

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Landasan Teori**

##### **1. Rancang Bangun**

Rancang bangun adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfirmasi dari komponen-komponen perangkat keras dan perangkat lunak dari semua sistem (Jogiyanto, 2005).

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002).

Menurut Maulani, dkk (2018), rancang bangun merupakan sebuah proses menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut.

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa rancang bangun adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfirmasi dari komponen-komponen perangkat keras dan perangkat lunak dari semua sistem. Dalam hal ini, penulis merujuk pengertian rancang bangun menurut Jogiyanto (2005).

## **2. *Smart Door Lock***

*Smart door lock* adalah sebuah kunci pintu yang untuk pengoperasiannya dapat dilakukan dengan cara yang tidak biasa. Dalam hal ini pengoperasian dapat dilakukan dengan menggunakan sidik jari, *password*, ketukan, komunikasi *bluetooth* bahkan dengan menggunakan jaringan internet. Fungsi utama dari *smart door lock* adalah untuk membatasi orang yang dapat mengakses pintu sehingga hanya orang-orang tertentu yang mendapat ijin dan mendapat wewenang yang dapat mengakses pintu tersebut. Dengan *smart door lock* ini maka keamanan rumah dapat lebih terjamin (Iman, 2018).

*Smart door lock* adalah kunci yang dapat digunakan untuk menciptakan suatu sistem pengaman yang praktis dan efisien pada rumah

maupun bangunan lainnya. Sistem pengamanan ini tidak hanya sekedar praktis namun dengan kualitas keamanan yang juga maksimal. Sangat berbeda dengan kunci yang dibuat dengan teknologi konvensional, *smart door lock* telah menerapkan metode-metode modern dalam penggunaan dan sistem operasinya, khususnya untuk sistem buka dan tutup pintu (Iglloadmin, 2019).

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa *Smart Door Lock* merupakan sebuah kunci pintu yang untuk pengoperasiannya dapat dilakukan dengan cara yang tidak biasa. Dalam hal ini pengoperasian dapat dilakukan dengan menggunakan sidik jari, *password*, ketukan, komunikasi *bluetooth* bahkan dengan menggunakan jaringan internet. Dalam hal ini, penulis merujuk pengertian *Smart Door Lock* menurut Iman (2018).

### 3. *Arduino*

*Arduino Uno* adalah sebuah rangkaian yang dikembangkan dari mikrokontroler berbasis *ATmega328*. *Arduino Uno* memiliki 14 kaki digital *input / output*, dimana 6 kaki digital diantaranya dapat digunakan sebagai sinyal PWM (*Pulse Width Modulation*). Sinyal PWM berfungsi untuk mengatur kecepatan perputaran motor. *Arduino Uno* memiliki 6 kaki analog *input*, kristal osilator dengan kecepatan jam 16 MHz, sebuah koneksi USB, sebuah konektor listrik, sebuah kaki *header* dari ICSP, dan sebuah tombol *reset* yang berfungsi untuk mengulang program

(Magdalena, dkk, 2013). Menurut Kadir (2013), *Arduino Uno* adalah produk berlabel *arduino* yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler *ATmega328* atau sebuah keping yang secara fungsional bertindak seperti sebuah komputer. *Arduino Uno* adalah pengganti mikrokontroler yang dapat diprogram dan dibuat dalam *board* mikrokontroler yang siap pakai dan di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler jenis AVR. Selain itu, *software* dan *hardware* yang digunakan bersifat *open source* (Syahwil, 2017).

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa *Arduino* adalah sebuah rangkaian yang dikembangkan dari mikrokontroler berbasis *ATmega328*. *Arduino Uno* memiliki 14 kaki digital *input / output*, dimana 6 kaki digital diantaranya dapat digunakan sebagai sinyal PWM (*Pulse Width Modulation*). Sinyal PWM berfungsi untuk mengatur kecepatan perputaran motor. *Arduino Uno* memiliki 6 kaki analog *input*, kristal osilator dengan kecepatan jam 16 MHz, sebuah koneksi USB, sebuah konektor listrik, sebuah kaki *header* dari ICSP, dan sebuah tombol *reset* yang berfungsi untuk mengulang program. Dalam hal ini, penulis merujuk pengertian *arduino* menurut Magdalena, dkk (2013).

Kelebihan *Arduino* diantaranya yaitu tidak perlu perangkat *chip programmer* karena didalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer, *Arduino* sudah memiliki sarana komunikasi USB, sehingga pengguna laptop yang tidak memiliki *port* serial/RS323 bisa menggunakannya. Bahasa pemrograman relatif mudah

karena *software Arduino* dilengkapi dengan kumpulan *library* yang cukup lengkap, dan *Arduino* memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada board *Arduino*. Misalnya *shield GPS*, *Ethernet*, *SD Card*, dll (Guntoro, dkk, 2013).

Menurut Kadir (2017), *Arduino* pertama kali diperkenalkan pada tahun 2005. Tim awal yang memprakasai *Arduino* adalah Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, dan David Mellis. Nama *Arduino* berasal dari nama kedai minum di Ivrea, Italia, yang menjadi tempat mereka berkumpul membahas proyek *Arduino*. Piranti *Arduino Uno* ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat diimplementasikan dengan menggunakan papan berukuran relatif kecil ini. Bahkan dengan penambahan komponen tertentu, piranti ini bisa digunakan untuk pemantauan kondisi pasien di rumah sakit dan pengendalian alat-alat di rumah (Gustomo, 2015).

Tabel 2.1 *Index Board Arduino* (Sumber: Gustomo, 2015)

Mikrokontroler	<i>Atmega328</i>
Tegangan pengoperasian	5 V
Tegangan <i>input</i> yang di sarankan	7-12 V
Batas tegangan <i>input</i>	6-20 V
Jumlah <i>pin</i> I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah <i>pin input</i> analog	6
Arus DC tiap <i>pin</i> I/O	40 Ma
Arus DC untuk <i>pin</i> 3.3 V	50 Ma
Memori <i>Flash</i>	32 KB ( <i>ATmega328</i> ), sekitar 0.5 KB digunakan oleh <i>bootloader</i>

Tabel 2.1 *Index Board Arduino (Lanjutan)*

SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz

*Hardware arduino uno* memiliki spesifikasi sebagai berikut:

- a) 14 *pin* IO Digital (*pin* 0-13)

Sejumlah *pin* digital dengan nomor 0-13 yang dapat dijadikan input atau output yang diatur dengan cara membuat program IDE.

- b) 6 *pin* Input Analog (*pin* 0-5)

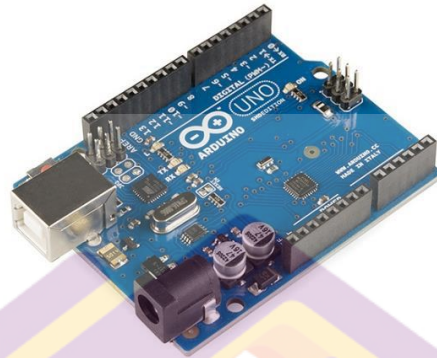
*Pin* analog bernomor 0-5 yang digunakan untuk membaca nilai *input* yang memiliki nilai analog dan mengubah ke dalam angka antara 0 dan 1023.

- c) 6 *pin* Output Analog (*pin* 3, 5, 6, 9, 10 dan 11)

Sejumlah *pin* yang sebenarnya merupakan *pin* digital, tetapi sejumlah *pin* tersebut dapat diprogram kembali menjadi *pin output* analog dengan cara membuat programnya pada IDE.

*Arduino Uno* menggunakan bahasa pemrograman C dalam membuat sebuah programnya. Bahasa C merupakan bahasa pemrograman yang sangat lazim dipakai sejak awal komputer dibuat dan sangat berperan penting dalam perkembangan *software*. Selain itu, bahasa C juga terkenal sangat ampuh karena kemampuannya mendekati bahasa *assembler*. Bahasa pemrograman C menghasilkan sebuah file kode objek yang sangat kecil dan dieksekusi sangat cepat, sehingga bahasa pemrograman ini sering digunakan pada sistem operasi dan pemrograman mikrokontroler, serta bersifat *multi-platform* dan

dapat dijalankan pada sistem operasi *Windows, Unix, Linux, MacOS* (Syahwil, 2017).



Gambar 2.1 *Arduino Uno*

(Sumber: B. Gustomo, 2015)

#### 4. ***Bluetooth HC-05***

*Bluetooth* merupakan teknologi komunikasi tanpa kabel jarak dekat untuk transfer data antar perangkat jarak dekat. *Bluetooth* ditemukan pertama kali pada tahun 1994 oleh Ericsson dari Swedia. Pada tahun 1998, *Bluetooth SIG (Special Interest Group)* mengembangkan spesifikasi terbuka untuk koneksi tanpa kabel jarak dekat (Sairam, dkk, 2002). Hingga kemudian *Bluetooth* ditetapkan sebagai salah satu standar terbuka untuk komunikasi radio digital jarak dekat. Teknologi *Bluetooth* ditetapkan dalam standar IEEE 802.15. Teknologi *Bluetooth* beroperasi pada frkuensi 2,4 GHz dengan jarak komunikasi antara 10 sampai 100 meter. Produk *Bluetooth* seri HC terdiri dari modul antarmuka serial *Bluetooth* dan adapter *Bluetooth*. Modul antarmuka serial *Bluetooth* sendiri dibagi menjadi dua, yaitu untuk keperluan industri (HC-03 dan HC-04) dan untuk keperluan sipil (HC05 dan HC-06).

