BAB I

PENDAHULUAN

A. Latar Belakang

Keamanan rumah merupakan aspek yang sangat penting dalam kehidupan sehari-hari. Sistem keamanan rumah mengandalkan kunci fisik untuk membuka pintu rumah secara tradisional memiliki beberapa kelemahan, seperti resiko kehilangan kunci, kunci yang dapat di palsukan, dan ketergantungan pada pengawasan manusia. Tindakan pencurian juga banyak terjadi saat penghuni rumah bepergian atau tingkat kesibukan masyarakat akan pekerjaan yang menyebabkan kurangnya perhatian terdapat keamanan rumah dari bahaya tindakan kriminal pencurian, sehingga ada perasaan khawatir atau was-was saat akan meningglkan rumah baik dalam waktu lama maupun waktu yang relatif sebentar(Samsugi et al., 2020). Sering kali penghuni rumah juga lupa sudah atau belum mengunci pintu rumah saat sudah bepergian dan mau memastikan namun telah berada jauh dari rumah. Oleh karena itu, diperlukan sebuah solusi yang lebih modern dan aman untuk meningkatkan keamanan rumah serta memantau keadaan pintu rumah dalam keadaan terkunci atau tidak.

Kemajuan teknologi dalam bidang *Internet of Things* (IoT) telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam sistem keamanan rumah. IoT memungkinkan perangkat-perangkat yang ada di rumah untuk saling terhubung dan berkomunikasi melalui internet, menciptakan lingkungan rumah yang lebih pintar, aman, dan efisien. Salah satu

penerapan teknologi IoT yang semakin populer adalah dalam pengembangan sistem kunci rumah pintar (*smart key home*). Beberapa penelitian telah dilakukan tentang pemanfaatan IoT dalam peningkatan keamanan rumah. Penelitian yang dilakukan (Mariza Wijayanti, 2022) *prototype smart home* dengan nodemcu esp8266 berbasis iot untuk mengukur suhu ruangan, bila sensor suhu ruangan melebihi 30ºC maka sensor akan memberikan masukan ke *NodeMCU* dan mengeluarkan output kipas otomatis menyala, sensor PIR akan bekerja bila ada pergerakan object sejauh kurang lebih 30 cm, *Sensor Magnetic Switch* akan bekerja bila ada yang membuka jendela jika kedua kutub terpisah lebih dari 3 cm dan akan memberikan notifikasi ke *smartphone* melalui aplikasi *blynk*. (Samsugi et al., 2020) sistem pengaman pintu otomatis dengan mikrokontroler arduino dan *module rf remote* cara kerja alat ini adalah Pintu yang sudah di beri keamanan *RF Remote* tidak akan berfugsi, apa bila di tombol *RF Remot*e tidak di tekan sandi tersebut yang telah dikenal pada mikrokontroler, bila kunci pintu yang sudah ada *RF Remot* akan menghidupkan proses membuka dan menutup pintunya. Hasil pengujian didapatkan modul *RF remote* dapat menerima sinyal frekuensi dan diinputkan pada arduino lalu di outputkan pada *solenoid door lock.*

Berdasarkan pembahasan diatas, maka dari itu penulis mengangkat judul tentang **"Rancang Bangun Smart Key Home berbasis Internet Of Things"**. Dalam penelitian ini, dirancang alat yang dapat digunakan untuk membuka dan menutup pintu lalu mengunci pintu secara otomatis menggunakan *smartphone* yang dapat diakses dari jarak jauh, alat ini juga

dilengkapi tombol *keypad* yang berfungsi untuk membuka pintu menggunakan *password* secara langsung di pintu alat. *Prototype* alat ini menggunakan Esp32 sebagai kontroller yang mengontrol komponen alat serta menghubungkan alat dengan *smartphone*. Dengan demikian pengguna dapat merasa lebih aman dan nyaman, meskipun sedang berada di luar rumah.

B. Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan yaitu:

1. Bagaimana merancang dan membangun alat *smart key home* berbasis *internet of things*?
2. Bagaimana cara mengontrol alat solenoid *door lock* menggunakan NodeMCU ESP32 dari jarak jauh?
3. Bagaimana cara mengkoneksikan *keypad* ke *NodeMCU* ESP32?
4. Bagaimana cara mengetahui pintu terbuka dan tertutup melalui perangkat *smartphone* android?
5. Bagaimana cara menghubungkan sensor *magnetic door switch* MC38 ke NodeMCU ESP32?

C. Tujuan Penelitian

1. Untuk mengetahui bagaimana cara merancang dan membangun alat *smart key home* berbasis *internet of things*.
2. Untuk mengetahui bagaimana cara mengontrol alat solenoid *door lock* menggunakan *NodeMCU ESP32* dari jarak jauh melalui jaringan internet.
3. Bagaimana cara mengkoneksikan *keypad* ke *NodeMCU ESP32.*

4. Bagaimana cara mengetahui pintu terbuka dan tertutup melalui perangkat *smartphone* android
5. Bagaimana cara menghubungkan sensor *Magnetic Door Switch* MC38 ke *NodeMCU ESP32*

D. Batasan Masalah

Pada Skripsi ini, pembahasan dibatasi pada :

1. Perancangan dan pengembangan *System Smart Key Home* berbasis *Internet Of Thing* untuk membuka dan mengunci pintu secara otomatis.
2. *System* yang dirancang hanya akan berfungsi pada satu pintu utama dan tidak mencakup pada jendela, pintu samping atau gerbang.
3. *Password* yang digunakan pada keypad 0-5v DC hanya memiliki 16 tombol

E. Manfaat Penelitian

Manfaat dari penelitian ini dapat diuraikan sebagai berikut yaitu:

1. Manfaat bagi penulis

Adapun manfaat bagi penulis adalah dapat mengimplementasikan ilmu yang telah didapatkan selama proses perkuliah.

2. Manfaat bagi masyarakat

Adapun manfaat bagi masyarakat yang dapat dijadikan referensi untuk pengembangan dan pembuatan alat keamanan rumah dengan memanfaatkan teknologi agar lebih dipermudah.

3. Manfaat bagi dunia akademik

Adapun manfaat bagi dunia akademik penelitian ini diharapkan dapat dimanfaatkan sebagai pembelajaran dan referensi bagi peneliti-peneliti selanjutnya.

F. Sistematika Penulisan

Adapun sistematika penulisan pada tugas akhir ini yaitu :

BAB I PENDAHULUAN

Berisi tentang uraian latar belakang pemilihan judul Rancang bangun *smart key home* berbasis *internet of things* , rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Dalam bab ini berisi beberapa teori-teori yang mendukung dalam pembahasan penyusunan tugas akhir ini serta bahasa pemrograman yang digunakan sehingga memudahkan penulis dalam menyelesaikan masalah.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan mengenai tahapan - tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu tempat penelitian dan waktu penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang atau dibuat.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai analisis sistem yang berjalan, analisis sistem yang diusulkan, rancangan sistem dan metode pengujian serta teknik pengujian.

BAB V PENUTUP

Pada bab ini menjelaskan mengenai kesimpulan dan saran guna menjadi referensi bagi para peneliti lain yang akan melanjutkan atau mengembangkan penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Rancang Bangun

Rancang bangun adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhankebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfirmasi dari komponen perangkat keras dan perangkat lunak dari semua sistem (Jogiyanto, 2005). Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002). Menurut Maulani, dkk (2018), rancang bangun merupakan sebuah proses menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut. Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa rancang bangun adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan fungsional, serta 7 menggambarkan bagaimana suatu sistem dibentuk yang dapat

berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfirmasi dari komponenkomponen perangkat keras dan perangkat lunak dari semua sistem. Dalam hal ini, penulis merujuk pengertian rancang bangun menurut Jogiyanto (2005)

2. Smart key

Smart key adalah cara untuk mengunci dan membuka pintu melalui teknologi kunci pintar, tanpa menggunakan jenis kunci fisik. Seperti mengunci dan membuka pintu menggunakan kunci digital, kunci otomatis, fitur buka kunci otomatis, sistem keamanan ini tidak hanya praktis, tetapi juga memiliki kualitas keamanan tertinggi. Kunci pintu pintar sangat berbeda dengan kunci teknologi tradisional dan menerapkan penggunaan dan sistem operasi terbaru, terutama untuk sistem yang membuka dan mengunci pintu (Iglooadmin, 2019).

Fungsi utama dari *smart key* adalah untuk membatasi orang yang dapat mengakses pintu sehingga hanya orang-orang tertentu yang mendapat ijin dan mendapat wewenang yang dapat mengakses pintu tersebut. Dengan *smart key* ini maka keamanan rumah dapat lebih terjamin (Iman, 2018).

3. Internet Of Things

Internet of things merupakan konsep penambahan jaringan yang terhubung ke suatu *system* yang bertujuan memperbesar atau memperluas jaringan internet yang tersambung kepada modul berupa mesin atau *interface* yang terhubung

secara terus menerus dengan membaca data dari *input* sensor atau *actuator*. Data yang dikirimkan baik data ADC, suhu atau data yang dikelola dari hasil kinerja modul tersendiri sehingga memungkinkan *controller* berkolaborasi dengan *system* lain dengan memanfaatkan *internet* (Yoyon Efendi,2018).

Internet of Things (IoT) adalah sebuah paradigma yang menghubungkan objek dan perangkat sehari-hari ke Internet, memungkinkan mereka untuk mengumpulkan dan bertukar data tanpa campur tangan manusia (Khan & Yuce, 2019; Bijapur, 2020). Jaringan perangkat yang saling terhubung ini mencakup kendaraan, peralatan, dan bangunan yang tertanam dengan sensor, perangkat lunak, dan konektivitas jaringan (Khan & Yuce, 2019). Aplikasi IoT menjangkau berbagai domain, termasuk rumah pintar, mobil yang terhubung, sistem perawatan kesehatan, dan pertanian pintar (Bijapur, 2020). Konsep ini bertujuan untuk meningkatkan keberadaan Internet di mana-mana dengan mengintegrasikan objek dengan sistem yang tertanam, memungkinkan interaksi tanpa batas antara perangkat dan manusia (Carretero & García Sánchez, 2013). IoT menawarkan manfaat seperti peningkatan layanan, penghematan energi, dan peningkatan layanan kesehatan, namun juga menimbulkan tantangan teknis dan etika (Carretero & García Sánchez, 2013). Evolusi IoT difasilitasi oleh kemajuan teknologi, termasuk perangkat seluler, sensor, dan infrastruktur jaringan yang lebih baik (Ivanov, 2014). Seiring dengan terus berkembangnya IoT, hal ini diperkirakan akan berdampak secara signifikan terhadap cara kita berinteraksi dengan internet dan lingkungan fisik kita.

Namun IoT bukan hanya terkait dengan pengiriman dan penerimaan data, IoT juga melibatkan pengumpulan, analisis, dan penggunaan data untuk berbagai tujuan. Salah satu aspek utama dari IoT adalah kemampuannya untuk menghubungkan berbagai perangkat fisik ke internet, sehingga memungkinkan interaksi antara perangkat tersebut, pengguna, dan lingkungan sekitarnya.

1. Cara kerja dan Komponen-komponen *internet of things*



Gambar 2. 1 Komponen-komponen *Internet Of Things*

Secara umum, sistem IoT bekerja dengan cara menghubungkan berbagai perangkat elektronik ke internet, sehingga perangkat-perangkat tersebut dapat saling berkomunikasi dan bertukar data. Berikut adalah komponen-komponen utama dalam sistem IoT dan bagaimana mereka bekerja:

a) Perangkat IoT

Perangkat IoT adalah perangkat elektronik yang dilengkapi dengan sensor, mikrokontroler, dan kemampuan terhubung ke internet. Contohnya adalah lampu pintar, kunci pintu otomatis, thermostat, dan lain-lain. Perangkat ini berfungsi untuk mengumpulkan data dari lingkungan sekitar.

b) Koneksi Internet

Koneksi internet untuk dapat saling berkomunikasi. Koneksi ini bisa melalui *WiFi*, *Bluetooth*, 4G/5G, atau teknologi nirkabel lainnya. Koneksi ini memungkinkan perangkat IoT untuk mengirim data yang dikumpulkan ke *cloud.*

c) Gateway

Gateway berfungsi sebagai penghubung antara perangkat IoT dengan cloud. *Gateway* menerima data dari perangkat IoT, lalu mengirimkannya ke cloud untuk diproses lebih lanjut. *Cloud Computing* Data yang dikirim oleh perangkat IoT akan disimpan dan diproses di *cloud computing*. Di sini, data akan dianalisis dan digunakan untuk mengontrol perangkat IoT.

d) Aplikasi dan Antarmuka Pengguna

Pengguna dapat mengakses dan mengendalikan perangkat IoT melalui aplikasi di smartphone atau komputer. Aplikasi ini terhubung ke *cloud* dan memungkinkan pengguna untuk memantau, menganalisis, dan mengendalikan perangkat IoT.

e) Implementasi *internet of things*

Dalam era yang semakin terhubung secara digital ini, *Internet of Things* (IoT) telah menjadi pendorong utama transformasi di berbagai sektor, mulai dari rumah tangga hingga industri. Implementasi IoT telah membuka pintu menuju kemungkinan baru dalam memantau, mengelola, dan mengoptimalkan berbagai aspek kehidupan dan bisnis. Dengan memanfaatkan teknologi ini, kita dapat meningkatkan efisiensi, produktivitas, dan kualitas layanan secara signifikan.

Tabel 2. 1 Implementasi IOT

No.	Bidang Iot	Contoh Implementasi
1	Implementasi *IoT* dalam Bidang Keamanan	Pemasangan kamera keamanan yang terhubung ke internet untuk memantau aktivitas di dalam dan di sekitar rumah. Pengguna dapat mengakses video secara real-time melalui aplikasi ponsel mereka
2	Implementasi *IoT* Dalam Manufaktur dan industri	Pemantauan Mesin, Pemasangan sensor pada mesin-mesin pabrik untuk memantau kinerja dan mencegah kerusakan yang tidak terduga. Data yang dikumpulkan dapat digunakan untuk perawatan preventif dan perencanaan produksi yang lebih efisien
3	Implementasi *IoT* dalam bidang pertanian dan perkebunan	Pemantauan Tanaman : Pemasangan sensor tanah, sensor kelembaban udara, dan sensor cuaca di lahan pertanian untuk memantau kondisi tanaman secara real-time. Data yang terkumpul dapat digunakan untuk mengatur irigasi dan memberikan nutrisi yang tepat dll.

