

RANCANG BANGUN TRAINER
SMART HOME MENGGUNAKAN 3 BAHASA PEMOGRAMAN DENGAN
MIKROKONTROLER NODEMCU ESP8266
BERBASIS *INTERNET OF THINGS*

TUGAS AKHIR

*Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana SI
Pada Departemen Teknik Elektronika Program Studi Pendidikan Teknik
Informatika Universitas Negeri Padang*



ISWORO SETYO NUGROHO

NIM. 2017/17076080

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2022

HALAMAN PERSETUJUAN TUGAS AKHIR

RANCANG BANGUN TRAINER *SMART HOME* MENGGUNAKAN 3 BAHASA
PROGRAMAN DENGAN *MIKROKONTROLER NODEMCU ESP8266*
BERBASIS *INTERNET OF THINGS*

Name : Isworo Setyo Nugroho
NIM/IM : 17076080/2017
Program Studi : Pendidikan Teknik Informatika
Jurusan : Teknik Elektronika
Fakultas : Teknik

Padang, 17 November 2022

Disetujui Oleh

Pembimbing,

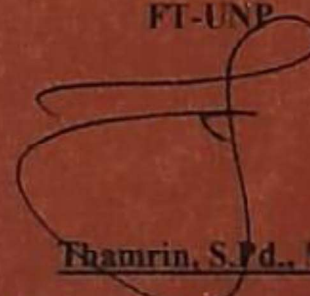

Ahmadul Hadi, S.Pd., M.Kom.

NIP. 197612092005011003

Mengetahui,

Kepala Departemen Teknik Elektronika

FT-UNP


Hamrin, S.Pd., MT.

NIP. 19770101 200812 1 001

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Isworo Setyo Nugroho
NIM/TM : 17076080/2017
Program Studi : Pendidikan Teknik Informatika
Jurusan : Teknik Elektronika
Fakultas : Teknik

Dengan ini saya menyatakan bahwa tugas akhir saya yang berjudul, "RANCANG BANGUN TRAINER *SMART HOME* MENGGUNAKAN 3 BAHASA PEMOGRAMAN DENGAN *MIKROKONTROLER NODEMCU ESP8266* BERBASIS *INTERNET OF THINGS*" adalah benar merupakan hasil karya saya sendiri sesuai dengan sitasi dan metode karya ilmiah dan bukan merupakan plagiat dari karya orang lain. Apabila sesuatu saat terbukti saya melakukan plagiat, maka saya bersedia di proses dan menerima sanksi akademis maupun hukum sesuai dengan hukum dan ketentuan yang berlaku, baik di Universitas Negeri Padang maupun di masyarakat negara.

Demikianlah pernyataan ini saya buat dengan penuh kesadaran dan rasa tanggung jawab sebagai anggota masyarakat ilmiah.

Padang, 17 November 2022

Saya yang menyatakan,



Isworo Setyo Nugroho

NIM.17076080

ABSTRAK

Isworo Setyo Nugroho : RANCANG BANGUN TRAINER *SMART HOME* MENGGUNAKAN 3 BAHASA PEMOGRAMAN DENGAN MIKROKONTROLER NODEMCU ESP8266 BERBASIS *INTERNET OF THINGS*

Penelitian ini bertujuan membuat *trainer* sistem kontrol meliputi lampu, kipas angin, stop kontak, pemanas air minum pada rumah tinggal yang berbasis teknologi *Internet of Things* (IoT). Sistem ini dirancang dengan *interface* dari 3 bahasa pemrograman (*Html, PHP dan Java*) sebagai media untuk melatih dan meningkatkan kemampuan siswa dalam melakukan pemrograman untuk mengontrol perelatan elektronik seperti lampu, kipas angin, stop kontak, pemanas air minum yang kemudian dapat dimonitoring secara *real time* melalui sebuah platform digital baik pada *smartphone* ataupun PC. Metode pembuatan alat dimulai dengan merancang diagram blok, memilih spesifikasi komponen berdasarkan diagram blok, mengintegrasikan hardware, dan pemrograman sistem. Output sistem dilengkapi dengan komponen yang berfungsi untuk mengendalikan kondisi peralatan rumah seperti lampu, kipas angin, Stop kontak, pemanas air minum. *Relay modul* digunakan sebagai saklar menghidupkan dan mematikan peralatan rumah seperti lampu, kipas angin, Stop kontak, pemanas air minum. Sistem dibuat berbasis *Internet of Things* (IoT), artinya sistem terhubung ke internet menggunakan *modul* ESP8266. Sistem mengakses data dari web Firebase yang kemudian ditampilkan dalam sebuah aplikasi *smartphone* sebagai sarana untuk memonitoring kondisi peralatan rumah. Hasil pengujian menunjukkan bahwa sistem yang telah dibuat dan dirancang dengan *interface* dari 3 bahasa pemrograman (*Html, PHP dan Java*) mampu mengontrol peralatan rumah seperti lampu, kipas angin, stop kontak, pemanas air minum.