HC-05 merupakan modul *Bluetooth SPP (Serial Port Protocol)* yang di desain untuk koneksi serial tanpa kabel secara transparan. Modul ini memiliki kualifikasi *Bluetooth V2.0+EDR (Enhanced Data Rate)* dengan kecepatan hingga 3 Mbps serta bekerja pada frekuensi radio 2.4 GHz sebagai *transceiver*. Modul *Bluetooth HC-05* memiliki fitur yaitu:

- a. Sensitivitas -80 dBm.
- b. Kekuatan transmit RF hingga +4 dBm.
- c. *Low Power*.
- d. Kontrol PIO.
- e. Antena terintegrasi.
- f. Antarmuka UART dengan *baud rate* yang dapat diprogram.
- g. Mendukung *baud rate* 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- h. Mendukung komunikasi dua arah (sebagai *transmitter* dan *receiver*).

## 5. QR Code

- a. Pengertian QR Code

QR Code adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994. Agar dapat membaca QR Code diperlukan sebuah pembaca atau pemindai berupa *software* yaitu QR Code Reader atau QR Code Scanner yang harus diinstal pada perangkat



telepon *mobile*. QR merupakan singkatan dari *Quick Response* atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, QR Code mampu menyimpan informasi secara horizontal dan vertical.

QR Code merupakan sebuah kode unik yang dapat berisi angka, huruf, dan simbol serta dilengkapi dengan *error correction*. Pada awalnya QR Code digunakan untuk pelacakan bagian kendaraan untuk perusahaan *manufacturing*. Tetapi sekarang QR Code telah digunakan untuk komersil yang ditujukan pada pengguna telepon seluler. QR Code dapat menyimpan informasi atau data lebih banyak secara horizontal dan vertikal (Damara, dkk, 2017). Menurut Tizhoosh (2015), *Quick Response Code* (QR Code) adalah jenis *Barcode* berbentuk dua dimensi yang berfungsi untuk menyimpan informasi dengan cepat dan mendapat tanggapan atau *respons* yang cepat juga. QR Code dapat menyimpan informasi seperti URL, nomor telepon, pesan SMS, atau teks apapun. QR Code merupakan hasil perkembangan dari *Barcode* yang dilakukan oleh Denso Wave. QR Code dan *Barcode* memiliki perbedaan yaitu pada penyimpanan data, *barcode* menyimpan data lebih pendek dibanding QR Code.

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa QR Code adalah jenis *Barcode* berbentuk dua dimensi yang

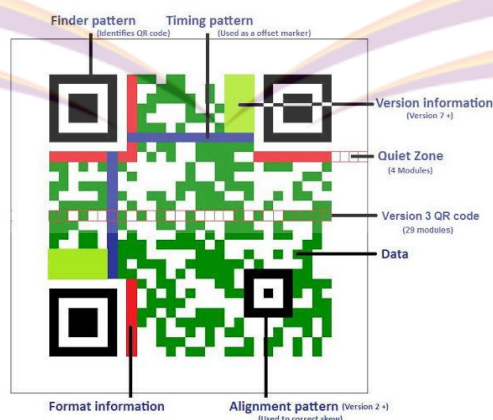
berfungsi untuk menyimpan informasi dengan cepat dan mendapat tanggapan atau *respons* yang cepat juga. *QR Code* dapat menyimpan informasi seperti URL, nomor telepon, pesan SMS, atau teks apapun. *QR Code* merupakan hasil perkembangan dari *Barcode* yang dilakukan oleh Denso Wave. *QR Code* dan *Barcode* memiliki perbedaan yaitu pada penyimpan data, *barcode* menyimpan data lebih pendek dibanding *QR Code*. Dalam hal ini, penulis merujuk pengertian *QR Code* menurut Tizhoosh (2015).

*QR Code* dapat diimplementasikan diberbagai keperluan seperti keperluan komersial digunakan sebagai pembayaran online, pemesanan tiket, promosi tiket atau kepentingan umum seperti berbagi informasi kartu mahasiswa, dan informasi produk makanan. Penggunaan *QR Code* secara umum melalui ponsel yang dilengkapi dengan aplikasi pembaca *QR Code* (*QR Scanner*, *QR Code reader*). Pola dan struktur dalam *QR Code* memiliki fungsi yang terdefinisi dimana terdiri dari garis simbol, *sampling grid*, dan koreksi kesalahan. Prosesnya informasi akan dilakukan *encoding* dalam *finder pattern*. Cara kerja *QR Code* awalnya data dilakukan *encode* berdasarkan *mode* dan *bit stream*. Aliran *bit stream* dibagi kedalam *codeword* yang nantinya akan dibagi ke dalam *blocks*. Semua *codeword* akan diletakan dalam sebuah matrix. Selanjutnya, *function patterns* menambahkan kedalam simbol dan akan terbentuk sebuah simbol *QR Code* (Kurniawan & Utomo, 2018).

## b. Struktur QR Code

Beberapa penjelasan struktur QR Code Menurut Ariadi (2011) antara lain:

- 1) *Finder Pattern* berfungsi untuk identifikasi letak QR Code.
- 2) *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- 3) Data berfungsi untuk menyimpan data yang dikodekan.
- 4) *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat QR Code, berbentuk modul hitam putih.
- 5) *Alignment Pattern* adalah pola yang berfungsi memperbaiki penyimpangan QR Code terutama *distorsi non linier*.
- 6) *Version Information* adalah versi dari sebuah QR Code.
- 7) *Quiet Zone* merupakan daerah kosong di bagian terluar QR Code yang mempermudah mengenali pengenalan QR oleh sensor CCD.
- 8) QR Code version adalah versi dari QR Code yang digunakan.



Gambar 2.2 Struktur QR Code

(Sumber: Ariadi, 2011)

c. Jenis-jenis QR Code

QR Code memiliki beberapa macam, yaitu sebagai berikut (Denso, 2019)

1) QR Code model 1

Jenis QR Code model 1 ini adalah QR Code asli yang dapat menampung 1.167 angka dengan maksimum 14 (73 x 73 modul).



Gambar 2.3 Jenis QR Code model 1

(Sumber: [www.qrcode.com](http://www.qrcode.com))

2) QR Code model 2

Model ini adalah penyempurnaan dari model 1 dengan versi terbesar 40 (177 x 177 modules), yang mampu menyimpan sampai 7.089 angka.

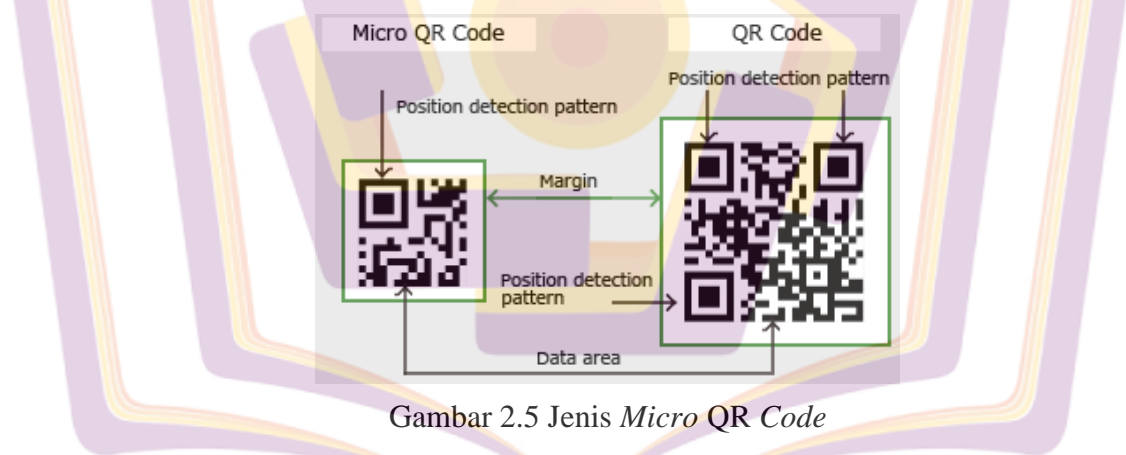


Gambar 2.4 Jenis QR Code model 2

(Sumber: [www.qrcode.com](http://www.qrcode.com))

### 3) *Micro QR Code*

Versi terbesar dari kode ini adalah M4 (17x17 modul) yang dapat menyimpan hingga 35 angka. Fitur utama dari *Micro QR Code* adalah hanya memiliki satu pola deteksi posisi, dibandingkan dengan regular *QR Code* yang memerlukan sejumlah tempat karena pola deteksi posisi yang terletak di tiga sudut simbol. *QR Code* biasa membutuhkan setidaknya empat modul yang lebar di sekitar simbol, sedangkan *Micro QR Code* hanya membutuhkan cukup dua modul *margin*. *Konfigurasi Micro QR Code* memungkinkan pencetakan di tempat lebih kecil dari *QR Code*.



Gambar 2.5 Jenis *Micro QR Code*

(Sumber: [www.qrcode.com](http://www.qrcode.com))

### 4) *iQR Code*

Kode yang dapat dihasilkan dari salah satu modul, persegi atau persegi panjang. Dan dapat di cetak sebagai kode inversi hitam putih atau kode pola *dot* (bagian penanda). Versi terbesar dari kode ini dapat mencapai 61 (422x422 modul), yang dapat menyimpan 40.000 angka.