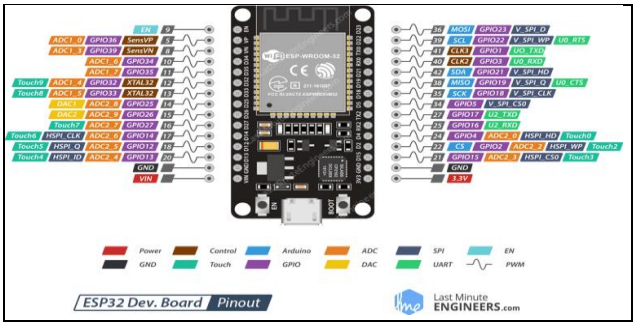
4. NodeMCU

ESP32 adalah sebuah single chip yang sudah terintegrasi dengan modul WiFi dan modul Bluetooth. Chip ESP32 dibuat oleh *Espressif System* dengan bantuan perusahaan semi konduktor TSMC yang menggunakan teknologi ultra-low power 40 nm. ESP32 dibuat untuk tujuan sebuah chip yang memiliki penggunaan sumber daya terendah dengan *performa radio frequency* (RF) yang tinggi, untuk 17 UIN Syarif Hidayatullah Jakarta menunjukkan ketangguhan, kekuatan dan dapat digunakan untuk berbagai macam implementasi. Beberapa fitur yang diunggulkan pada ESP32 adalah solusi untuk alat-alat dengan sumber daya yang sangat rendah atau hemat sumber daya dan juga ESP32 dapat digunakan untuk solusi alat-alat *Internet of Things (IoT)* karena mudahnya integrasi ESP32 dengan modul-modul eksternal lainnya (Raihan, 2022).

ESP32 banyak digunakan untuk berbagai macam mikrokontroler seperti ESP32 DOIT Devkit v1 yang digunakan pada penelitian ini. Beberapa mikrokontroler lainnya yang menggunakan chip ESP32 seperti ESP32 CAM yaitu ESP32 yang sudah terintegrasi dengan modul kamera, terdapat juga ESP32 DevkitC V4, LoRa ESP32, dan WeMos D1 R32 yang merupakan gabungan dari ESP32 dengan Arduino UNO. Perbedaan dari versi ESP32 tersebut terdapat pada konfigurasi pin GPIO, jumlah pin GPIO (analog dan digital), dan port yang digunakan seperti port USB.

Tabel 2. 2 Spesifikasi dari NodeMCU Esp32

Mikrokontroller	***ESP32***
Prosesor	Xtensa single/dual-core 32-bit LX6 160/240 MHz
SRAM	520 KB
Memori Flash	4 MB
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth v4.2 / Bluetooth *Low Energy*
Tegangan Operasi	3.3V
Tegangan Input	7-12V (Vin) / 5V (USB)
Pin GPIO	30 atau 36 (sesuai model)
Digital IO	25
ADC	15
DAC	2
Capacitive Touch	9
UART	3
SPI	2
12C	3
Kontroler USB	CP2102



Gambar 2. 2 NodeMCU ESP32

Terdapat beberapa alasan menggunakan ESP32 pada penelitian ini seperti ESP32 sudah terinte grasi dengan modul Wi-Fi sehingga tidak memerlukan modul Wi-Fi tambahan ketika menggunakan konektivitas Wi-Fi. Alasan lainnya, harga ESP32 terjangkau dengan harga kisaran 70 ribu Rupiah pada bulan Juni pada tahun 2021 ini. ESP32 juga memiliki pin GPIO yang memenuhi kriteria untuk alat yang 19 UIN Syarif Hidayatullah Jakarta dibuat pada penelitian ini. Beberapa ESP32 memiliki mikroprosesor dual *core* sehingga memungkinkan 2 program berjalan dalam waktu yang bersamaan (multitasking)

5. Kabel Jumper

Kabel *jumper* merupakan kabel elektrik yang mempunyai pin konektor di setiap ujungnya dan memungkinkan untuk menghubungkan dua komponen yang melibatkan Esp8266 tanpa memerlukan solder. Intinya, kegunaan kabel jumper ini digunakan sebagai konduktor listrik untuk menyambungkan rangkaian listrik.

Kabel jumper biasanya digunakan pada breadboard atau alat *prototyping* lainnya supaya lebih mudah untuk mengutak-atik rangkaian. Konektor yang terdapat pada ujung kabel terdiri dari konektor jantan (*male connector*) dan konektor betina (*female connector*). Konektor female berfungsi untuk menusuk dan konektor male berfungsi untuk ditusuk.Sederhananya, kabel *jumper* bekerja untuk menghantarkan arus listrik dari satu komponen ke komponen lainnya yang dihubungkan. Hal ini terjadi karena di ujung dan di dalam kabel terdapat konduktor listrik kecil yang memang fungsinya untuk menghantarkan listrik.

Tabel 2. 3 Spesifikasi kabel jumper

Type kabel	26 AWG
Panjang kabel	20cm
Kongigurasi	Female to male



Gambar 2 1 Kabel jumper

6. *Keypad*

Keypad membrane 4x4 adalah tombol-tombol yang disusun secara matriks (baris x kolom) sehingga dapat mengurangi penggunaan pin input. *Keypad* memiliki 12 tombol dimana keypad memiliki konfigurasi 4 baris (input scanning) dan 4 kolom (output scanning). *Keypad* berfungsi sebagai input kode *password* untuk membuka s*olenoid door lock*

Tabel 2. 4 Spesifikasi *keypad membrane 3x4*

Jumlah tombol	4 baris x 4 kolom (total 16 tombol)
Koneksi	female 2.54mm pitch
Tegangan oper	3V-5V DC



Gambar 2. 3 *Keypad 4x4*

7. Solenoid *door lock*

Solenoid door lock 12V DC adalah salah satu s*olenoid* yang difungsikan khusus sebagai *solenoid* untuk membuka pintu elektronik. *Solenoid* ini mempunyai dua sistem kerja, yaitu *normaly close* (NC) dan *Normaly Open* (NO). perbedaannya adalah jika cara kerja solenoid NC apabila diberi tegangan, maka *solenoid NO* adalah kebalikannya dari *solenoid* NC. Biasanya kebanyakan *Solenoid door lock* membutuhkan inputan kerja 12V DC. Berarti dibutuhkan *power supply* 12V dan sebuah *relay* untuk mengaktifkannya.

Tabel 2. 5 Spesifikasi *Solenoid door lock* 12V DC

Tegangan	12VDC
Arus	0.35A
Dimensi	27x27x18mm
Panjang lacth	10mm
Bentuk energy	Intermittent
Waktu buka kunci	1 detik



Gambar 2. 4 *Solenoid door lock* 12v DC

7. *Buzzer*

Buzzer adalah sebuah elektronika yang berfungsi mengubah listrik menjadi getaran suara. Pada dasarnya cara kerja *buzzer* hampir sama dengan *load*

speaker, Buzzer terdiri dari kumparan yang berpasangan pada diafragma. *Buzzer* bisa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat.

Tabel 2. 6 Spesifikasi *Buzzer*

Diameter	22mm/0,86
Tegangan operasi	3 - 24V DC
Ketinggian alarm	10mm/0,39
2 jarak lubang pemasangan	30mm/1,18
2 panjang kabel	90mm/3.54
Jenis bel	Piezoelektrik
Tingkat tekanan suara	95 Db
Tingkat Tegangan	12V DC
Peringkat Arus	maks 10mA
Frekuensi	3900 ± 500Hz



Gambar 2. 5 *Buzzer*

8. Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak *Saklar/Switch*). *Relay* menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (low

power) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A .

Adapun Spesifikas Relay 2 cannel sebagai berikut:

1. Tipe relay adalah SPDT (Single Pole Double Throw): 1 COMMON, 1 NC (Normally Close), dan 1 NO (Normally Open)
2. Memakai SMD Optocoupler isolation, yang berkinerja stabil dengan arus pemicu (trigger current) hanya sebesar 5mA
3. Tegangan sinyal pemicu sebesar 5V DC
4. Dapat disetting untuk mendeteksi *high* atau *low* dengan mengubah jumper
5. Dirancang dengan toleransi keamanan, bahkan jika arus pemicu putus, *relay* tidak akan bekerja
6. Dilengkapi lampu indikator *Power* (hijau) dan Status *Relay* (merah)
7. Mudah dipasang, menggunakan terminal untuk pemasangan kabel.
8. Ukuran: 50x41x18.5mm
9. Dilengkapi 4 lobang baut berdiameter 3.1mm berjarak 44.5mm x 35.5mm



Gambar 2. 6 Relay V5 2 C*hannel*

10. Motor servo

Motor Servo merupakan sebuah motor yang memiliki sistem umpan balik tertutup. Sistem umpan balik ini akan menginformasikan posisi dari motor ke rangkaian kontrol yang ada di dalam motor servo. Berbeda dengan motor DC pada umumnya, pada motor servo ini selain terdapat serangkaian gear, juga terdepat potensiometer dan rangkaian kontrol. Potensiometer berfungsi sebagai pembatas dari putara motor servo sehingga putaran motor dapat diatur sesuai dengan kebutuhan. Pengaturan sudut putaran motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kabel ke kaki motor dibagian sinyal (Novi Lestari, 2017).

Motor servo merupakan komponen untuk mengubah energi listrik menjadi energi mekanik. Terdapat permanen magnet motor DC servo yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dua medan magnet. Salah satu medan dihasilkan oleh magnet permanen dan yang satu lagi dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Pada umumnya motor servo memiliki 2 jenis. Jenis yang pertama adalah motor servo standar. Motor tipe ini hanya mampu berputar 180 derajat. Motor servo tipe ini sering digunakan pada sistem robotika, misalnya untuk membuat "*Robot Arm*" (Robot Lengan). Jenis yang kedua adalah motor servo *continuous*. Motor tipe ini mampu berputar hingga 360 derajat. Motor servo tipe ini sering

digunakan pada *Mobile Robot*. Tipe motor servo ini tertulis pada bagian badan motor.



Gambar 2. 7 Motor servo

Tabel 2. 7 Spesifikasi motor servo

Berat	55 gram
Dimensi	40,7 x 19,7 x 42,9 mm approx
Torsi	9,4 kgfcm (4,8 V), 11 kgfcm (6 V)
Kecepatan kerja	0,17 s/60 (4,8 V), 0,14 s/60 (6 V)
Tegangan kerja	4,8 V a 7,2 V
Rentang suhu	0º C

11. Sensor *Magnetic Door Switch*

Sensor Magnet MC-38 adalah sensor untuk mendeteksi bukaan/tutupan pintu yang bekerja berdasarkan prinsip elektromagnetik. Dalam kondisi normal sensor dan magnet tidak dalam keadaan berdekatan yaitu saklar berada dalam pada kondisi terbuka (*open circuit*), sedangkan untuk kondisi aktif sensor dan magnet berdekatan atau pada pintu tertutup dan saklar berada pada kondisi tertutup (*closed circuit*) dengan nilai hambatan ±4Ω. Saklar ini berupa sensor yang dipasangkan dengan sebuah magnet serta dikemas dalam kotak plastik yang siap ditempel yang dapat diaplikasikan langsung ke pintu, jendela, laci, lemari dan

sebagainya berbahan nonmetal. Pada komponen sensor terdapat kabel yang dapat langsung dihubungkan dengan Mikrokontroler, atau dapat juga digunakan sebagai saklar untuk mengaktifkan rangkaian elektronika lainnya



Gambar 2. 8 Sensor *Magnetic Door Switch* MC-38

Tabel 2. 8 Spesisfikasi Sensor *Magnetic Door Switch*

Rated current	100mA
Rated voltage	200V DC
Resistance	±4Ω
Dimension	28x15x0,9cm

12. Arduino IDE

Arduino IDE adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware-nya* memiliki prosesor *Atmel* AVR dan softwarenya memiliki bahasa pemrograman sendiri.

Arduino ide merupakan perangkat lunak *open source* yang digunakan untuk menulis kode, perangkat lunak ini dibuat menggunakan java dan dapat bekerja di berbagai platform seperti windows, mac, dan linux.

Untuk memprogram *board* Arduino, dibutuhkan aplikasi *IDE (Integrated Development Environment)* bawaan dari Arduino. Aplikasi

ini berguna untuk membuat, membuka, dan mengubah *source code* Arduino berisi logika dan algoritma yang akan diupload ke dalam IC mikrokontroller

1. Fungsi Arduino IDE

(*Integrated Development Environment*) dapat menulis *sketch*. *Sketch* ditulis dalam suato editor sebuah teks dan disimpan dengan ekstensino. Teks editor yang ada pada Arduino IDE yang memiliki fitur untuk cut and paste dan search and replace sehingga programmer dapat lebih mudah dalam melakukan penulisan kode program. Tidak hanya itu saja *software* Arduino IDE juga mempunyai fitur *message box*, yang mempunyai warna hitam guna menampilkan sebuah status seperti pesan error, compile, dan melakukan program. Di bagian paling kanan dari software ini juga menunjukkan sebuah board yang terkonfigurasi sekaligus COM Ports yang mau digunakan.

