

**PEMBANGUNAN *EMBEDDED SYSTEM SMART LOCK* BERBASIS
NODEMCU DAN RFID**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan
Jenjang Strata Satu (S1)
pada Program Studi Teknik Informatika**

Oleh

Cindy Veronika Hutahean

361461011



SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER INDONESIA

MANDIRI

BANDUNG

2021

LEMBAR PENGESAHAN
PEMBANGUNAN *EMBEDDED SYSTEM SMART LOCK*
BERBASIS NODEMCU DAN RFID

Oleh
Cindy Veronika Hutahean
361461011

Tugas Akhir ini telah diterima dan disahkan untuk
memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI

Bandung, Juni 2021
Disahkan oleh

Ketua Program Studi,



Chalifa Chazar, S.T.,M.T.
NIDN. 0421098704

Dosen Pembimbing,

A handwritten signature in black ink, which appears to be 'Chalifa', written over a faint circular stamp.

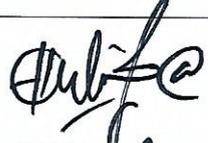
Chalifa Chazar, S.T.,M.T.
NIDN. 0421098704

LEMBAR PERSETUJUAN REVISI
PEMBANGUNAN *EMBEDDED SYSTEM SMART LOCK*
BERBASIS NODEMCU DAN RFID

Oleh
Cindy Veronika Hutahean
361461011

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, Juni 2021
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Chalifa Chazar, S.T., M.T.	Pembimbing	
2.	Hendra Gunawan, S.T., M.Kom.	Penguji 1	
3.	Novi Rukhviyanti, S.T., M.Si.	Penguji 2	

Mengetahui
Ketua Program Studi Teknik Informatika



Chalifa Chazar, S.T., M.T.
NIDN. 0421098704

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Juni 2021

Yai  taaan



Cindy Veronika Hutahean

361461011

ABSTRAK

PEMBANGUNAN *EMBEDDED SYSTEM SMART LOCK* BERBASIS NODEMCU DAN RFID

Oleh

Cindy Veronika Hutahean

361461011

Pintu menjadi sarana keluar dan masuk dimana awal dari kegiatan sehari-hari dimulai. Pintu yang dijaga dengan baik akan membuat penghuni merasa aman saat berada didalam maupun diluar dan untuk menjaga keamanan tersebut maka dibutuhkan suatu sistem keamanan yang baik dimana penerapan teknologi pada keamanan pintu akan meningkatkan keamanan dan kenyamanan pengguna dibandingkan dengan keamanan kunci konvensional yang saat ini mudah sekali dibobol, sehingga sistem ini dirasakan kurang praktis dan kurang modern untuk saat ini. Gagasan untuk melakukan suatu penelitian terkait pembuatan suatu kunci pintu pintar dengan tingkat keamanan yang lebih tinggi dan rahasia yakni dengan adanya pembangunan *embedded system smart lock* berbasis NodeMCU dan RFID dapat menjadi solusi dari permasalahan kunci konvensional dimana Modul RFID MRFC522 sebagai sensor dan kunci elektronik untuk membuka pintu dan NodeMCU ESP8266 (Amica) sebagai board utama, dengan komponen pendukung berupa Servo sebagai motor penggerak slot kunci pintu. Pemograman inti menggunakan software Arduino IDE dan sistem yang dirancang pun berbasis IoT dengan bantuan komunikasi Wi-fi dan software blynk sebagai monitoring dan notifikasi. Metode pengembangan yang dilakukan dalam penelitian menggunakan metode waterfall dengan lima tahapan proses, diantaranya perencanaan, analisis, desain, implementasi hingga penggunaan dimana setiap tahap akan mempengaruhi dan menjadi dasar dari keberlangsungan tahap pengembangan berikutnya.

Kata kunci: *Embedded System, Radio Frequency Identification (RFID), Smart Lock, NodeMCU, Blynk*

ABSTRACT

DEVELOPMENT OF EMBEDDED SYSTEM SMART LOCK BASED ON NODEMCU AND RFID

By

Cindy Veronika Hutahean

361461011

The door is a means of entering and leaving where the beginnings of daily activities begin. A well-guarded door will make resident feel safe both inside and outside and to maintain that security, a good security system is needed where the application of technology to door security will increase user security and comfort compared to conventional lock security which is currently very easy to break. To conduct a research related to the manufacture of a smart door lock with a higher level of security and secret, namely the development of an embedded smart lock system based on NodeMCU and RFID can be a solution to conventional lock problems where the MRFC522 RFID Module is used as a sensor and electronic key to open doors and locks. NodeMCU ESP8266 (Amica) as the main board, with supporting components in the form of a Servo as the driving motor for the door lock slot. The core programming uses Arduino IDE software and the designed system is based on IoT with the help of Wi-fi communication and blynk software as monitoring and notification. The development method carried out in this study uses the waterfall method with five stages of the process, including planning, analysis, design, implementation to use where each stage will influence and become the basis for the sustainability of the next development stage.

Keywords: Embedded System, Radio Frequency Identification (RFID), Smart Lock, NodeMCU, Blynk

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas kebaikan dan cinta kasih-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pembangunan *Embedded System Smart Lock* Berbasis NodeMCU dan RFID”. Adapun skripsi ini bertujuan untuk memenuhi salah satu syarat kelulusan di Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika & Komputer Indonesia Mandiri.

Dalam penyusunan skripsi ini penulis menyadari bahwa telah banyak pihak yang membantu baik berupa pemberian materi maupun dukungan moril, karena itu penulis menyampaikan rasa terima kasih kepada semua pihak yang telah mengulurkan tangannya sehingga tugas akhir ini dapat diselesaikan.

Penulis menyadari isi dari skripsi ini masih jauh dari sempurna, meskipun demikian penulis berharap skripsi ini dapat sesuai dengan tujuan yang diharapkan berguna bagi penulis khususnya dalam dunia pendidikan umumnya dan bagi perkembangan ilmu teknologi dimasa yang akan datang.

Bandung, Juni 2021

Cindy Veronika Hutahean
361461011

UCAPAN TERIMA KASIH

Dalam penyusunan skripsi ini tidak terlepas dukungan dari berbagai pihak, oleh sebab itu penulis secara khusus mengucapkan banyak terima kasih kepada:

1. Tuhan Yang Maha Esa atas hikmat dan berkat-Nya sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan baik.
2. Ibu Chalifa Chazar, S.T., M.T. selaku dosen pembimbing sekaligus Kaprodi Teknik Informatika yang senantiasa mencurahkan segenap ilmu, waktu, dan tenaga untuk memberikan bimbingan, arahan, masukan yang sangat berharga baik dalam penulisan skripsi maupun dalam perkuliahan.
3. Adies Gunardi rekan yang membantu membimbing penulis mengenal sistem modul yang digunakan disaat penulis kesulitan dalam pembangunan perangkat keras.
4. Bapak & Ibu dosen yang telah memberikan penulis ilmu selama penulis melakukan kegiatan pembelajaran di STMIK-IM Bandung.
5. Kepala dan seluruh staff Administrasi, BAAK, BAUK dan staff Perpustakaan STMIK-IM Bandung.
6. Serta masih banyak lagi pihak-pihak yang sangat berpengaruh dalam proses penyelesaian skripsi yang yang tidak bisa penulis sebutkan satupersatu.

DAFTAR ISI

SURAT PERNYATAAN.....	i
ABSTRAK	ii
<i>ABSTRACT</i>	iii
KATA PENGANTAR	iv
UCAPAN TERIMA KASIH.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI	7
2.1 <i>Lock dan Key</i>	7
2.2 <i>Internet of Things (IoT)</i>	8
2.3 <i>Smart Lock</i>	9
2.4 <i>Radio Frequency Identification (RFID)</i>	10
2.4.1 <i>Arsitektur Radio Frequency Identification (RFID)</i>	10
2.4.2 <i>Klasifikasi Sistem Radio Frequency Identification (RFID)</i>	12
2.4.2.1 <i>Frekuensi Operasi Sistem</i>	12
2.4.2.2 <i>Klasifikasi Sumber Daya Tag</i>	15
2.5 Servo Motor.....	16
2.5.1 <i>Mekanisme Servo Motor</i>	17
2.5.2 <i>Micro Servo 9g SG90</i>	18

2.6	<i>Embedded System</i>	19
2.6.1	Kategori <i>Embedded System</i>	19
2.6.2	Kelebihan dan Kekurangan <i>Embedded System</i>	21
2.7	Mikrokontroler	22
2.7.1	Jenis-Jenis Mikrokontroler	25
2.7.2	Pemanfaatan Mikrokontroler	28
2.8	Hubungan <i>Embedded System</i> dengan Mikrokontroler	29
2.9	<i>Node Micro Controller Unit</i> ESP8266.....	29
2.9.1	Kelebihan NodeMCU ESP8266	31
2.9.2	<i>Hardware</i> NodeMCU ESP8266 (<i>Amica</i>).....	32
2.10	Arduino IDE	36
2.11	Blynk	38
2.12	<i>Use Case Diagram</i>	40
2.13	<i>Sequence Diagram</i>	42
2.14	<i>Activity Diagram</i>	43
2.15	<i>Fishbone Diagram</i>	44
BAB III ANALISIS DAN PERANCANGAN		46
3.1	Studi Literatur.....	46
3.2	Analisis Masalah	47
3.3	Skema Analisis	49
3.4	Analisis Sistem Yang Diusulkan.....	51
3.5	Perancangan Sistem.....	52
3.5.1	Desain Sistem.....	53
3.5.2	Desain Alat.....	53
3.5.3	Perancangan Alur Kerja Sistem	54
3.5.4	<i>Use Case Diagram</i>	54
3.6	Analisis Skenario.....	56
3.6.1	Skenario <i>Use Case</i> Menambah Registrasi	56
3.6.2	Skenario <i>Use Case</i> Menghapus Registrasi.....	57
3.6.3	Skenario <i>Use Case</i> Melakukan <i>Scan Tag</i>	57
3.6.4	Skenario <i>Use Case</i> Mendapat Notifikasi	57
3.7	<i>Sequence Diagram</i> <i>Smart Lock</i>	58

BAB IV IMPLEMENTASI DAN PENGUJIAN	60
4.1 Implementasi	60
4.1.1 Antarmuka Perangkat Keras.....	61
4.2 Metode Pengujian.....	62
4.3 Pengujian	63
4.3.1 Tahapan Rangkaian dan Pengujian Perangkat Keras.....	63
4.3.2 Pengujian NodeMCU ESP-8266 dengan RFID dan Servo.....	65
4.3.3 Analisis Pengujian Perangkat Keras	66
4.3.4 Pengujian Fungsi.....	70
BAB V KESIMPULAN DAN SARAN	72
5.1 Kesimpulan.....	72
5.2 Saran.....	72
DAFTAR PUSTAKA	73

DAFTAR TABEL

Tabel 2.1 Perbandingan LF, HF dan UHF (Fennani et al., 2011).....	15
Tabel 2.2 <i>Use Case Diagram</i> (Hendini, 2016)	41
Tabel 2.3 <i>Sequence Diagram</i> (Hendini, 2016).....	42
Tabel 2.4 <i>Activity Diagram</i> (Hendini, 2016)	43
Tabel 3.1 Referensi Jurnal terkait RFID, <i>embedded system</i> , <i>smart lock</i> , mikrokontroler.....	46
Tabel 3.2 Langkah Analisis	50
Tabel 3.3 Analisis Masalah Terhadap <i>Smart Lock</i>	51
Tabel 3.4 Deskripsi <i>use case</i>	55
Tabel 3.5 Deskripsi aktor pada <i>use case</i>	56
Tabel 4.1 Pengujian Perangkat Keras	66
Tabel 4.2 Pengujian Fungsi.....	70

DAFTAR GAMBAR

Gambar 1.1 Metode <i>Waterfall</i>	4
Gambar 2.1 Arsitektur RFID.....	11
Gambar 2.2 Alokasi frekuensi RFID	13
Gambar 2.3 Micro Servo 9g SG90.....	18
Gambar 2.4 Dimensi dan spesifikasi Micro Servo 9g SG90	18
Gambar 2.5 Diagram Blok Umum Mikrokontroler	23
Gambar 2.6 Mikrokontroler Intel 8051	25
Gambar 2.7 Mikrokontroler PIC	25
Gambar 2.8 Mikrokontroler ARM	26
Gambar 2.9 ATmega328P.....	27
Gambar 2.10 NodeMCU ESP8255	30
Gambar 2.11 <i>Board</i> NodeMCU ESP-8266 (Amica).....	35
Gambar 2.12 <i>Software</i> Arduino IDE.....	37
Gambar 2.13 Sistem Blynk	39
Gambar 3.1 <i>Fishbone</i> diagram.....	48
Gambar 3.2 Skema analisis	49
Gambar 3.3 <i>Activity diagram</i>	52
Gambar 3.4 Desain Sistem.....	53
Gambar 3.5 Desain Alat <i>Smart Lock</i>	54
Gambar 3.6 <i>Use Case</i> diagram.....	55
Gambar 3.7 <i>Sequence</i> Diagram Menambah Registrasi.....	58
Gambar 3.8 <i>Sequence</i> Diagram Menghapus Registrasi	59
Gambar 3.9 <i>Sequence</i> Diagram <i>Scan Tag</i> ID dan Notifikasi.....	59

Gambar 4.1 Implementasi Perangkat Keras.....	62
Gambar 4.2 Pengujian Program dan Status pada Serial Monitor.....	65
Gambar 4.3 Posisi Alat Saat Melakukan Pengujian.....	67
Gambar 4.4 Serial Monitor	67
Gambar 4.5 Servo Bergerak setelah UID diterima	68
Gambar 4.6 Notifikasi pada User	69
Gambar 4.7 Pencatatan Data Akses	70

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin tingginya permintaan akan teknologi dalam berbagai aspek kebutuhan manusia dimana kemajuan akan perangkat otomatis ikut berkembang. Salah satu dari kebutuhan teknologi tersebut ialah *smart home* yang merupakan bagian dari inovasi *Internet of Things*. Setiap penghuni rumah pastilah ingin memiliki rumah yang nyaman dan aman bagi dirinya dan keluarganya. Di Indonesia sendiri keamanan rumah masih bersifat konvensional, pagar berduri yang dibangun tinggi, gerbang dengan teralis besi, memasang *CCTV* bahkan mempekerjakan seorang *security*. Kunci pintu konvensional membutuhkan anak kunci sebagai akses masuk ke rumah, dimana anak kunci sendiri sangatlah mudah untuk digandakan, pembobolan dengan jenis kunci seperti pun sangat rawan menjadikan kunci pintu ini memiliki keamanan yang rendah.

Maka dari itu dibutuhkan sistem kunci pintu yang lebih efisien, praktis dan jauh lebih aman, sistem kunci pintar dengan metode diantaranya menggunakan *Radio Frequency Identification* (RFID) sebagai tanda pengenalan, teknik RFID memfasilitasi penggunaan identifikasi nirkabel otomatis melalui *tag* pasif dan aktif elektronik dengan *readers* yang sesuai (Hameed *et al.*, 2015).

Board utama menggunakan NodeMCU, *board* elektronik yang berbasis *chip* ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (WiFi) sebagai pusat dimana sensor-sensor yang digunakan akan berkomunikasi, *board* ini pun dikhususkan untuk “*Connected to Internet*”, dimana sangat ideal untuk membangun proyek berbasis *Internet of Thing*.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya maka yang menjadi permasalahan dalam penelitian ini:

1. Bagaimana membuat sistem kunci pintu yang lebih aman dibandingkan kunci pintu konvensional?
2. Bagaimana menerapkan NodeMCU dan RFID agar mejadi suatu sistem keamanan pintu pintar.