Gambar 2.6 Jenis iQR Code

(Sumber: [www.qrcode.com](http://www.qrcode.com))

5) SQRC

Jenis QR Code ini dilengkapi dengan membaca fungsi pembatas. Ini dapat digunakan untuk menyimpan informasi pribadi untuk mengelola informasi internal perusahaan dan sejenisnya.



Gambar 2.7 Jenis SQRC

(Sumber: [www.qrcode.com](http://www.qrcode.com))

6) LogoQ

Jenis QR Code yang dapat menggabungkan fitur desain tingkat tinggi seperti ilustrasi, huruf dan logo. QR Code ini menggunakan logika *Since proprietary*.



Gambar 2.8 Jenis LogoQ

(Sumber: [www.qrcode.com](http://www.qrcode.com))

#### d. Manfaat QR Code

Manfaat yang terdapat pada QR Code menurut Denso (2011) antara lain:

1) Kapasitas tinggi dalam menyimpan data

Sebuah QR Code tunggal dapat menyimpan sampai 7.089 angka.

2) Ukuran yang kecil

QR Code dapat menyimpan jumlah data yang sama dengan barcode 1D dan tidak memerlukan ruang besar.

3) Dapat mengoreksi kesalahan

Tergantung tingkat koreksi kesalahan yang dipilih, data pada QR Code yang kotor atau rusak sampai 30% dapat diterjemahkan dengan baik.

4) Banyak jenis data

Menangani angka, abjad, simbol, karakter bahasa Jepang, Cina, Korea dan data biner.

5) Kompensasi distorsi

QR Code tetap dapat dibaca pada permukaan melengkung.

6) Kemampuan menghubungkan

Sebuah *QR Code* dapat dibagi hingga 16 simbol yang lebih kecil agar sesuai dengan ruang. Simbol-simbol kecil yang dibaca sebagai kode tunggal apabila dibaca atau *scan* menurut urutan.

## 6. *Smartphone*

*Smartphone* adalah telepon selular dengan mikroprosesor, memori, layar dan modem bawaan. *Smartphone* merupakan *ponsel* multimedia yang menggabungkan fungsionalitas PC dan *handset* sehingga menghasilkan *gadget* yang mewah, di mana terdapat pesan teks, kamera, pemutar musik, video, *game*, akses email, tv digital, *search engine*, pengelola informasi pribadi, fitur GPS, jasa telepon *internet* dan bahkan terdapat telepon yang juga berfungsi sebagai kartu kredit (Williams & Sawyer, 2011).

*Smartphone* (*ponsel* pintar) adalah sebuah telepon *mobile* yang dibangun dengan sistem operasi komputer, memiliki kemampuan kalkulasi canggih beserta konektifitas membuatnya lebih dari sekedar telepon. Mengkombinasikan sejumlah fungsi *Asisten Digital Personal* (PDA) dengan sebuah *mobile phone*. Pada model selanjutnya fungsi media player, kamera digital, kamera video *pocket*, dan navigasi GPS telah ditambahkan untuk menghadirkan sebuah perangkat multi-guna. *Smartphone modern* juga menyertakan layar sentuh beresolusi tinggi dan *web browser* guna menampilkan halaman *web* standar beserta situs yang mendukung perangkat *mobile*. Akses data kecepatan tinggi disediakan oleh koneksi *Wi-Fi* dan



*mobile broadband*. Sistem operasi yang umum digunakan oleh *Smartphone* adalah Android milik Google (Kompasiana, 2015).

*Smartphone* adalah telepon yang *Internet-enabled*, biasanya menyediakan fungsi *Personal Digital Assistant* (PDA) seperti fungsi kalender, buku agenda, buku alamat, kalkulator, dan catatan. (Gary & Misty, 2007).

Berdasarkan beberapa pendapat di atas, maka dapat disimpulkan bahwa *Smartphone* merupakan sebuah telepon *mobile* yang dibangun dengan sistem operasi komputer, memiliki kemampuan kalkulasi canggih beserta konektifitas membuatnya lebih dari sekedar telepon. Mengkombinasikan sejumlah fungsi *Asisten Digital Personal* (PDA) dengan sebuah *mobile phone*. Pada model selanjutnya fungsi media player, kamera digital, kamera video *pocket*, dan navigasi GPS telah ditambahkan untuk menghadirkan sebuah perangkat multi-guna. Dalam hal ini, penulis merujuk pengertian *Smartphone* menurut Kompasiana (2015).

## 7. *Android*

### a. Pengertian *Android*

*Android* adalah sistem operasi untuk telepon seluler yang berbasis *Linux* dan dikembangkan oleh Google. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan dan mengembangkan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. *Android* umum digunakan di *smartphone* dan juga tablet PC. Fungsinya

sama seperti sistem operasi *Symbian* di Nokia, *iOS* di Apple dan *BlackBerry OS* (Safaat, 2012). Menurut Silvia, Haritman dan Muladi (2014), *Android* adalah *platform open source* yang komprehensif dan dirancang untuk *mobile devices*. Dikatakan komprehensif karena *Android* menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. Sistem *Android* menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan.

*Android* merupakan sebuah sistem operasi yang dikeluarkan oleh Google khususnya untuk *smartphone* dan tablet. *Android* juga mempunyai toko dimana terdapat 1 miliar pengguna aktif. Berbicara tentang pemrograman tentunya tidak lepas dari *Integrated Development Environment (IDE)* yang bisa dipakai oleh para pengembang. *Android* menjadi sistem operasi yang banyak digunakan untuk *Smartphone* saat ini karena bersifat *open source* sehingga memberikan sebuah kemudahan bagi para pembuat maupun pengembang aplikasi (Imaduddin & Permana, 2017).

Berdasarkan beberapa pengertian di atas, maka dapat disimpulkan bahwa *Android* adalah *platform open source* yang komprehensif dan dirancang untuk *mobile devices*. Dikatakan komprehensif karena *Android* menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. *Android* menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap

tersimpan meskipun *device* dimatikan. Dalam hal ini, penulis merujuk pengertian *Android* menurut Silvia, Haritman dan Muladi (2014).

b. *Arsitektur Android*

Secara garis besar arsitektur *Android* dapat dijelaskan sebagai berikut (Safaat, 2015):

1) *Application dan Widgets*

*Application dan Widgets* merupakan *layer* yang berhubungan dengan aplikasi saja. Pada *layer* ini terdapat sebuah aplikasi inti termasuk *klien, email, program, SMS, kalender, peta, browser, kontak* dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman Java.

2) *Application Frameworks*

*Android* adalah *open development platform* yaitu *Android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses berbagai perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya pengembang dapat dengan mudah menggunakan kembali komponen yang sudah digunakan. *Application Frameworks* adalah *layer* dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan disistem operasi *Android*, pada *layer* ini aplikasi dapat dirancang dan dibuat,

seperti *content-providers* yang berupa SMS dan panggilan telepon. Komponen dalam *Application Frameworks* yaitu *views*, *content-provider*, *resource manager*, *notification manager*, *activity manager*.

### 3) *Libraries*

*Libraries* adalah layer dimana fitur-fitur *Android* berada, biasanya pada pembuatan aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Arsitektur *Android* ini berjalan diatas kernel *layer* ini meliputi *library C/C++* inti seperti *Libc* dan *SSL*, serta *libraries* media untuk pemutaran media audio dan video, *libraries* untuk manajemen tampilan, *libraries graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D, *libraries SQLite* untuk dukungan *database*, *libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*, *libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view*, dan *libraries 3D* yang mencakup implementasi *OpenGL*, *ES 1.0 API's*.

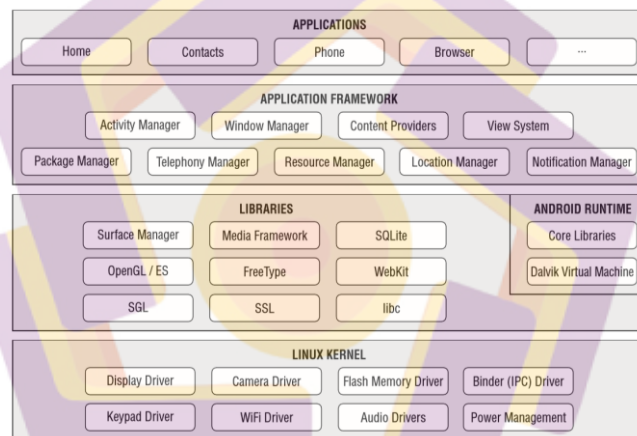
### 4) *Android Run Time*

*Layer* yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan sebuah mesin yang membentuk dasar kerangka aplikasi *Android*. Di dalam *Android Run Time* dibagi menjadi dua bagian, yaitu *Core Libraries* yang berfungsi sebagai penerjemah bahasa *Java* atau *C* dan *Dalvik Virtual Machine* (DVM) yang bertugas untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan

pengembangan yang mampu membuat *Linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

#### 5) *Linux Kernel*

*Linux Kernel* adalah inti dari *operating system* dari *Android*. Berisi file-file yang mengatur sistem *processing*, *memory*, *resource*, *drivers* dan sistem-sistem operasi\ lainnya. *Linux kernel* yang digunakan *Android* adalah *Linux kernel release 2.6*.



