

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Berdasarkan Konsep Keamanan dengan *Android*

Dalam membuat penelitian ini terdapat beberapa referensi yang di ambil oleh penulis dari penelitian – penelitian terdahulu, yaitu:

Penelitian yang dilakukan oleh Slamet Winardi, dkk (2016) tentang Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan *Android* Berbasis Arduino. Mereka membahas tentang rancang bangun sistem keamanan pintu rumah dengan memanfaatkan *Android* sebagai pengendali yang berisi perintah berupa *Voice Recognition* (suara berisi perintah buka atau tutup), yang kemudian *bluetooth* sebagai penghubung antara *Android* ke pusat kendali yaitu Arduino akan melanjutkan perintah ke *Solenoid door lock* sehingga dapat memberikan akses untuk membuka dan menutup kunci pada pintu.

Penelitian yang dilakukan Jeril H.Lontoh, dkk (2017) tentang Rancang Bangun Kunci Pintu Elektronik Menggunakan *bluetooth* Berbasis *Android*. Mereka membahas tentang penerapan sistem pengamanan pintu rumah dengan menggunakan *Android* sebagai media pengontrol untuk membuka dan mengunci pintu rumah dengan *bluetooth* sebagai media penghubung antara *Android* dengan Arduino sebagai pusat kendali dan selanjutnya Arduino akan memberikan perintah kepada motor *DC servo* untuk bergerak membuka dan menutup kunci pintu sesuai dengan perintah dari aplikasi pada *Android*.

Penelitian yang dilakukan Ai Fitri Silvia, dkk (2014) tentang Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino Dan Android. Mereka membahas tentang penerapan sistem pengendali pintu gerbang otomatis dengan menggunakan *Android* sebagai media pengontrol untuk membuka dan menutup pintu gerbang dengan *bluetooth* sebagai media penghubung antara *Android* dengan Arduino sebagai pusat kendali dan selanjutnya Arduino akan memberikan perintah kepada motor *DC servo* untuk bergerak membuka dan menutup pintu gerbang sesuai dengan perintah dari aplikasi pada *Android*.

Penelitian yang dilakukan Ahmad Jufri (2016) tentang Rancang Bangun Dan Implementasi Kunci Pintu Elektronik Menggunakan Arduino dan Android. Dalam jurnal yang dibahas penulis menjelaskan bahwa untuk menggantikan kunci pintu manual penulis menggunakan *Android* sebagai alat untuk membuka kunci dan menutup kunci. Untuk dapat melakukannya pengguna harus *login* terlebih dahulu agar bias mengontrol kunci pintu melalui *Bluetooth*.

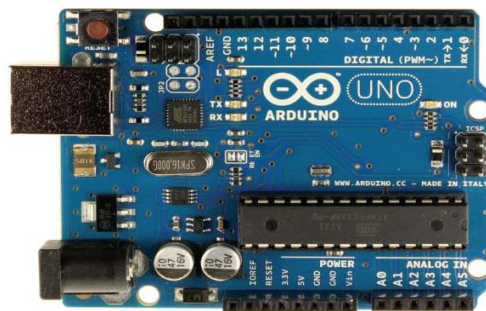
Penelitian yang dilakukan oleh Ade Septryani, dkk (2017) tentang Rancang Bangun Aplikasi Kunci Pintu Otomatis Berbasis Mikrokontroler Menggunakan *Smartphone Android*. Mereka membahas tentang percobaan penerapan aplikasi *QRCodeScanner* terhadap *Smartphone Android* dengan fasilitas koneksi oleh *Bluetooth*. Dengan demikian mereka berharap bahwa dengan pemanfaatan *Smartphone Android* sebagai alat pengganti anak kunci pintu maka akan di dapat hasil yang lebih canggih, praktis ,dan efisien.

II.2. Uraian Teoritis

Adapun teori – teori yang digunakan sebagai landasan dalam penulisan proposal skripsi ini adalah sebagai berikut:

II.2.1. *Microcontroller Arduino Uno*

Arduino merupakan sebuah pengendali mikro single-board yang bersifat open-source, di turunkan dari Wiring Platform dan dirancang untuk memudahkan pengguna elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR (*Automatic Voltage Regulator*) atau Atmel ARM (*Acorn RISC Machine*) dan softwernya memiliki bahasa pemrograman sendiri. Contoh seketsa *Board Arduino uno* bisa di lihat pada Gambar II.1. (Jeril H.Lontoh dkk, 2017).