1.3 Tujuan Penelitian

1. Membuat *embedded system* keamanan pintu dengan konsep *IoT*.
2. Mengetahui cara penerapan *NodeMCU* dan *RFID* agar menjadi suatu sistem keamanan pintu pintar.

1.4 Batasan Masalah

1. Menggunakan RFID *tag* dan *reader*
2. *Node Micro Controller Unit*

3. Kunci elektronik
4. Tidak membahas jika kunci elektronik tidak ada

1.5 Metodologi Penelitian

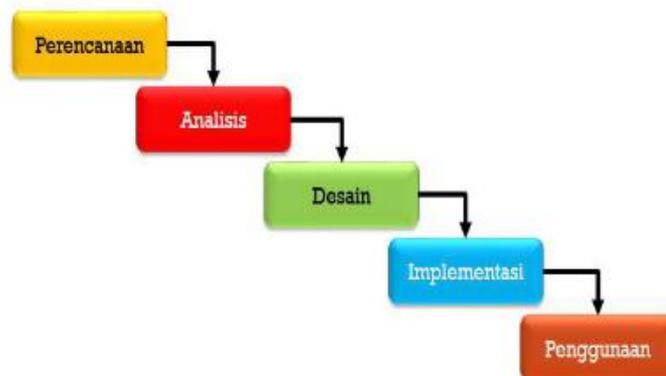
Metode penelitian yang dilakukan mencakup metode-metode yang digunakan penulis untuk memperoleh data baik dari narasumber dan sumber terkait lainnya, adapun metode yang digunakan yaitu:

1. Studi Literatur

Data-data yang dibutuhkan dalam penelitian dapat diperoleh dari sumber pustaka atau dokumen dari penelitian terdahulu ataupun dari sumber lainnya yang berhubungan dengan penelitian yang sedang dilakukan. Menurut Zed (2014), pada riset pustaka (*library research*), penelusuran pustaka tidak hanya untuk langkah awal menyiapkan kerangka penelitian (*research design*) akan tetapi sekaligus memanfaatkan sumber-sumber perpustakaan untuk memperoleh data penelitian (Melfianora, 2019).

2. Metode yang digunakan pada pengembangan sistem menggunakan model *waterfall*.

Metode WATERFAL



Gambar 1.1 Metode *Waterfall* (Maulani *et al.*, 2020)

Setiap tahap akan mempengaruhi dan menjadi dasar dari keberlangsungan tahap pengembangan berikutnya, sama seperti air terjun yang mengalir dari atas kebawah. Tahap baru akan dilakukan setelah tahap sebelumnya rampung dan kesalahan tahap sebelumnya akan berdampak kuat pada tahap berikutnya.

Aktivitas-aktivitas dalam *waterfall* model adalah sebagai berikut:

1. Proses perencanaan merupakan tahapan awal dari proses pembangunan penelitian ini, yaitu apa saja yang akan dibutuhkan dimulai dari kebutuhan perangkat lunak, perangkat keras dan siapa saja yang akan terlibat.
2. Dalam tahapan analisis bertujuan mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau

informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

3. Tahapan desain merupakan perancangan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.
4. Setelah 3 tahap sebelumnya dilakukan maka dilakukan tahap implementasi yaitu direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.
5. Tahap penggunaan merupakan tahapan terakhir dari metode pengembangan dari penelitian ini, dimana seluruh sistem sudah dapat digunakan oleh pihak-pihak yang terlibat.

1.6 Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan tugas akhir ini antara lain:

BAB 1. PENDAHULUAN

Bab ini berisi penjelasan umum mengenai usulan penelitian yang dilakukan dalam pengerjaan tugas akhir. Di dalamnya berisi latar belakang masalah, identifikasi masalah, tujuan penelitian, lingkup penelitian, metodologi pengerjaan, dan sistematika penulisan.

BAB 2. LANDASAN TEORI

Bab ini berisi definisi-definisi, teori-teori, serta konsep-konsep dasar yang diperlukan untuk menganalisa situasi yang diteliti. Di dalam bab ini dikemukakan hasil-hasil penelitian yang termaktub di buku-buku teks ataupun makalah-makalah di jurnal-jurnal ilmiah yang terkait yang relevan sebagai referensi pengerjaan penelitian ini.

BAB 3. ANALISIS DAN PERANCANGAN

Bab ini berisi mengenai analisis kebutuhan alat dan perancangan alat berdasarkan kebutuhan yang telah dipaparkan.

BAB 4. IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan mengenai pembuatan kode program *Embedded system* dan pengujian *Embedded system* yang telah dibangun.

BAB 5. KESIMPULAN DAN SARAN

Pada bagian ini akan dijelaskan mengenai kesimpulan dan saran dari semua tahapan yang dilewati dalam pembangunan *Embedded System*.

BAB II

LANDASAN TEORI

2.1 *Lock dan Key*

Lock adalah perangkat pengikat mekanis atau elektronik yang dilepaskan oleh objek fisik (kunci, kartu kunci, sidik jari, kartu RFID, token keamanan, koin, dll.), Dengan memberikan informasi rahasia (seperti permutasi angka atau huruf atau kata sandi), atau dengan kombinasinya atau hanya dapat dibuka dari satu sisi.

Key adalah perangkat yang digunakan untuk mengoperasikan *lock* (seperti untuk mengunci atau membukanya). Kunci tipikal adalah sepotong kecil logam yang terdiri dari dua bagian: bit atau bilah, yang meluncur ke jalan pasak kunci dan membedakan antara kunci yang berbeda, dan busur, yang dibiarkan menonjol sehingga torsi dapat diterapkan oleh pengguna. Dalam implementasinya yang paling sederhana, sebuah kunci mengoperasikan satu *lock* atau sekumpulan *lock* yang memiliki kunci yang sama. *Key* berfungsi sebagai token keamanan untuk akses ke area terkunci sedangkan *lock* dimaksudkan hanya untuk memungkinkan orang yang memiliki kunci yang benar untuk membukanya dan mendapatkan akses.

2.2 *Internet of Things (IoT)*

Internet of Things (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Pada dasarnya IoT (*Internet of Things*) mengacu pada benda yang dapat diidentifikasi secara unik sebagai *representative virtual* dalam struktur berbasis internet (Hidayati *et al.*, 2018).

Cara Kerja IoT (*Internet of Things*) adalah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan user dan dalam jarak berapa pun. Agar tercapainya cara kerja IoT (*Internet of Things*) tersebut diatas internet menjadi penghubung di antara kedua interaksi mesin tersebut, sementara user hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Manfaat yang didapatkan dari konsep IoT (*Internet of Things*) ialah pekerjaan yang dilakukan bisa menjadi lebih cepat, mudah dan efisien. Sistem dasar dari IoT terdiri dari 3 hal yaitu (Hidayati *et al.*, 2018):

- a. *Hardware/fisik (Things)*
- b. Koneksi Internet
- c. *Cloud Data Center*, tempat untuk menyimpan atau menjalankan aplikasinya.

Sejak mulai dikenalnya internet pada tahun 1989, mulai banyak hal kegiatan melalui internet, pada tahun 1990 John Romkey menciptakan perangkat pemanggang roti yang bisa dinyalakan dan dimatikan melalui Internet. WearCam diciptakan pada tahun 1994 oleh Steve Mann. Pada tahun 1997 Paul Saffo

memberikan penjelasan singkat pertama tentang sensor dan masa depan. Tahun 1999 Kevin Ashton menciptakan *The Internet of Things*, direktur eksekutif Auto ID Centre, MIT. Mereka juga menemukan peralatan berbasis RFID (*Radio Frequency Identification*) global yang sistem identifikasi pada tahun yang sama. Penemuan ini disebut sebagai sebuah lompatan besar dalam *commercializing IoT*. Tahun 2000 LG mengumumkan rencananya menciptakan kulkas pintar yang akan menentukan sendiri apakah bisa atau tidak makanan yang tersimpan di dalamnya diisi ulang. Pada tahun 2003 RFID mulai ditempatkan pada tingkat besar-besaran di militer AS di Program Savi mereka. Pada tahun yang sama melihat raksasa ritel Walmart untuk menyebarkan RFID di semua toko-toko di seluruh dunia untuk lebih besar. Pada tahun 2005 arus publikasi utama seperti *The Guardian*, *Amerika* ilmiah dan *Boston Globe* mengutip banyak artikel tentang IoT. Pada tahun 2008 kelompok perusahaan meluncurkan IPSO Alliance untuk mempromosikan penggunaan *Internet Protocol (IP)* dalam jaringan dari "*Smart object*" dan untuk mengaktifkan *Internet of Things*.

2.3 Smart Lock

Smart Lock adalah kunci elektromekanis yang dirancang untuk melakukan operasi penguncian dan pembukaan kunci pada pintu ketika menerima instruksi tersebut dari perangkat resmi menggunakan protokol nirkabel dan kunci kriptografi untuk menjalankan proses otorisasi. Memonitor akses dan mengirimkan peringatan untuk berbagai peristiwa yang dipantau dan beberapa peristiwa penting lainnya

yang terkait dengan status perangkat. Kunci pintar dapat dianggap sebagai bagian dari *smart home*.

Dalam kunci elektronik, kunci yang digunakan bukanlah kunci fisik tetapi *smartphone* atau kunci khusus yang dikonfigurasi secara eksplisit untuk tujuan ini yang secara nirkabel melakukan otentikasi yang diperlukan untuk membuka kunci pintu secara otomatis.

2.4 Radio Frequency Identification (RFID)

Teknologi RFID mampu mengubah aplikasi manajemen, penilaian, pengendalian inventaris dan *database* pencatatan. Identifikasi otomatis dan kontrol proses adalah kunci untuk meningkatkan efisiensi. Menggunakan teknologi RFID dapat mencapai bagian dari identifikasi otomatis, tetapi bagaimana mengintegrasikan RFID dengan sistem adalah bagian yang tidak dapat dijelaskan. Untuk mempersiapkan perubahan RFID, kita harus memperluas ide-ide untuk menggabungkan sistem kontrol proses baru yang terintegrasi dengan teknologi RFID (Hameed *et al.*, 2015).

2.4.1 Arsitektur Radio Frequency Identification (RFID)

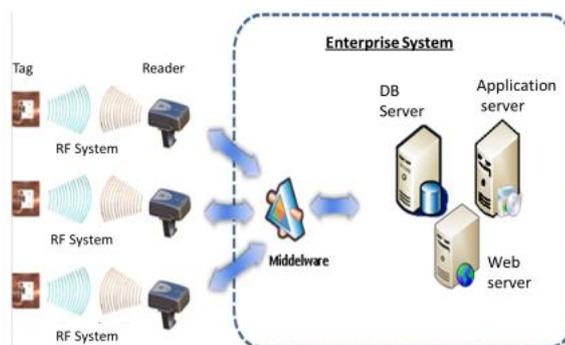
Tujuan utama RFID adalah untuk mengidentifikasi suatu objek secara unik. Yang dilengkapi dengan *chip* yang dikenal sebagai tag RFID dan berisi antena dan sirkuit terintegrasi yang berisi memori yang digunakan untuk menyimpan

informasi. Untuk mengatur sistem RFID, diperlukan tiga komponen utama yaitu (Fennani, Hamam and Dahmane, 2011):

Tag: adalah *chip* dengan antena dan memori yang akan ditempelkan atau disuntikkan ke dalam suatu objek (item, hewan). Setiap *tag* memiliki kode pengenalnya sendiri yang akan digunakan untuk mengidentifikasi objek yang terkait dengannya secara unik.

Reader: adalah peralatan yang mengumpulkan informasi dari *tag*, memfilternya, dan mentransfernya ke unit pemrosesan.

Server / host: adalah bagian pemrosesan dari sistem RFID. *Host* adalah komputer yang berjalan, selain aplikasi perusahaan, aplikasi yang disebut *middleware* yang merupakan antarmuka antara sistem RF (*Radio Frequency*) RFID (terutama *reader*) dan lapisan aplikasi. *Host* bisa sesederhana satu unit pemrosesan atau seluruh sistem perusahaan yang terdiri dari server (*web, database*).



Gambar 2.1 Arsitektur RFID (Fennani, Hamam and Dahmane, 2011)

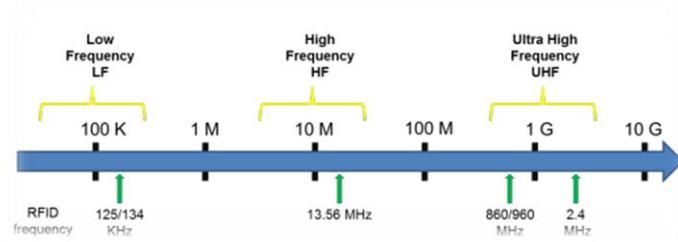
Prinsip operasi sistem RFID sederhana. *Reader* memulai proses komunikasi dengan memancarkan gelombang elektromagnetik, gelombang ini akan dicegat oleh antena *tag* RFID. Arus yang diinduksi akan dibuat di *tag* dan akan mengaktifkan sirkuit terintegrasi, memungkinkannya untuk berkomunikasi. Untuk mengirimkan informasi yang disimpan dalam *chip*-nya, *tag* memodulasi sinyal *digital* menggunakan modulasi fase atau amplitudo pada pembawa frekuensi yang telah ditentukan sebelumnya. *Reader* mengubah informasi yang diterima dalam kode biner dan mengarahkannya ke *host* tempat akan diproses. Gambar 2.1 menjelaskan prinsip komunikasi antara berbagai komponen sistem RFID (Fennani, Hamam and Dahmane, 2011).

2.4.2 Klasifikasi Sistem *Radio Frequency Identification (RFID)*

Sistem RFID dapat diklasifikasikan berdasarkan dua kriteria yaitu (Fennani, Hamam and Dahmane, 2011):

2.4.2.1 Frekuensi Operasi Sistem

Karakteristik gelombang elektromagnetik berubah dari satu pita ke pita lainnya. Oleh karena itu, sistem RFID mungkin berperilaku berbeda bergantung pada frekuensi pengoperasian sehingga erat kaitannya aplikasi teknologi RFID dengan pilihan *carrier* yang ditunjukkan pada Gambar 2.2



Gambar 2.2 Alokasi frekuensi RFID (Fennani, Hamam and Dahmane, 2011)

A. RFID Frekuensi Rendah (LF)

Pada dasarnya pembawa yang paling umum digunakan oleh sistem RFID frekuensi rendah adalah 125 KHz dan 134,2 KHz. Salah satu karakteristik utama dari jenis sistem RFID ini adalah sinyalnya tidak terlalu terpengaruh oleh logam sehingga ideal untuk identifikasi objek logam seperti mobil, peralatan, perkakas, dan wadah logam. Kisaran pembacaan bervariasi dari beberapa sentimeter hingga sekitar 2 meter dan pada dasarnya bergantung pada antena *tag*. Gelombang elektromagnetik di LF dapat dipengaruhi oleh gangguan listrik dari mesin di lingkungan industri. Kecepatan transfer data relatif rendah (sekitar 70ms untuk perintah *read*) sehingga membatasi dukungan untuk membaca beberapa transponder secara bersamaan (Fennani, Hamam and Dahmane, 2011).

B. RFID Frekuensi Tinggi (HF)

RFID *Passive* HF (*High Frequency*) beroperasi pada frekuensi 13,56 MHz yang diterima dunia. Ini berarti bahwa sistem RFID HF akan beroperasi di seluruh dunia. Namun, beberapa karakteristik sinyal lainnya, terutama daya dan *bandwidth*

sudah dinormalisasi menurut wilayah. Pada frekuensi tinggi, sinyal menembus dengan cukup baik seperti kebanyakan material termasuk air dan jaringan tubuh. Namun, lebih dipengaruhi oleh logam yang diisyaratkan oleh LF. Dibandingkan dengan LF, HF memungkinkan label menurunkan biaya, membaca beberapa label sekaligus dan memiliki kecepatan komunikasi yang lebih baik. Ukuran antena tergantung pada panjang gelombang sinyal yang berbanding terbalik dengan frekuensi (semakin tinggi frekuensi semakin pendek panjang gelombang) (Fennani, Hamam and Dahmane, 2011).