Gambar 2.9 Arsitektur *Android*

(Sumber: <https://bierpinter.com>)

#### c. Versi *Android*

Versi *Android* yang pernah dirilis adalah sebagai berikut (Clinton, 2018):

##### 1) *Android 1.0*

Versi komersial *Android* pertama kali hadir pada September 2008 di ponsel besutan T-Mobile, G1 (HTC Dream). Pada waktu debutnya ini, *Android* hanya memiliki nomor versi (*Android 1.0*) tanpa embel-embel nama cemilan. *Android 1.0* sudah dibekali *notifikasi pull-down*

dan *widjets* di layar *home screen*, serta kehadiran *Google Play Store* pertama yang waktu itu bernama *Android Market*.

#### 2) *Android 1.5 Cupcake*

Mulai dari versi ini, *Android* menggunakan nama makanan manis untuk setiap versi yang diluncurkan. *Android 1.5 Cupcake* sendiri dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *featured phone* pada saat itu. *Android Cupcake* untuk pertama kalinya menghadirkan *on-screen keyboard* menggantikan papan ketik fisik yang sebelumnya dipakai oleh perangkat *Android*. Perubahan besar lainnya adalah kemampuan merekam video yang baru ditambahkan *Android Cupcake*. Google turut membuka *SDK widget Android* sehingga *developer* pihak ketiga bisa membuat *widget* sendiri untuk sistem operasi tersebut.

#### 3) *Android 1.6 Donut*

*Android Donut* dirilis pada 15 September 2009 dengan menghadirkan dukungan untuk jaringan *CDMA* dan beragam ukuran layar. *Donuts* menandai titik dimana *Android* sudah bisa disematkan dalam di perangkat dengan beberapa ukuran layar yang berbeda. Selain itu, untuk memudahkan pengguna dalam menjelajahi antarmuka ponsel, muncul fitur baru berupa *search box* yang ada dalam *home screen*.

#### 4) *Android 2.0 Eclair*

Versi *Android Eclair* rilis pada 26 Oktober 2009 yang bernama *Eclair* (versi 2.0), untuk pertama kalinya membawa fitur baru untuk

mempermudah pengguna dalam berpergian, yakni *Google Map*. Fitur peta yang kelak sangat populer itu datang lengkap dengan *turn-by-turn navigation* dan panduan suara (*voice guidance*) yang masih ada hingga sekarang. Muncul tak lama setelah Donut pada tahun yang sama, *Eclair* turut menambah dukungan HTML5 di *browser*, berikut kemampuan untuk memutar video. *Lock screen* turut dirombak dengan menambahkan fitur *swipe to unlock*.

#### 5) *Android 2.2 Froyo*

Hadir pada 20 Mei 2010, *Android Froyo (Frozen Yoghurt, versi 2.2)* melakukan debutnya pada ponsel seri Nexus pertama besutan Google, Nexus One. *Froyo* kebanyakan memoles fungsi *Android* yang sudah ada sebelumnya, seperti dukungan *homescreen* yang bisa ditambahkan hingga 5 buah. Beberapa peningkatan lain termasuk tambahan, *Voice Action*, dukungan *mobile hotspot* dan kunci PIN untuk *lock screen*, melengkapi *pattern lock* yang sebelumnya sudah ada di *Android*.

#### 6) *Android 2.3 Gingerbread*

Rilis pada 6 Desember 2010, *Android Gingerbread (versi 2.3)* menambah dukungan kamera depan sehingga pengguna perangkat *Android* pun kini bisa melakukan *selfie*. Google juga mulai menerapkan fitur *battery management* untuk menginformasikan pengguna soal aplikasi mana yang menguras baterai. *Keyboard* turut diperbaiki dengan warna baru dan peningkatan dukungan *multi-touch*.

7) *Android 3.0 Honeycomb*

Rilis pada 22 Februari 2011, Google merilis sistem operasi *Android Honeycomb* (versi 3.0) yang khusus ditujukan bagi perangkat tablet. Debutnya dilakukan di Motorola Xoom. *Honeycomb* mengusung beberapa perubahan dibanding OS Android untuk *smartphone*, seperti warna tema biru bukan hijau dan *preview* untuk *widget*. Tombol-tombol *navigasi* pun ditampilkan *Honeycomb* langsung di layar (*virtual buttons*) sehingga perangkat tak membutuhkan tombol fisik.

8) *Android 4.0 Ice Cream Sandwich*

Pada 19 Oktober 2011, *Android Ice Cream Sandwich* (versi 4.0), memboyong fitur-fitur *Honeycomb* ke *smartphone*. Termasuk tombol *navigasi virtual buttons* yang bisa ditampilkan langsung di layar dan gestur sapuan dengan jari untuk menutup aplikasi. *Ice Cream Sandwich* turut menghadirkan sejumlah fitur baru seperti *face unlock*, rekaman penggunaan data internet, aplikasi *e-mail*, kalender baru, serta dukungan *Near-Field Communication* (NFC).

9) *Android 4.1 Jelly bean*

Pada 27 Juni 2012 muncul versi baru, yakni *Jelly bean* (versi 4.1). Google mulai menerapkan teknologi asisten digital *Google Now* yang bisa diakses dengan sapuan jari dari *home screen*. Google turut mengimplementasikan "*Project Butter*" yang bertujuan memperlancar proses *navigasi* di *Android* lewat teknik *triple buffering* grafis. Hasilnya, *stuttering* di *Android* berkurang sehingga terasa lebih mulus.



#### 10) *Android 4.4 Kitkat*

Pada 31 Oktober 2013, Google merilis sistem operasi *Android Kitkat* (versi 4.4) bersamaan dengan kemunculan ponsel seri Nexus 5. Tampilan *Kitkat* diperbaiki supaya tampak lebih modern, dengan akses warna putih dan desain ulang aplikasi-aplikasi bawaan dengan warna lebih terang. *Kitkat* ikut memperkenalkan frasa perintah suara “OK Google” yang digunakan untuk memanggil Google *Now* dengan ucapan, kapanpun dikehendaki oleh pengguna.

#### 11) *Android 5.0 Lollipop*

Kemunculan perangkat Google Nexus 6 pada 2014 sekaligus menandai hadirnya *Android Lollipop* (versi 5.0). Di OS inilah Google mulai menerapkan filosofi desain antarmuka “*Material Design*” yang serba “*flat*” di *Android*. Di luar tampilan ada banyak perubahan lain di *Android Lollipop*, termasuk *Android Runtime* menggantikan *Dalvik VM* dan dukungan format gambar RAW.

#### 12) *Android 6.0 Marshmallow*

*Android Marshmallow* (versi 6.0) dirilis pada 28 Mei 2015 dan membawa sejumlah perubahan tampilan lebih jauh. Menu *app*, diperbaiki dan menggunakan latar belakang putih. Ada juga *search bar* untuk mempermudah *user* dalam menemukan aplikasi yang dicari. *Android* menambahkan dukungan resmi terhadap sensor pemindai sidik jari. *Permissions* aplikasi tidak lagi diminta secara sekaligus, tapi dirinci satu per satu sehingga bisa lebih diteliti oleh pengguna.

### 13) *Android 7.0 Nougat*

*Nougat* (versi 7.0) diluncurkan pada tanggal 23 Agustus 2016. *Nougat* adalah salah satu *upgrade* terbesar untuk sistem operasi *Android*. Google menambahkan dukungan *multi-window* sehingga pengguna bisa menjalankan dua aplikasi secara bersamaan dalam jendela terpisah yang diposisikan secara berdampingan. Fitur *Data Saver* pada *Android Nougat* membantu pengguna dalam memonitor dan membatasi penggunaan kuota data internet, tambahan lain termasuk mode VR.

### 14) *Android 8.0 Oreo*

*Android Oreo* (versi 8.0) yang dirilis pada 2017 menambah lebih banyak fitur *multi-tasking* seperti *Picture-in-Picture*. Bagian notifikasi ikut diperbarui. Pengguna bisa mengatur notifikasi mana saja yang akan ditampilkan. Selain OS *Android* versi “reguler”, Google juga membuat *Android* versi “ringan” bernama *Android Go Edition* yang diambil dari basis *Android Oreo*. OS ini khusus ditujukan supaya bisa dijalankan dengan lancar oleh perangkat-perangkat *entry level Android Go*.

### 15) *Android 9.0 Pie*

*Android Pie* diresmikan pada 6 Agustus 2018. *Android Pie* (versi 9.0) mengganti tiga tombol *navigasi* dengan tombol tunggal berbentuk *elips*. *Android Pie* lebih mengandalkan gestur gerakan-gerakan jari untuk melakukan hal-hal seperti *multitasking*. Fitur lain termasuk

“Digital Wellbeing” untuk menginformasikan soal pola pemakaian perangkat, *Adaptive Battery* untuk membatasi pemakaian baterai oleh aplikasi, dan *App Action* yang langsung menjalankan fitur aplikasi dari *app drawer*.