Menu-menu yang ada pada software Arduino IDE

a. Verifiy

Verifiy yang tersedia pada Arduino IDE digunakan sebagai pemeriksaan kode program yang telah kami buat. Apakah sudah benar atau belum, jika belum nanti akan muncul sebuah *message box* yang memberitahukannya**.**

b. *Menu*

Menu upload idgunakan sebagai untuk untuk mengupload *sketch* yang telah selesai ke board Arduino. Jika menggunakan *menu* upload maka akan otomatis Arduino IDE akan melakukan sebuah *verify* lebih dulu sebelum melakukan proses upload.

c. *New*

Menu new berfungsi sebagai membuat halaman *sketch* baru.

d. *Open*

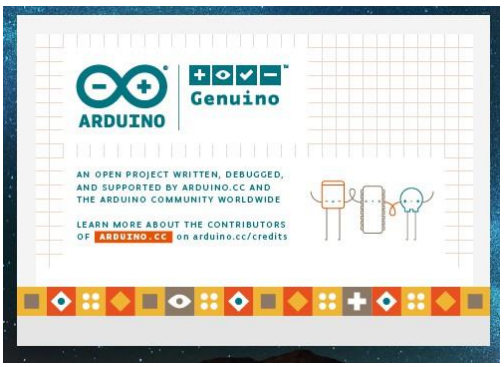
Menu open fungsinya digunakan untuk membuat proyek *sketch* yang sudah tersimpan di dalam folder sebelumnya.

e. *Save*

Menu save berguna untuk melakukan penyimpanan *sketch* yang sudah dibuat.

f. *Serial Monitor*

Menu *serial monitor* digunakan untuk membuka *serial monitor* untuk menampilkan sebuah jendela yang dikirimkan atau dipertukarkan antara *skecth* dan Arduino pada port serialnya.



Gambar 2. 9 Arduino

13. Bahasa Pemrograman berbasis bahasa C

Program Arduino sendiri menggunakan bahasa C. Walaupun banyak sekali terdapat bahasa pemrograman tingkat tinggi (*high level language*) seperti pascal, basic, cobol, dan lainnya. Walaupun demikian, sebagian besar dari para programer

profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul, berikut alasan-alasannya :

a. Bahasa C merupakan bahasa

Telah terbukti dapat menyelesaikan program-program besar seperti pembuatan sistem operasi, pengolah gambar (seperti pembuatan game) dan juga pembuatan kompilator bahasa pemrograman baru.

b. Bahasa C merupakan bahasa yang portabel

Dapat dijalankan di beberapa sistem operasi yang berbeda. Sebagai contoh program yang kita tulis dalam sistem operasi *windows* dapat kita kompilasi didalam sistem operasi linux dengan sedikit ataupun tanpa perubahan sama sekali.

c. Bahasa C merupakan bahasa yang sangat popular

Banyak digunakan oleh programer berpengalaman sehingga kemungkinan besar library pemrograman telah banyak disediakan oleh pihak luar/lain dan dapat diperoleh dengan mudah.

d. Bahasa C merupakan bahasa yang bersifat modular

Tersusun atas rutin-rutin tertentu yang dinamakan dengan fungsi *(function)* dan fungsi-fungsi tersebut dapat digunakan kembali untuk pembuatan program-program lainnya tanpa harus menulis ulang implementasinya.

e. Bahasa C merupakan bahasa tingkat menengah

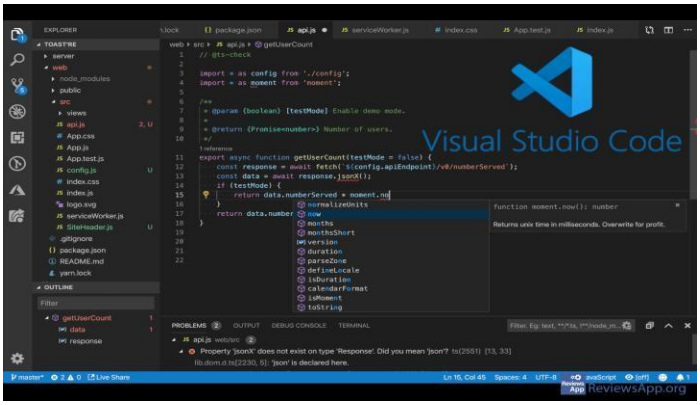
Bahasa tingkat menengah sehingga mudah untuk melakukan *interface* (pembuatan program antar muka) ke perangkat keras. Struktur penulisan

program dalam bahasa C harus memiliki fungsi utama, yang bernama main. Fungsi inilah yang akan dipanggil pertama kali pada saat proses eksekusi program. Artinya apabila kita mempunyai fungsi lain selain fungsi utama, maka fungsi lain tersebut baru akan dipanggil pada saat digunakan.

Oleh karena itu bahasa C merupakan bahasa prosedural yang menerapkan konsep runtutan (program dieksekusi per baris dari atas ke bawah secara berurutan), maka apabila kita menuliskan fungsi-fungsi lain tersebut dibawah fungsi utama, maka kita harus menuliskan bagian prototipe (*prototype*), hal ini dimaksudkan untuk mengenalkan terlebih dahulu kepada kompiler daftar fungsi yang akan digunakan di dalam program. Namun apabila kita menuliskan fungsi-fungsi lain tersebut diatas atau sebelum fungsi utama, maka kita tidak perlu lagi untuk menuliskan bagian prototipe diatas.

14. *Visual Code Studio*

Visual Studio Code adalah aplikasi code editor buatan *Microsoft* yang dapat dijalankan di semua perangkat desktop secara gratis. Kelengkapan fitur dan ekstensi membuat code editor ini menjadi pilihan utama para pengembang. *Visual Studio Code* bahkan mendukung hampir semua sistem operasi seperti *Windows, Mac OS, Linux*, dan lain sebagainya.



Gambar 2. 10 *Visual studio code*

15. Cardova

Apache Cordova adalah *framework open source* yang memungkinkan pengembang untuk mengembangkan aplikasi mobile menggunakan teknologi *web* standar seperti HTML, CSS dan *JavaScript*. Cardova bertindak sebagai jembatan antara aplikasi web dan perangkat keras yang ada di perangkat mobile.



Gambar 2. 11 Cardova

16. HTML (*HyperText Markup Language*)

(Abdul Kadir, 2018), HTML 5 adalah kepanjangan dari *HyperText Markup Language* versi 5, merupakan *HTML* baru penerus dari HTML 4, dan XHTML*1*. HTML*i* dikembangkan oleh badan lain yaitu WHATMG *(Web Hypertext Application Technology Working Group)* merupakan pengembangan bahasa *HTML* yang lebih baik, lebih berarti atau semantik *(semantic meaning*) yang

sebelumnya adalah bahasa markup sederhana menjadi sebuah *flatform* canggih, penuh *fitur* dan kaya akan antarmuka pemrograman aplikasi yang disebut *API (Application Programming Interface).* HTML5 menawarkan berbagai fitur menarik yang tidak didukung oleh *HTML* sebelumnya antara lain sebagai berikut :

a. *Canvas*. Memungkinkan pembuatan gambar dalam kanvas. Jadi, gambar tidak lagi diambil dari gambar utuh, melainkan bisa disusun sendiri, bak menggunakan *program paint*.
b. *Header*. Ber guna untuk menyatakan suatu judul, yang bisa diisi dengan logo dan nama perusahaan.
c. *Footer.* Merupakan kebalikan dari *header*. Sebagai catatan kaki, elemen ini berguna untuk menaruh informasi di bagian bawah halaman *Web*.
d. *Time*. Elemen ini berguna untuk menyajikan informasi tentang waktu.
e. *Audio.* Memungkinkan penyajian player untuk memutar suar.
f. *Video.* Memungkinkan player untuk memainkan video.



Gambar 2. 12 HTML

17. Javascript

(Wahana Komputer, 2017), *JavaScript* merupakan salah satu bahasa *script Web*site yang paling banyak digunakan untuk menambah manipulasi *script* HTML dari CSS pada sisi *clie*

nt/browser. *JavaScript* mampu memberikan *fungsionalitas* lebih pada *Website*, seperti *validasi form*, berkomunikasi dengan *server*, serta membuat *Website* lebih *interaktif* dan *animatif*.

JavaScript digunakan pada banyak browser, seperti *Internet Explorer, Firefox, Chrome, Opera, Safari*, dan sebagainya. Hampir seluruh browser mendukung *JavaScript* sehingga tidak perlu khawatir kode *JavaScript* yang digunakan pada *Website* tidak berfungsi.

18. *HypertText Preprocessor*

Hypertext Preprocessor adalah sebuah bahasa pemrograman *server side scripting* yang bersifat *open source*. Sebagai sebuah *scripting language*, *PHP* menjalankan instruksi pemrograman saat proses r*untime*. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. *PHP* merupakan bahasa pemrograman *server-side*, maka *script* dari *PHP* nantinya akan diproses di *server*. Jenis *server* yang sering digunakan bersama dengan *PHP* antara lain *Apache, Nginx*, dan *LiteSpeed*. Selain itu, *PHP* juga merupakan bahasa pemrograman yang bersifat *open source*. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka.



Gambar 2. 13 *HyperText Preprocessor*

19. MySQL

MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis *website*. *MySQL* dibagi menjadi dua lisensi, pertama adalah *Free Software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya.

MySQL termasuk ke dalam RDBMS *(Relational Database Management Sistem)*. Sehingga, menggunakan tabel, kolom, baris, di dalam struktur *database* - nya. Jadi, dalam proses pengambilan data menggunakan metode relational *database*. Dan juga menjadi penghubung antara perangkat lunak dan *database* server.

Secara garis besar, fungsi dari *MySQL* adalah untuk membuat dan mengelola *database* pada sisi server yang memuat berbagai informasi dengan menggunakan bahasa *SQL*. Fungsi lain yang dimiliki adalah memudahkan pengguna dalam mengakses data berisi informasi dalam bentuk *String* (teks), yang dapat diakses secara personal maupun publik dalam web.

Hampir seluruh penyedia server web atau *host* menyediakan fasilitas untuk *MySQL* dalam pengembangan aplikasi berbasis *website* untuk dikelola oleh *web developer*. Kemudian, antarmuka dari *MySQL* adalah *PHPMyAdmin*. Yang berfungsi untuk menghubungkan antara bahasa pemrograman *PHP* dengan *MySQL* untuk proses pengelolaan basis data pada *web*.



Gambar 2. 14 *MySQL*

20. Bahasa Pemrograman Java

Bahasa pemrograman Java adalah bahasa pemrograman yang sangat populer dan luas digunakan untuk pengembangan berbagai jenis perangkat lunak, termasuk aplikasi Android. Berikut adalah beberapa poin penting tentang pengertian bahasa pemrograman Java yang digunakan untuk membuat aplikasi Android:

a. Objek-Oriented: Java adalah bahasa pemrograman berorientasi objek, yang berarti pemrograman dilakukan dengan membuat objek yang memiliki atribut dan perilaku tertentu.
b. Platform-Independent: Java dirancang agar dapat berjalan di berbagai platform, termasuk Android. Kode Java dikompilasi menjadi *bytecode* yang dapat dijalankan di mesin virtual Java (JVM), sehingga aplikasi Java dapat dijalankan di berbagai sistem operasi.
c. *Widely Used*: Java adalah salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Kebanyakan aplikasi Android dikembangkan menggunakan Java, baik untuk Android *Native Development Kit* (NDK) maupun Android *Software Development Kit* (SDK).
d. Android *Development*: Untuk pengembangan aplikasi Android, Java

digunakan bersama dengan Android SDK. Android SDK menyediakan berbagai pustaka dan alat yang diperlukan untuk membangun aplikasi Android, dan Java digunakan untuk menulis kode aplikasi itu sendiri.

e. Fitur-fitur Khusus Android: Java untuk pengembangan Android memiliki beberapa fitur dan pustaka khusus yang disediakan oleh Android SDK, seperti komponen UI (*User Interface*) seperti *Activities* dan *Fragments*, pengelolaan siklus hidup aplikasi, akses ke layanan sistem Android, dan lain sebagainya.

21. *Sotfware Development Kit* (SDK)

SDK atau *Sotfware Development Kit* adalah subuah library pengembangan software yang digunakan untuk mengembangkan atau memebuat aplikasi untuk paket *software* tertentu, *software framework*, *hardware platfrom*, sistem komputer konsul *video game*, sistem operasi atau platfrom sejenisnya. SDK laksana kotak alat lengkap yang berisi berbagai komponen penting, seperti:

a. Pustaka: Kumpulan kode pra-tertulis yang menyediakan fungsi dan fitur umum yang dapat digunakan *developer* untuk menghemat waktu dan tenaga.

b. Alat Bantu: Beragam program yang membantu *developer* dalam proses pengembangan, seperti debugger, compiler, dan emulator.

c. Dokumentasi: Panduan dan petunjuk penggunaan yang lengkap untuk membantu *developer* memahami dan menggunakan SDK dengan efektif.

d. Contoh Kode: Cuplikan kode yang menunjukkan cara menggunakan SDK untuk berbagai tugas, membantu *developer* mempelajari cara kerjanya.

Manfaat Menggunakan SDK:

a. Mempercepat Pengembangan: Menyediakan komponen siap pakai dan dokumentasi yang lengkap, sehingga *developer* dapat fokus pada logika aplikasi dan menghemat waktu pengembangan.
b. Meningkatkan Konsistensi: Memastikan aplikasi yang dibangun mengikuti standar dan praktik terbaik platform, menghasilkan aplikasi yang stabil dan mudah dipelihara.
c. Memudahkan Integrasi: Menyediakan akses ke fitur dan layanan platform, memungkinkan *developer* untuk mengintegrasikan aplikasi mereka dengan mulus ke dalam ekosistem platform.
d. Meningkatkan Kualitas: Alat bantu dan dokumentasi yang lengkap membantu *developer* dalam menulis kode yang lebih bersih, teruji, dan bebas bug.

22. JDK (*Java Development Kit*)

JDK atau *Java Development Kit* adalah sebuah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *java* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *java Runtime Environment* (JRE), JDK menyimpan berbagai komponen penting yang bekerja sama untuk menghadirkan pengalaman pengembangan Java yang mulus dan efisien.

Komponen Utama JDK:

a. Java Compiler: Sang maestro di balik transformasi kode *Java* menjadi bytecode, format yang dapat dipahami dan dieksekusi oleh *Java* Virtual Machine (JVM).

b. *Java Virtual Machine* (JVM): Mesin ajaib yang mengubah *bytecode* menjadi instruksi yang dapat dijalankan oleh sistem operasi, memungkinkan aplikasi java berjalan di berbagai platform.
c. *Java Runtime Environment* (JRE): Subset dari JDK yang berisi JVM dan pustaka inti Java, memungkinkan aplikasi Java untuk dieksekusi tanpa memerlukan JDK lengkap.
d. *Documentation*: Panduan lengkap dan tutorial yang menjelaskan detail bahasa pemrograman *Java*, API, dan alat pengembangan yang disertakan dalam JDK.
e. *Tools*: Beragam program yang membantu developer dalam proses pengembangan, seperti debugger, profiler, dan alat dokumentasi.
f. *Libraries*: Kumpulan kode siap pakai yang menyediakan fungsionalitas umum dan teruji, membantu *developer* menghemat waktu dan meningkatkan produktivitas.