C. RFID Frekuensi Sangat Tinggi (UHF)

UHF memiliki frekuensi antara 300MHz dan 3GHz. RFID UHF menggunakan 433 MHz, 2,45 GHz dan terutama pada pita 860 - 956 MHz. Penggunaannya lebih baru dibandingkan dengan HF dan LF. UHF menggunakan antena yang lebih kecil dengan komunikasi jarak jauh dengan kecepatan yang lebih baik. Untuk mengirim sinyal, *tag* UHF merefleksikan sinyal yang masuk dari *reader* setelah memodulasi, teknik ini disebut *backscatter* dan diadopsi sebagai pengganti induksi. Sistem UHF RFID jauh lebih dipengaruhi oleh air dan logam sehubungan dengan LF dan HF. RFID UHF juga memungkinkan pengidentifikasian jumlah *tag* yang lebih besar secara bersamaan (hingga 1000 *tag* pada saat yang sama) berkat protokol penyiaran bitnya yang digunakan untuk tujuan anti-collision (Fennani, Hamam and Dahmane, 2011).

Tabel 2.1 Perbandingan LF, HF dan UHF (Fennani, Hamam and Dahmane, 2011)

<i>Type</i>	<i>LF</i>	<i>HF</i>	<i>UHF</i>
<i>Range: (Passive)</i>	<i>Few Centimetres</i>	<i>50 cm</i>	<i>6 m</i>
<i>Advantages</i>	<ul style="list-style-type: none"> - <i>Non affected by water</i> - <i>Non affected by metals</i> - <i>Frequency use without restriction</i> 	<ul style="list-style-type: none"> - <i>Non affected by water</i> - <i>Non affected by metals</i> - <i>Multiple tag read</i> - <i>Non affected by electrical noise</i> 	<ul style="list-style-type: none"> - <i>Long range</i> - <i>Standard</i> - <i>Easy to product with lowcost (5 cents)</i>
<i>Drawbacks</i>	<ul style="list-style-type: none"> - <i>Non affected by water</i> - <i>Noise</i> - <i>Low rate (70ms pour lire untag)</i> 	<ul style="list-style-type: none"> - <i>Range < 1m</i> - <i>Less efficient than LF (water dand metals)</i> 	<ul style="list-style-type: none"> - <i>Absorbed by water</i> - <i>Reflected by metals</i> - <i>Limited memory</i> - <i>Interference with many applications</i>
<i>Applications</i>	<ul style="list-style-type: none"> - <i>Animal tracking</i> - <i>Identification</i> 	<ul style="list-style-type: none"> - <i>Credit card, Access control card</i> - <i>Passports</i> 	<ul style="list-style-type: none"> - <i>Industry</i> - <i>Retail chain</i>

2.4.2.2 Klasifikasi Sumber Daya Tag

A. RFID Pasif

Dalam sistem RFID pasif, *tag* menggunakan energi sinyal elektromagnetik yang berasal dari *reader* untuk memberi daya pada sirkuit terintegrasi dan untuk mengirim respons kembali. Dikarenakan energi yang terbatas ini membatasi secara signifikan tidak hanya rentang pembacaan antara *reader* dan *tag*, tetapi juga kapasitas komputasi teknologi Informasi dan Komunikasi sirkuit terintegrasi.

B. RFID Aktif

Tag aktif menggunakan baterai untuk menyalakan sirkuit internal dan respons sinyal. Kapasitas perhitungannya jauh di atas kemampuan *tag* pasif dan jarak transmisi ditingkatkan dan dapat mencapai beberapa ratus meter tergantung pada frekuensi yang digunakan. *Tag* jenis ini dapat digabungkan dengan sensor dan

dapat memiliki fitur-fitur canggih yang terkadang *tag* dapat menyematkan sirkuit yang rumit dan menaikkan harga per unitnya.

C. RFID Semi Aktif

Sistem ini didasarkan pada *tag* semi-aktif yang menggunakan baterai untuk memberi daya pada sirkuit terintegrasi untuk mencapai pemrosesan khusus tetapi modul transmisi masih bekerja dengan cara pasif (yaitu menarik energi dari sinyal pembaca untuk modul transmisi dayanya). Namun waktu yang diperlukan untuk beralih dari keadaan tidak aktif ke aktif tidak dapat diterima dan terkadang memerlukan untuk menunggu. *Tag* jenis ini biasanya digabungkan dengan sensor.

2.5 Servo Motor

Motor servo adalah aktuator putar atau aktuator linier yang memungkinkan kontrol yang tepat dari posisi sudut atau linier, kecepatan, dan percepatan. Ini terdiri dari motor yang cocok digabungkan ke sensor untuk umpan balik posisi. Ini juga membutuhkan pengontrol yang relatif canggih, seringkali modul khusus yang dirancang khusus untuk digunakan dengan servo motor.

Servo motor digunakan dalam aplikasi seperti robotika, mesin CNC atau manufaktur otomatis. Kebanyakan motor servo modern dirancang dan dipasok di sekitar modul pengontrol khusus dari pabrikan yang sama. Pengontrol juga dapat dikembangkan di sekitar mikrokontroler untuk mengurangi biaya untuk aplikasi *volume* besar.

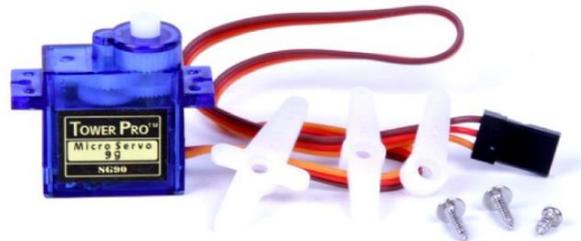
2.5.1 Mekanisme Servo Motor

Motor servo adalah mekanisme servo *loop* tertutup yang menggunakan umpan balik posisi untuk mengontrol gerakan dan posisi akhirnya. *Input* ke kontrolnya adalah sinyal (baik *analog* atau *digital*) yang mewakili posisi yang diperintahkan untuk poros *output* (Wikipedia, 2020).

Motor dipasangkan dengan beberapa jenis *encoder* posisi untuk memberikan umpan balik posisi dan kecepatan. Dalam kasus yang paling sederhana, hanya posisi yang diukur. Posisi *output* yang diukur dibandingkan dengan posisi perintah, *input* eksternal ke pengontrol. Jika posisi keluaran berbeda dari yang dibutuhkan, sinyal kesalahan dihasilkan yang kemudian menyebabkan motor berputar ke salah satu arah, sesuai kebutuhan untuk membawa poros keluaran ke posisi yang sesuai. Saat posisi mendekati, sinyal kesalahan berkurang menjadi nol dan motor berhenti.

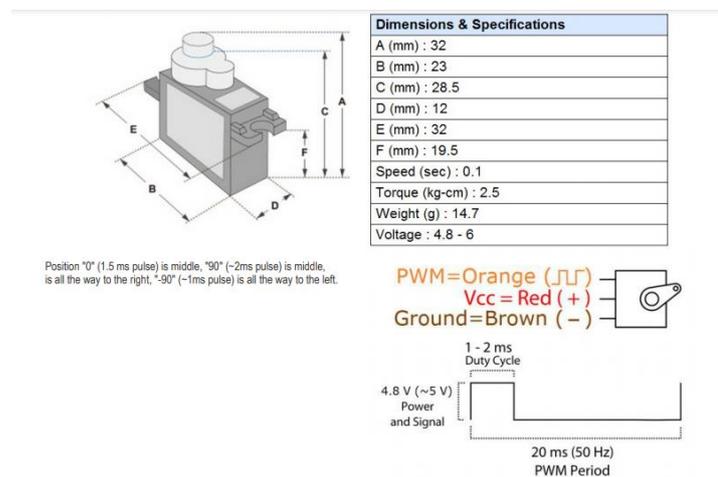
Modul penggerak untuk motor servo adalah komponen industri standar. Desainnya adalah cabang elektronika daya, biasanya didasarkan pada jembatan H MOSFET atau IGBT tiga fase. Modul standar ini menerima satu arah dan jumlah pulsa (jarak rotasi) sebagai *input*. Dimana bisa termasuk pemantauan suhu berlebih, torsi berlebih dan fitur deteksi *stall*. Karena jenis *encoder*, rasio *gearhead*, dan dinamika sistem secara keseluruhan adalah khusus aplikasi, lebih sulit untuk menghasilkan pengontrol keseluruhan sebagai modul *off-the-shelf* dan ini sering diimplementasikan sebagai bagian dari pengontrol utama.

2.5.2 Micro Servo 9g SG90



Gambar 2.3 *Micro Servo 9g SG90* (Cheung, 2018)

Servo ini kecil dan ringan dengan daya keluaran tinggi. Servo dapat berputar kira-kira 180 derajat (90 derajat di setiap arah), dan bekerja seperti jenis standar tetapi lebih kecil. Dapat menggunakan kode servo, perangkat keras atau pustaka untuk mengontrol servo ini. Servo jenis ini memiliki 3 *horns* (lengan) dan perangkat keras. (Cheung, 2018)



Gambar 2.4 Dimensi dan spesifikasi *Micro Servo 9g SG90* (Cheung, 2018)

2.6 *Embedded System*

Sistem tertanam dapat didefinisikan secara luas sebagai perangkat yang berisi perangkat keras dan komponen perangkat lunak yang digabungkan erat untuk melakukan fungsi tunggal, merupakan bagian dari sistem yang lebih besar, tidak dimaksudkan untuk dapat diprogram secara independen oleh pengguna, dan diharapkan bekerja dengan minimal atau tidak ada interaksi manusia. Dua karakteristik tambahan yang sangat umum dalam sistem tertanam: operasi reaktif dan sangat terbatas (Kadionik, 2013).

Kebanyakan sistem tertanam berinteraksi langsung dengan proses atau lingkungan, membuat keputusan dengan cepat, berdasarkan masukan mereka. Ini mengharuskan sistem harus reaktif, merespons secara *real-time* untuk memproses masukan guna memastikan pengoperasian yang benar. Selain itu, sistem ini beroperasi dalam lingkungan terbatas di mana memori, daya komputasi, dan catu daya terbatas. Selain itu, persyaratan produksi, dalam banyak kasus karena volume, menjadi kendala biaya tinggi pada desain. Ini adalah definisi luas yang menyoroti berbagai macam sistem yang termasuk di dalamnya (Kadionik, 2013).

2.6.1 *Kategori Embedded System*

Embedded System dapat diklasifikasikan berdasarkan fungsi dan performansinya yaitu sebagai berikut:

A. Mandiri (*Standalone*)

Perangkat *standalone* dapat berfungsi secara independen dari perangkat keras lainnya. Tidak terintegrasi ke dalam perangkat lain. Contoh : Kotak TiVo untuk merekam siaran televisi. Sedangkan DVR (*digital video recorder*) merupakan sistem *embedded* yang terintegrasi dengan DVD *player*. *Standalone* juga dapat merujuk pada program *software* yang tidak memerlukan *software* selain sistem operasi untuk menjalankannya.

B. Real Time

Sistem *embedded* dengan tugas-tugas spesifik yang dilakukan dalam periode waktu spesifik disebut dengan sistem *real time*. Sistem *real time* terdiri dari *hard real time system* dan *soft real time system*.

C. Hard real time

Hard real time adalah sistem yang harus melaksanakan tugas dengan *deadline* yang tepat. Contoh dari sistem *hard realtime* adalah sistem yang harus membuka katup dalam 30 milidetik ketika kelembaban udara melintasi ambang batas tertentu. Jika katup tidak dibuka dalam 30 milidetik maka akan menimbulkan masalah.

Sistem *hard real-time* sering kali digunakan sebagai alat pengontrol untuk aplikasi yang *dedicated*, mempunyai batasan waktu yang tetap yang sudah didefinisikan dengan baik. Pemrosesan harus selesai dalam batasan-batasan yang sudah didefinisikan, atau sistem akan gagal.

D. *Soft real time*

Soft real time adalah sistem yang tidak memerlukan *deadline*. Contoh dari *soft realtime* seperti DVD *player*, jika diberikan suatu perintah dari *remote control* maka akan mengalami *delay* selama beberapa milidetik untuk menjalankan perintah tersebut. *Delay* ini tidak akan berakibat sesuatu yang serius. Sistem *soft real-time* mempunyai lebih sedikit batasan waktu yang keras, dan tidak mendukung *deadline* dengan menggunakan batas akhir.

2.6.2 Kelebihan dan Kekurangan *Embedded System*

Terdapat beberapa kelebihan dan kekurangan dari *embedded system*, yaitu:

1. Kelebihan *embedded system*

- Mudah untuk produksi massal.
- Sangat dapat diandalkan.
- Meningkatkan kualitas produk.
- Menggunakan operasi sumber daya rendah.
- Beroperasi sangat cepat, dan portabel serta berukuran kecil.

2. Kekurangan *embedded system*

- Tidak ada ruang untuk peningkatan teknologikal.
- Sulit untuk dipelihara.
- Sulit untuk mengambil kembali file yang disematkan.
- Memiliki daya tahan catu daya yang lebih rendah jika dioperasikan dengan baterai.

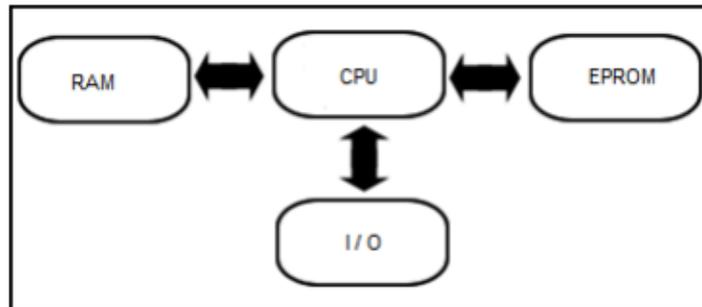
2.7 Mikrokontroler

Mikrokontroler adalah sebuah system komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip IC*, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan *system computer* yang mempunyai satu atau beberapa tugas yang sangat spesifik. Elemen mikrokontroler tersebut diantaranya adalah (Chamim, 2010):

- a. Pemroses (*processor*)
- b. Memori
- c. Input dan output

Kadangkala pada mikrokontroler ini beberapa *chip* digabungkan dalam satu papan rangkaian. Perangkat ini sangat ideal untuk mengerjakan sesuatu yang bersifat khusus, sehingga aplikasi yang diisikan ke dalam komputer ini adalah aplikasi yang bersifat *dedicated*. Jika dilihat dari harga, mikrokontroler ini harga umumnya lebih murah dibandingkan dengan komputer lainnya, karena perangkatnya relatif sederhana (Chamim, 2010).

Mikrokontroler mempunyai ruang alamat tersendiri yang disebut memori. Memori dalam mikrokontroler terdiri atas memori program dan memori data dimana keduanya terpisah, yang memungkinkan pengaksesan data memori dan pengalamatan 8 bit, sehingga dapat langsung disimpan dan dimanipulasi oleh mikrokontroler dengan kapasitas akses 8 bit. Program memori tersebut bersifat hanya dapat dibaca (ROM/EPROM). Sedangkan untuk data memori dapat menggunakan memori eksternal (RAM) (Chamim, 2010).



Gambar 2.5 Diagram Blok Umum Mikrokontroler (Güven *et al.*, 2017)

Pada Gambar 2.5 di atas tampak suatu mikrokontroler standar yang tersusun atas komponen-komponen sebagai berikut (Electronics hub, 2013):

1. *Central Processing Unit (CPU)*

CPU merupakan bagian utama dalam suatu mikrokontroler. CPU pada mikrokontroler ada yang berukuran 8 bit ada pula yang berukuran 16 bit. CPU ini akan membaca program yang tersimpan di dalam ROM dan melaksanakannya.