#### d. Komponen Aplikasi *Android*

Komponen aplikasi *android* adalah bagian penting dari sebuah aplikasi *android*. Setiap komponen dalam aplikasi *android* mempunyai peranan yang berbeda-beda, sesuai dengan keperluan aplikasi. Ada 4 jenis komponen aplikasi *android* (Asmono, 2013) yang memiliki peranan atau tujuan yang berbeda di antaranya adalah:

##### 1) *Activities*

*Activities* biasanya berfungsi untuk menyajikan *user interface (UI)* kepada *user* untuk melakukan interaksi. Untuk pindah dari satu *activity* ke *activity* lain dapat melakukannya dengan satu *event*, misalnya klik tombol, atau memilih *opsi*.

##### 2) *Services*

*Services* merupakan komponen yang berjalan di *background* saat melakukan pekerjaannya. *Service* tidak memiliki *user interface*. Contohnya pada layanan pemutar musik yang dapat menjalankan satu memutar lagu di *background* saat pengguna menjalankan aplikasi yang berbeda.

### 3) *Content providers*

*Content providers* adalah antarmuka yang digunakan untuk berbagi data antar aplikasi. Dengan adanya *Content providers programmer* dapat menyimpan data dalam sistem file, *SGLite* dan lain-lain. Sebagai contoh *Android* menyediakan aplikasi kontak pengguna. Aplikasi apapun yang memerlukan kontak pengguna atas izin pengguna dapat mengaksesnya.

### 4) *Broadcast receivers*

*Broadcast receivers* merupakan sebuah komponen yang merespon pengumuman ke *user* dari sistem *Android*. Hal ini berfungsi untuk mengingatkan *user* itu sendiri. Misalnya baterai lemah, pesan masuk, telepon seluler telah dimatikan atau lain sebagainya. *Broadcast receivers* merespon dengan menghidupkan lampu latar, *led light*, getaran, suara dan lain-lain.

## 8. **Arduino IDE**

### a. Pengertian *Arduino IDE*

*Arduino IDE* adalah *software* yang disediakan oleh *arduino.cc* yang ditujukan sebagai perangkat pengembangan *sketch* yang digunakan sebagai program di papan *Arduino*. *Integrated Development Environment (IDE)* berarti bentuk alat pengembangan program yang terintegrasi sehingga berbagai keperluan disediakan dan dinyatakan dalam bentuk antarmuka berbasis *menu*. Dengan menggunakan *Arduino IDE*,

bisa dilakukan penulisan *sketch*, memeriksa kesalahan, dan mengunggah *sketch* yang sudah dikompil ke papan *Arduino* (Kadir, 2015).

#### b. Bagian-bagian *Arduino* IDE

*Arduino* IDE memiliki bagian-bagian seperti berikut (Sinaryuda, 2017):

##### 1) *Verify*

Pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi di-*upload* ke *board Arduino*, biasakan untuk memverifikasi terlebih dahulu *sketch* yang dibuat. Jika ada kesalahan pada *sketch*, akan muncul *error*. Proses *Verify / Compile* mengubah *sketch* ke *binary code* untuk di-*upload* ke mikrokontroler.

##### 2) *Upload*

*Upload* berfungsi untuk *mengupload sketch* ke *board Arduino*. Walaupun kita tidak mengklik tombol *verify*, maka *sketch* akan di-*compile*, kemudian langsung di-*upload* ke *board*. Berbeda dengan tombol *verify* yang hanya berfungsi untuk memverifikasi *source code* saja.

##### 3) *New Sketch*

Berfungsi untuk membuka *window* dan membuat *sketch* baru.

##### 4) *Open Sketch*

Berfungsi membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE *Arduino* akan disimpan dengan *ekstensi file .ino*

5) *Save Sketch*

*Save Sketch* berfungsi untuk menyimpan *sketch*, tapi tidak disertai dengan mengkompile.

6) Serial Monitor

Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara *arduino* dengan *sketch* pada *port* serialnya. Serial monitor ini sangat berguna sekali ketika ingin membuat program atau melakukan *debugging* tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan *error*.

7) Keterangan Aplikasi

Pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal “*Compiling*” dan “*Done Uploading*”. Ketika proses *compile* dan *upload sketch* ke board Arduino.

8) *Console log*

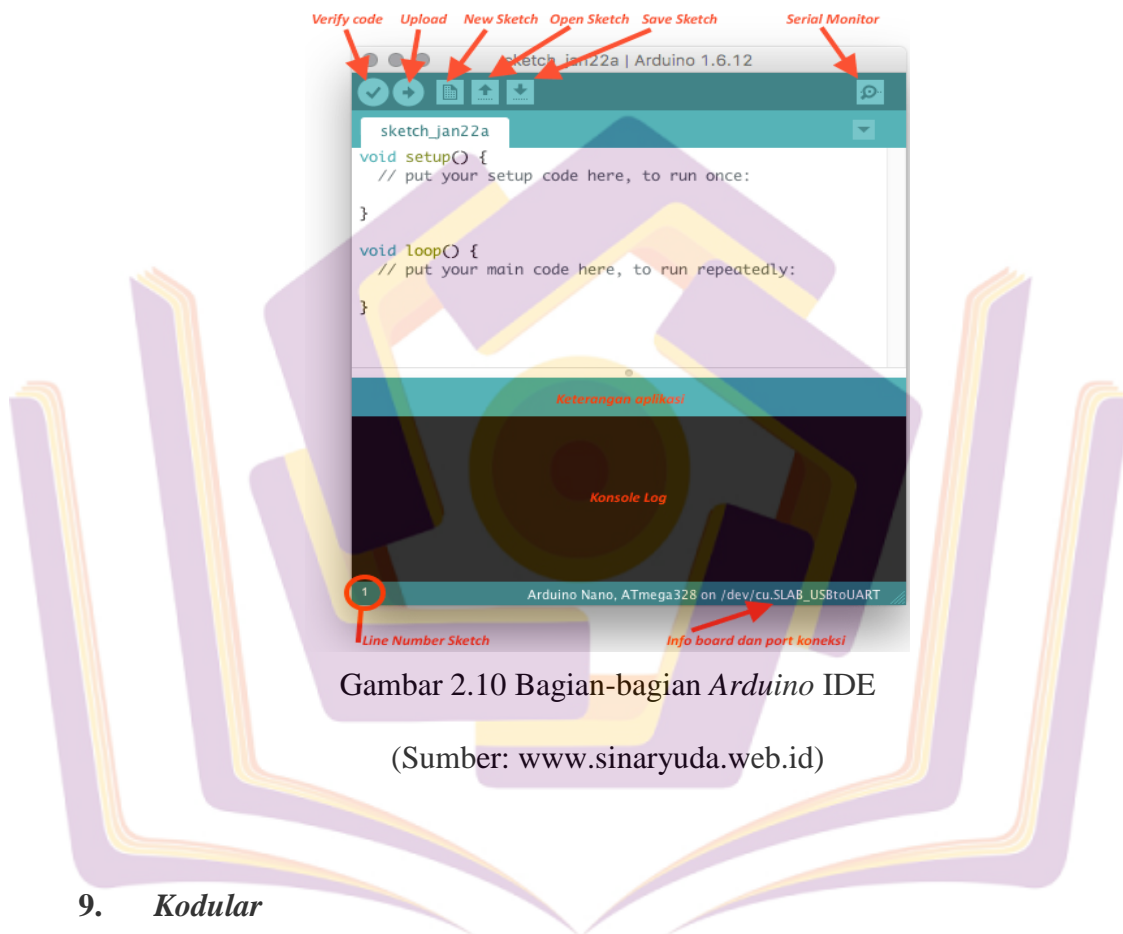
Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal, ketika aplikasi mengkompile atau ketika ada kesalahan pada *sketch* yang dibuat, maka informasi *error* dan baris akan diinformasikan di bagian ini.

9) Baris *Sketch*

Pada bagian ini *tools* akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.

## 10) Informasi *Board* dan *Port*

Bagian yang berfungsi untuk menampilkan sebuah tulisan atau pesan yang menginformasikan *port* yang sedang dipakai oleh *board* Arduino.



Gambar 2.10 Bagian-bagian *Arduino IDE*

(Sumber: [www.sinaryuda.web.id](http://www.sinaryuda.web.id))

## 9. *Kodular*

*Kodular* adalah situs *web* yang menyediakan *tools* untuk membuat aplikasi *Android* dengan menggunakan *block programming*. *Kodular* menyediakan kelebihan fitur yakni *Kodular Store* dan *Kodular Extension IDE* yang bisa memudahkan *developer* melakukan unggah (*upload*) aplikasi *Android* ke dalam *Kodular Store*, melakukan dalam pembuatan blok program *extension IDE* sesuai dengan keinginan *developer*. *Kodular* ini bisa

melakukan kustom tema (*theme*) sesuai dengan keinginan *developer*. File eksistensi dari *Kodular* adalah (*.aia*) dan *plugin* eksistensinya (*.aix*). *Plugin* eksistensi isi berisi beberapa kode perintah dalam bahasa pemrograman *Java* (*.java*) yang akan mengkonversi menjadi file *plugin* eksistensi (*.aix*), ini berguna bagian *extension*. *Kodular* memiliki fitur tambahan yang tidak dimiliki oleh *tools* lain yang serupa, yakni *Copy Screen*. Fitur ini berfungsi untuk menyalin satu *screen* ke *screen* lain yang isinya sama. Sehingga tidak perlu menghabiskan waktu dengan membuat *screen* baru tapi isinya sama dengan *screen* sebelumnya. Kelebihan lainnya yaitu memiliki fitur komponen *pallette* lebih kompleks, membuat program hanya dengan melakukan *drag* dan *drop* pada program *blocks* yang ada, memiliki fitur *plugin monetize* bawaan dari *Kodular* sendiri, memiliki berbagai fitur *plugin monetize* sebagai penghasilan uang, dan tidak perlu *instal software* tambahan (Lestari, 2019).