Manfaat Utama JDK:

a. Membangun Aplikasi Java yang Kuat: JDK menyediakan semua alat yang diperlukan untuk membangun aplikasi Java yang skalabel, aman, dan berkinerja tinggi.
b. Mempercepat Pengembangan: Pustaka siap pakai dan alat bantu yang intuitif membantu *developer* menulis kode lebih cepat dan efisien.
c. Meningkatkan Kualitas Kode: Alat bantu seperti compiler dan debugger membantu *developer* dalam menghasilkan kode yang bersih, teruji, dan bebas bug.

d. Memudahkan Integrasi: JDK menyediakan akses ke berbagai API dan layanan, memungkinkan *developer* untuk mengintegrasikan aplikasi mereka dengan platform dan sistem eksternal.
e. Dukungan Platform Luas: Aplikasi Java yang dibangun dengan JDK dapat berjalan di berbagai platform, termasuk *Window*s, *macOS*, *Linux*, dan *Android*.

Meskipun sering disalah artikan sebagai hal yang sama, JDK dan JRE memiliki perbedaan penting. JDK adalah seperangkat lengkap alat pengembangan yang diperlukan untuk membangun aplikasi Java, sedangkan JRE adalah subset dari JDK yang hanya berisi komponen yang diperlukan untuk menjalankan aplikasi Java.

JDK sangat penting bagi developer yang ingin:

a. Membangun aplikasi *Java* dari awal: JDK menyediakan semua alat dan sumber daya yang diperlukan untuk memulai pengembangan Java.
b. Mengembangkan applet *Java*: JDK memungkinkan developer untuk membuat applet Java yang dapat dijalankan di browser web.
c. Membuat aplikasi web *Jav*a: JDK menyediakan alat dan pustaka yang diperlukan untuk membangun aplikasi web Java yang dinamis dan interaktif.
d. Mengembangkan aplikasi enterprise *Java*: JDK menawarkan fitur-fitur canggih yang diperlukan untuk membangun aplikasi enterprise Java yang skalabel dan aman.

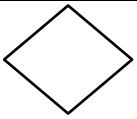
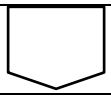
23. *Flowchart*

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program, secara garis besar setiap perancangan *Flowchart* selalu terdiri dari tiga bagian, yaitu input, proses dan output.berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart.*

Salah satu fungsi utama *flowchart* adalah untuk memvisualisasikan proses secara keseluruhan. Dengan mewakili langkah-langkah dan keputusan dalam bentuk diagram, *flowchart* membantu orang untuk memahami urutan tindakan yang diperlukan dalam suatu proses.

Flowchart sangat membantu dalam menggambarkan proses yang kompleks secara visual sehingga lebih mudah dipahami oleh orang-orang yang terlibat. Ini juga memungkinkan untuk mengidentifikasi masalah atau kekosongan dalam proses dengan lebih mudah. Flowchart membantu dalam memahami alur kerja atau proses dengan cara yang jelas dan terstruktur. Ini membuatnya lebih mudah bagi orang untuk memahami langkah-langkah yang diperlukan dalam suatu proses. Representasi visual dari langkah-langkah dalam flowchart membuatnya lebih mudah untuk melihat keseluruhan proses. Ini membantu dalam mengidentifikasi bagian-bagian yang kompleks atau masalah yang mungkin terjadi

Tabel 2. 9 *Symbol flowchart*

No	Simbol	Nama	Fungsi
1		*Terminator*	Permulaan/akhir program
2		Garis Alir (*Flow Line*)	Arah aliran program
3		*Preparation*	Proses inisialisasi program/pemberian nilai awal
4		Proses	Proses perhitungan/proses pengolahan data
5		Input/Output Data	Proses input/output data, parameter, informasi
6		*Predefined process (sub program)*	Permulaan sub program/proses menjalankan sub program
7		*Decision*	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk data langkah selanjutnya
8		*On Page Connect or*	Penghubung bagian-bagian *Flowchart* yang berada pada halaman berbeda
9		*Off Page Connect or*	Permulaan/akhir programan

24. *Firebase*

Google Cloud Messaging (GCM) adalah layanan komunikasi push cloud-to-device, sejak terintegrasi dengan *Firebase* berubah namanya menjadi *Firebase Cloud Messaging* (FCM) biasa disebut *Firebase. Firebase* memiliki beberapa fitur, diantaranya adalah realtime database yang disimpan secara *cloud*, layanan ini menggunakan *Application* program interface (API), data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, apabila ada perubahan pada data yang tersimpan, maka setiap user yang terhubung akan menerima pembaruan data secara otomatis. Format waktu yang dapat

dipergunakan pada *Firebase* adalah *TIMESTAMP* (*time since the Unix epoch*).

Firebase adalah platform pengembangan aplikasi yang dibuat oleh Google. Ini menawarkan berbagai layanan untuk membantu pengembang membangun, meningkatkan, dan mengelola aplikasi *mobile* dan *web*, dengan fokus pada pengembangan yang cepat dan mudah.

Berikut beberapa fungsi utama *Firebase*:

1. *Database Realtime* : *Firebase* menyediakan database *cloud NoSQL* yang dapat disinkronkan secara *real-time* antara pengguna dan perangkat mereka. Ini sangat berguna untuk aplikasi yang memerlukan pertukaran data secara langsung antara pengguna.
2. Autentikasi : *Firebase* menyediakan layanan autentikasi yang aman dengan beberapa metode login seperti *email/password, Google, Facebook*, dan banyak lagi. Ini memungkinkan pengembang untuk dengan mudah mengimplementasikan sistem autentikasi di aplikasi mereka tanpa perlu mengembangkan infrastruktur sendiri.
3. *Hosting* : *Firebase Hosting* memungkinkan pengembang untuk dengan mudah menerapkan situs *web* dan aplikasi *web* statis mereka. Ini menyediakan hosting global cepat dan dapat diandalkan untuk aplikasi *web*.
4. *Cloud Firestore & Cloud Storage* : *Firebase* menyediakan layanan database NoSQL dan penyimpanan file di *cloud. Cloud Firestore* adalah database yang sangat fleksibel dan dapat disesuaikan, sementara *Cloud Storage* memungkinkan penyimpanan dan pengambilan file secara efisien.
5. *Cloud Functions : Firebase Cloud Functions* memungkinkan pengembang

untuk menulis dan mengelola kode backend tanpa harus mengelola infrastruktur server mereka sendiri. Fungsi ini dapat digunakan untuk menangani peristiwa seperti perubahan database, autentikasi pengguna, atau tindakan lainnya.

6. *Analytics* : *Firebase Analytics* memberikan wawasan tentang perilaku pengguna dan kinerja aplikasi. Ini membantu pengembang untuk memahami bagaimana pengguna berinteraksi dengan aplikasi mereka dan mengambil keputusan yang lebih baik dalam pengembangan aplikasi selanjutnya.

7. *Testing dan Monitoring* : *Firebase* menyediakan alat untuk melakukan pengujian aplikasi dan memantau kinerjanya. Ini termasuk uji A/B, pemberitahuan kerusakan, pelaporan bug, dan banyak lagi.

8. *Cloud Messaging : Firebase Cloud Messaging* (FCM) memungkinkan pengembang untuk mengirim pemberitahuan push dan pesan kepada pengguna mereka di semua platform, baik itu *Android*, iOS, atau web.

Fungsi-fungsi ini membuat *Firebase* menjadi platform yang sangat berguna bagi pengembang yang ingin membangun aplikasi mobile dan web yang kuat, aman, dan terukur.

Firebase menyediakan berbagai alat dan layanan yang memungkinkan pengembang untuk membangun dan menerapkan aplikasi dengan cepat. Fitur seperti autentikasi pengguna, penyimpanan data secara , dan hosting web yang dioptimalkan membuat proses pengembangan menjadi lebih efisien.

Firebase Realtime Database adalah layanan *database cloud* yang menyimpan dan mensinkronkan data antara pengguna secara *real-time*. Ini sangat

berguna untuk aplikasi yang memerlukan respons cepat dan sinkronisasi data secara langsung antara pengguna.



Gambar 2. 15 *Firebase*

25. Metode *Black Box*

Metode *black box*, dalam konteks pengujian perangkat lunak, merujuk pada pendekatan di mana pengujian dilakukan tanpa memperhatikan struktur internal atau logika dari perangkat lunak yang diuji. Dalam metode ini, pengujian dilakukan hanya berdasarkan input dan output yang dihasilkan oleh perangkat lunak, tanpa memperhatikan bagaimana perangkat lunak tersebut melakukan proses internalnya. Dalam keseluruhan, metode *black box* adalah pendekatan yang berguna dalam pengujian perangkat lunak yang fokus pada fungsionalitas eksternal dan pengalaman pengguna, tanpa memperhatikan detail internal implementasi perangkat lunak.

Berikut adalah contoh pengujian *blackbox testing study* kasus Pengujian *Black Box* Menyiram tanaman cabai Berbasis *Internet of Things* (IoT). Berikut penjelasan dari tabel uji coba:

Tabel 2. 10 Contoh pengujian black box penyiraman tanaman cabai

No	Uji coba	Hasil	Keterangan
1	Jika keadaan tanah kering maka pompa air akan menyala dan penyiraman dilakukan	sukses	Berhasil karena lampu indikator pada relay channel k1 menyala dan status pada aplikasi menyiram yang berarti perintahberfungsi.



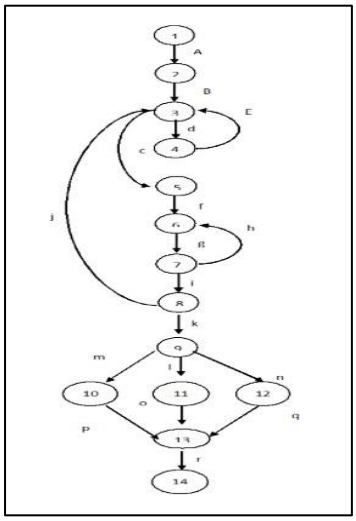
Gambar 2. 16 Contoh *Blackbox Testing* Pengujian Alat

26. Metode *White Box*

Metode *white box*, juga dikenal sebagai pengujian struktural atau pengujian berbasis kode, adalah pendekatan dalam pengujian perangkat lunak di mana pengujian dilakukan dengan memperhatikan struktur internal dan logika dari perangkat lunak yang diuji. Dalam metode ini, pengujian dilakukan dengan memeriksa kode sumber perangkat lunak untuk memastikan bahwa semua jalur kode dieksekusi dan bahwa semua kondisi pengujian terpenuhi. Dalam keseluruhan, metode *white box* adalah pendekatan yang berguna dalam pengujian perangkat lunak yang fokus pada struktur dan logika internal perangkat lunak. Ini memungkinkan pengujian yang lebih menyeluruh dan membantu mengidentifikasi

masalah yang mungkin terjadi dalam kode sumber.

Berikut adalah contoh penguujian *white box* pada Pengujian *White Box* pada Program Monitoring Suhu dan Kelembaban Udara Menggunakan Teknik Basis Path (Nugroho et al., 2018).



Gambar 2. 17 Contoh *White Box Testing Flow Graph*

Setelah flow graph dibuat, selanjutnya adalah perhitungan cyclomatic complexity untuk mendapatkan banyaknya jalur yang dapat dilalui oleh program tersebut. Berikut cara untuk menghitung *cyclomatic complexity* dari *flow graph* diatas: Diketahui jumlah node yang ada adalah 14 dan jumlah edge yang ada adalah 18. Dengan rumus yang telah dijelaskan sebelumnya didapatkan:

V(G) = E – N + 2

V(G) = 18 – 14 + 2

V(G) = 6

Dari perhitungan diatas didapatkan hasil V(G)=6, yang berarti jalur yang dimiliki oleh *flow graph* diatas adalah 6 (enam) jalur. Dapat diketahui jalur yang dapat dilalui program tersebut adalah sebagai berikut:

Path 1 = 1-2-3-5-6-7-8-9-10-13-14

Path 2 = 1-2-3-5-6-7-8-9-11-13-14

Path 3 = 1-2-3-5-6-7-8-9-12-13-14

Path 4 = 1-2-3-4-3-5-6-7-8-9-10-13-14

Path 5 = 1-2-3-5-6-7-6-7-8-9-11-13-14

Path 6 = 1-2-3-5-6-7-8-3-5-6-7-8-9-12-13-14

Tabel 2. 11 *Graph Matric*

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		a												
2			b											
3				d	c									
4			e											
5						f								
6							g							
7						h		i						
8			j						k					
9										m	l	n		
10													p	
11													o	
12													q	
13														r
14														

Tabel 2. 12 *Connection Matric*

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Koneksi
1		1													=1-1=0
2			1												=1-1=0
3				1	1										=2-1=1
4			1												=1-1=0
5						1									=1-1=0
6							1								=1-1=0
7						1		1							=2-1=1
8			1						1						=2-1=1
9										1	1	1			=3-1=2
10													1		=1-1=0
11													1		=1-1=0
12													1		=1-1=0
13														1	=1-1=0
14															

27. UML *(Unified Modeling Language)*

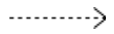
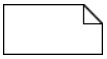
Unified Modeling Language adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. *UML* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumen tasikan sistem piranti lunak.