2. *Read Only Memory (ROM)*

ROM merupakan suatu memori (alat untuk mengingat) yang sifatnya hanya dibaca saja. Dengan demikian ROM tidak dapat ditulisi. Dalam dunia mikrokontroler ROM digunakan untuk menyimpan program bagi mikrokontroler tersebut. Program tersimpan dalam format biner ('0' atau '1'). Susunan bilangan biner tersebut bila telah terbaca oleh mikrokontroler akan memiliki arti tersendiri.

3. *Random Acces Memory (RAM)*

Berbeda dengan ROM, RAM adalah jenis memori selain dapat dibaca juga dapat ditulis berulang kali. Tentunya dalam pemakaian mikrokontroler ada semacam data yang bisa berubah pada saat mikrokontroler tersebut bekerja. Perubahan data tersebut tentunya juga akan tersimpan ke dalam memori. Isi pada RAM akan hilang jika catu daya listrik hilang.

4. *Input / Output (I/O)*

Untuk berkomunikasi dengan dunia luar, maka mikrokontroler menggunakan terminal I/O (port I/O), yang digunakan untuk masukan atau keluaran.

5. *Komponen lainnya*

Beberapa mikrokontroler memiliki *timer/counter*, ADC (*Analog to Digital Converter*), dan komponen lainnya. Pemilihan komponen tambahan yang sesuai dengan tugas mikrokontroler akan sangat membantu perancangan sehingga dapat mempertahankan ukuran yang kecil. Apabila komponen komponen tersebut belum ada pada suatu mikrokontroler, umumnya komponen tersebut masih dapat ditambahkan pada sistem mikrokontroler melalui port-portnya.

2.7.1 Jenis-Jenis Mikrokontroler

A. 8051 Mikrokontroler



Gambar 2.6 Mikrokontroler Intel 8051 (Khadse, Gawai and Faruk, 2014)

Mikrokontroler 8051 adalah mikrokontroler 8bit yang ditemukan pada tahun 1981 oleh Intel Corporation. Tersedia dalam 40 pin DIP yaitu paket *dual in line*. Merupakan mikrokontroler dasar tetapi masih banyak perusahaan yang memproduksi jenis mikrokontroler tersebut. Jenis 8051 yang lebih tua memiliki 12 *clock* per instruksi yang membuatnya lamban sedangkan 8051 baru-baru ini memiliki 6 *clock* per instruksi. Mikrokontroler 8051 tidak memiliki bus memori internal dan konverter A / D dan mikrokontroler tersebut adalah prosesor CISC, 8051 menggunakan arsitektur Von Neuman (Khadse, Gawai and Faruk, 2014).

B. Mikrokontroler PIC



Gambar 2.7 Mikrokontroler PIC (Khadse, Gawai and Faruk, 2014)

Peripheral interface controller (PIC) adalah keluarga mikrokontroler dengan teknologi *Microchip* USA dengan arsitektur Harvard. Awalnya ini dikembangkan sebagai perangkat pendukung untuk komputer PDP (pemroses data program) untuk mendukung perangkat periferan dan oleh karena itu dinamai PIC. Mikrokontroler PIC adalah prosesor RISC. Siklus mesin PIC hanya terdiri dari 4 pulsa clock berbeda dengan 12 pulsa clock pada mikrokontroler Intel 8051 (Khadse, Gawai and Faruk, 2014).

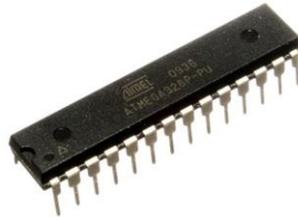
C. Mikrokontroler ARM



Gambar 2.8 Mikrokontroler ARM (Khadse, Gawai and Faruk, 2014)

ARM adalah mikrokontroler 32bit yang intinya dirancang oleh ARM Limited dengan arsitektur RISC. ARM memiliki arsitektur von Neumann (program dan RAM dalam ruang yang sama). Mikrokontroler ARM digunakan dalam penghematan daya dan beroperasi dalam konsumsi daya yang sangat rendah. Mikrokontroler ARM banyak digunakan di *handset modern* untuk komunikasi seluler. Ini juga digunakan di berbagai sistem tertanam lainnya seperti iPOD, unit permainan genggam, *driver disk* dan sebagainya. 8051 dan PIC membutuhkan banyak siklus *clock* per instruksi. AVR dan ARM menjalankan sebagian besar instruksi dalam satu siklus *clock* (Khadse, Gawai and Faruk, 2014).

D. Mikrokontroler AVR



Gambar 2.9 ATmega328P (Khadse, Gawai and Faruk, 2014)

AVR adalah mikrokontroler *chip* tunggal RISC 8-bit arsitektur Harvard RISC yang dimodifikasi, yang dikembangkan oleh Atmel pada tahun 1996. AVR adalah singkatan dari Alf-Egil Bogen dan prosesor RISC Vegard Wollan. AVR hanya membutuhkan satu jam per instruksi (Khadse, Gawai and Faruk, 2014).

Mikrokontroler AVR diklasifikasikan menjadi tiga jenis:

1. *TinyAVR*: lebih sedikit memori, ukuran kecil, hanya cocok untuk aplikasi yang lebih sederhana.
2. *MegaAVR*: yang paling populer yang memiliki jumlah memori yang baik (hingga 256 KB), jumlah periferil *inbuilt* lebih tinggi dan cocok untuk aplikasi sedang hingga kompleks.
3. *XmegaAVR*: digunakan secara komersial untuk aplikasi kompleks, yang membutuhkan memori program besar dan kecepatan tinggi.

2.7.2 Pemanfaatan Mikrokontroler

Mikrokontroler digunakan dalam produk dan perangkat yang dikontrol secara otomatis, seperti sistem kontrol mesin mobil, perangkat medis implan, *remote control*, mesin kantor, peralatan, perkakas listrik, mainan, dan sistem tertanam lainnya. Dengan mengurangi ukuran dan biaya dibandingkan dengan desain yang menggunakan mikroprosesor, memori, dan perangkat *input / output* terpisah, mikrokontroler membuatnya ekonomis untuk mengontrol lebih banyak perangkat dan proses secara *digital*. Mikrokontroler sinyal campuran umum digunakan, mengintegrasikan komponen analog yang diperlukan untuk mengontrol sistem elektronik *non-digital*. Dalam konteks *Internet of Things*, mikrokontroler adalah alat pengumpulan data yang ekonomis dan populer, merasakan dan menggerakkan dunia fisik sebagai perangkat tepi.

1. Penerapan mikrokontroler dalam perangkat *Day to Day Life*:

- Perangkat penginderaan dan pengontrol cahaya
- Perangkat penginderaan dan pengontrol suhu
- Deteksi kebakaran dan perangkat keamanan
- Perangkat instrumentasi industri
- Perangkat kontrol proses

2. Penerapan mikrokontroler pada alat kontrol industri:

- Perangkat instrumentasi industri
- Perangkat kontrol proses

3. Penerapan mikrokontroler dalam alat pengukur dan pengukuran:

- Pengukur Volt
- Mengukur benda berputar
- Pengukur arus
- Sistem pengukuran genggam

2.8 Hubungan *Embedded System* dengan Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil di dalam satu sirkuit yang berisi inti prosesor, memori, dan input/output peripheral yang dapat diprogram, dan di desain khusus untuk *embedded system*. Mikrokontroler berukuran kecil dan murah, sangat berbeda dengan apa yang ada di komputer. Perbedaan tersebut dikarenakan menyesuaikan kebutuhan, mikrokontroler ini didesain untuk suatu tujuan khusus, bukan untuk melakukan pekerjaan-pekerjaan rumit seperti yang ada di komputer sebenarnya.

Embedded system tidak membutuhkan kapasitas data dan *address bus* yang terlalu besar, itu sebabnya mikrokontroler ini berukuran lebih kecil, dan harga harus terjangkau karena akan dipasangkan ke barang-barang elektronik yang harganya juga terjangkau.

2.9 *Node Micro Controller Unit ESP8266*

NodeMCU adalah sebuah *board* elektronik yang berbasis chip ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet

(WiFi). Terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun *controlling* pada proyek IOT. NodeMCU ESP8266 dapat diprogram dengan *compiler*-nya Arduino, menggunakan Arduino IDE. Bentuk fisik dari NodeMCU ESP 8266, terdapat port USB (mini USB) sehingga akan memudahkan dalam pemrogramannya. NodeMCU ESP8266 merupakan modul turunan pengembangan dari modul *platform* IoT (*Internet of Things*) keluarga ESP8266 tipe ESP-12. Secara fungsi hampir menyerupai dengan *platform* arduino, tetapi yang membedakan yaitu dikhususkan untuk “*Connected to Internet*“. Untuk saat ini modul NodeMCU sudah terdapat 3 tipe versi antara lain (Hidayati *et al.*, 2018):

Versi NodeMCU ESP8266



Gambar 2.10 NodeMCU ESP8255

Adapun data teknis *board* NodeMCU ESP8266 adalah sebagai berikut (Make-It.ca, 2020):

- Mikrokontroler: Tensilica 32-bit RISC CPU Xtensa LX106

- Tegangan Operasi: 3.3V
- Tegangan *Input* (Rekomendasi): 7-12V
- Pin *digital I/O* (DIO): 16
- Pin *analog Input* (ADC): 1
- *Universal Asynchronous Receiver/Transmitter*: 1
- *Serial Peripheral Interface*: 1
- *Inter-Integrated Circuit*: 1
- *Flash Memory*: 4 MB
- SRAM: 64 KB
- *Clock Speed*: 80 MHz
- USB-TTL berdasarkan CP2102 disertakan di papan, mengaktifkan *Plug n Play*
- Antena PCB
- Modul berukuran kecil

2.9.1 Kelebihan NodeMCU ESP8266

NodeMCU ESP8266 sudah dilengkapi dengan modul WiFi ESP-01, dimana sebelumnya ESP-01 hanya berbentuk *chip* saja, namun pada NodeMCU disertakan pula USB UART yang memudahkan pengguna dalam memasukan program pada modul dan sudah dilengkapi dengan modul WiFi, sehingga cocok untuk proyek IoT.

Kelebihan NodeMCU diantaranya:

- Murah

NodeMCU relatif murah dibandingkan dengan *platform* mikrokontroler lainnya.

- Lintas *platform*

NodeMCU menggunakan *default* program bernama pemrograman Lua, namun tetap dapat menggunakan bahasa C dari Arduino menggunakan Arduino Software (IDE) yang berjalan pada sistem operasi Windows, Macintosh OSX, dan Linux. Kebanyakan sistem mikrokontroler terbatas pada Windows.

- Lingkungan pemrograman yang sederhana dan jelas

Perangkat lunak Arduino (IDE) mudah digunakan untuk pemula, namun cukup fleksibel untuk dimanfaatkan oleh pengguna tingkat lanjut.

- Sumber terbuka yang dapat diperluas

NodeMCU dirilis sebagai alat sumber terbuka, tersedia untuk ekstensi oleh pemrogram berpengalaman maupun tidak.

2.9.2 Hardware NodeMCU ESP8266 (Amica)

- Pin Daya

Terdiri dari 4 pin. Pin VIN dan 3 pin 3.3V.

- VIN dapat digunakan untuk langsung memasok NodeMCU / ESP8266 dan periferalnya. Daya yang dikirim pada VIN diatur

melalui regulator *onboard* pada modul NodeMCU. Dapat juga dapat memasok 5V yang diatur ke pin VIN.

- Pin 3.3V adalah *output* dari regulator tegangan *onboard* dan dapat digunakan untuk memasok daya ke komponen eksternal.

- GND

Merupakan pin *ground* NodeMCU / ESP8266

- Pin I2C

Digunakan untuk menghubungkan sensor dan periferan I2C. Baik I2C *Master* dan I2C *Slave* didukung. Fungsionalitas antarmuka I2C dapat direalisasikan secara terprogram dan frekuensi *clock* maksimumnya adalah 100 kHz. Perlu dicatat bahwa frekuensi *clock* I2C harus lebih tinggi dari frekuensi *clock* paling lambat dari perangkat *slave*.

- Pin GPIO

NodeMCU / ESP8266 memiliki 17 pin GPIO yang dapat digunakan untuk fungsi seperti I2C, I2S, UART, PWM, IR *Remote Control*, Lampu LED dan Tombol secara terprogram. Setiap GPIO yang mendukung digital dapat dikonfigurasi ke *pull-up* atau *pull-down internal*, atau disetel ke impedansi tinggi. Saat dikonfigurasi sebagai *input*, ini juga dapat diatur ke pemacu tepi atau pemacu level untuk menghasilkan interupsi CPU.

- ADC Channel

NodeMCU tertanam dengan SAR ADC presisi 10-bit. Kedua fungsi tersebut dapat diimplementasikan dengan menggunakan ADC. Pengujian

tegangan catu daya pin VDD3P3 dan pengujian tegangan *input* pin TOUT. Namun, tidak dapat diimplementasikan pada waktu yang bersamaan.

- Pin UART

NodeMCU / ESP8266 memiliki 2 antarmuka UART (UART0 dan UART1) yang menyediakan komunikasi asinkron (RS232 dan RS485) dan dapat berkomunikasi hingga 4,5 Mbps. UART0 (pin TXD0, RXD0, RST0 & CTS0) dapat digunakan untuk komunikasi. Namun, UART1 (pin TXD1) hanya menampilkan sinyal transmisi data sehingga biasanya digunakan untuk mencetak *log*.

- Pin SPI

NodeMCU / ESP8266 memiliki 2 SPI (SPI dan HSPI) dalam *mode slave* dan *master*. SPI ini juga mendukung fitur SPI dengan tujuan umum berikut:

- Mode waktu transfer format SPI
- Hingga 80 MHz dan *clock* terbagi 80 MHz
- Hingga 64-Byte FIFO

- Pin SDIO

NodeMCU / ESP8266 memiliki fitur *Secure Digital Input / Output Interface* (SDIO) yang digunakan untuk langsung menghubungkan kartu SD. 4-bit 25 MHz SDIO v1.1 dan 4-bit 50 MHz SDIO v2.0 didukung.

- Pin PWM

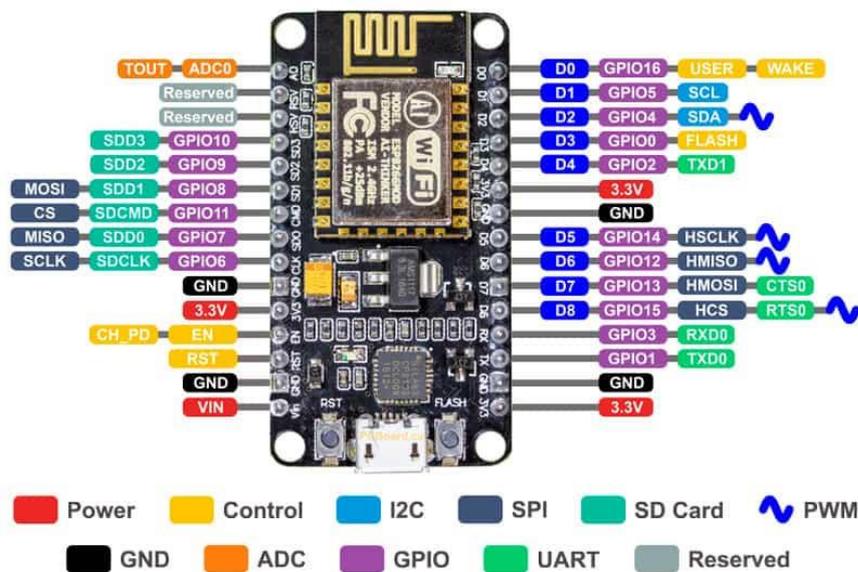
Memiliki 4 saluran *Pulse Width Modulation* (PWM). *Output* PWM dapat diimplementasikan secara terprogram dan digunakan untuk menggerakkan

motor digital dan LED. Rentang frekuensi PWM dapat disesuaikan dari 1000 μ s hingga 10000 μ s (100 Hz dan 1 kHz).