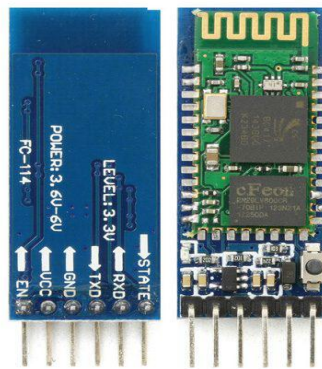
**Gambar II.1. Sketsa Board Arduino Uno
(Sumber: Jeril H. Lontoh dkk, 2017)**

Komponen utama dari rangkaian Arduino Uno adalah IC Mikrokontroler sebagai prosesnya. Mikrokontroler yang digunakan dalam sistem ini adalah mikrokontroler dengan jenis AVR seri ATmega 328. Mikrokontroler ini mempunyai 20 pin yang meliputi 14 pin I/O digital dengan 6 pin yang dapat berfungsi sebagai output PWM (*Pulse Width Module*) dan 6 pin I/O analog.

II.2.2. *Bluetooth Module HC-06*

Bluetooth Module HC-05 merupakan module komunikasi nirkabel pada frekuensi 2,4GHz dengan pilihan koneksi bias sebagai slave, ataupun sebagai master. Sangat mudah di gunakan dengan mikrokontroler untuk membuat aplikasi *wireless*. *Interface* yang di gunakan adalah serial *RXD*, *TXD*, *VCC*, dan *GND*. *Build in LED* sebagai indikator koneksi *Bluetooth Module*.

Tegangan input antara 3,6-6V, jangan menghubungkan nya dengan daya lebih dari 7V. Arus saat unpaired sekitar 30mA, dan saat paired (terhubung) sebesar 10mA. 4 pin intervace 3,3V dapat di hubung kan langsung ke berbagai macam mikrokontroler (khusus Arduino, 8051, 8535, *AVR*, *PIC*, *ARM*, *MSP430*, etc.). Jarak efektif jangkauan jaringan ini sampai dengan 10m, bias lebih dari 10m namun kualitas koneksi kurang baik. Contoh dari Bluetooth Module HC-05 bisa di lihat pada Gambar II.3. (Jeril H.Lontoh dkk, 2017).



Gambar II.3. Bluetooth Module HC-05
(Sumber: Jeril H. Lontoh dkk, 2017)

II.2.3. *Smartphone*

Smartphone merupakan kombinasi fungsi dari perangkat komunikasi dan perangkat penunjang kebutuhan digital lifestyle dengan beberapa fitur multimedia dan organizer. Seiring perkembangan zaman, *smartphone* sekarang ditunjang dengan fitur GPS untuk navigasi, NFC untuk komunikasi instan dalam pertukaran data. Pada umumnya *smartphone* memiliki prosesor yang cukup tinggi berkat teknologi *SoC (System on Chip)* yang menghadirkan kemampuan *hardware* yang tinggi namun dengan ukuran yang kompak. (Angger dkk, 2017).

II.2.4. *Android*

Android merupakan salah satu operasi sistem pada perangkat *mobile* yang berlogo hijau seperti yang kita bias lihat pada (gambar : II.3. Logo Android). Dalam pengembangan aplikasi *android* menggunakan platform java sebagai bahasa pemrogramannya. Google bekerjasama dengan lebih dari 47 perusahaan lain yang tergabung dalam OHA yaitu (*Open Handset Alliance*) untuk membuat standar pada perangkat *mobile*. (Angger dkk, 2017).

Berikut ini adalah beberapa fitur yang terdapat pada *smartphone* dengan operating sistem *android*:

1. *Touch screen*. Dengan menggunakan fitur ini, mengkiatkan navigasi menu menjadi lebih mudah dan efisien dengan hanya menyentuh layar.
2. *Multipage* Sangat berguna untuk keperluan *multitasking* Pengguna dapat berpindah *page* tanpa perlu menutup *page* sebelumnya untuk digunakannanti.
3. Merupakan sistem operasi terbuka (*open source*) sehingga dapat dimodifikasi

bahkan membuat dengan bebas.

4. Memiliki kualitas grafik dan suara yang bagus karena standar yang digunakan seperti MP3 dengan grafik 3D.
5. Terdapat perangkat pendukung seperti, *wifi*, *bluetooth*, kamera, dan GPS.
6. Untuk beberapa *smartphone* juga telah di bekali *Finger Print* sebagai pengaman.

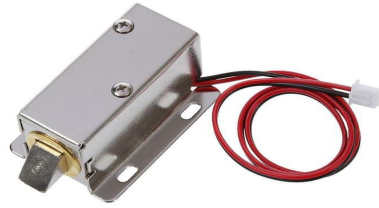
Contoh dari logo *Android* bisa di lihat pada Gambar II.4. (Angger dkk, 2017).



(Gambar : II.4. Logo Android)
(Sumber : Angger Dimas Bayu Sadewo dkk, 2017)

II.2.5. Solenoid

Solenoid adalah salah satu jenis kumparan terbuat dari kabel panjang yang di lilitkan secara rapat dan dapat diasumsikan bahwa panjangnya jauh lebih besar daripada diameternya. Dalam kasus *solenoid*, ideal panjang kumparan adalah tak terhingga dan dibangun dengan kabel yang saling berhimpitan dalam lilitannya, dan medan magnet di dalamnya adalah seragam dan paralel terhadap sumbu solenoid. Contoh bisa di lihat pada Gambar II.5. Solenoid Lock. (Ade Septriyanti dkk, 2017).