UML merupakan metodologi untuk mengembangkan sistem OOP dan sekelompok tool untuk mendukung pengembangan sistem tersebut. *UML* mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar *OOP* sejak tahun 1980-an. Seperti bahasa - bahasa lainnya, *UML* mendefinisikan notasi dan Syntax/semantik. Setiap bentuk memiliki makna tertentu, dan *UML* Syntax mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari 3 notasi yang telah ada sebelumnya : *Grady Booch OOD (Object – Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique)* dan Ivar Jacobson *OOSE (Object – Oriented Software Engineering). UML* memungkinkan pengguna untuk memodelkan struktur sistem perangkat lunak, termasuk kelas, objek, antarmuka, dan hubungan antara entitas-entitas tersebut. Adapun daftar simbol *UML* yaitu :

Tabel 2. 13 *Symbol Use Case* Diagram

No	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.

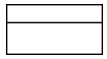
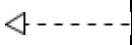
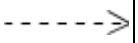
Lanjutan Tabel 2. 14

No	GAMBAR	NAMA	KETERANGAN
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*Independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak *(Descendent*) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (*Ancestor*).
4		*Include*	Menspesifikasikan bahwa Use Case sumber secara Eksplisit.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen- elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 15 *Symbol Class* Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

Lanjutan Tabel 2.14

No.	GAMBAR	NAMA	KETERANGAN
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan Antara objek satu dengan objek lainnya

Tabel 2. 16 *Symbol Squence* Diagram

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi. untuk menunjukkan kapan suatu objek atau entitas terlibat dalam proses dan kapan mereka tidak aktif.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi untuk menunjukkan bagaimana objek berinteraksi satu sama lain dan bertukar informasi selama jalannya proses.

Tabel 2. 17 *Symbol State Chart* Diagram

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.

Lanjutan Tabel 2.16

No	GAMBAR	NAMA	KETERANGAN
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai Atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 18 *Symbol Activity* Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing- masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

B. Kajian Hasil Penelitihan Terdahulu

1. (Yusman et al., 2019)"Rancang Bangun *Smart Home* pada Keamanan Pintu menggunakan Sensor *PIR* Berbasis *Website".* Penelitian ini berfokus pada pemanfaatan sensor PIR untuk mendeteksi gerakan mencurigakan di sekitar pintu, dan juga dapat dimonitoring dari jarak jauh melalui sebuah website. Selain itu, sistem ini juga memanfaatkan Keypad 4x4 untuk akses pintu.

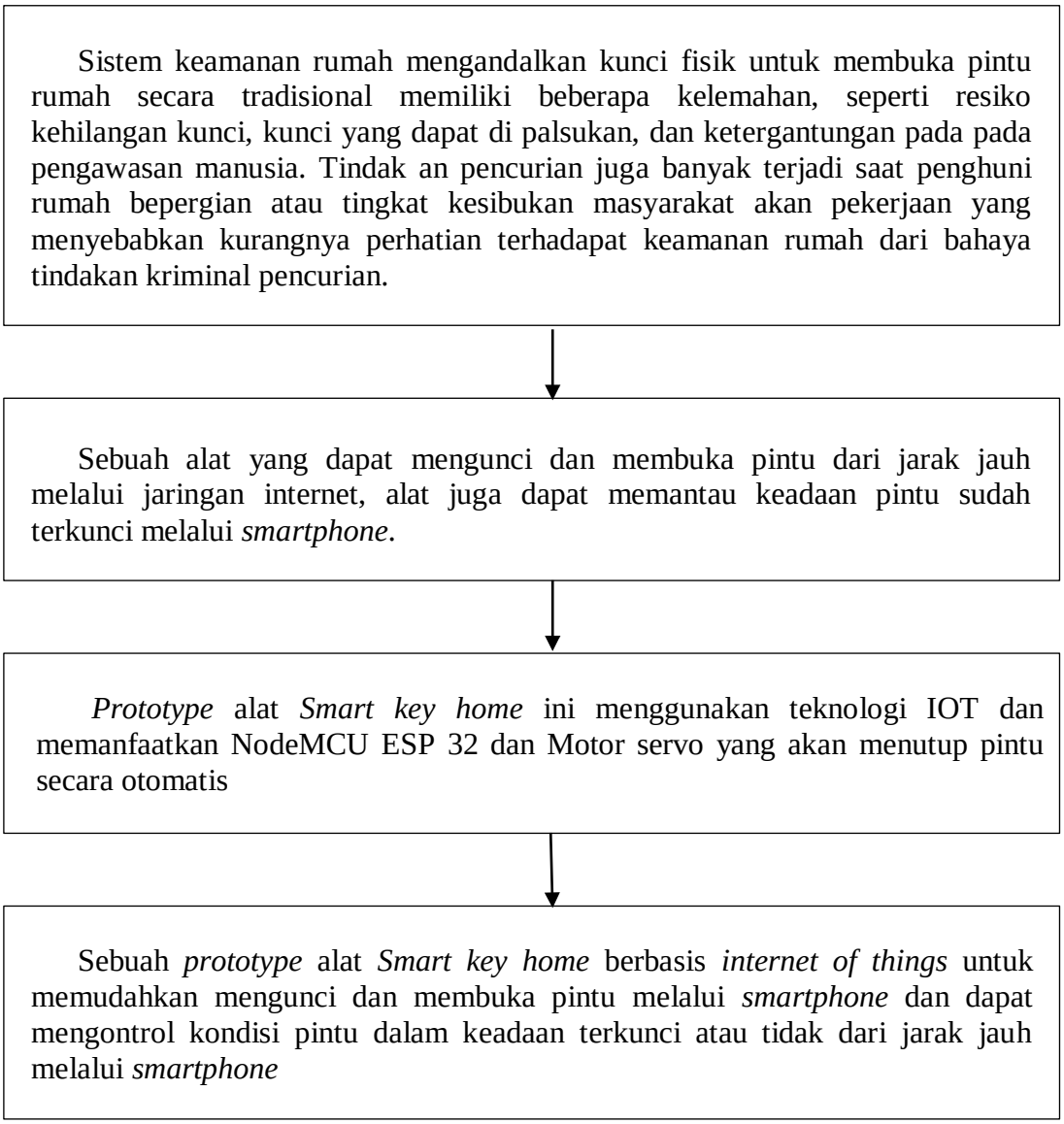
Dengan demikian, sistem ini dapat di monitor dari jarak jauh, sehingga pemilik rumah dapat dengan mudah memantau keamanan rumah melalui aplikasi *website.* Selain itu, sistem ini juga akan dilengkapi dengan alarm darurat yang akan memberikan notifikasi ke aplikasi Telegram jika terjadi situasi darurat, seperti pembobolan pintu.

2. (Novianti, 2019)"Rancang Bangun Pintu Otomatis dengan Menggunakan RFID "Penelitian ini menggunakan metode riset dan pengembangan,yaitu metode yang bertujuan menghasilkan atau mengembangkan produk tertentu. Metode ini diterapkan pada prosedur penelitian menjadi 9 tahap yaitu : (1) mulai, (2)potensi masalah, (3)pengumpulan informasi, (4)perencangan alat, (5)desain alat, (6)pembuatan alat, (7) uji coba alat, (8) pengumpulan data, (9) analisis data. Dengan adanya masalah ini membuat Tugas Akhir dengan Judul "Membuat pintu otomatis menggunakan *RFID* melalui *Arduino*". Sehingga alat bantu ini bisa mendapat jaminan keamanan. Adapun Tujuan dari Penelitian ini yaitu: merancang dan membangun pintu otomatis menggunakan sensor *RFID* serta membuat sistem keamanan yang efektif dari pintu otomatis dengan sensor *RFID.*
3. (Fathoni & Khotimah, 2023)"Rancang bangun *smart home* berbasis IoT menggunakan *bot* Telegram dan modul NodeMCU ESP 32. Prototipe ini berupa sebuah saklar yang dapat dikendalikan jarak jauh dengan memanfaatkan fitur *chat* pada Telegram. Modul ini diintegrasikan melalui jaringan *wi-fi* yang tersedia pada sebuah rumah untuk mendapatkan akses internet. Prototipe ini dapat dihubungkan dengan peralatan kelistrikan rumah

tangga seperti lampu, kipas angin, AC atau peralatan elektronik lainya agar dapat dikendali jarak jauh. Setelah selesai dibuat, sistem ini kemudian diuji kepada narasumber ahli untuk mengetahui kinerja alat yang dibuat serta pengujian tingkat kelayakan alat.

4. (Amini et al., 2021) " Perancangan keamanan ruangan dengan sensor pir dan *magnetic door switch* berbasis web" Sistem kamanan ruangan pada saat ini masih menggunakan kunci pintu biasa dan tidak memiliki sebuah pesan atau notifikasi jika adanya seseorang membuka pintu ataupun memasuki ruangan tanpa melapor ke petugas keamanan. Sebuah sistem keamanan dengan menggunakan sensor *Passive Infrared* dan *Magnetic Door Switch* yang dapat mendeteksi jika seseorang membuka pintu dan adanya gerakan didalam ruangan dapat mengatasi masalah tersebut. Sistem keamanan ruangan ini nantinya dapat memberikan sebuah notifikasi berupa email dan memberikan peringatan berupa alarm jika pintu ruangan terbuka.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian *eksperimen* Menurut Arboleda, penelitian *eksperimen* adalah penelitian di mana peneliti dengan sengaja melakukan manipulasi terhadap satu atau lebih *variabel* dengan suatu cara yang dapat mempengaruhi *variabel* tersebut.

Menurut Kerlinger, penelitian eksperimen adalah penelitian di mana peneliti melakukan manipulasi dan kontrol terhadap satu atau lebih *variabel* bebas sekaligus pengamatan terhadap *variabel-variabel* lain yang terikat untuk menemukan variasi yang muncul karena adanya manipulasi tersebut.

B. Waktu Dan Tempat Penelitian

Penelitian dilakukan ±2 bulan dan lokasi penelitian dilakukan di rumah Jln. Pendidikan, Kec.Tiroang, Kota Pinrang.

C. Alat dan bahan penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan alat dan bahan guna mendukung kegiatan penelitian tersebut. Alat dan bahan yang digunakan sebagai berikut:

1. Perangkat keras

Perangkat keras yang digunakan untuk membuat *prototype* rancang Bangun *Smart Key Home berbasis* IOT dapat dilihat pada tabel berikut:

Tabel 3. 1 Spesifikasi Perangkat Keras

NO	Spesifikasi	
1.	Merk laptop	Laptop *Lenovo*
2.	Processor Laptop	Intel Celeron N4000 COMPUTE CORES 2.6 GHz
3.	RAM Laptop	4,00 GB (HDD 1TB)
4.	Mikrokontoroller	ESP32 (NodeMCU)
5.	Perangkat Tambahan	Motor servo,Magnetic door switch,kabel jumper, keypad,relay,solenoid look door,adaptor,buzzer, breadbroard

2. Perangkat lunak

Perangkat lunak yang digunakan untuk membuat prototipe rancang bangun *smart key home* berbasis iot dapat dilihat sebagai berikut:

Tabel 3. 2 Spesifikasi Perangkat Lunak

No.	Spesifikasi	
1	Sistem operasi	Windows 10
2	Tool Pemrograman	*Arduino* IDE, Visual Code Studio
3	Bahasa Pemrograman	C
4	*Database*	*Firebase*

Alat dan bahan yang digunakan dalam pembuatan *prototype* alat *smart key home* berbasis *internet of things* yaitu :

1. Alat
    a. NodeMCU Esp32
    b. Motor servo
    c. Kabel *jumper*
    d. Relay
    e. Adaptor

f. *Beardboard*

g. Solenoid door lock

h. Keypad

i. Buzzer

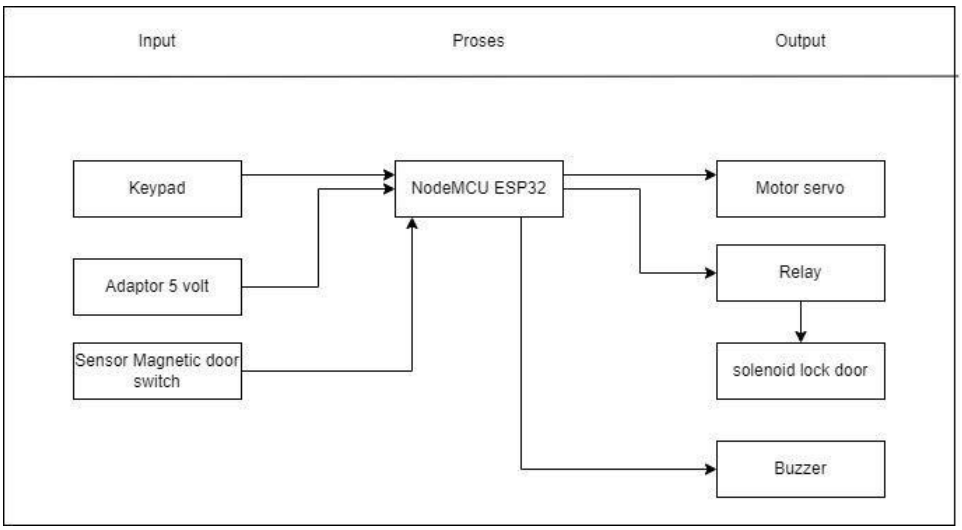
j. Sensor *magnetic door switch*

2. Bahan

Data yang digunakan dalam penelitian ini berasal dari hasil upaya pengumpulan data para peneliti, yang kemudian akan dilakukan analisis tambahan untuk landasan perancangan sistem.

D. Perancangan Sistem

Perancangan alat *smart key home* berbasis *internet of things* meliputi perancangan perangkat keras dan perancangan perangkat lunak. Adapun blok diagram yang akan dibuat dapat dilihat pada gambar dibawah ini.



Gambar 3. 1 Blok diagram *system*

Komponen-komponen pada blok diagram tersebut, antara lain:

1. Motor servo

   Motor servo digunakan untuk menggerakan mekanisme pintu,sehingga dapat membuka atau menutup pintu secara otomatis berdasarkan perintah dari sistem.

2. *keypad*

   keypad yang berfungsi untuk membuka pintu menggunakan *password* secara langsung di pintu alat.

3. Relay V5 2 *channel*

   Berfungsi untuk mengatur aliran listrik kepada *solenoid door lock*.

4. NodeMCU ESP32

   Berfungsi sebagai mikrokontroler untuk memproses input dari sensor yang kemudian menghasilkan output seperti yang diinginkan.