- Pin Kontrol & WAVE

Digunakan untuk mengontrol NodeMCU / ESP8266. Pin ini termasuk:

- EN: *Chip* ESP8266 diaktifkan saat pin EN ditarik TINGGI. Saat ditarik *LOW chip* bekerja dengan daya minimum.
- RST: Pin RST digunakan untuk mereset *chip* ESP8266.
- WAKE: Pin WAKE digunakan untuk membangunkan *chip* dari tidur.



Gambar 2.11 Board NodeMCU ESP-8266 (Amica)

2.10 Arduino IDE

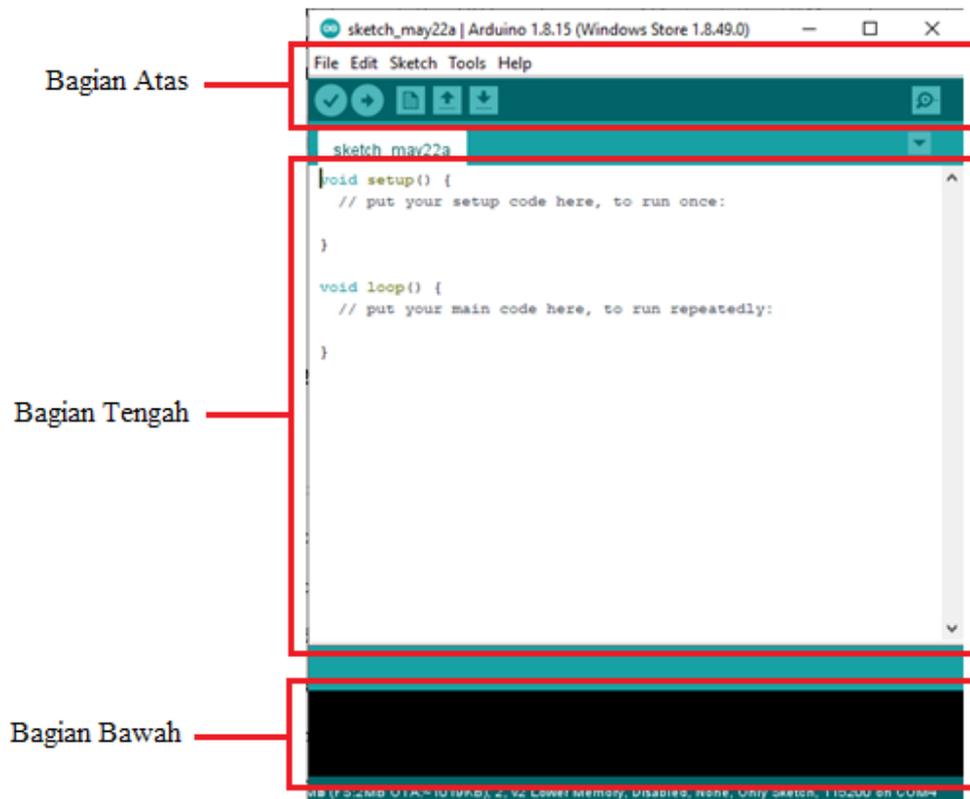
Arduino *integrated development environment* (IDE) merupakan aplikasi lintas platform yang ditulis dalam Java. Berasal dari IDE untuk proyek bahasa pemrograman, pemrosesan dan proyek *Wiring*. Dirancang untuk memperkenalkan pemrograman kepada pengguna dan pendatang baru lainnya yang tidak terbiasa dengan pengembangan perangkat lunak. Ini termasuk editor kode dengan fitur-fitur seperti penyorotan sintaks, pencocokan *brace*, dan indentasi otomatis, dan menyediakan mekanisme satu-klik sederhana untuk menyusun dan memuat program ke papan modul yang sedang dikerjakan. Program yang ditulis dengan IDE disebut "*sketch*" (Zlatanov, 2016).

Arduino IDE mendukung bahasa pemrograman C dan C ++ menggunakan aturan khusus organisasi kode. Arduino IDE menyediakan pustaka perangkat lunak yang disebut "*Wiring*" dari proyek *Wiring*, yang menyediakan banyak prosedur input dan output umum. Sketsa Arduino C / C ++ tipikal terdiri dari dua fungsi yang dikompilasi dan ditautkan dengan program *stub main ()* menjadi program eksekutif siklik yang dapat dieksekusi (Zlatanov, 2016):

- *setup ()*: fungsi yang berjalan satu kali pada awal program dan dapat menginisialisasi pengaturan.
- *loop ()*: fungsi yang dipanggil berulang kali hingga *board* mati.

Setelah kompilasi dan ditautkan dengan GNU *toolchain*, juga disertakan dengan distribusi IDE, Arduino IDE menggunakan program *avrdude* untuk mengubah kode yang dapat dieksekusi menjadi file teks dalam pengkodean

heksadesimal yang dimuat ke papan oleh program pemuat di *firmware* papan (Zlatanov, 2016).



Gambar 2.12 *Software* Arduino IDE

Berikut ini adalah jendela utama IDE Arduino, terdiri dari tiga bagian yaitu:

1. Bagian atas yakni *Toolbar*, pada bagian atas terdapat menu *file*, *edit*, *sketch*, *tools*, dan *help*.
2. Bagian tengah, yaitu tempat penulisan kode program atau *sketch*.
3. Bagian bawah berupa jendela pesan (*message windows*) atau tes konsol yang berisi status dan pesan *error*.

IDE arduino terdiri dari:

1. *Editor Program*

Sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

2. *Compiler*

Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

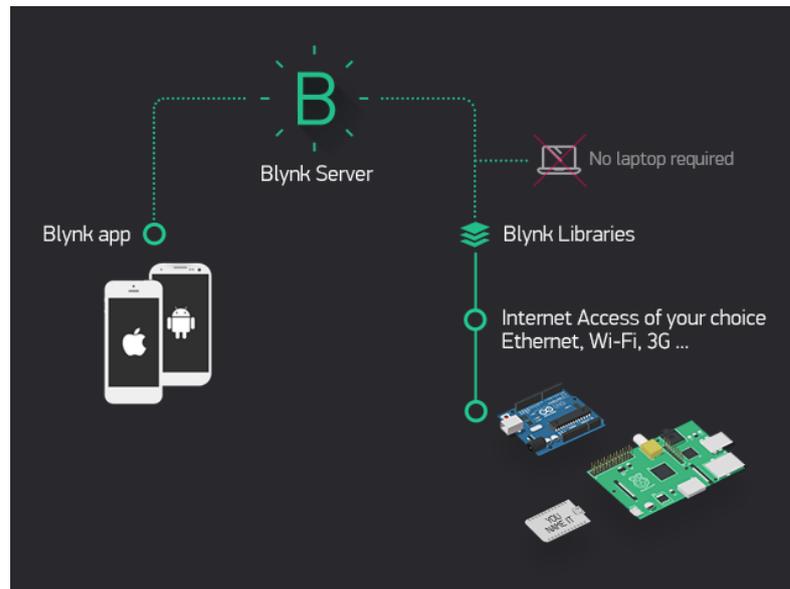
3. *Uploader*

Sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan mikrokontroler.

2.11 Blynk

Blynk bekerja melalui Internet. Ini berarti perangkat keras harus dapat terhubung ke internet. Beberapa *board*, seperti Arduino Uno akan membutuhkan Ethernet atau Wi-Fi *Shield* untuk berkomunikasi, yang lain sudah mendukung Internet: seperti ESP8266, Raspberri Pi dengan dongle WiFi, *Particle Photon* atau SparkFun Blynk *Board* (Blynk, 2021).

Blynk dirancang untuk *Internet of Things*. Dimana dapat mengontrol perangkat keras dari jarak jauh, menampilkan data sensor, menyimpan data, melihatnya secara visual dan hal lainnya (Blynk, 2021).



Gambar 2.13 Sistem Blynk (Blynk, 2021)

Ada tiga komponen utama dalam *platform* Blynk (Blynk, 2021):

- Aplikasi Blynk: Memungkinkan membuat antarmuka untuk proyek menggunakan berbagai *widget* yang disediakan.
- Blynk *Server*: Bertanggung jawab atas semua komunikasi antara smartphone dan perangkat keras. Dapat juga menggunakan Blynk Cloud atau menjalankan server Blynk pribadi secara lokal. *Open-source*, dimana dapat dengan mudah menangani ribuan perangkat dan bahkan dapat digunakan pada Raspberry Pi.
- Blynk *Libraries*: Untuk semua *platform* perangkat keras populer, memungkinkan komunikasi dengan server dan memproses semua perintah yang masuk dan keluar.

Fitur pada Blynk (Blynk, 2021):

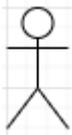
- API & UI untuk semua perangkat keras & perangkat yang didukung
- Koneksi ke *cloud* menggunakan:
 - Wifi
 - Bluetooth dan BLE
 - Ethernet
 - USB (Serial)
 - GSM
- *Set Widget* yang mudah digunakan
- Manipulasi pin langsung tanpa penulisan kode
- Mudah diintegrasikan dan menambahkan fungsionalitas baru menggunakan pin virtual
- Pemantauan data riwayat melalui *widget SuperChart*
- Komunikasi perangkat-ke-perangkat menggunakan *Bridge Widget*
- Mengirim *email*, *tweet*, pemberitahuan *push*, dll.

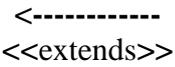
2.12 Use Case Diagram

Diagram *use case* merupakan gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya. *Use case* adalah teknik penemuan kebutuhan perangkat lunak yang dikenalkan pertama kali dalam metode pendekatan berbasis objek yang dikembangkan oleh Jacobson dan kawan-kawan pada tahun 1990an. Saat ini, diagram *use case* menjadi salah satu tipe dari Diagram *Unified*

Modeling Language (UML) berbasis tingkah laku. Menurut Alistair Cockburn, *use case* mendeskripsikan tentang tingkah laku sistem di berbagai kondisi dan bagaimana sistem tersebut menanggapi permintaan pengguna. *Use case* didefinisikan dari sudut pandang aktor yang terlibat. Aktor merupakan orang atau alat yang menggunakan sistem tersebut atau bisa juga sebagai apapun yang berinteraksi dengan sistem.

Tabel 2.2 *Use Case Diagram* (Hendini, 2016)

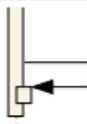
Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p><i>Actor</i> atau Aktor adalah Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor 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 aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
<p>----- <<include>></p>	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>

	Extend, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
---	---

2.13 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* yaitu: (Hendini, 2016)

Tabel 2.3 *Sequence Diagram* (Hendini, 2016)

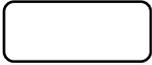
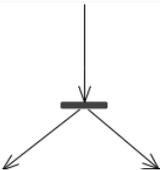
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

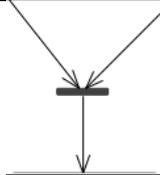
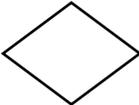
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

2.14 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu (Hendini, 2016):

Tabel 2.4 *Activity Diagram* (Hendini, 2016)

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir aktivitas.
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p>

2.15 *Fishbone Diagram*

Diagram tulang ikan atau *fishbone diagram* adalah salah satu metode / *tool* di dalam meningkatkan kualitas. Sering juga diagram ini disebut dengan diagram Sebab-Akibat atau *cause effect diagram*. Penemunya adalah seorang ilmuwan jepang pada tahun 60-an. Bernama Dr. Kaoru Ishikawa, ilmuwan kelahiran 1915 di Tikyo Jepang yang juga alumni teknik kimia Universitas Tokyo sehingga sering juga disebut dengan diagram ishikawa. Metode tersebut awalnya lebih banyak digunakan untuk manajemen kualitas yang menggunakan data verbal (*non-numerical*) atau data kualitatif. Dr. Ishikawa juga ditengarai sebagai orang pertama yang memperkenalkan 7 alat atau metode pengendalian kualitas (*7 tools*). Yakni *fishbone diagram*, *control chart*, *run chart*, *histogram*, *scatter diagram*, *pareto chart*, dan *flowchart*. Dikatakan *Diagram Fishbone* (Tulang Ikan) karena memang berbentuk mirip dengan tulang ikan yang moncong kepalanya menghadap ke kanan. Diagram ini akan menunjukkan sebuah dampak atau akibat dari sebuah

permasalahan, dengan berbagai penyebabnya. Efek atau akibat dituliskan sebagai moncong kepala. Sedangkan tulang ikan diisi oleh sebab-sebab sesuai dengan pendekatan permasalahannya. Dikatakan diagram *Cause and Effect* (Sebab dan Akibat) karena diagram tersebut menunjukkan hubungan antara sebab dan akibat. Berkaitan dengan pengendalian proses statistik, diagram sebab-akibat dipergunakan untuk menunjukkan faktor-faktor penyebab (sebab) dan karakteristik kualitas (akibat) yang disebabkan oleh faktor-faktor penyebab itu (Murnawan and Mustofa, 2014).

BAB III

ANALISIS DAN PERANCANGAN

3.1 Studi Literatur

Studi literatur dilakukan agar memahami konsep dan teori perancangan dan implementasi yang digunakan dalam membuat sistem yang lebih aman dari kunci konvensional dengan integrasi antara NodeMCU sebagai *board* utama, RFID sebagai akses kunci pintu, servo sebagai motor dan *software* Blynk sebagai *push* notifikasi.

Tabel 3.1 Referensi Jurnal terkait RFID, *embedded system*, *smart lock*, mikrokontroler

No.	Literatur	Pembahasan
1.	Arifaldy Satriadi, Wahyudi, dan Yuli Christiyono "Perancangan <i>Home Automation</i> Berbasis NodeMCU" Vol. 8, NO. 1, Maret 2019	Membuat sistem otomatis mengontrol lampu, kipas, AC yang disimbolkan oleh tiga buah lampu, dan juga pintu pagar yang dikendalikan oleh Motor DC melalui web server menggunakan situs hosting jogjahost.com dan menggunakan bahasa pemrograman web PHP dan menggunakan sinyal HSPA+ dan 4G sebagai koneksi internet pada NodeMCU ESP-12E.
2.	Anggher Dea Pangestu , Feby Ardianto , Bengawan Alfaresi "Sistem Monitoring Berbasis Arduino NodeMCU ESP8266" Vol 4 No 1, Juni 2019	Memonitoring beban listrik rumah tangga menggunakan NodeMCU ESP8266 secara real-time dengan modul tambahan LCD Keypad Shield, Current Sensor ACS712.
3.	Ridwan Alief , Darjat, dan Sudjadi "Pemanfaatan Teknologi RFID Melalui Kartu Identitas Dosen Pada Prototipe Sistem Ruang Kelas Cerdas" 16, (2), 2014	Membuat suatu sistem ruang kelas cerdas yang memanfaatkan RFID melalui kartu identitas dosen sebagai media akses penggunaan ruangan kelas menggunakan mikrokontroler Atmega 8535, sensor Pyroelectric Infrared (PIR).
4.	No. Jufri "Rancang Bangun dan Implementasi Kunci Pintu Elektronik Menggunakan Arduino dan Android" Vol. 7, No. 1, Tahun 2016	Merancang bangun kunci elektronik yang dikendalikan dari smartphone Android dengan komunikasi melalui bluetooth dengan board utama menggunakan Arduino.