Kata Kunci : *Trainer, Smart Home, NodeMCU ESP8266, Firebase.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah rabbil'alam, puji dan syukur peneliti ucapkan kepada Allah Subhanahu Wata'ala yang telah melimpahkan rahmat dan karunia-Nya, sehingga peneliti dapat menyelesaikan laporan tugas akhir yang berjudul – RANCANG BANGUN TRAINER *SMART HOME* MENGGUNAKAN 3 BAHASA PEMOGRAMAN DENGAN *MIKROKONTROLER NODEMCU ESP8266* BERBASIS *INTERNET OF THINGS*- ini dengan baik. Shalawat beserta salam tidak henti-hentinya peneliti hadiahkan kepada Rasulullah Salallahu'Alaihi Wasallam dan kepada keluarga, para sahabat, dan orang-orang yang memperjuangkan risalah beliau sampai akhir zaman.

Penelitian laporan tugas akhir ini bertujuan sebagai syarat untuk menyelesaikan studi dan memperoleh gelar sarjana (S1) di Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang. Dalam penyusunan dan penelitian laporan tugas akhir ini peneliti banyak mendapatkan bantuan, bimbingan dan perhatian dari berbagai pihak. Maka dari itu pada kesempatan ini peneliti ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. Fahmi Rizal, M.Pd., M.T selaku Dekan Fakultas Teknik Universitas Negeri Padang.
2. Bapak Thamrin, S.Pd., M.T selaku Ketua Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang dan juga Dosen Pembimbing serta Penguji

yang telah banyak memberikan bantuan dan bimbingan dengan sabar dan ikhlas dalam penelitian laporan tugas akhir ini.

3. Ibu Delsina Faiza, S.T., M.T selaku Sekretaris Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
4. Bapak Ahmaddul Hadi, S.Pd, M.Kom selaku Dosen Penasihat Akademik.
5. Bapak/Ibu Dosen, Staf Pengajar, dan Administrasi di Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
6. Rekan-rekan seperjuangan Prodi Pendidikan Teknik Informatika 2017 serta semua pihak yang telah membantu dan memotivasi dalam proses penyelesaian Tugas Akhir ini.
7. Teman teman organisasi kemahasiswaan Universitas Negeri Padang (HJM HIMANIKA, BPM FT, MPM KM) tempat bertukar pikiran dan gagasan dalam keadaan senang ataupun duka. Yang senang tiasa membantu dan mensupport dalam menjalankan aktivitas.
8. Teman teman Pengurus Forum Lembaga Legislatif Mahasiswa Indonesia dari Periode 90, 01 dan 12 yang telah membantu dan memotivasi dalam proses penyelesaian Tugas Akhir ini
9. Teman teman himpunan mahasiswa alwashliyah yang telah banyak membantu, memotivasi dan memberikan support dalam penyelesaian tugas akhir ini.
10. Abang Marta Suhendra S.Pd, M.Pd , dan juga kakak Nazwita Dewi Putri. M.Biomed yang telah banyak membantu, memotivasi serta memfasilitasi setiap pengerjaan tugas akhir ini.

11. Teristimewa untuk kedua orang tua dan keluarga yang telah memberikan kasih sayang, semangat, dorongan, motivasi, serta perhatiannya sehingga peneliti dapat menyelesaikan laporan tugas akhir ini.