5. *Solenoid lock door*

   Berfungsi sebagai pengunci pintu secara elektronik

6. Adaptor 5 *volt*

   Berfungsi sebagai tegangan listrik yang terhubung ke *NodeMCU* ESP32

7. Sensor *magnetic door switch*

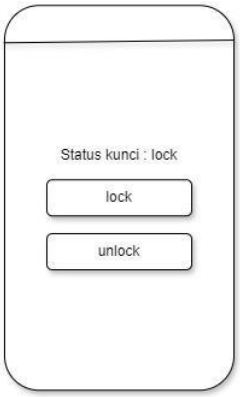
   Untuk mendeteksi status terbuka atau tertutupnya pintu yang dilengkapi dengan bagian magnet

8. *Buzzer*

   Berfungsi sebagai indicator bahwa proses telah selesai atau terjadi suatu kesalahan input.

9. Desain tampilan aplikasi

Aplikasi *i*ni direncanakan dibuat menggunakan *visual code studio,* dengan dibuatnya aplikasi ini untuk memantau keadaan kunci rumah dari jarak jauh. Perancangan tampilan aplikasi ini akan dibuat seperti pada gambar dibawah ini.



Gambar 3. 2 Desain tampilan aplikasi

E. Metode Pengumpulan Data

Metode pengumpulan data yang tepat yaitu dengan mempertimbangkan penggunaannya berdasarkan jenis data dan sumbernya. Data yang *objektif* dan relevan dengan pokok permasalahan penelitian merupaka indikator keberhasilan suatu penelitian. Pengumpulan data penelitian ini dilakukan dengan cara sebagai berikut:

1. Observasi

Merupakan metode pengumpulan data dengan cara mengadakan pengamatan langsung kepada objek penelitian yaitu dengan mengunjungi dan mengamati alat *smart key home* berbasis iot atau yang sudah ada sebeleumnya.

2. Wawancara

Merupakan teknik pengumpulan data dengan cara mengadakan tanya jawab atau wawancara langsung kepada narasumber pada penelitian ini, peneliti

melakukan wawancara langsung kepada pembuat alat *smart key home* berbasis *internet of things.*

3. Studi pustaka

Mengumpulkan data dengan mempelajari masalah yang berhubungan dengan objek yang diteliti, bersumber dari buku-buku pedoman, literatur yang disusun oleh para ahli untuk melengkapi data yang diperlukan dalam penelitian baik secara *offline* maupun *online*.

F. Metode Pengujian

Setelah perangkat yang dibutuhkan telah terpasang, selanjutnya dilakukan pengujian alat *smart key home* berbasis *internet of things* sebagai pengujian sistem secara keseluruhan. Apabila terdapat masalah pada pengujian sistem yang telah dibuat maka akan dilakukan identifikasi *erorr* dan melakukan perancangan dan perakitan ulang. Selain pengujian di atas dilakukan pula pengujian *black box* dan juga pengujian *white box.*

1. *Blackbox testing*

Blackbox testing yaitu untuk menguji program alat yang dibuat apakah sudah berhasil atau tidak.

2. *Whitebox testing*

Whitebox testing yaitu untuk mengetahui struktur dan alur apakah prototipe alat *smart key home* berbasis *internet of things* apakah sudah sesuai dengan ketentuan kemudian aplikasi tersebut untuk memperoleh hasil.

BAB IV

HASIL DAN PEMBAHASAN

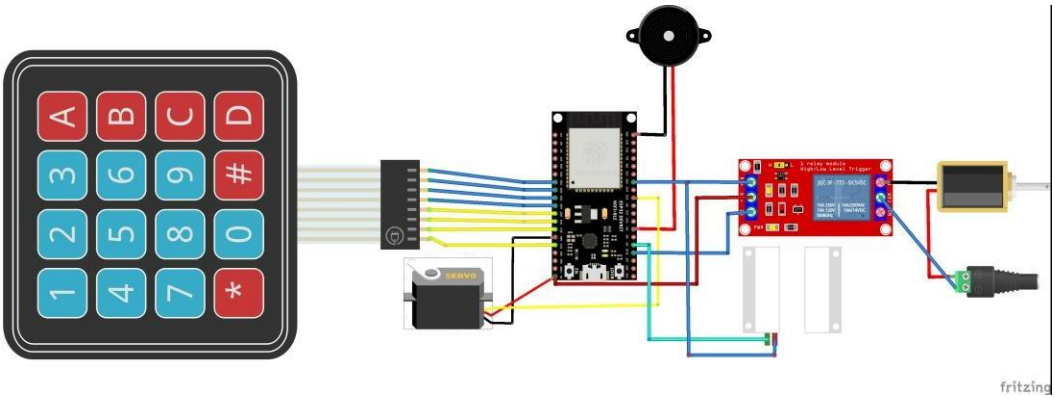
A. Perancangan sistem

Pada penjelasan hasil dan analisi, penulis menjelaskan tentang beberpa komponen yang digunakan, dalam hal ini penulis menjelaskan tentang prinsip kerja komponen yang digunakan, bagaimana menghubungkan komponen yang satu dengan yang lain agar dapat menghasilakna suatu sistem yang dirancangkan sesuai apa yang menjadi rancangan penulis, pada penulis ini membagi 2 (dua) pokok pembahasan dan menganalisa secara mendalam baik secara *hardware* (perangkat keras) dan *software* (perangkat lunak), dimana dapat dijelaskan sebagai berikut.

1. Perangkat perancangan keras (*Hardware*)

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektroniknya. Rancangan ini menjelaska beberapa rancanganan komponen yang digunakan dan pengkoneksian komponen dengan searching beberapa referensi agar dapat memberikan tingkat akurasi. Hasil ini dimaksud agar sistem rancang bangun *smart key home* berasis *internet of things* dapat berjalan sesuai dengan deskripsi awal yang telah dirancanakan.

Adapun uraian masing-masing perangkat keras yang digunakan sebagai berikut:



Gambar 4. 1 Skema Keseluruhan Rangkaian Alat

Pada gambar diatas merupakan skema keseluruhan rangkaian alat yang menunjukan sistem *Smart Key Home Berbasis Internet Of Things* yang menggunakan *NodeMCU* ESP32, motor servo, *buzzer*, sensor *magnetic switch*, dan *relay* sebagai komponen utama sistem sebagai saklar penghubung *solenoid lock door*. Untuk lebih jelas berdasarkan fungsinya sebagai berikut ini:

1. ESP32 sebagai pengendali utama pada sistem Alat *smart key home* berbasis *internet of things*, tugasnya menganalisis data inputan yang berasal dari keypad dengan cara memberi output kepada *relay* dan *buzzer*.

2. *Keypad* sebagai perangkat input yang terdiri dari tombol huruf, angka, dan karakter. Fugsinya untuk memasukkan inputan pinkepada ESP32.

3. *Relay* sebagai penghubung arus listrik untuk menjalankan solenoid. *Relay* dikontrol langsung oleh ESP32.

4. *Buzzer* sebagai media pemberitahuan berupa alarm. Fungsinya untuk memberitahukan *user* bahwa *password* yang di masukkan salah atau salah.

5. Motor Servo Berfungsi untuk membuka dan menutup pintu secara otomatis berdasarkan perintah dari sistem

6. *Solenoid door lock* sebagai alat yang digunakan untuk membuka dan mengunci pintu secara otomatis ketika di perintahkan oleh ESP32 melalui *relay*.

7. Sensor *manetic switch* sebagai sensor status yang bisa dikirim ke *firebase* untuk memantau dari jarak jauh.

8. Konektor berfungsi untuk menyalurkan daya Listrik

Tabel 4. 1 Keterangan Pin Keypad ke ESP32

No.	Pin Keypad	NodeMCU ESP32
1	C4	D13
2	C3	D12
3	C2	D14
4	C1	D27
5	R4	D26
6	R3	D25
7	R2	D33
8	R1	D32

Tabel 4. 2 Keterangan Pin *Relay* ke ESP32

No	Pin Relay	Nodemcuesp32
1	VCC	VIN ESP32
2	GND	GND ESP32
3	IN	D15 ESP32

Tabel 4. 3 Keterangan Pin Relay ke *Solenoid lock door*

No	Pin Relay	Pin Solenoid Lock Door
1	NO	GND

Tabel 4. 4 Keterangan Pin *Relay* ke Konektor

No	Pin Relay	Konektor
1	COM	GND

Tabel 4. 5 Keterangan Pin *Solenoid Lock Door* ke Konektor

No	Pin solenoid lock door	Konektor
1	VIN	VIN

Tabel 4. 6 Keterangan Pin *Buzzer* ke ESP32

NO	Pin *Buzzer*	Pin ESP32
1	I/O	Pin D04
2	GND	GND

Tabel 4. 7 Keterangan Pin Motor servo ke ESP32

NO	Pin Motor servo	Pin esp32
1	VCC	VIN
2	GND	GND
3	GPIO	D18

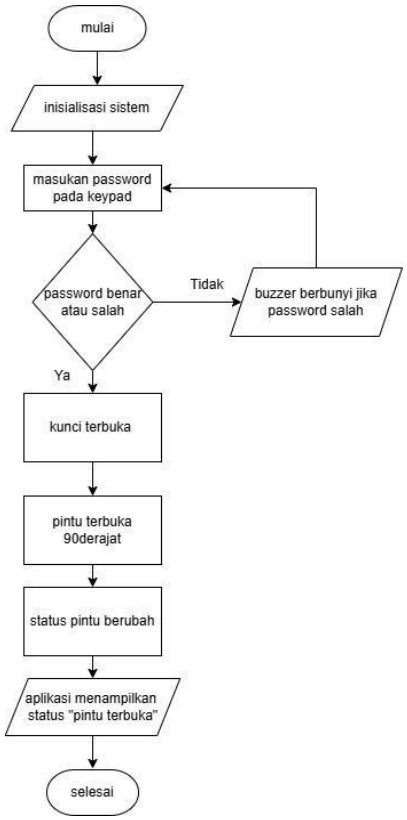
Tabel 4. 8 Keteranaga Pin Sensor *magnetic switch* ke ESP32

NO	Pin Sensor *magnetic switch*	Pin esp32
1	VCC	VIN
2	GPIO	D02

2. Perancangan perangkat lunak *(software*)

a. *Flowchart*

1. *Flowchart* Membuka Menggunakan *Keypad*



Gambar 4. 2 *Flowchart* Membuka Menggunakan *Keypad*

Pada gambar 4.2 merupakan perancangan perangkat lunak yag dimulai dari menginisialisasi semua komponen (ESP32, *Keypad, Servo, Solenoid*) Pengguna memasukan *password* melalui *keypad* kemudian ESP32 membaca *password* jika pin benar maka *solenoid* terbuka dan pintu terbuka 90 derajat, jika *password* salah maka *buzzer* akan berbunyi dan kembali memasukan *password* pada *keypad* kemudian status pintu berubah di aplikasi android.

2. Flowchart Menutup Menggunakan Keypad



Gambar 4. 3 Menutup Pintu Menggunakan *Keypad*

Pada gambar 4.3 merupakan perancangan perangkat lunak yag dimulai dari menginisialisasi semua komponen (ESP32, *Keypad, Servo, Solenoid*), kemudian pengguna menekan satu tombol tutup pada *keypad* untuk menutup pintu setelah *solenoid* mungunci pintu servo akan menutup pintu dan status diperbarui ke “Pintu tertutup” kemudian akan ditampilkan di aplikasi .

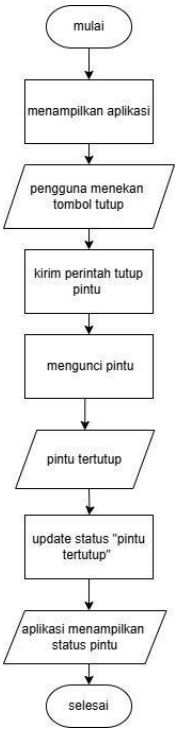
3. *Flowchart* Membuka Pintu Menggunakan Aplikasi *Android*



Gambar 4. 4 *Flowchart* Membuka Pintu Menggunakan Aplikasi *Android*

Pada gambar 4.4 *Flowchart* membuka pintu menggunakan aplikasi android yang dimulai dari menampilkan aplikasi kemudian pengguna menekan tombol buka pada aplikasi, lalu aplikasi mengirimkan sinyal atau perintah ke sistem pintu untuk membuka kunci pintu dan pintu terbuka 90 derajat dan aplikasi akan menampilkan dan mempembarui status pintu menjadi ”pintu terbuka”.