(Gambar : II.5.Solenoid Lock)
(Sumber : Selamat Winardi dkk, 2016)

II.2.6. 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* seperti pada Gambar II.6 sebagai berikut:



Gambar : II.6. Bentuk Fisik Buzzer
(Sumber : Prihono, dkk.,2009)

II.2.7. Relai

Relai adalah suatu peranti yang bekerja berdasarkan asas elektromagnetik untuk menggerakkan sejumlah kontaktor (saklar). Kontaktor akan tertutup (*off*) atau terbuka (*on*) karena induksi magnet yang dihasilkan kumparan ketika dialiri listrik (Lena dan Putrawan, 2014). Relai terdiri dari *coil* dan *contact*, *coil* adalah gulungan kawat yang mendapat arus listrik, sedangkan *contact* adalah sejenis saklar yang dipengaruhi dari ada tidaknya arus listrik pada *coil*. Adapun bentuk fisik dari relai seperti pada Gambar II.7.



Gambar : II.7. Bentuk Fisik Relai
(Sumber :Eko Saputro,2016)

II.3. Unified Modelling Language (UML)

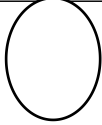
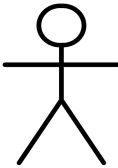
Menurut Ade Hendini (2016), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan




alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut:

II.3.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel II.1.

Tabel II.1 Simbol Use Case Diagram

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan system sebagai unit-unit yang bertukar pesan antar unit dengan aktif yang dinyatakan dengan menggunakan kata kerja.</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau system yang lain yang mengaktifkan fungsi dari target system. Untuk mengidentifikasi system, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target system.</p>



	<p>Asosiasi antara system dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara system dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila system berinteraksi secara pasif dengan system.</p>
	<p><i>Include</i>, merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain.</p>


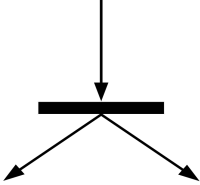
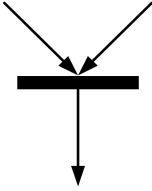
(Sumber: Ade Hendini, 2016)

II.3.2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel II.2.

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan
	<p><i>Start Point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.</p>
	<p><i>End Point</i>, akhir aktivitas.</p>

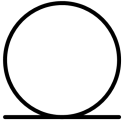
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> / percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.

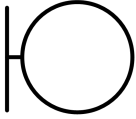
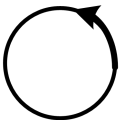

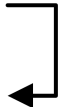
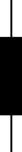

(Sumber: Ade Hendini, 2016)

II.3.3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol Sequence Diagram

Gambar <i>graphia</i>	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.</p>
	<p><i>Control Class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar class.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</p>

(Sumber: Ade Hendini, 2016)

II.3.4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*. Simbol – simbol *Class Diagram* dapat dilihat pada Tabel II.4.

Tabel II.4. Simbol Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Mempunyai arti minimal n maksimal n

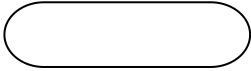


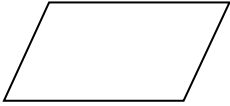
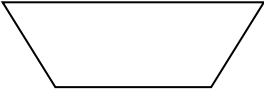
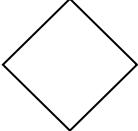
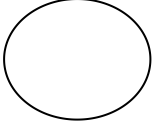


(Sumber: Ade Hendini, 2016)

II.3.5. Flowchart

Menurut Santoso dan Radna Nurmalina (2017), *flowchart* adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang.

Flowchart membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah. Simbol-simbol flowchart dapat di lihat pada Tabel II.5.

Tabel II.5. Simbol Flowchart

	<p>Terminator Simbol Yaitu simbol dari suatu permulaan (Start) atau simbol dari akhir (Stop) dari suatu kegiatan.</p>
	<p>Flow Direction Simbol Yaitu simbol yang di gunakan untuk menghubungkan antara simbol satu dengan lainnya atau <i>connecting line</i>.</p>
	<p>Processing Simbol Simbol yang menunjukkan pengolahan oleh komputer.</p>
	<p>Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.</p>
	<p>Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak di lakukan oleh komputer.</p>
	<p>Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.</p>
	<p>Connector Simbol Yaitu simbol untuk keluar masuk atau penyambungan proses pada lembar/halaman yang berbeda.</p>
	<p>Simbol Manual Input Simbol untuk memasukkan data secara manual on-line keyboard.</p>
	<p>Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output di cetak ke kertas.</p>

(Sumber: Santoso dkk, 2017)