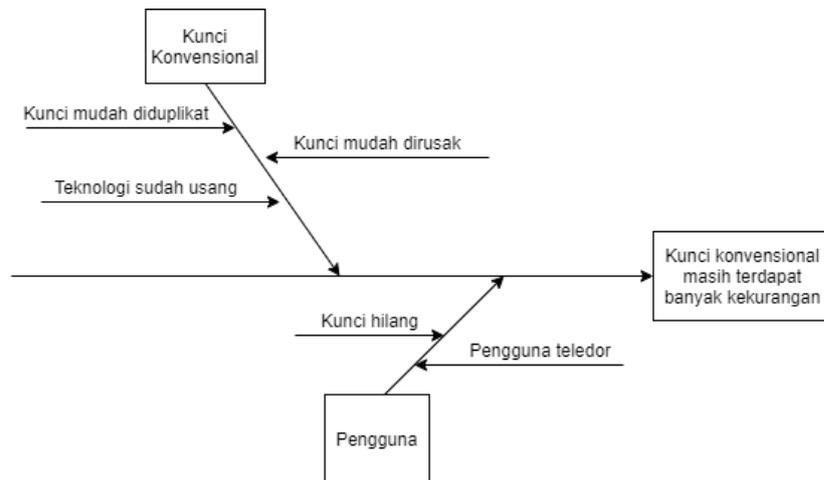
No.	Literatur	Pembahasan
5.	Silfia Rifka, Firdaus, dan Waldito Febri Ramadhan “Penerapan <i>Embedded System</i> pada Sistem Pintar Pengendali Multi Perangkat 51-61 dalam Kelas berbasis Intel Galileo dan Web” Vol. 14, No. 1, April 2018	Memanfaatkan modul mikrokontroler Intel Galileo dan teknologi Internet of Things (IoT) sebagai pengendali, dimana perangkat yang dikendalikan terhubung dengan sistem pengendali jarak jauh berbasis web yang diakses melalui smart devices (smart phone, tablet, dan laptop). Penelitian ini sudah selesai pada tahapan pembuatan pengendali multi perangkat yang ada dalam ruangan belajar (kelas) meliputi lampu, AC, LCD proyektor dan IP camera. Manfaat paling utama menggunakan web sebagai pengendali jarak jauh adalah dapat diakses dengan sistem operasi yang berbeda dengan memanfaatkan internet dan browser yang sudah terpasang pada smart devices.

3.2 Analisis Masalah

Analisis masalah pada penelitian ini digambarkan menggunakan diagram *fishbone* untuk mengorganisir penyebab yang mungkin terjadi dari suatu masalah dalam format yang terstruktur, terdapat 3 komponen pada *fishbone* diagram yaitu sebagai berikut :

1. *Head* (kepala), merupakan fokus masalah yang sedang diamati, dideskripsikan dalam kotak dibagian kepala pada diagram. *Head* terletak pada bagian kanan diagram.
2. *Spine* (tulang belakang), ditunjukkan dengan arah panah yang menunjukan penyebab dari masalah yang terjadi pada bagian *head*.
3. *Bones* (tulang atau duri), menunjukan kategori utama yang menyebabkan permasalahan yang terjadi. Jika terdapat *bones* yang lebih kecil, memperlihatkan kedalam penyebab dari permasalahan tersebut. *Bones* yang saling terkoneksi memperlihatkan bahwa penyebab permasalahan tersebut saling berhubungan.

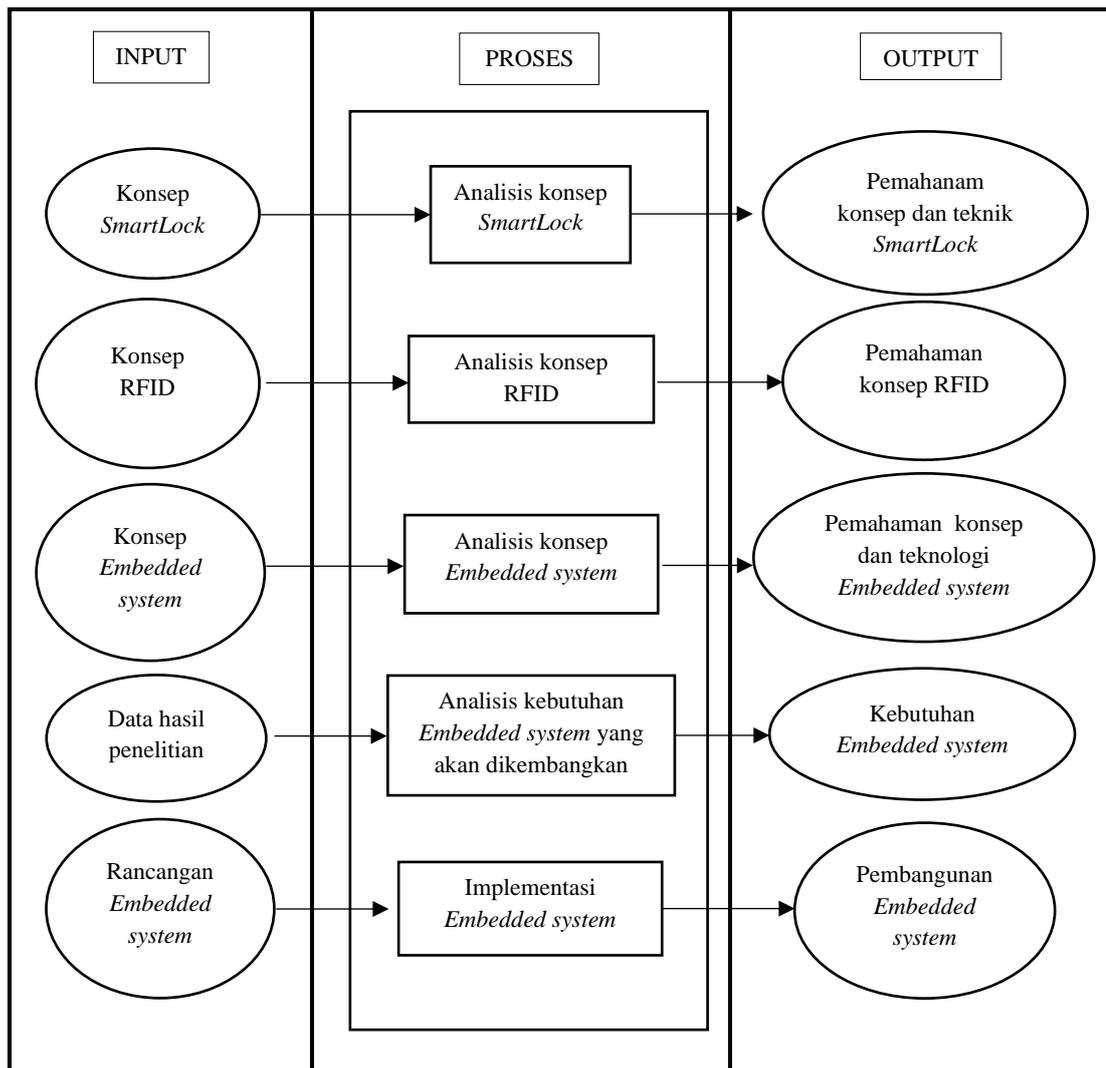
Berikut merupakan Gambar 3.1 *Fishbone diagram* untuk analisis pembangunan *Embedded system* untuk membuat *smartlock*.



Gambar 3.1 *Fishbone diagram*

3.3 Skema Analisis

Berikut ini adalah skema analisis yang dilakukan dalam pengerjaan penelitian ini.



Gambar 3.2 Skema analisis

Pada gambar 3.2 terdapat skema analisis penelitian dimana pada skema tersebut terdapat beberapa komponen yang didapatkan dari literatur yang digunakan

selama melakukan tahapan analisis penelitian ini. Skema analisis menjelaskan adanya inputan yang berupa kebutuhan untuk membangun *Smart Lock* kemudian dari *input* tersebut dilakukan proses analisis untuk mendapatkan *output* yang dijadikan referensi selanjutnya. Proses analisis disesuaikan dengan literatur yang digunakan sebagai langkah untuk kemudian disesuaikan dengan studi kasus yang menjadi objek penelitian tugas akhir. Penjelasan detail mengenai skema analisis diatas dapat dilihat pada tabel 3.2 langkah analisis.

Tabel 3.2 Langkah Analisis

Langkah Analisis	Objek Analisis	Hasil Analisis	Maksud Analisis
Analisis <i>SmartLock</i>	Studi literatur	Memahami konsep <i>SmartLock</i>	Untuk memahami konsep <i>SmartLock</i>
Analisis RFID	Studi literatur	Memahami RFID	Untuk memahami konsep RFID dan bagaimana menerapkan konsep ini dalam membuat <i>smartlock</i>
Analisis <i>Embedded system</i>	Studi literatur	Memahami <i>Embedded system</i>	Untuk memahami konsep <i>Embedded system</i> untuk diterapkan dalam konsep <i>Internet of Things</i>
	Eksplorasi	Memahami <i>Embedded system</i> dan teknologi yang dapat digunakan	Untuk lebih memahami pengembangan <i>Embedded system</i> yang akan diterapkan dalam konsep <i>Internet of Things</i>
Analisis kebutuhan <i>Embedded system</i> yang akan dikembangkan	Analisis aktifitas	Gambaran mengenai alur proses yang berjalan pada <i>smartlock</i>	Untuk mengetahui teknik dan proses dalam sistem <i>smartlock</i> .
	Analisis teknologi	Pemahaman mengenai teknologi - teknologi yang dapat digunakan dalam pengembangan <i>Embedded system</i> untuk <i>Internet of Things</i>	Agar dapat memilih teknologi yang sesuai untuk pengembangan <i>smartlock</i> berbasis <i>Internet of Things</i>

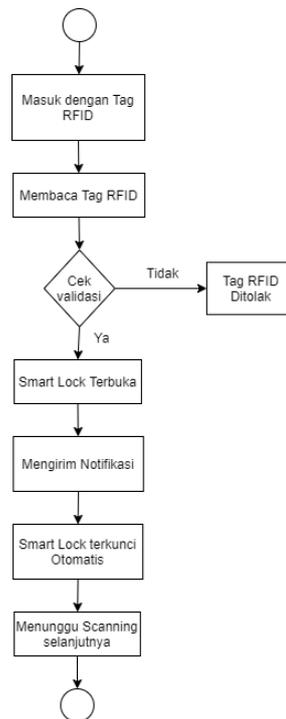
Berikut ini merupakan Tabel 3.3 hasil analisis masalah dalam membuat *smart lock*. Terdapat masalah penyebab dan solusi untuk menyelesaikan permasalahan terhadap keamanan pintu.

Tabel 3.3 Analisis Masalah Terhadap *Smart Lock*

No	Masalah	Penyebab	Solusi
1	Dibutuhkannya sistem <i>Smart Lock</i> yang dapat diakses dengan RFID	Pengguna dapat mengakses <i>Smart Lock</i> dengan RFID	Membangun <i>Smart Lock</i> dengan konsep RFID agar tercipta sistem kunci pintu yang lebih aman
2	Dibutuhkannya konsep RFID untuk membangun <i>system Smart Lock</i>	Agar dapat meningkatkan keamanan pintu dan membuka kunci pintu tanpa alat konvensional	

3.4 Analisis Sistem Yang Diusulkan

Dilihat dari permasalahan yang terjadi di atas, sistem yang diusulkan dalam menyelesaikan permasalahan tersebut adalah dengan membuat sistem *Smart Lock*. Sistem ini dibangun untuk dapat menyelesaikan beberapa permasalahan yang terjadi pada penelitian ini. Adapun alur *Activity Diagram* dari sistem yang diusulkan oleh penulis adalah sebagai berikut:



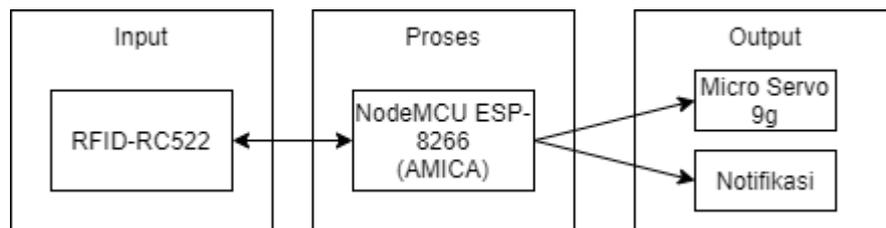
Gambar 3.3 Activity diagram

3.5 Perancangan Sistem

Perancangan sistem merupakan suatu sistem kegiatan yang dilakukan untuk mendesain suatu sistem yang mempunyai tahapan-tahapan kerja yang tersusun secara logis, dimulai dari pengumpulan data yang diperlukan guna pelaksanaan perancangan tersebut. Langkah selanjutnya adalah menganalisis data yang telah dikumpulkan guna menentukan batasan-batasan sistem, kemudian melangkah lebih jauh lagi yakni merancang sistem tersebut. Berikut ini rancangan sistem informasi:

3.5.1 Desain Sistem

Berikut ini merupakan gambaran dari diagram blok pada sistem perancangan *Embedded-system smart lock* yang dapat dilihat pada gambar 3.3:



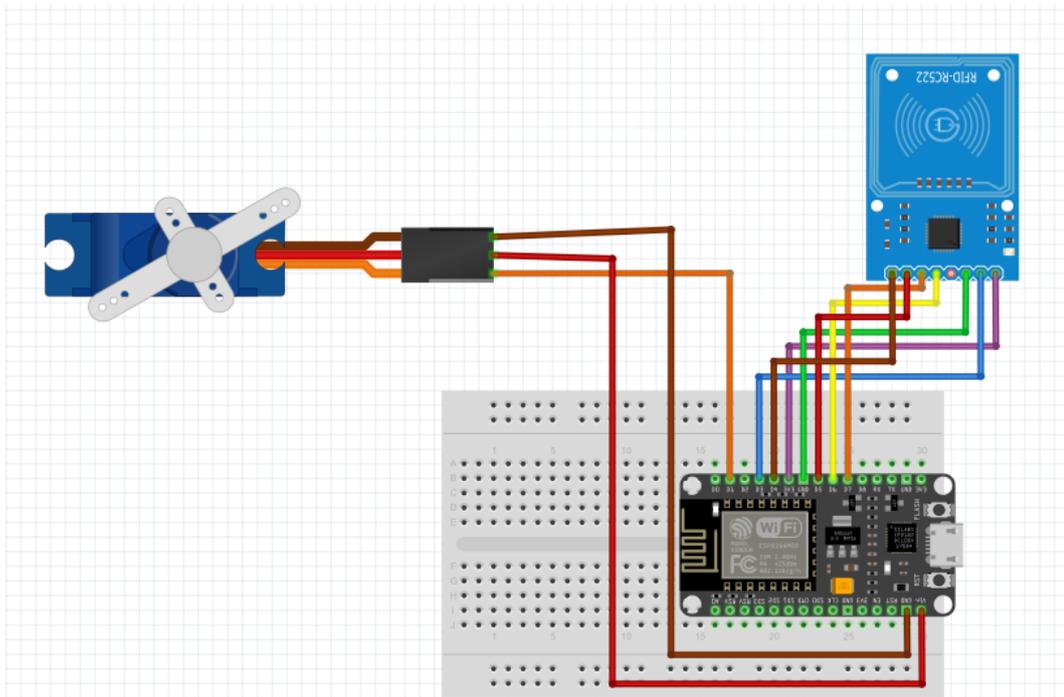
Gambar 3.4 Desain Sistem

Dibawah ini merupakan penjelasan dari beberapa komponen yang digunakan yaitu:

1. RFID digunakan sebagai *reader* untuk *key tag* akses.
2. NodeMCU sebagai pusat kendali dari seluruh rangkaian sistem *monitoring* RFID, dimana NodeMCU akan menerima data yang dikirim oleh RFID dan juga sebagai pusat kendali untuk komunikasi data antara RFID, dan servo.
3. Servo sebagai motor penggerak slot kunci jika akses sudah diverifikasi RFID dan NodeMCU memberikan perintah.