4. *Flowchart* menutup pintu menggunakan aplikasi android



Gambar 4. 5 Flowchart menutup pintu menggunakan aplikasi

Pada gambar 4.5 *Flowchart* menutup pintu menggunakan aplikasi android yang dimulai dari menampilkan aplikasi di layar *smartphone* pengguna. kemudian pengguna menekan tombol tutup pintu aplikasi, lalu aplikasi mengirimkan perintah kesistem pintu untuk menutup dan mengunci pintu setelah pintu tertutup sepenuhnya maka aplikasi akan menampilkan dan mempembarui status pintu menjadi ”pintu tertutup”.

b. Pembuatan Software

Aplikasi IDE (*Integrate Deplopment Environmen*) Arduino adalah software yang digunakan sebagai media untuk memprogram board Arduino. Arduino IDDE adalah *software* yang digunakan untuk membuat sketsa pembrograman atau istilah lain Arduino IDE sebagai media pembrograman pada board yang ingin deprogram. Arduino IDE bermanfaat untuk mengedit membuat,

mengunggah ke papan terperinci, dan mengkodekan paket tertentu. Arduino IDE dibuat dari Bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++ (wiring), yang membuat operasi enter/output lebih mudah. Didalam program Arduino terdapat *library* dan coding program berikut. Berikut penerapan codingannya:

1. Deklarasi library

Bagian ini mengimpor Pustaka yang diperlukan untuk perangkat keras yang digunakan

a. **Firebase.h** : digunakan untuk berkemunikasi dengan *firebase realtime* database.
b. **Keypad.h** : digunakan untuk membaca input dari keypad 4x4
c. **ESP32Servo.h** : digunakan untuk mengendalikan motor servo yang menggerakan pintu
d. **Preferensi.h** : digunakan untuk menyimpan kata sandi secara permanen di memori ESP32
e.

```
#include <Keypad.h>
#include <ESP32Servo.h>
#include <Preferences.h>  // Library untuk penyimpanan di ESP32
```

2. Kongfigurasi Koneksi

Kode berikut digunakan untuk menghubungkan ESP32 ke jaringan WiFi dan Firebase:

a. **WIFI_SSID** adalah nama jaringan WiFi yang akan digunakan ESP32 untuk terhubung ke internet.
b. **WIFI_PASSWORD** adalah kata sandi dari jaringan WiFi yang digunakan.
c. **REFERENCE_URL** adalah URL dari *Firebase Realtime Database* tempat ESP32 akan mengirim dan menerima data.
d. **AUTH_TOKEN** adalah kunci autentikasi *Firebase* yang digunakan untuk mengakses database dengan izin tertentu.

```
#define WIFI_SSID "faizfauzan"
#define WIFI_PASSWORD "nio12345"
#define REFERENCE_URL "https://smartkeyhome-d8928-default-rtdb.firebaseio.com/"
#define AUTH_TOKEN "COeBhNCTzT3NuZWXyolq11c0j326GQYydi67LPp7"
```

3. Kongfirasi Perangkat Keras

Beberapa komponen elektronik yang digunakan dalam sistem ini memiliki konfigurasi sebagai berikut:

1. Motor Servo Untuk Menutup Pintu

```
// Pin untuk servo dan sudut awal
const int servoPin = 18; // Sesuaikan dengan pin ESP32 yang digunakan
const int bukaSudut = 90; // Sudut untuk membuka pintu
const int tutupSudut = 0; // Sudut untuk menutup pintu
```

a. **servoPin** adalah pin GPIO yang digunakan untuk mengontrol motor servo.

b. **bukaSudut** adalah sudut 90° yang digunakan saat membuka pintu.

c. **tutupSudut** adalah sudut 0° yang digunakan saat menutup pintu.

2. Buzzer Sebagai Indikator

BuzzerPin adalah pin GPIO yang digunakan untuk mengaktifkan buzzer sebagai indikator suara

```
const int buzzerPin = 4;
```

3. Konfigurasi *Keypad* 4x4

Keys adalah *array* yang merepresentasikan susunan tombol pada *keypad*. ***rowPins*** dan **colPins** adalah *array* yang menyimpan nomor pin GPIO yang digunakan untuk baris dan kolom *keypad*.

```
const byte ROWS = 4; // Jumlah baris pada keypad
const byte COLS = 4; // Jumlah kolom pada keypad

char keys[ROWS][COLS] = {
  {'D', 'C', 'B', 'A'},
  {'#', '9', '6', '3'},
  {'0', '8', '5', '2'},
  {'*', '7', '4', '1'},
};

byte rowPins[ROWS] = {13, 12, 14, 27};
byte colPins[COLS] = {26, 25, 33, 32};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
```

4. Kunci *Solenoid* Dan Sensor *Magnetic*

a. SolenoidPin adalah pin GPIO yang digunakan untuk mengontrol solenoid lock.

b. MagneticPin adalah pin GPIO yang digunakan untuk membaca status sensor magnetik yang mendeteksi apakah pintu dalam keadaan terbuka atau tertutup.

c. StatusPintu adalah variabel yang menyimpan status pintu.

5. Fungsi Untuk Membunyikan *Buzzer*

Fungsi ini akan mengaktifkan *buzzer* selama 100 milidetik dan kemudian mematikannya sebagai indikator suara ketika tombol pada *keypad* ditekan atau saat password salah

```
void buzzer() {
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
  delay(100);
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
}
```

6. Fungsi Untuk Mengunci Wifi

Fungsi ini menghubungkan ESP32 ke jaringan WiFi dengan menunggu hingga status *WiFi* terhubung sebelum melanjutkan ke proses berikutnya.

```
void setupWiFi() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(1000);

  Serial.println();
  Serial.print("Connecting to: ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("-");
    delay(500);
  }

  Serial.println();
  Serial.println("WiFi Connected");
  Serial.println();
}
```

7. *Void Taskkeypad* untuk memeriksa kata sandi masukan

a. Pengguna memasukkan kata sandi menggunakan *keypad.*

Jika tombol (#)ditekan, sistem akan memeriksa apakah kata sandi benar. Jika benar, pintu akan terbuka. Jika salah, *buzzer* akan berbunyi sebagai indikator kesalahan.

```
void taskKeypad(void *pvParameters) {
  while (true) {
    char key = keypad.getKey();
    if (key) {
      Serial.print("Input password: ");
      Serial.println(key);
      buzzer();

      if (key == '#') {
        if (inputPassword == password) {
          Serial.println("✅ Password BENAR! Membuka kunci...");
          buzzer();
          digitalWrite(solenoidPin, HIGH);
          servo.write(bukaSudut);
          fb.setString("perintah", "");
          delay(unlockDuration);
          digitalWrite(solenoidPin, LOW);
        } else {
          Serial.println("❌ Password SALAH!");
          buzzer();
        }
        inputPassword = "";
      } else {
        inputPassword += key;
      }
    }
    delay(50);
  }}
```

8. Task Firebase Unutk Memantau Status Pintu

a. Membaca sensor magnet untuk mengetahui apakah pintu terbuka atau tertutup dan mengirim statusnya ke *Firebase*. Jika perintah "tutup" diterima dari *Firebase*, pintu akan ditutup.

```
void taskFirebase(void *pvParameters) {
  while (true) {
    int readMagnetic = digitalRead(magneticPin);
    if (readMagnetic == 0) {
      statusPintu = "Pintu Terbuka";
    } else  {
      statusPintu = "Pintu Tertutup";
    }
    Serial.println(statusPintu);
    fb.setString("status", statusPintu);
    String perintah = fb.getString("perintah");
    if (perintah == "tutup"){servo.write(tutupSudut);}
    delay(5000);  // Kirim status setiap 5 detik
  }
}
```

9. Fungsi Setup() Untuk Inisialisasi Sistem

a. Menghubungkan ESP32 ke WiFi.

b. Mengatur perangkat keras seperti servo, *solenoid*, *buzzer*, dan sensor *magnet.*

c. Memulai *taskKeypad* dan *taskFirebase* pada inti yang berbeda.

```
void setup() {
  setupWiFi();
  servo.attach(servoPin);
  servo.write(tutupSudut);
  pinMode(solenoidPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(magneticPin, INPUT);
  digitalWrite(solenoidPin, LOW);

  preferences.begin("storage", false);

  if (preferences.isKey("password")) {
    password = preferences.getString("password", "1234");
  }
  xTaskCreatePinnedToCore(taskKeypad, "Keypad Task", 10000, NULL, 1, NULL, 0);
  xTaskCreatePinnedToCore(taskFirebase, "Firebase Task", 10000, NULL, 1, NULL, 1);
}

void loop() {
  delay(1000);
}
```

10. Program Perulangan

loop()dibiarkan kosong karena semua tugas berjalan di dalam task multitasking ESP32.

```
void loop() {
  delay(1000);
}
```

B. Rancangan Sistem dan Alat

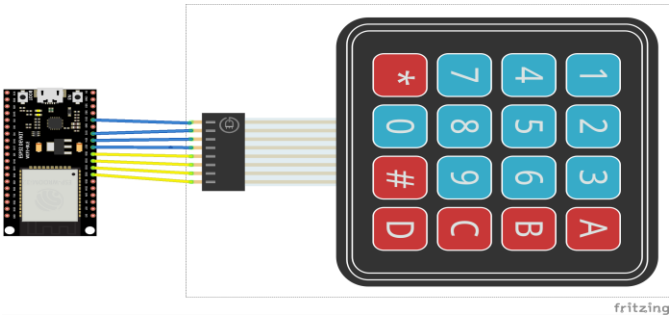
Pada perancangan sistem dan alat ini membahas mengenai hasil rancangan perangkat lunak dan perangkat keras serta pengujian perangkat

1. Perancangan Keseluruhan Perangkat Keras (*Hardware*)

Pada perancangan perangkat keras (*Hardware*) menjelaskan beberapa alat yang digunakan pada penelitian ini. Adapun beberapa rangkaian perangkat keras yang digunakan sebagai berikut:

a. Rangkaian *Keypad*

Pada rangkaian ini penulis menggunakan *keypad* 4x4 sebagai perangkat input yang terdiri dari tombol huruf,angka, dan karakter



Gambar 4. 6 Rangkaian *Keypad* 4x4

Pada rangkaian keypad yang dilihat pada gambar 4.1 pin (D13) ESP32 dihubungkan ke pin (C4) *Keypad*, pin (D12) ESP32 dihubungkan ke pin (C3) *Keypad*, pin (D14) dihubungkan ke pin (C2) *Keypad*, pin (D27) ESP32

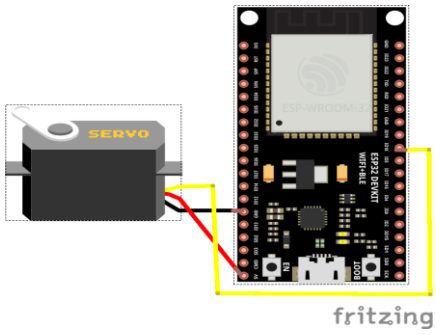
dihubungkan ke pin (C1) *Keypad*, kemudian pin (D26) dihubungkan ke pin (R4) *Keypad*, pin (D25) ESP32 dihubungkan ke pin (R3) *Keypad*, pin (D33) ESP32 dihubungkan ke pin (R2) *Keypad*, pin (D32) ESP32 dihubungkan ke pin (R1) *Keypad*.

Tabel 4. 9 Rangkaian NodeMCU ESP32 dengan Keypad

No.	NodeMCU ESP32	Keypad
1	D13	C4
2	D12	C3
3	D14	C2
4	D27	C1
5	D26	R4
6	D25	R3
7	D33	R2
8	D32	R1

b. Rangkaian Motor Servo

Pada rangkaian ini penulis menggunakan motor servo yang berfungsi untuk menggerakan mekanisme pintu, seingga dapat dibuka atau di tutup secara otomatis berdasarkan perintah dari sistem

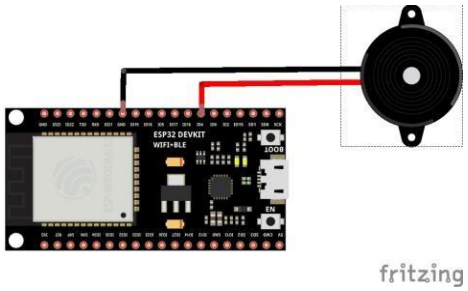


Gambar 4. 7 Rangkaian Motor Servo

Pada rangkaian Motor Servo pada gambar 4.7 di atas kabel warna kuning dihubungkan ke pin (D18), kabel warna merah dihubungkan ke pin (VIN), kemudian kabel warna coklat dihubungkan ke pin (GND)

c. Rangkaian *Buzzer*

Pada rangkaian ini penulis menggunakan *buzzer* sebagai alarm untuk memberikan peringatan atau pemberitahuan.

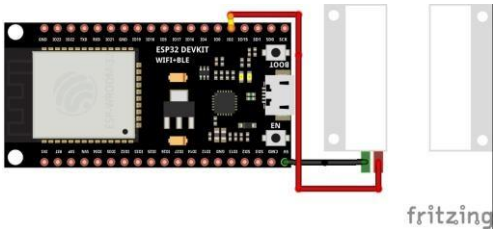


Gambar 4. 8 Rangkaian *Buzzer*

Pada rangkain Buzzer yang dapat diliat di gambar 4.8 kabel merah pada *buzzer* dihubungkan ke pin (GND) ESP32, kemudian kabel hitam pada *buzzer* dihubungkan ke pin (D04) ESP32.

d. Rangkaian Sensor *magnetic switch*

Pada rangkaian Sensor *magnetic switch* ini digunakan untuk mendeteksi status pada pintu yang dilengkapi dengan bagian magnet

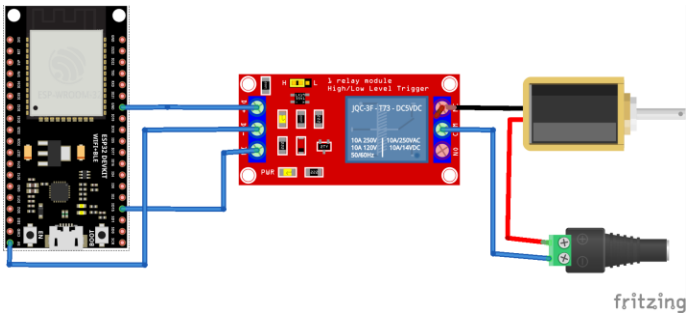


Gambar 4. 9 Rangkaian Sensor *Magnetic Switch*

Pada rangkaian *Sensor magnetic switch* bisa diliat pada gambar 4.9 kabel merah pada sensor *magnetic switch* dihubungkan ke pin (D02), kemudian kabel hitam pada sensor *magnetic switch* dihubungkan ke pin (VIN).

e. Rangkaian Solenoid lock door

Pada rangkaian ini penulis mengunakan *solenoid lock door* sebagai penguncian dan membuka pintu secara elektronik

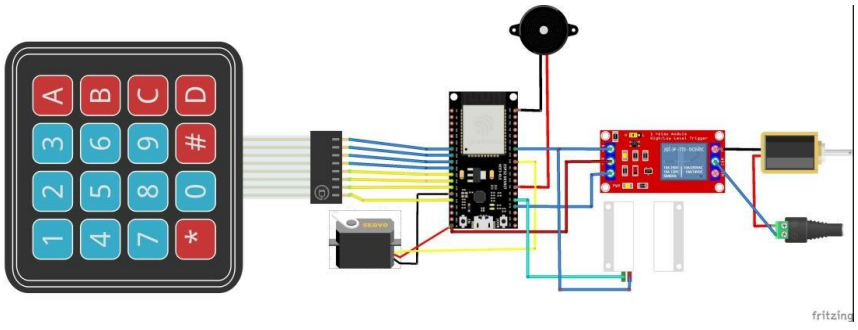


Gambar 4. 10 *Solenoid Lock Door*

Pada rangkaian *solenoid lock door* yang dapat diliat pada gambar diatas yang menggunakan *relay* sebagai saklar listrik untuk mengendalikan *solenoid lock door* yang dimana pin (-) relay dihubungkan ke pin VIN(-) ESP32, pin (+) relay dihubungkan ke pin GND(+) ESP32, kemudian pin (IN) *relay* dihubungka ke pin (D15) ESP32, kemudian kabel hitam pin (-) *solenoid lock door* dihubungkan ke pin (NC) *relay*, pin (+) *solenoid lock door* di hubungkan ke pin (+) kabel konektor, pin (COM) *relay* dihubungkan ke pin (-) kabel konektor.

