *offline (local host)*. Dengan cara mengatur tampilan aplikasi (*user interface*) pada *web GUI (graphical user interface) builder*, kemudian menspesifikasikan *behavior* aplikasi yang ingin anda buat dengan menyusun *block* yang sesuai. *AppInventor* pada dasarnya bekerja secara *online* melalui *browser* internet, tetapi diperlukan beberapa *software* pendukung paket *Java* untuk membuka *block designer*, emulator untuk menjalankan aplikasi yang dibuat, dan lainnya [21].

2.2.12 Quality of Service

Quality of Service (QoS) adalah metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha yang dilakukan untuk mendefinisikan karakteristik dan sifat dari satu servis. QoS berfungsi untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis (Ferguson & Huston, 1998). QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda .

- a. *Delay* adalah waktu yang dibutuhkan paket untuk mencapai tujuan. Rumus untuk menghitung rata-rata delay sebagai berikut :

$$\text{Rata - rata delay} = \frac{\text{Total delay}}{\text{Total paket yang diterima}} \quad (2.1)$$

Tabel 2.3 Kategori Delay [22]

Kategori <i>Latency</i>	<i>Delay</i> (ms)	Indeks
Sangat bagus	<150 ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	> 450 ms	1

- b. *Throughput* yaitu kecepatan transfer data efektif, yang diukur dalam bps (bit per second). *Throughput* adalah jumlah total kedatangan paket yang sukses dan

diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu yang telah ditentukan. Umumnya *throughput* selalu berkaitan dengan *bandwith*.

$$Throughput = \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} \quad (2.2)$$

Tabel 2.4 Kategori *Throughput* [22]

Kategori <i>Throughput</i>	<i>Throughput</i>	Indeks
Sangat bagus	>2,1 Mbps	4
Bagus	1200 kbps – 2,1 Mbps	3
Cukup	700 – 1200 kbps	2
Kurang baik	338-700 kbps	1
Buruk	0-338 kbps	0

c. *Packet loss* didefinisikan sebagai kegagalan transmisi paket IP mencapai tujuannya, atau juga biasa disebut paket yang hilang. Untuk mengetahui *Packet loss* dapat menggunakan rumus sebagai berikut :

$$Packet Loss = \frac{\text{Paket data yang dikirim} - \text{paket data yang diterima}}{\text{paket data yang dikirim}} \times 100\% \quad (2.3)$$

Tabel 2.5 Kategori *Packet loss* [22]

Kategori <i>Packet Loss</i>	<i>Packet Loss</i>	Indeks
Sangat bagus	0-2%	4
Bagus	3-14%	3
Sedang	15-24%	2
Jelek	>25%	1