Semoga bantuan yang telah diberikan dapat menjadi amalan yang baik dan mendapatkan imbalan dari Allah Subhanahuwata`ala, Aamiin. Peneliti menyadari bahwa penelitian laporan tugas akhir ini masih jauh dari kesempurnaan, untuk itu segala kritik dan saran bersifat membangun sangat peneliti harapkan demi perbaikan penelitian kedepannya.

Padang, 17 November 2022

Peneliti

DAFTAR ISI

HALAMAN PERSETUJUAN TUGAS AKHIR.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN TUGAS AKHIR.....	Error! Bookmark not defined.
SURAT PERNYATAAN	iii
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Identifikasi Masalah.....	6
C. Batasan Masalah	6
D. Rumusan Masalah.....	7
F. Manfaat Tugas Akhir	8
BAB II LANDASAN TEORI.....	10
A. <i>Internet of Things (IOT)</i>	10
B. <i>Smart Home</i>	11
C. Model Perancangan <i>ADDIE</i>	12
D. Perangkat Keras	13
1. <i>NodeMCU ESP 8266</i>	13
2. <i>Relay Module</i>	14

3.	<i>Catu Daya Switching</i>	17
4.	<i>Router (Modem)</i>	17
5.	Saklar Tukar	18
6.	Stop kontak	19
7.	<i>Smartphone Android</i>	19
E.	Perangkat Lunak	20
1.	Android Studio.....	20
2.	Arduino IDE.....	21
3.	Firestore	22
4.	<i>Google Assistant</i>	23
5.	Html (<i>Hypertext Markup Language</i>)	24
6.	PHP (<i>Hypertext Preprocessor</i>)	24
7.	Java	24
BAB III ANALISIS DAN PERANCANGAN SISTEM		25
A.	Analisis Sistem.....	26
1.	Analisis Sistem Berjalan	26
2.	Analisis Sistem yang Diusulkan	29
B.	Perancangan Sistem	31
1.	<i>Use Case Diagram</i>	31
2.	<i>Activity Diagram</i>	32
3.	<i>Sequence Diagram</i>	33
C.	Perancangan Alat	34
1.	Diagram Blok Rangkaian.....	35

2. Flowchart Sistem Kontrol	37
D. Prinsip Kerja Alat	37
E. Perancangan Perangkat Keras	38
1. Skema Rangkaian Sensor <i>RFID</i>	38
2. Skema Rangkaian <i>Relay Module</i>	39
F. Perancangan Perangkat Lunak	40
G. Rancangan Fisik Alat	40
H. Perancangan Sistem Antarmuka	42
BAB IV HASIL DAN PEMBAHASAN.....	44
A. Hasil Pembuatan Alat	44
1. Bentuk Fisik Alat Keseluruhan.....	44
2. Posisi Mikrokontroler Node MCU	45
3. Posisi Saklar.....	46
4. Posisi <i>Power Supply</i>	46
B. Persiapan Pengoperasian Trainer <i>Smart Home</i>	47
C. Pengujian Perangkat Keras	48
1. Pengukuran <i>Power Supply</i> dan <i>IC</i> Tegangan.....	48
2. Pengujian Driver <i>Relay</i>	50
3. Pengujian Sensor Sistem Keamanan.....	53
D. Pengujian Perangkat Lunak	53
1. Pengujian Pengiriman Data Ke Web Server.....	53
2. Pengujian Perangkat melalui HTML yang telah dibangun.....	57
3. Pengujian Perangkat melalui WEB PHP yang telah dibangun.....	58

4. Pengujian Perangkat melalui aplikasi android yang telah dibangun.	59
5. Pengujian perangkat menggunakan perintah suara.....	60
BAB V PENUTUP	62
A. Kesimpulan	62
B. Saran.....	62
DAFTAR PUSTAKA	64
LAMPIRAN	67