## 10. Solenoid

*Solenoid* adalah aktuator yang mampu melakukan gerakan *linier*. *Solenoid* dapat bekerja secara elektromekanis (AC/DC), *hidrolik*, *pneumatik* atau di dorong semua operasi pada prinsip-prinsip dasar yang sama. Dengan memberikan sumber tegangan, maka *solenoid* dapat menghasilkan gaya yang *linier*. Sistem kerja *Solenoid* adalah didalamnya terdapat kawat melingkar pada inti besi. Ketika arus listrik melalui kawat ini, maka terjadi medan magnet untuk menghasilkan energi yang bisa mendorong inti besi. Poros dalam dari *solenoid* adalah piston seperti silinder terbuat dari besi atau baja,



disebut *pluger* (setara dengan sebuah dinamo) medan magnet kemudian menerapkan kekuatan untuk *pluger*, baik menarik atau *repling* (kembali posisi). Ketika medan magnet dimatikan, pegas *pluger* akan kembali ke posisi semula (Huda, 2015).

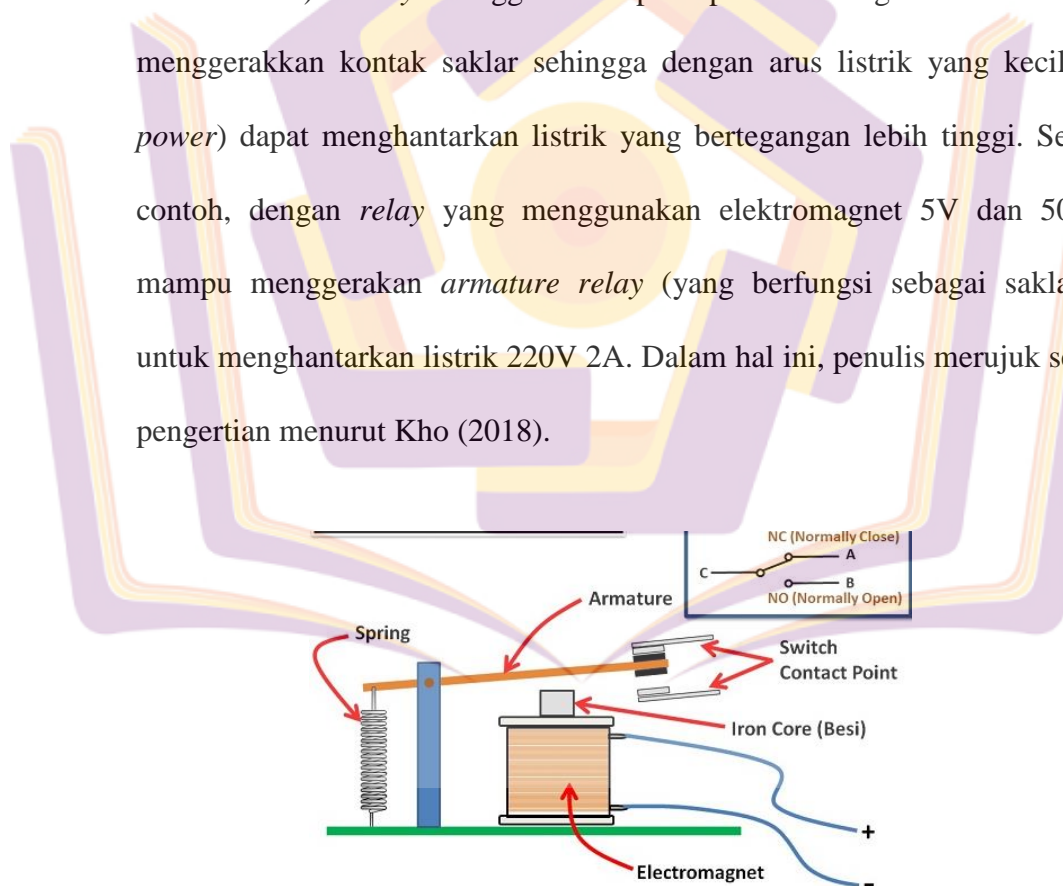
## 11. Relay

*Relay* adalah saklar (*switch*) yang dioperasikan 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 (Kho, 2018).

*Relay* adalah komponen elektronika yang mempunyai fungsi untuk memutus dan menghubungkan arus listrik pada rangkaian. Pada *relay* terdapat dua komponen yang mendukung kinerjanya yaitu kumparan atau elektromagnet dan bagian kontak (saklar). Cara kerja *relay* memanfaatkan prinsip elektromagnetik sebagai penggerak kontak saklar. Apabila kontak *point* terhubung, maka arus listrik dapat mengalir, jika kontak *point* terbuka maka aliran arus listrik terputus. Menurut Wicaksono (2016), *Relay* merupakan saklar elektrik yang sistem kerjanya menggunakan

elektromagnetik untuk memindahkan saklar dari posisi *off* (mati) ke posisi *on* (hidup). Daya yang diperlukan untuk mengaktifkan *relay* relatif kecil, tetapi *relay* dapat mengendalikan sesuatu yang membutuhkan daya lebih besar.

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa *Relay* adalah saklar (*switch*) yang dioperasikan 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. Dalam hal ini, penulis merujuk sebuah pengertian menurut Kho (2018).



Gambar 2.11 Struktur *Relay*

(Sumber: <https://teknikelektronika.com>)

Berdasarkan Gambar 2.11 Struktur *Relay*, besi (*iron core*) yang dililit oleh kumparan *coil* yang berfungsi untuk mengendalikan besi tersebut. Jika kumparan *coil* diberikan arus listrik, maka akan menimbulkan sebuah gaya elektromagnet yang kemudian 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 *open* atau tidak terhubung. Pada saat tidak dialiri arus listrik, *armature* akan kembali lagi ke posisi awal (NC). *Coil* yang digunakan oleh *relay* untuk menarik *contact point* ke posisi *close* hanya membutuhkan arus listrik yang relatif kecil.

## 12. *Liquid Crystal Display (LCD) 16 x 2*

### a. Pengertian *Liquid Crystal Display (LCD)*

Merupakan jenis media tampil menggunakan kristal cair sebagai penampil utama. LCD banyak digunakan diberbagai bidang, misalnya alat elektronik seperti televisi, kalkulator, atau layar komputer. LCD berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat (Mundandar, 2012).

LCD 16×2 mempunyai 16 *pin* dan untuk pengkabelanya adalah sebagai berikut (Anugrah, 2016):

- 1) Kaki 1 dan 16 terhubung dengan *Ground (GND)*.
- 2) Kaki 2 dan 15 terhubung dengan *VCC (+5V)*.
- 3) Kaki 3 dari LCD 16×2 adalah *pin* yang digunakan untuk mengatur

kontras kecerahan LCD. Jadi kita bisa memasang sebuah trimpot 103 untuk mengatur kecerahannya. Pemasangannya seperti terlihat pada rangkaian tersebut. Karena LCD akan berubah kecerahannya jika tegangan pada *pin* 3 ini di turunkan atau dinaikan.

- 4) *Pin* 4 (RS) dihubungkan dengan *pin* mikrokontroler.
- 5) *Pin* 5 (RW) dihubungkan dengan GND.
- 6) *Pin* 6 (E) dihubungkan dengan *pin* mikrokontroler.
- 7) Sedangkan *pin* 11 hingga 14 dihubungkan dengan *pin* mikrokontroler sebagai jalur datanya.

b. Cara Kerja LCD 16 x 2

Menurut Mundandar (2012) cara kerja LCD 16 x 2 yaitu pada aplikasi umumnya RW diberi logika rendah "0". Bus data terdiri dari 4 *bit* atau 8 *bit*. Jika jalur data 4 *bit* maka yang digunakan ialah DB4 sampai dengan DB7. *Interface* LCD merupakan sebuah *parallel* bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8 *bit* dikirim ke LCD secara 4 *bit* atau 8 bit pada satu waktu. Jika mode 4 *bit* yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8 *bit* (pertama dikirim 4 *bit* MSB lalu 4 *bit* LSB dengan pulsa *clock* EN setiap *nibble* nya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus mengatur EN ke kondisi *high* "1" dan kemudian

mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

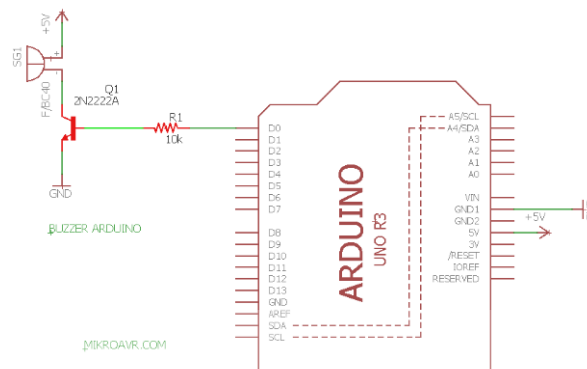
Saat jalur lainnya sudah siap, EN harus diatur ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan *set* EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan yaitu data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diatur ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get LCD status* (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diatur ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara *parallel* baik 4 *bit* atau 8 *bit* merupakan 2 mode operasi *primer*. Untuk membuat sebuah aplikasi *interface* LCD, menentukan mode operasi merupakan hal yang paling penting.