3.5.2 Desain Alat

Berikut ini merupakan gambaran desain pada alat *smart lock* yang akan dibuat. Berikut dapat dilihat desain alat *smartlock* pada gambar 3.5



Gambar 3.5 Desain Alat *Smart Lock*

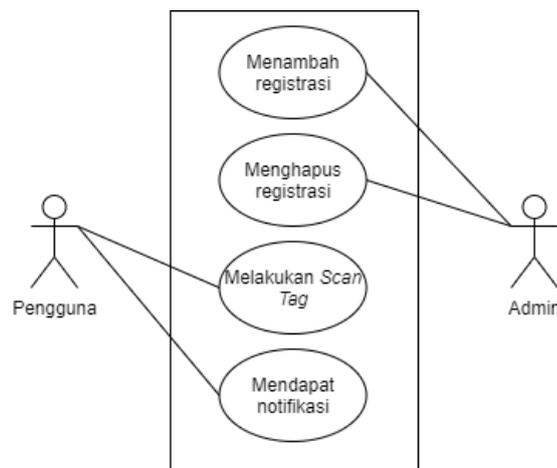
3.5.3 Perancangan Alur Kerja Sistem

Perancangan alur kerja sistem disini penulis akan memberikan penjelasan mengenai fitur-fitur apa saja yang ada di dalam sistem serta bagaimana mengoperasikannya. Untuk perancangannya sendiri akan gambarkan ke dalam diagram UML yang terdiri dari *use case* diagram, *sequence* diagram. Berikut adalah perancangan alur kerja sistem.

3.5.4 *Use Case Diagram*

Use Case Diagram adalah gambaran grafis dari beberapa atau semua *actor*,

use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, *actor*, dan *system*. Berikut adalah *use case diagram* sistem.



Gambar 3.6 *Use Case diagram*

Berikut ini merupakan penjelasan dari diagram *use case* yang akan dipaparkan pada tabel di bawah ini.

Tabel 3.4 Deskripsi *use case*

No.	Nama Use Case	Deskripsi Use Case
1.	Menambahkan Registrasi	Pada bagian <i>Use case</i> ini berfungsi untuk menambahkan registrasi dari pengguna yang akan menambahkan <i>Tag</i> RFID
2.	Menghapus Registrasi	Pada bagian <i>Use case</i> ini berfungsi untuk menghapus registrasi dari pengguna yang akan menghapus <i>Tag</i> RFID
3.	Melakukan <i>Scan Tag</i>	Pada bagian <i>Use case</i> ini berfungsi untuk pengguna jika akan melakukan scan tag untuk membuka <i>Smart Lock</i>
4.	Mendapat notifikasi	Pada bagian <i>Use case</i> ini berfungsi untuk pengguna akan mendapatkan notifikasi jika <i>Smart lock</i> telah terbuka

Selain dari fungsi terdapat juga *actor* yang akan berinteraksi langsung dengan fungsi-fungsi tersebut. Tabel di bawah ini akan menjelaskan mengenai *actor* yang akan berinteraksi langsung dengan perangkat.

Tabel 3.5 Deskripsi aktor pada *use case*

No	Aktor	Keterangan
1.	Admin	Admin merupakan aktor yang dapat melakukan tambah dan hapus data registrasi pada <i>smart lock</i>
2.	Pengguna	Pengguna merupakan aktor yang dapat menggunakan manfaat dari <i>smart lock</i>

3.6 Analisis Skenario

Skenario bertujuan untuk menggambarkan proses interaksi antara sistem atau fungsi aplikasi dengan pengguna. Pada skenario ini penjelasannya dilakukan setiap fungsional aplikasi.

3.6.1 Skenario *Use Case* Menambah Registrasi

<i>Monitoring Smartlock</i>	
Identifikasi	
Nama	Menambah Registrasi
Aktor	Admin
Kondisi Awal	Admin mengakses embedded system melalui Arduino IDE
Aksi Aktor	Reaksi Sistem
1. Melakukan tambah registrasi data Tag RFID	2. Data registrasi <i>Tag</i> RFID bertambah
Kondisi Akhir	Data registrasi <i>Tag</i> RFID pada NodeMCU bertambah

3.6.2 Skenario Use Case Menghapus Registrasi

<i>Monitoring Smart Lock</i>	
Identifikasi	
Nama	Menghapus Registrasi
Aktor	Admin
Kondisi Awal	Admin mengakses <i>embedded system</i> melalui Arduino IDE
Aksi Aktor	Reaksi Sistem
1. Melakukan hapus registrasi data <i>Tag</i> RFID	2. Data registrasi <i>Tag</i> RFID berkurang
Kondisi Akhir	Data registrasi <i>Tag</i> RFID pada NodeMCU berkurang

3.6.3 Skenario Use Case Melakukan Scan Tag

<i>Monitoring Smart Lock</i>	
Identifikasi	
Nama	Melakukan <i>Scan Tag</i>
Aktor	Pengguna
Kondisi Awal	Pengguna melakukan <i>scan tag</i> RFID pada <i>system smart lock</i>
Aksi Aktor	Reaksi Sistem
1. Melakukan scan <i>Tag</i> RFID	2. Sistem Melakukan <i>scan tag</i> RFID pengguna
	3. Sistem melakukan validasi <i>tag</i>
	4. <i>Smart lock</i> terbuka
Kondisi Akhir	<i>Smart Lock</i> terbuka setelah pengguna melakukan <i>scan Tag</i> RFID

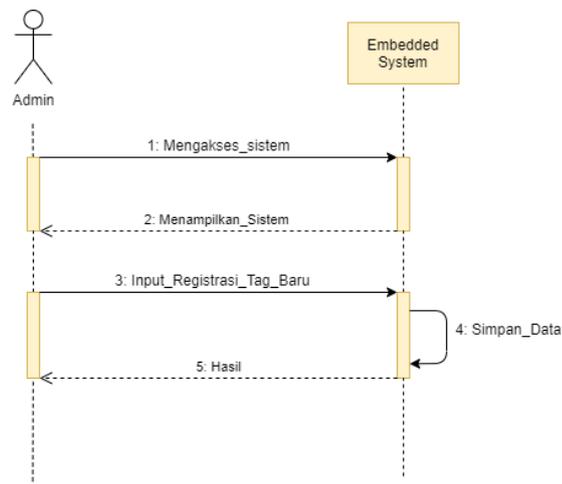
3.6.4 Skenario Use Case Mendapat Notifikasi

<i>Monitoring</i>	
Identifikasi	
Nama	Mendapat Notifikasi
Aktor	Pengguna
Kondisi Awal	<i>Smart lock</i> Telah terbuka
Aksi Aktor	Reaksi Sistem
1. Selesai Melakukan <i>scan Tag</i> RFID	2. Sistem mengirim notifikasi kepada pengguna
Kondisi Akhir	Pengguna mendapat notifikasi dari <i>Smart Lock</i>

3.7 Sequence Diagram Smart Lock

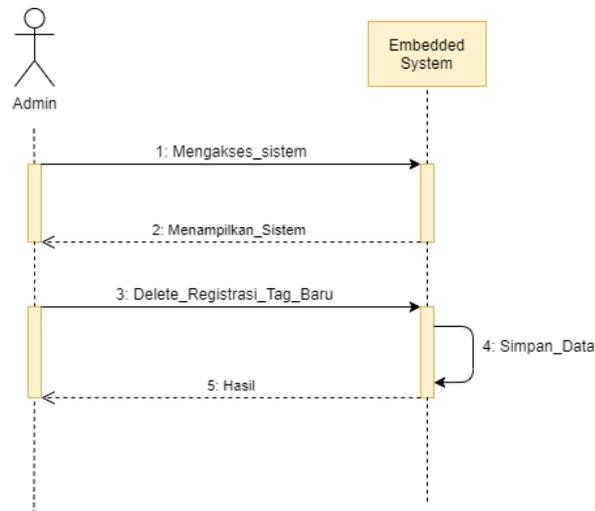
Sequence diagram merupakan sebuah diagram yang menggambarkan interaksi antar objek yang sesuai dengan urutannya, *sequence diagram* ini menunjukkan serangkaian pesan yang dilakukan oleh objek-objek. Berikut ini merupakan beberapa *sequence diagram Smart Lock* yang dapat dilihat pada gambar dibawah ini:

1. Sequence Diagram Menambah Registrasi



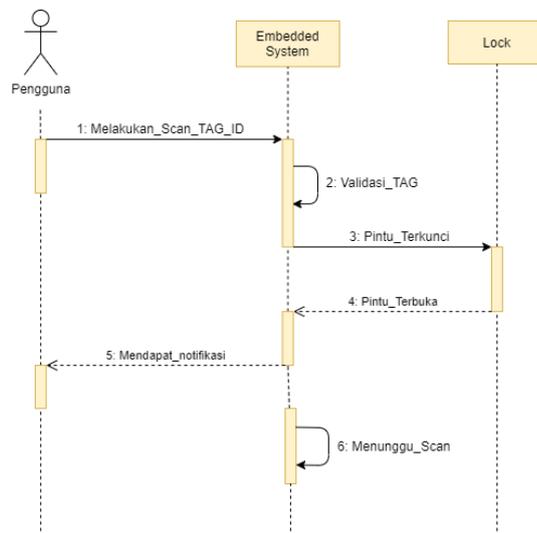
Gambar 3.7 *Sequence Diagram* Menambah Registrasi

2. Sequence Diagram Menghapus Registrasi



Gambar 3.8 Sequence Diagram Menghapus Registrasi

3. Sequence Diagram Scan Tag ID dan Notifikasi



Gambar 3.9 Sequence Diagram Scan Tag ID dan Notifikasi

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi perangkat sistem *smartlock* berbasis mikrokontroler yang dibangun dengan beberapa jenis implementasi terdiri dari implementasi perangkat lunak dan perangkat keras sebagai pendukung dalam membangun *smartlock* menggunakan NodeMCU ESP8266, *tag* dan *reader* RFID dan servo.

Dalam pembangunannya membutuhkan beberapa perangkat lunak sebagai pendukung, diantaranya yaitu:

1. Sistem Operasi

Sistem Operasi yang digunakan yaitu Windows 10.

2. Arduino IDE

Arduino IDE sebagai bahasa pemrograman yang digunakan untuk membangun kode-kode program pada mikrokontroler.

3. Serial Monitor

Serial Monitor digunakan sebagai media untuk melihat hasil program yang telah dibuat di NodeMCU ESP8266.

4. Draw.io

Digunakan untuk membuat sebuah perancangan UML (*Unified Modeling Language*).

Sementara untuk implementasi perangkat keras sebagai pendukung dalam pembangunannya diantaranya yaitu :

5. NodeMCU ESP8266

NodeMCU digunakan sebagai pusat kendali dari seluruh rangkaian sistem.

6. RFID *Reader*

RFID digunakan sebagai pembaca *tag* yang dapat mengakses sistem.

7. Tag RFID

Tag akses digunakan sebagai kunci untuk mengakses sistem.

8. Servo

Servo digunakan sebagai motor penggerak slot kunci.

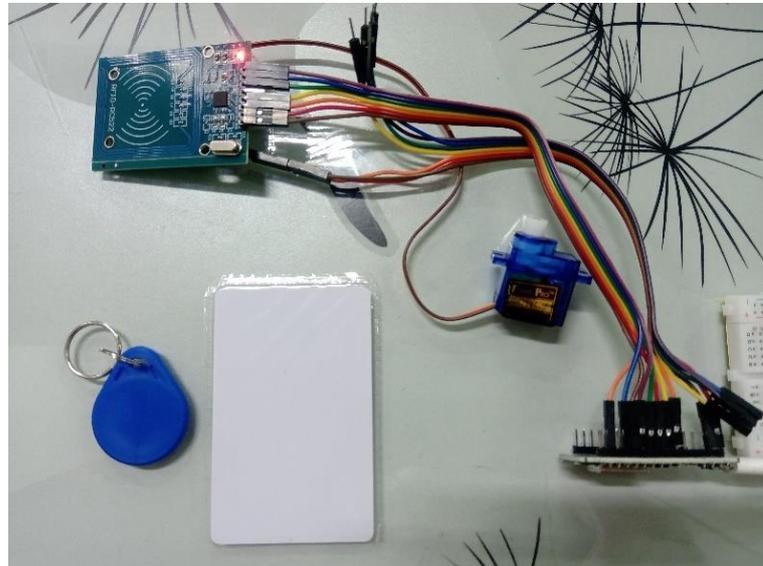
4.1.1 Antarmuka Perangkat Keras

Antarmuka perangkat keras merupakan hasil dari rangkaian seluruh perangkat berdasarkan hasil dari analisis dan perancangan yang telah dilakukan sebelumnya, rangkain perangkat tersebut dapat dilihat pada poin-poin berikut ini :

1. Implementasi Perangkat Keras NodeMCU ESP8266, RFID reader, dan servo

Digunakan sebagai pusat kendali dari semua perangkat keras serta proses data yang telah dihasilkan dari sensor yang selanjutnya data tersebut akan dikirimkan ke web service menggunakan NodeMCU lalu dikirimkan datanya melalui jaringan wifi.

Berikut ini merupakan rangkaian alat sistem *smart lock*:



Gambar 4.1 Implementasi Perangkat Keras

4.2 Metode Pengujian

Pengujian terhadap *software* menggunakan pengujian *gray box* yang merupakan teknik untuk menguji aplikasi dengan pengetahuan yang terbatas tentang cara kerja internal aplikasi. Dalam pengujian perangkat lunak, istilah semakin banyak tahu semakin baik membawa banyak bobot saat menguji aplikasi. Metodologi dimulai dengan mengidentifikasi semua persyaratan *input* dan *output* ke sistem komputer. Informasi ini ditangkap dalam dokumentasi persyaratan perangkat lunak (Maulani *et al.*, 2020).

Dalam pengujian dengan menggunakan metode *gray box testing*, yang dilakukan menggunakan arduino IDE guna mengetahui fungsionalitas dari suatu

software dan untuk menemukan kesalahan dalam beberapa kategori, diantaranya adalah sebagai berikut:

1. Mengamati fungsi-fungsi di setiap modul, seperti fungsi untuk berkomunikasi dengan memperhatikan fungsionalitasnya.
2. Memperhatikan kemungkinan kesalahan yang dapat terjadi saat *compiling* program, melakukan *debug* ataupun *running* program
3. Mengamati performa sehingga aplikasi dapat digunakan dengan baik dan mendukung sistem yang dibuat.
4. Mengamati kemungkinan kesalahan inisialisasi fungsi yang digunakan dalam berinteraksi dengan piranti lain

4.3 Pengujian

Pengujian merupakan tahapan yang dilakukan untuk mengetahui kinerja dari perangkat keras dan perangkat lunak yang telah dilakukan sebelumnya. Setelah dilakukan pengujian akan dilakukan analisa terhadap hasil pengujian, hal ini dilakukan untuk memastikan bahwa alat yang dibangun telah bekerja sesuai dengan perancangan yang dilakukan.

4.3.1 Tahapan Rangkaian dan Pengujian Perangkat Keras

Sebelum melakukan pengujian perangkat keras, terlebih dahulu mempersiapkan komponen-komponen perangkat keras yang akan diuji, kemudian

merangkai komponen-komponen tersebut menjadi sebuah alat *smart lock*. Selanjutnya tahap pengujian dilakukan untuk mengetahui tingkat keberhasilan atau tingkat kegagalan dari perangkat *smart lock* yang telah dibuat. Pengujian perangkat keras ini dilakukan dalam beberapa tahapan diantaranya yaitu:

1. Menentukan dan merangkai pin *wiring* untuk koneksi servo dengan NodeMCU.
2. Menentukan dan merangkai pin *wiring* untuk koneksi RFID *Reader* dengan NodeMCU.
3. Mencari library dan menginstal dari setiap modul pada arduino IDE.
4. Membuat *code* program dan menentukan parameter dari setiap modul.
5. Mengupload *code* program yang telah disiapkan kedalam NodeMCU.
6. Membuat tampilan Blynk menurut analisis yang sudah ditampilkan.