DAFTAR TABEL

Tabel 1. Analisis proses bisnis	26
Tabel 2. Analisis pelaku bisnis	27
Tabel 3 . Analisis Aturan Bisnis.....	28
Tabel 4. Analisis Masalah dan Solusi	28
Tabel 5. Analisis <i>User</i>	29
Tabel 6. Analisis Prosedur.....	30
Tabel 7. Pengukuran Tegangan	48
Tabel 8. Hasil pengukuran tegangan <i>relay</i>	52
Tabel 9. Hasil pengujian <i>user Interface</i> HTML	57
Tabel 10. Hasil pengujian <i>user Interface</i> PHP	58
Tabel 11. Hasil pengujian <i>user Interface</i> Android	59
Tabel 12. Hasil pengujian <i>Voice Recognition</i>	60

DAFTAR GAMBAR

Gambar 1. Alur Pengembangan Model <i>ADDIE</i>	13
Gambar 2. Board NodeMCU ESP8266.....	14
Gambar 3. Skema rangkaian <i>Relay Module</i>	16
Gambar 4. <i>Relay Module</i>	16
Gambar 5. Model catu daya	17
Gambar 6. Router	18
Gambar 7. Saklar tukar.....	18
Gambar 8. Stop kontak.....	19
Gambar 9. Logo Android	20
Gambar 10. Logo Android Studio.....	21
Gambar 11. Tampilan Menu Arduino IDE	22
Gambar 12. Logo Firebase	23
Gambar 13. Alur Pengembangan Model <i>ADDIE</i>	25
Gambar 14. flowmap Yang Diusulkan.....	30
Gambar 15. Use Case Diagram	31
Gambar 16. Activity Diagram	32
Gambar 17. Sequence Diagram.....	33
Gambar 18. Skema <i>Smart Home</i>	34
Gambar 19. Blok diagram rancangan alat.....	35
Gambar 20. <i>Flowchart sistem</i>	37
Gambar 21. Skema Rangkaian Sensor <i>RFID</i>	39

Gambar 22. Skema Rangkaian <i>Relay</i>	39
Gambar 23. Proses Perancangan Perangkat Lunak Pada <i>Mikrokontroler</i>	40
Gambar 24. Desain tampak dari depan	40
Gambar 25. Desain tampak dari kiri	41
Gambar 26. Desain Tampak dari kanan	41
Gambar 27. Desain tampak keseluruhan.....	41
Gambar 28. Gambar tampak atas	41
Gambar 29. Rancangan sistem antarmuka	42
Gambar 30. Tampak keseluruhan.....	44
Gambar 31. Posisi Doorlock dan Posisi Kipas Angin.....	44
Gambar 32. Posisi Lampu dan Posisi Pemanas Air Minum.....	45
Gambar 33. Posisi Node MCU.....	45
Gambar 34. Posisi Saklar	46
Gambar 35. Posisi <i>Power Suply</i>	47
Gambar 36. Pengukuran pada catu daya	49
Gambar 37. Pengukuran Pada <i>IC 9V</i>	49
Gambar 38. Pengukuran Pada <i>IC 5V</i>	50
Gambar 39. Setup Rangkaian <i>Relay</i>	51
Gambar 40. pengukuran saat diberikan logika <i>LOW</i>	51
Gambar 41. pengukuran saat diberikan logika <i>HIGH</i>	52
Gambar 42. No ID kartu yang muncul saat <i>scanning</i>	53
Gambar 43. SSID Dan Password Pada Pemograman Arduino IDE.....	54
Gambar 44. Tampilan Serial Monitor Terhubung Dengan Akses Poin	54

Gambar 45. Token <i>FIREBASE_HOST</i> Dan <i>FIREBASE_AUTH</i>	55
Gambar 46. Tampilan Serial monitor Data Yang Dikirim.....	56
Gambar 47. Tampilan <i>Database</i> Pada Firebase	56
Gambar 48. Tampilan Button pada <i>user interface</i>	57
Gambar 49. Tampilan Button pada <i>user interface</i>	58
Gambar 50. Tampilan Button pada <i>user interface</i>	59
Gambar 51. Tampilan <i>Google Voice Recognition</i>	60