Mode 8 *bit* sangat baik digunakan saat kecepatan menjadi keutamaan dalam sebuah aplikasi dan minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 *pin* untuk data). Sedangkan mode 4 *bit* minimal hanya

membutuhkan 7 *bit* (3 *pin* untuk kontrol, 4 *pin* untuk data). *Bit* RS digunakan untuk memilih data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika *bit* ini di *set* ( $RS = 1$ ), maka *byte* pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika *bit* ini direset ( $RS = 0$ ), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

### 13. *Buzzer*

*Buzzer* adalah sebuah komponen elektronika yang mengubah energi listrik menjadi energi mekanik atau getaran. Energi getaran ini akan menghasilkan suara. *Buzzer* juga biasanya digunakan untuk indikator suara untuk alarm, *input keypad*, dan pemberitahuan kerusakan pada sebuah sistem elektronik, seperti di *motherboard* komputer. *Buzzer* ini biasanya memiliki tegangan kerja antara 3 *volt* sampai dengan 12 *volt*, biasanya jika kita beli sudah di jelaskan tegangan kerjanya. jika tegangan kerja di bawah 5 *volt* bisa kita langsung hubungkan ke *Arduino*. Tapi jika tegangan kerjanya menggunakan 12 *volt*, maka kita membutuhkan rangkaian *driver* untuk *buzzer* (Sitepu, 2018).



Gambar 2.12 Rangkaian *Buzzer* Pada *Arduino*

(Sumber: <https://mikroavr.com>)

Gambar 2.12 Rangkaian *Buzzer* Pada *Arduino*, menjelaskan bahwa pada pin D0 terhubung ke basis *Transistor* NPN 2N2222, disini rangkaian dibuat sebagai aktif *high*, ketika PIN D0 logika 1, maka *buzzer* akan menyala, dan ketika PIN D0 logika 0 maka *Buzzer* akan padam.

## B. Penelitian Sebelumnya

1. Penelitian yang dilakukan oleh Lontoh, Mamahit & Tulung (2017) dengan judul Rancang Bangun Kunci Pintu Elektronik Menggunakan *Bluetooth* Berbasis *Android*. Tujuan dari penelitian ini adalah menciptakan sebuah sistem keamanan pada kunci rumah dengan memanfaatkan *Bluetooth* dan aplikasi rancangan sendiri yang telah terinstal pada *Smartphone Android*. Penelitian ini menghasilkan sistem keamanan kunci pintu elektronik yang memanfaatkan *Bluetooth* dan *Android* yang dapat menjadikan sebuah pilihan untuk keamanan rumah

yang praktis dan dari segi keamanannya dapat dipercaya serta tidak mudah untuk diakses oleh orang lain yang tidak memiliki akses. Kunci pintu elektronik ini menggunakan jaringan *Bluetooth* sebagai media penghubung antara mikrokontroler *Arduino* dengan aplikasi yang ada pada *Smartphone Android*. Sistem kunci pintu ini akan membuka ketika tombol buka pada aplikasi yang terdapat di *Smartphone Android* di tekan atau klik. Kelemahan pada sistem sebelumnya, yaitu tidak ada media validasi tambahan untuk pengamanan data, selain itu belum terdapat indikator pintu terbuka dan terkunci. Perbedaan dengan penelitian yang akan dilakukan adalah menerapkan *QR Code* sebagai media untuk membuka kunci pintu menggantikan kunci konvensional atau kunci pintu biasa, serta mengimplementasikan *solenoid* sebagai alat kunci elektronik yang akan bekerja otomatis (membuka kunci dan menutup kunci) apabila dialiri arus listrik.

2. Penelitian yang dilakukan oleh Jufri (2016) berjudul Rancang Bangun Dan Implementasi Kunci Pintu Elektronik Menggunakan *Arduino* dan *Android*. Tujuan penelitian ini adalah menghasilkan sebuah alat kunci pintu elektronik yang dapat dikendalikan dari *Smartphone Android* dengan komunikasi serial melalui modul *Bluetooth HC-05*. Penelitian ini menggunakan mikrokontroler *Arduino Pro Mini* sebagai pengendali *solenoid* dan modul *Bluetooth HC-05* yang berfungsi untuk *transfer* data antar piranti *Arduino* dan *Android*. Pada penelitian ini menggunakan komunikasi serial melalui *Bluetooth* dengan konfigurasi



*baud rate* 9600. Metode penelitian yang digunakan yaitu metode penelitian tindakan. Metode penelitian tindakan mengacu pada model Kurt Lewin. Tahapan dari model penelitian tindakan adalah perencanaan, tindakan, pengamatan dan refleksi. Kelemahan dari sistemnya yaitu belum menerapkan validasi tambahan untuk keamanan data, validasi atau proses membuka pintu hanya dengan menghubungkan *Bluetooth* pada *Smartphone Android* dengan *module HC-05*. Perbedaan dari penelitian yang akan dilakukan yaitu mengembangkan media komunikasi *transfer* data dengan modul *Bluetooth* berupa *QR Code* sebagai media validasi data.

3. Penelitian yang dilakukan oleh Najjar (2017) dengan judul Rancang Bangun Keamanan Pintu Berbasis *Arduino Uno* Dengan *Quick Response Code* Pada Ruang Laboratorium Komputer di SMK Negeri 1 Tambelang. Tujuan Penelitian ini yaitu menerapkan *QR Code* sebagai keamanan akses pintu masuk Laboratorium, sehingga ruangan tersebut dapat terkontrol dengan aman serta nyaman dan tidak sembarang orang bisa masuk kedalam ruangan. Penelitian ini menghasilkan alat akses kunci yang memanfaatkan *Bluetooth* dan *QR Code* sebagai media pengiriman data. Penelitian ini menggunakan *module HC-06* sebagai media komunikasi *Arduino* dengan *Smartpone Android*. *HC-06* memiliki kelemahan dimana jenis *module Bluetooth* ini tidak dapat dijadikan sebagai *master*, hanya dapat digunakan sebagai *slave* saja. Pada penelitian ini indikator pintu terbuka dan terkunci hanya

ditampilkan pada *serial monitor* yang terdapat di *software Arduino IDE*. Perbedaan dengan penelitian yang akan dilakukan adalah menggunakan *solenoid smart door lock* yang dikendalikan dan digerakan oleh *relay*. Kemudian *QR Code* ditampilkan pada *LCD SPI TFT*.

4. Penelitian yang dilakukan oleh Pradana, Gusti & Rahardjo (2019) yang berjudul Rancang Bangun *Vending Machine* Menggunakan *QR Code* Berbasis Mikrokontroler. Tujuan dari penelitian ini adalah untuk mengembangkan sebuah *Vending Machine* dengan memanfaatkan teknologi *QR Code* yang terintegrasi dengan *Smartphone* berbasis *android*. Penelitian ini menghasilkan sebuah *Vending Machine* yang dapat dirancang menggunakan *Arduino Nano* sebagai mikrokontroler utama, motor *servo* sebagai penahan minuman dan mengeluarkan minuman ketika diperintahkan oleh *Arduino Nano*, serta modul *Bluetooth HC-05* yang berperan sebagai *master* dan *slave* untuk komunikasi antara *Vending Machine* dengan *Smartphone* melalui aplikasi *android*. Sistem ini akan bekerja ketika *Arduino Nano* memproses data serial yang dikirimkan oleh *smartphone android*. Pada penelitian ini aplikasi belum terdapat fitur pilih *Bluetooth* yang tersedia. Perbedaan dengan penelitian yang akan dilakukan adalah mengimplementasikan *QR Code* untuk membuat sebuah sistem keamanan pintu rumah bersama dengan *Arduino Uno R3* sebagai pengendali dan *Solenoid Smart Door Lock* sebagai kunci otomatis.

5. Penelitian yang dilakukan oleh Guntoro, Somantri, & Haritman (2013) dengan judul Rancang Bangun *Magnetic Door Lock* Menggunakan *Keypad* Dan *Solenoid* Berbasis Mikrokontroler *Arduino Uno*. Tujuan penelitian ini adalah untuk membuat alat *magnetic door lock* berbasis *Arduino Uno* sebagai keamanan rumah seperti pintu, lemari, loker, brankas, dan lainnya secara elektronik tanpa harus menggunakan kunci konvensional. Penelitian ini menghasilkan sebuah *magnetic door lock* menggunakan *keypad* sebagai alat untuk membuka pintu. Kelemahan dari sistem pada penelitian ini adalah tidak memiliki *display* untuk menunjukkan tampilan kode *password* yang dimasukkan serta sistem ini tidak memiliki algoritma pemrograman untuk sistem proteksi, sehingga orang yang tidak memiliki akses dapat mencoba-coba menekan kode *password* melalui *keypad* hingga mendapatkan *password* yang *valid* atau benar. Perbedaan dengan penelitian yang akan dilakukan yaitu menggunakan *QR Code* sebagai media pembuka pintu.