Untuk memastikan perangkat yang telah dirangkai dapat bekerja dengan baik ada beberapa hal yang harus dilakukan, diantaranya yaitu:

1. Memastikan library terinstall dengan baik dan cocok dengan modul yang digunakan.
2. Mengecek fungsi servo dengan mengupload code cek fungsi servo ke NodeMCU melalui Arduino IDE.
3. Mengecek fungsi RFID dengan mengupload code cek fungsi RFID ke NodeMCU melalui Arduino IDE dan menampilkan hasilnya pada serial monitor.
4. Membuat dan mengecek koneksi antara blynk dan NodeMCU.

5. Memastikan adanya notifikasi dan monitor di Blynk berjalan dengan baik.
6. Mengecek ulang program keseluruhan dan memperbaiki jika ada kesalahan.
7. Peranti siap digunakan.

4.3.2 Pengujian NodeMCU ESP-8266 dengan RFID dan Servo

RFID digunakan untuk mendeteksi UID pada *key tag*, sebelum melakukan pengujian langkah pertama yang harus dilakukan yaitu meng-upload program yang sudah dibuat kedalam NodeMCU untuk mengaktifkan RFID agar dapat membaca UID pada key tag. Pada program yang telah diupload ke NodeMCU terdapat sebuah perintah yang dapat mengatur kerja servo, NodeMCU akan mengirimkan data UID *key tag* dan status servo melalui wifi, kemudian akan diproses data yang telah diterima dari sensor dan akan muncul notifikasi pada perangkat klien.

The image shows two windows side-by-side. The left window is a serial monitor displaying the output of a NodeMCU program. The right window is an IDE showing the source code for the program.

```

11:18:30.790 -> Card UID: 88 144 202 147
11:18:31.068 -> Message : User 1
11:17:00.711 -> [1872] Connected to WiFi
11:17:00.746 -> [1873] IP: 192.168.43.129
11:17:00.782 -> [1873]
11:17:00.702 ->
11:17:00.818 ->
11:17:00.818 ->
11:17:00.850 ->
11:17:00.894 ->
11:17:00.918 ->
11:17:00.918 -> [1461] Connecting to blynk-cloud.com:80
11:17:00.963 -> [1419] Ready (ping: 6ms).
11:17:00.091 -> Scan..
11:17:19.091 -> Card UID: 124 33 218 03
11:17:19.364 -> Message : unregistered user
  
```

```

// latihan_rfid_nodemcu_v1.4
if (content.substring(1) == "88 144 202 147")
{
  Serial.println("User 1");
  Blynk.virtualWrite(V2, "User 1 mengakses pintu");
  Blynk.virtualWrite(V2, "Authorized access");
  Serial.println();
  delay(500);
  myServo.write(180);
  delay(5000);
  myServo.write(0);
}
else
if (content.substring(1) == "124 33 218 03")
{
  Serial.println("User 2");
  Blynk.virtualWrite(V2, "User 2 mengakses pintu");
  Blynk.virtualWrite(V2, "Authorized access");
  Serial.println();
  delay(500);
  myServo.write(180);
  delay(5000);
  myServo.write(0);
}
// else
// if (content.substring(1) == "27 A4 C3 A7")
// {
//   Serial.println("User 3");
//   Blynk.virtualWrite(V2, "User 3 mengakses pintu");
//   Blynk.virtualWrite(V2, "Authorized access");
//   Serial.println();
//   delay(500);
//   myServo.write(180);
// }
// else return;
  
```

Gambar 4. 2 Pengujian Program dan Status pada Serial Monitor

Pada pengujian yang telah dilakukan akan mendapatkan sebuah data dari hasil pengujian yang akan dijelaskan pada tabel 4.1

Tabel 4.1 Pengujian Perangkat Keras

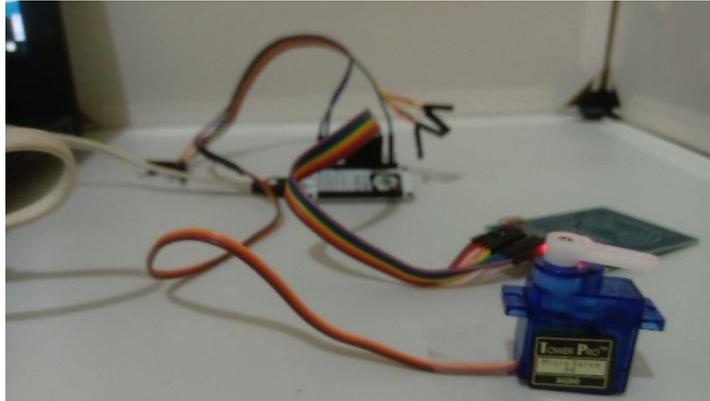
Sensor	Data	Status
RFID	UID	Servo bergerak/tidak dan notifikasi

Berdasarkan dari hasil pengujian pada tabel 4.1 perangkat berjalan sesuai dengan yang diinginkan dimana sensor dapat mendeteksi UID dan memverifikasi apakah servo bergerak atau tidak dan mengirimkan notifikasi.

4.3.3 Analisis Pengujian Perangkat Keras

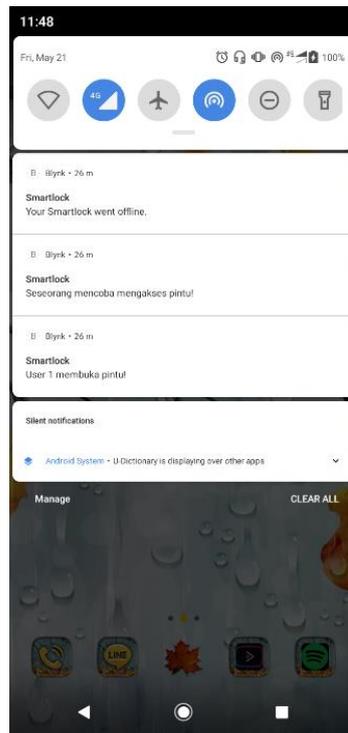
Analisis pengujian ini dilakukan untuk mengetahui kinerja dari perangkat keras secara keseluruhan, pada saat melakukan pengujian semua perangkat keras bekerja dengan baik, ketika sensor mendeteksi UID akan langsung diolah ke NodeMCU dan memberikan status ke servo apakah servo bergerak atau tidak. Berikut ini merupakan gambar-gambar dari proses pengujian yang telah dilakukan:

Jika UID pada key tag terdaftar maka NodeMCU akan memberi perintah pada servo untuk bergerak membuka slot kunci dan jika tidak terdaftar maka servo tetap pada posisi awal.



Gambar 4.5 Servo Bergerak setelah UID diterima

NodeMCU juga akan mengirimkan notifikasi pada user tentang status yang sedang terjadi melalui wifi.



Gambar 4.6 Notifikasi pada User



Gambar 4.7 Pencatatan Data Akses

4.3.4 Pengujian Fungsi

Berikut adalah tabel pengujian yang telah dilakukan pada penelitian ini:

Tabel 4.2 Pengujian Fungsi

No.	Skenario Pengujian	Langkah Pengujian	Pengujian Data	Hasil Pengujian
1	Membaca UID	1. <i>Key Tag</i> diarahkan pada RFID reader	1. RFID Reader membaca UID	Berhasil

2	Pergerakan Servo	1. UID terdaftar 2. Key Tag diarahkan pada RFID reader	1. UID terdaftar maka servo bergerak 2. UID tidak terdaftar maka servo tetap pada posisi awal	Berhasil
3	Notifikasi	1. <i>Key Tag</i> diarahkan pada RFID reader	1. User mendapatkan notifikasi	Berhasil

Berdasarkan dari hasil pengujian pada tabel 4.2 dapat disimpulkan bahwa semua fitur-fitur yang digunakan serta NodeMCU berhasil diimplementasikan, dari hasil pengujian yang telah dilakukan dari semua fungsi telah berjalan dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian yang dikerjakan adalah sebagai berikut:

1. Telah dihasilkannya *Embedded System* sebagai *smart lock*.
2. Alat *smartlock* dapat dibangun menggunakan RFID reader dan kunci elektronik sebagai akses, dan servo menjadi motor penggerak slot kunci dengan board utama menggunakan NodeMCU yang telah menunjang konsep IoT karena sudah dilengkapi dengan modul WiFi untuk mempermudah akses menggunakan pintu, menjaga keamanan dan monitoring akses pintu.

5.2 Saran

Dibutuhkannya perkembangan berkelanjutan tentang pembangunan *Embedded System smart lock* ini, supaya sistem yang dibangun dapat lebih mencakup segala aspek yang lebih luas. Dimana penghapusan dan penambahan user bisa dilakukan user sendiri, menambahkan sensor tambahan sebagai akses menggunakan sistem.

DAFTAR PUSTAKA

Blynk (2021) <https://docs.blynk.cc/>. Available at: <https://docs.blynk.cc/> (Accessed: 26 May 2021).

Chamim, A. N. N. (2010) 'Penggunaan Microcontroller Sebagai Pendeteksi Posisi Dengan Menggunakan Sinyal Gsm', *Jurnal Informatika*, 4(1), pp. 430–439.

Cheung, P. (2018) 'Servo Motor SG90', *Micro motor*, 1(2), p. 180. Available at: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf.

Electronics hub (2013) 'Microcontroller Basics, Types and Applications', p. 1. Available at: <https://www.electronicshub.org/microcontrollers/> (Accessed: 5 February 2021).

Fennani, B., Hamam, H. and Dahmane, A. O. (2011) 'RFID overview', *Proceedings of the International Conference on Microelectronics, ICM*, (May 2019). doi: 10.1109/ICM.2011.6177411.

Güven, Y. *et al.* (2017) 'Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications', *Research Inventy: International Journal of Engineering And Science*, 6(December), pp. 38–44.

Hameed, S. *et al.* (2015) 'Radio Frequency Identification (RFID) Based Attendance & Assessment System with Wireless Database Records', *Procedia - Social and Behavioral Sciences*, 195, pp. 2889–2895. doi: 10.1016/j.sbspro.2015.06.414.

Hendini, A. (2016) 'Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang', *Jurnal Khatulistiwa Informatika*, 2(9), pp. 107–116. doi: 10.1017/CBO9781107415324.004.

Hidayati, N. *et al.* (2018) 'Prototype Smart Home Dengan Modul NodeMCU ESP8266 Berbasis Internet of Things (IoT)', *Teknik Informatika Universitas Islam*

Majapahit, pp. 1–9. Available at: http://repository.unim.ac.id/265/2/JURNAL_5.14.04.11.0.097_NURUL_HIDAYATI_LUSITA_DEWI.pdf.

Kadionik, P. (2013) *Introduction to Embedded Systems, Communicating Embedded Systems*. doi: 10.1002/9781118557624.ch1.

Khadse, R., Gawai, N. and Faruk, B. M. (2014) ‘Overview and Comparative Study of Different Microcontrollers’, 2(XII). Available at: https://dt01-s1.123dok.com/pdf/123dok_us/pdf/2020/05_18/a2xiif1589745484.pdf?X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=94NFDWF3B17T3R35S85K%2F20210202%2F%2Fs3%2Faws4_req-uest&X-Amz-Date=20210202T100452Z&X-Amz-S.

Make-It.ca (2020) *NodeMCU ESP8266 Specifications, Overview and Setting Up, Make-It.ca*. Available at: <https://www.make-it.ca/nodemcu-arduino/nodemcu-details-specifications/> (Accessed: 11 May 2021).

Maulani, J. *et al.* (2020) ‘Penerapan Metode Waterfall Pada Pengembangan Aplikasi’, *Technologia*, 11(2), pp. 64–70.

Melfianora (2019) ‘Penulisan Karya Tulis Ilmiah dengan Studi Literatur’, *Open Science Framework*, pp. 1–3. Available at: osf.io/efmc2.

Murnawan, H. and Mustofa (2014) ‘Perencanaan Produktivitas Kerj dari Hasil Evaluasi Produktivitas degan Metode Fishbone di Perusahaan dengan Metode Fishbone di Perusahaan Percetakan Kemasan PT. X’, *Jurnal Teknik Industri HEURISTIC*, 11(1), pp. 27–46. Available at: <https://core.ac.uk/download/pdf/229336394.pdf>.

Wikipedia (2020) *Servomotor - Wikipedia*. Available at: https://en.wikipedia.org/wiki/Servomotor#cite_ref-11 (Accessed: 22 May 2021).

Zlatanov, N. (2016) ‘Arduino and Open Source Computer Hardware and Software’,

Journal of Water, Sanitation and Hygiene for Development, 10(11), pp. 1–8. doi:
10.13140/RG.2.1.1071.7849.

LAMPIRAN

```
#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#define BLYNK_PRINT Serial

constexpr uint8_t RST_PIN = D3;

constexpr uint8_t SS_PIN = D4;

MFRC522 mfrc522(SS_PIN, RST_PIN);

Servo myServo;

char auth[] = "MOnul33_TLbNsm117Wnl-G_EzmYf5Hwr";

char ssid[] = "Nama Wifi";

char pass[] = "Pass";
```

```
SimpleTimer timer;
```

```
WidgetTerminal terminal(V2);
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
    Blynk.begin(auth, ssid, pass);
```

```
    myServo.attach(5);
```

```
    myServo.write(0);
```

```
    SPI.begin();
```

```
    mfrc522.PCD_Init();
```

```
    Serial.println("Scan..");
```

```
    timer.setInterval(1000L, smartlock);
```

```
}
```

```
void loop()
```

```
{
```

```
    Timer.run();
```

```
    Blynk.run();
```

```
}
```

```
void smartlock()
```

```
{
```

```
if (! mfr522.PICC_IsNewCardPresent())
    return;

if (! mfr522.PICC_ReadCardSerial())
    return;

Serial.print("UID:");

Blynk.virtualWrite(V2, "UID :");

String content= "";

byte letter;

for (byte i = 0; i < mfr522.uid.size; i++)
{
    Serial.print(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfr522.uid.uidByte[i], DEC);
    Blynk.virtualWrite(V2,mfr522.uid.uidByte[i]);
    content.concat(String(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfr522.uid.uidByte[i], DEC));
}

Serial.println();

Serial.print("Message : ");
```

```
content.toUpperCase();
```

```
if (content.substring(1) == "55 164 202 167")
```

```
{
```

```
Serial.println("Cindy mengakses pintu");
```

```
Blynk.virtualWrite(V2, "Cindy mengakses pintu");
```

```
Blynk.virtualWrite(V2, "akses diterima");
```

```
Blynk.notify("Cindy membuka pintu!");
```

```
Serial.println();
```

```
delay(500);
```

```
myServo.write(180);
```

```
delay(5000);
```

```
myServo.write(0);
```

```
}
```

```
else
```

```
if (content.substring(1) == "124 33 225 00")
```

```
{
```

```
Serial.println("User 2 mengakses pintu");
```

```
Blynk.virtualWrite(V2, "User 2 mengakses pintu");
```

```
Blynk.virtualWrite(V2, "akses diterima");
```

```
Blynk.notify ("User 2 membuka pintu!");

Serial.println();

delay(500);

myServo.write(180);

delay(5000);

myServo.write(0);

}

// else

// if (content.substring(1) == "UID")

// {

// Serial.println("User 3");

// Blynk.virtualWrite(V2, "User 3 mengakses pintu");

// Blynk.virtualWrite(V2, "Akses Diterima");

// Serial.println();

// delay(500);

// myServo.write(180);

// delay(5000);

// myServo.write(0);

// }
```

```
else
{
Serial.println("unregistered user");

Blynk.virtualWrite(V2, "unregistered user");

Blynk.virtualWrite(V2, "Akses Ditolak!");

Blynk.notify ("Seseorang mencoba mengakses pintu!");

}
}
```