DAFTAR LAMPIRAN

Lampiran 1. Source Code	67
Lampiran 2. Source Code menggunakan PHP 1	71
Lampiran 3. Source Code http client.....	73
Lampiran 4. Source Code menggunakan android	75
Lampiran 5. Source Code terhubung ke Firebase	78
Lampiran 6. Modul Trainer	85

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Di era globalisasi saat ini ilmu teknologi informasi selalu berkembang dan semakin maju. Hampir seluruh aspek kehidupan kita bergantung dengan kemajuan teknologi, tentu hal ini terjadi karena sejatinya teknologi itu dikembangkan untuk mempermudah aktivitas dan pekerjaan kita. Perkembangan teknologi digital yang sangat pesat saat ini telah mengubah tatanan dan pola kehidupan manusia. Manusia yang dulu berkomunikasi menggunakan kentongan berevolusi menggunakan surat hingga kini telah berkembang menggunakan ponsel pintar (Rachman, 2017). Perkembangan teknologi digital sudah semakin merata di berbagai sektor kehidupan mulai dari sektor perdagangan, pendidikan, kesehatan, pemerintahan, pertanian hingga perbankan. Serta tidak dapat dibantahkan perkembangan itu semua tidak terlepas dari pengaruh internet dalam proses perkembangannya.

Menurut (Behmann, 2015), perkembangan internet dan teknologi di era revolusi industri 4.0 saat ini banyak melahirkan produk-produk digital salah satunya yaitu *Internet of things (IOT)*. *Internet of things (IOT)* merupakan salah satu pembaharuan dalam era dunia industri 4.0 yang akan menjadi tren besar di masa depan. *Internet of things* atau *IOT* adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar, mengadopsi komputasi yang bersifat mobile dan konektivitas dengan menghubungkannya ke dalam kehidupan sehari-hari. *IOT* berkaitan dengan *DOT (disruption of things)* sebagai pengantar perubahan atau transformasi penggunaan internet yang dari sebelumnya yaitu *internet of people* menjadi *Internet of M2M (Maching-to-Machine)*. *Internet of things (IOT)*

bisa dimanfaatkan pada gedung untuk mengendalikan peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dari jarak jauh melalui jaringan nirkabel. Sehingga, kemajuan teknologi dapat dimanfaatkan dan dipelajari serta diterapkan dalam kehidupan sehari-hari. Dengan adanya *Internet of things (IOT)* maka terciptalah sistem yang disebut *Smart Home*. *Smart Home* adalah sistem yang dapat mengontrol dan memantau perangkat ruangan secara berkala.

Smart Home meliputi alat-alat listrik rumah tangga, sistem penerangan serta sistem keamanan rumah yang semuanya dikendalikan dan dipantau secara langsung sesuai keinginan oleh pemilik rumah. Sistem *Smart Home* yang telah berkembang saat ini ada yang menggunakan instalasi kabel dan tanpa kabel (M. D. Putro, 2017). *Smart Home* identik dengan perangkat pintar, untuk sejarah perangkat pintar sendiri dimulai dengan ditemukannya *remote control* berbasis *infrared* pada tahun 1898. Hingga pada tahun 1966 sampai 1967 baru lahirlah *ECHO IV* dan komputer dapur, meskipun produk ini tidak pernah dijual secara komersial *ECHO IV* adalah perangkat cerdas pertama yang dapat digunakan untuk mengontrol suhu rumah dan menyalakan serta mematikan peralatan listrik. Setahun kemudian baru komputer dapur dikomersialkan tetapi produk ini pada akhirnya juga tidak laku dipasaran. Setelah penemuan *mikroprosesor* pada awal 1970-an maka mulai bermunculanlah model *Smart Home* yang mulai dikembangkan pada tahun 1975 di Amerika Serikat menggunakan konsep gelombang radio untuk mengirimkan data.