Berdasarkan penelitian yang dilakukan oleh Lontoh, Mamahit & Tulung (2017) mengatakan bahwa dengan menggunakan *Smartphone Android* sebagai media untuk membuka pintu rumah menggantikan kunci konvensional lebih mudah dan praktis, selain itu dengan *Android* yang bersifat *open source* akan dapat dimanfaatkan dengan baik untuk melakukan inovasi dalam bidang keamanan. Dengan memanfaatkan *Smartphone Android* sebagai media pembuka kunci tentu akan memberikan sebuah alternatif bagi pemilik rumah, disamping itu penggunaan *smartphone android* yang sudah umum dikalangan masyarakat tentu menjadi sebuah kemudahan bagi pemilik rumah, karena tidak harus lagi membawa kunci konvensional ketika pergi meninggalkan rumah, serta dari segi keamanan memiliki keamanan yang baik. Penelitian oleh Guntoro, Somantri, & Haritman (2013) menjelaskan bahwa dengan menggunakan *keypad* sebagai media untuk membuka pintu masih terdapat sebuah kelemahan yaitu tidak adanya algoritma pemrograman untuk sistem proteksi, sehingga orang lain dapat mencoba-coba menekan kode *password* melalui *keypad* hingga akhirnya mendapatkan kode *password* yang sesuai atau benar.

Berdasarkan penelitian oleh Najjar (2017) dan penelitian yang dilakukan oleh Pradana, Gusti & Rahardjo (2019). Dengan menggunakan *QR Code* sebagai media pembuka pintu akan memberikan sebuah sistem keamanan yang lebih tinggi dan kuat. *QR Code* mampu memperbaiki kesalahan sampai dengan 30%. Selain itu saat *QR Code* kotor atau rusak,

data yang ada akan tetap bisa terbaca, pembacaan kode QR dapat dilakukan dari sudut mana saja, dan hasilnya tetap akan sama. *QR Code* juga dapat menyimpan berbagai jenis data, kombinasi dari berbagai karakter dan jenis data dapat dilakukan, sehingga dengan *QR Code* akan memberikan sebuah keamanan yang lebih baik dari media-media lain. Untuk pembacaan atau proses *scan QR Code* membutuhkan waktu 3-4 detik.



Tabel 2.2 Perbandingan Penelitian Sebelumnya

No.	Peneliti	Judul Penelitian	Persamaan	Perbedaan	
				Penelitian Terdahulu	Penelitian Sekarang
1.	Lontoh, Mamahit & Tulung (2017)	Rancang Bangun Kunci Pintu Elektronik Menggunakan <i>Bluetooth</i> Berbasis <i>Android</i> .	Media komunikasi data menggunakan <i>Bluetooth</i> dan <i>Android</i> . <i>Arduino</i> yang digunakan sama yaitu <i>Arduino Uno R3</i> .	<ul style="list-style-type: none"> <li>• Teknologi <i>QR Code</i> belum diterapkan sebagai media untuk membuka kunci pintu rumah.</li> <li>• Pintu masih menggunakan <i>motor servo dc</i> sebagai penggerak pintu.</li> <li>• Belum adanya media validasi tambahan untuk pengamanan data.</li> <li>• Tidak terdapat indikator untuk pintu terbuka dan pintu terkunci.</li> </ul>	<ul style="list-style-type: none"> <li>• Menerapkan <i>QR Code</i> sebagai media untuk validasi dan membuka kunci pintu rumah.</li> <li>• Kunci pintu menggunakan <i>solenoid</i> dengan menggunakan <i>relay</i> sebagai pengendali.</li> <li>• Menggunakan kode <i>random</i> atau acak yang berguna sebagai validasi keamanan data pada aplikasi <i>Smart QReader</i> dan pada alat <i>Smart Door Lock</i>.</li> <li>• LCD, LED dan <i>Buzzer</i> sebagai indikator pintu terbuka dan terkunci.</li> </ul>

Tabel 2.2 Perbandingan Penelitian Sebelumnya (Lanjutan)

No.	Peneliti	Judul Penelitian	Persamaan	Perbedaan	
				Penelitian Terdahulu	Penelitian Sekarang
2.	Jufri (2016)	Rancang Bangun Dan Implementasi Kunci Pintu Elektronik Menggunakan <i>Arduino</i> dan <i>Android</i> .	Menggunakan modul <i>Bluetooth HC-05</i> sebagai media pengiriman data dan penghubung antara <i>Smartphone Android</i> dengan <i>Arduino</i>	<ul style="list-style-type: none"> <li>• <i>Arduino</i> yang digunakan adalah tipe <i>Arduino Pro Mini</i>.</li> <li>• Validasi data yang berguna untuk membuka pintu hanya menggunakan <i>Bluetooth</i>.</li> <li>• Tidak terdapat indikator atau tanda ketika pintu membuka dan ketika pintu terkunci.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan tipe atau jenis <i>Arduino Uno R3</i>, yang memiliki fitur serta keamanan yang lebih baik.</li> <li>• Menambah media validasi data berupa <i>QR Code</i> dan kode khusus untuk keamanan akses kunci pintu dan aplikasi <i>QR Code Reader</i>.</li> <li>• LED, LCD, dan <i>Buzzer</i> berguna sebagai indikator pintu terbuka dan terkunci.</li> </ul>
3.	Najar (2017)	Rancang Bangun Keamanan Pintu Berbasis <i>Arduino Uno</i> Dengan <i>Quick Response Code</i> Pada Ruang	Menggunakan <i>QR Code</i> sebagai media untuk membuka dan memvalidasi data, serta menggunakan	<ul style="list-style-type: none"> <li>• Menggunakan <i>Bluetooth HC-06</i>.</li> <li>• Tidak terdapat indikator membuka dan mengunci pintu.</li> <li>• Aplikasi <i>QR Code</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan <i>Bluetooth HC-05</i> yang bisa digunakan untuk <i>slave</i> atau <i>master</i> (penerima atau pengirim data).</li> </ul>

Tabel 2.2 Perbandingan Penelitian Sebelumnya (Lanjutan)

No.	Peneliti	Judul Penelitian	Persamaan	Perbedaan	
				Penelitian Terdahulu	Penelitian Sekarang
		Laboratorium Komputer di SMK Negeri 1 Tambelang.	<i>Bluetooth</i> yang berguna untuk menghubungkan perangkat <i>Smartphone Android</i> dengan mikrokontroler <i>Arduino</i> .	<p><i>reader</i> menggunakan <i>App Inventor2</i>.</p> <ul style="list-style-type: none"> <li>Gambar <i>QR Code</i> masih harus di print atau di tampilkan secara manual.</li> </ul>	<ul style="list-style-type: none"> <li>Aplikasi <i>QR Code Reader</i> dibuat dengan <i>software Kodular</i>.</li> <li>Menambahkan indikator pintu terbuka dan terkunci dengan LED, LCD, dan <i>Buzzer</i>.</li> <li>Gambar <i>QR Code</i> ditampilkan pada LCD TFT ST7735.</li> </ul>
4.	Pradana, Gusti & Rahardjo (2019)	Rancang Bangun <i>Vending Machine</i> Menggunakan <i>QR Code</i> Berbasis Mikrokontroler.	<i>Bluetooth</i> berfungsi sebagai media komunikasi data, dan <i>Smartphone android</i> berfungsi sebagai media membaca <i>QR Code</i> .	<ul style="list-style-type: none"> <li>Aplikasi <i>QR Code Reader</i> belum terdapat fitur untuk memilih <i>Bluetooth</i> yang tersedia.</li> <li>Penggerak pintu masih menggunakan <i>motor servo dc</i></li> </ul>	<ul style="list-style-type: none"> <li>Menambahkan fitur pilih <i>Bluetooth</i> dan fitur <i>exit</i> aplikasi.</li> <li>Kunci pintu menggunakan <i>solenoid</i> yang dikendalikan oleh <i>relay</i> dan komponen ini bisa mengunci otomatis.</li> </ul>



Tabel 2.2 Perbandingan Penelitian Sebelumnya (Lanjutan)

No.	Peneliti	Judul Penelitian	Persamaan	Perbedaan	
				Penelitian Terdahulu	Penelitian Sekarang
				<ul style="list-style-type: none"> <li>• <i>Arduino</i> yang digunakan adalah tipe <i>Arduino Nano</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan tipe atau jenis <i>Arduino Uno R3</i>, yang memiliki fitur serta keamanan yang lebih baik.</li> </ul>
5.	Guntoro, Somantri & Haritman (2013)	Rancang Bangun <i>Magnetic Door Lock</i> Menggunakan <i>Keypad</i> Dan <i>Solenoid</i> Berbasis Mikrokontroler <i>Arduino Uno</i> .	Menggunakan <i>Arduino Uno R3</i> sebagai pengendali dan <i>solenoid</i> sebagai kunci pintu elektronik.	<ul style="list-style-type: none"> <li>• Media validasi atau untuk membuka kunci menggunakan <i>keypad</i>.</li> <li>• Belum menerapkan smartphone android sebagai media untuk membuka pintu.</li> <li>• Tidak terdapat proses atau fitur pengamanan data.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan <i>Bluetooth HC-05</i> yang bisa digunakan untuk <i>slave</i> atau <i>master</i> (penerima atau pengirim data).</li> <li>• Menggunakan smartphone android sebagai pembaca atau validasi <i>QR Code</i>.</li> <li>• Menggunakan kode acak atau <i>random</i> yang terdapat pada aplikasi <i>Smart QReader</i> dan Alat <i>Smart Door Lock</i>.</li> </ul>