f. Rangkaian Keseluruan

Berikut merupakan keseluruhan perangkat keras (*Hardware*) rancang bangun *smart key home* berbasis *internet of things* :

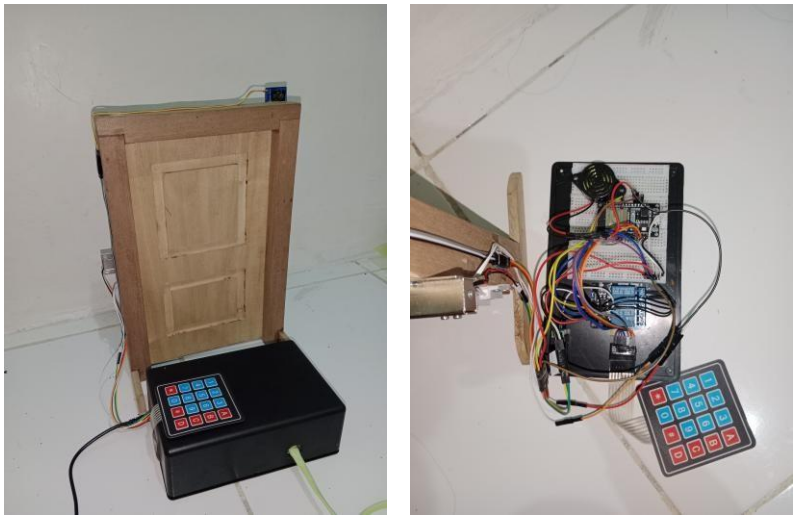


Gambar 4. 11 Rangkaian Alat Keseluruhan

Pada rangkaian alat yang dapat diliat pada gambar 4.11 merupakan rangkaian secara keseluruhan yang menghubungkan motor servo, *buzzer*, sensor *magnetic switch*, dan *relay* sebagai komponen utama sistem sebagai saklar penghubung *solenoid lock door*.

2. Rancangan Prototype

Pada bagian ini dijelaskan mengenai hasil rancang bangun *prototype smart key home* berbasis *internet of things*.



Gambar 4. 12 Prototype Alat

Pada *prototype* rancang bangun *smart key home* berbasis *internet of things* ini menggunakan *keypad* sebagai alat untuk membuka pintu secara manual menggunakan kode pin yang terhubung ke ESP32 dan *Solenoid Lock Door* sebagai alat pengunci pintu, ESP32 sebagai pengontrol komponen alat serta sebagai penghubung ke *firebase* dan alat melalui jaringan wifi, *relay* untuk mengaliri listrik pada *Solenoid Lock Door* yang dikontrol oleh ESP32, *Buzzer* sebagai alarm peringatan dan motor servo untuk menggerakan pintu.

3. Koneksi Data Firebase

1. Pilih "*data base*" di menu database
2. Pilih *realtime database*, lalu atur mode akses menjadi "*start in test mode*"
3. Agar ESP32 bisa menulis data tanpa autentigikasi terkebih dahulu.
4. Api key "COeBhNCTzT3NuZWXyolq11c0j326GQYydi67LPp7" dan Database URL "https://smartkeyhome-d8928-default-rtdb.firebaseio.com/"
5. Buka project setting,servie,accounts
6. Salin database URL yang akan digunakan dalam kode ESP32



Gambar 4. 13 Tampilan Database Firebase

Pada gambar 4.10 menunjukkan *firebase database*, terdapat dua entitas utama dalam struktur yaitu perintah dan status

a. Perintah : intruksi yang dikirim dari aplikasi untuk mengontrol pintu (membuka dan menutup), jika perintah "pintu tertutup" maka pintu tertutup atau perintah "pintu terbuka" maka pintu akan terbuka dan ESP32 akan membaca perintah dari *firebase*
b. Status : setelah ESP32 menjalankan perintah dari *Firebase*, sistem akan memperbarui status sesuai kondisi pintu yang sebenarnya.

4. Pengujian Alat

Penggujian alat bertujuan untuk mengetahui apakah berfungi atau tidaknya rangkaian alat *Smart Key Home* Berbasis *Internet Of Things* . Adapun pengujian dilakukan dengan

Tabel 4. 10 Hasil Pengujian Alat

No	Input keypad	Perintah firebase	Hasil Uji Coba	Keterangan
1	0	0	0	Tidak ada input dari keypad atau Firebase
2	1	0	1	Keypad input benar, pintu terbuka
3	0	1	1	Firebase perintahkan buka, walau keypad tidak aktif
4	1	1	1	Keypad dan Firebase aktif, pintu terbuka
5	0	0	0	Hanya keypad benar, cukup untuk membuka pintu
6	1	0	1	Hanya keypad benar, cukup untuk membuka pintu
7	0	1	1	Kontrol jarak jauh aktif, pintu terbuka
8	1	1	1	Sistem lokal & online aktif, pintu terbuka

Keteragan angka

Input keypad

1= Password benar

0= Password salah/ belum diinput

Perintah firebase buka/tutup

1= Perintah buka dari aplikasi IOT

0= Tidak ada perintah

Output pintu

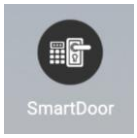
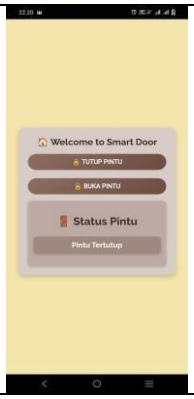
1= pintu terbuka

0= pintu terkunci

5. Pengujian Black box

Berikut ini merupakan pengujian *black box* dari aplikasi monitoring status pintu

Tabel 4. 11 Hasil Pengujian *Black Box*

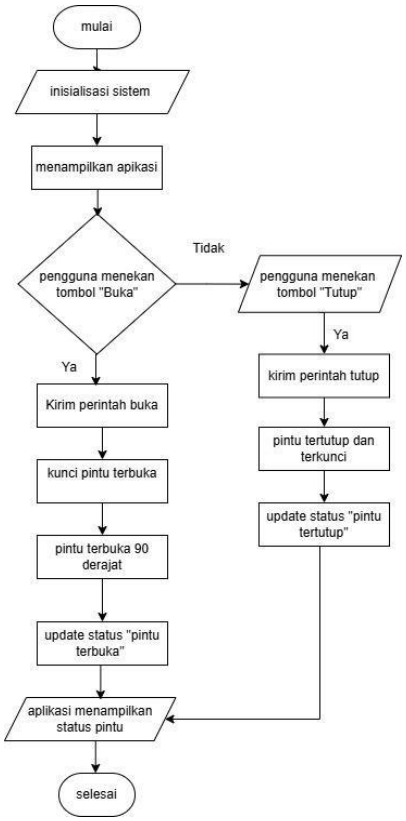
No	Uji coba	Hasil yang dharapkan	Hasil
1	Membuka aplikasi android 	Tombol "Tutup", “buka” dan “status pintu”	
2	Tombol “tutup pintu” 	Pintu tertutup , status di aplikasi berubah menjadi "Pintu Tertutup"	
3.	Tombol “buka pintu” 	Pintu terbuka, status di aplikasi berubah menjadi “pintu terbuka”	
4.	Status pintu	Status di aplikasi harus sesuai dengan kondisi pintu yang sebenarnya	

6. Pengujian White box

Halaman Monitoring (Aplikasi Android)

1. Tampilan aplikasi android

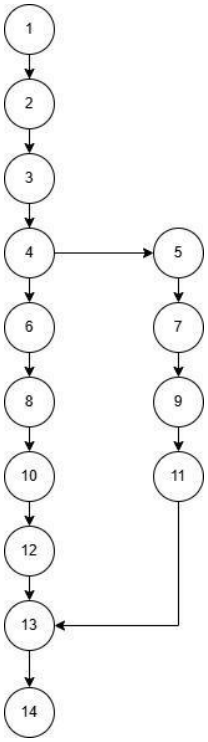
a. *Flowchart*



Gambar 4. 14 *Flowchart* sistem tampilan aplikasi

Berdasarkan gambar 4.14 monitoring aplikasi dimulai ketika pengguna membuka aplikasi yang menampilkan dua informasi, tombol “tutup pintu” dan status pintu kemudian pengguna memilih untuk menutup pintu dengan menekan tombol yang tersedia setelah tombol ditekan, aplikasi mengirimkan perintah ke ESP32, lalu *firebase* menerima perintah dan mempeberbarui status pintu sesuai permintaan pengguna setelah status di perbarui, aplikasi akan menampilkan data terbaru dari firebase. Pengguna melihat tampilan terbaru mengenai status pintu setelah diperbarui firebase.

b. *Flowgraph*



Gambar 4. 15 *Flowgraph* sistem tampilan aplikasi

Berdasarkan gambar 4.15 yang disajikan diatas dapat dilakukan proses perhitungan sebagai berikut:

(a) Menghitung *Clomatic Complexcit* dari *Edge* dan *Node*

Dengan rumus v(g) = E-N+2

Dengan edge = 15

Dengan node =14

Dengan predikat node =2

Penyelesaian:

V(G) = E-N+2

=15-14+2

=3

Predikat Node = P+1

= 2+1

(b) Berdasarkan perhitungan *Cylomatic Complexcity* dari *Flowgraph* di atas memiliki region = 3

(c) Independent path pada *Flowgraph* diatas adalah:

Parh 1= 1-2-3-4-5-6--8-10-11-12-13-14

Parh 2 = 1-2-3-4-5-7-9-11-12-13-14

Parh 3 = 1-2-3-4-6-8-10-11-12-13-11-13-14

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil pembahasan dan hasil alat yang telah dilakukan pada alat tersebut maka dapat ditarik kesimpulan Sistem Smart Key Home berbasis IoT sebagai berikut.

1. *Prototype Smart Key Home* berbasis IoT telah dibuat untuk membuka dan mengunci pintu secara otomatis menggunakan smartphone serta secara manual menggunakan *keypad*.
2. keypad digunakan sebagai alternatif untuk membuka pintu dan menutup pintu, dan *buzzer* akan berbunyi jika ada kesalahan input.
3. Sistem dilengkapi dengan tombol reset untuk mengubah jika terjadi kesalahan input, serta tombol tutup pada *keypad* untuk mengunci pintu secara manual.
4. Minimal PIN yang digunakan pada keypad adalah 4 digit
5. Keypad lebih responsif dibandingkan aplikasi, karena langsung berkomunikasi dengan sistem tanpa melalui internet.
6. Pengujian menunjukkan bahwa sistem bekerja dengan baik, meskipun perlu stabilitas jaringan.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat diperlukan perbaikan dan pengembangan penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk mengembangkan selanjutnya

1. Pasang fitur keamanan tambahan seperti sensor sidik jari atau pemindai wajah supaya pintu lebih aman dan tidak mudah dibuka orang lain.
2. Sambungkan sistem ini dengan perangkat *smart home*, seperti alarm dan kamera *CCTV*, supaya rumah lebih aman dan bisa dipantau dengan lebih mudah.
3. Perbaiki koneksi internetnya supaya sistem bisa jalan lebih lancar, terutama kalau jaringan tiba-tiba lemot atau putus.

DAFTAR PUSTAKA

Amini, S., Informatika, T., Informasi, F. T., Luhur, U. B., Utara, P., Lama, K., Switch, M. D., Ruangan, K., Keamanan, P., Dengan, R., Pir, S., Magnetic, D., Switch, D., & Web, B. (2021). Perancangan Keamanan Ruangan Dengan Sensor Pir. *Juli*, *4*(2), 50–56.

Fathoni, A. N., & Khotimah, K. (2023). Smart Home Design Based on IoT Using Telegram Messenger Bot and NodeMCU ESP32. *TELKA: Jurnal Telekomunikasi, Elektronika, Komputasi, Dan Kontrol*, *9*(1), 34–43.

Mariza Wijayanti. (2022). Prototype Smart Home Dengan Nodemcu Esp8266 Berbasis Iot. *Jurnal Ilmiah Teknik*, *1*(2), 101–107. https://doi.org/10.56127/juit.v1i2.169

Novianti, T. (2019). Rancang Bangun Pintu Otomatis dengan Menggunakan RFID. *Jurnal Teknik Elektro Dan Komputer TRIAC*, *6*(1), 1–6. https://doi.org/10.21107/triac.v6i1.4878

Nugroho, A. M., Wibowo, R. J., Islamey, E. V, & Nurnisa, H. (2018). Pengujian White Box pada Program Monitoring Suhu dan Kelembaban Udara Menggunakan Teknik Basis Path. *Academia.Edu*.

Raihan, T. M. (2022). Sistem pemantauan kualitas air menggunakan Esp32 dengan Fuzzy Logic Sugeno Berbasis Android. *Sistem Pemantauan Kualitas Air Menggunakan Esp32 Dengan Fuzzy Logic Sugeno Berbasis Android*.

Samsugi, S., Yusuf, A. I., & Trisnawati, F. (2020). Sistem Pengaman Pintu Otomatis Dengan Mikrokontroler Arduino Dan Module Rf Remote. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, *1*(1), 1–6.

Yusman, Y., Bakhtiar, B., & Sari, U. (2019). Rancang Bangun Sistem Smart Home dengan Arduino Uno R3 Berbasis Internet of Things (IoT). *Jurnal Litek : Jurnal Listrik Telekomunikasi Elektronika*, *16*(1), 25. https://doi.org/10.30811/litek.v16i1.1466