Seiring dengan berjalannya waktu mulailah berkembangnya *Mikrokontroler* berbasis *Bluetooth* maka berkembang pulalah peralatan *Smart Home* yang terintegritas dengan jaringan *Bluetooth* yang mana dengan mengunakan *kontroler*

berbasis jaringan *Bluetooth* ini peralatan sudah dapat dikendalikan dan dimonitoring tanpa kabel dengan menggunakan komputer. Akan tetapi dengan menggunakan *kontroler* berbasis *Bluetooth* ini jumlah perangkat yang dapat dikendalikan terbatas dan juga terbatasnya jarak kontrol dikarenakan jangkauan dari jaringan *Bluetooth* tidak lebih dari 10 meter. Sehingga ini menjadi kendala dalam pengendalian dan memonitoring dari status perangkat elektronik rumah tangga.

Kemudian saat ini dengan telah berkembangnya *Mikrokontroler* dari yang berbasis *Bluetooth* menjadi berbasis Wi-fi, salah satu contohnya yaitu *NodeMCU ESP8266* yang dapat terkoneksi dengan jaringan internet memungkinkan pengembangan *Smart Home* terintegrasi menggunakan jaringan internet. *NodeMCU* sebagai pusat kontrol peralatan elektronik rumah tangga dapat dikontrol dengan menggunakan perintah suara (*Voice Control*) melalui *Google Assistant* pada *Smartphone android* dengan menggunakan jaringan Wi-Fi sebagai media komunikasi data. Alat dirancang bukan untuk menggantikan saklar manual pada umumnya melainkan untuk menjadi *alternative* lain untuk mengendalikan peralatan elektronik sehingga lebih mudah dalam penggunaannya.

Tradisi masyarakat Indonesia yang melakukan mudik atau meninggalkan rumah dalam waktu yang lama menyebabkan borosnya penggunaan listrik, Rata-rata pemilik meninggalkan rumah dalam keadaan lampu menyala. Sehingga ini menyebabkan tagihan listrik membengkak, serta dilain waktu kondisi seperti ini dapat menyebabkannya korsleting yang mampu menimbulkan korban jiwa serta materil. Lalu dengan saklar konvensional pemilik rumah tidak dapat memonitoring

status perangkat dan harus kembali kerumah untuk memastikan kondisi sebenarnya yang mana ini sangat tidak efisien untuk pemilik rumah ketika sudah dalam perjalanan meninggalkan rumah.

Penggunaan saklar konvensional juga dinilai tidak ramah untuk semua kalangan khususnya bagi penyandang disabilitas, dikarenakan konsep yang ditawarkan dari penggunaan saklar konvensional itu pengguna dituntut untuk menghampiri kemudian menyentuh saklar secara langsung untuk mengoperasikannya. Kemudian bagaimana dengan penyandang kursi roda yang kesulitan menjangkau posisi saklar? Atau bagaimana dengan orang sakit yang hanya bisa berbaring ditempat tidur tetapi masih bisa berbicara serta mengoperasikan *smartphone* miliknya?

Konsep *Smart Home* menjadi konsep rumah masa depan yang mana fitur yang ditawarkan tidak hanya berbasis aplikasi via *android* saja namun juga telah berkembang dengan *voice command* yaitu pengoperasian peralatan elektronik rumah tangga dengan menggunakan perintah suara, dimana pada saat itu pengguna tidak harus lagi bergerak mendekati sebuah peralatan rumah tangga dan menekan tombol yang ada untuk menghidupkan atau mematikan peralatan tersebut, melainkan cukup dengan menggunakan perintah suara oleh pemilik rumah melalui teknologi *voice recognition* sistem ini dapat menghidupkan dan mematikan peralatan rumah tangga dengan pusat kontrol menggunakan *Mikrokontroler NodeMCU ESP8266*.

Dalam menangani masalah tersebut peneliti terdorong untuk memanfaatkan fitur *Google Assistant* pada *Smartphone android* sebagai *trigger* untuk mengendalikan sistem yang menjadi *alternative* lain pengganti saklar yang ada

sekarang, jadi fitur *Google Assistant* pada *Smartphone android* dapat di fungsikan untuk perintah masukan berupa suara dengan tujuan sebagai pengendali untuk menghidupkan dan mematikan lampu rumah, menghidupkan dan mematikan kipas angin, menghidupkan dan mematikan Stop kontak, membuka dan mengunci pintu rumah serta menghidupkan dan mematikan pemanas air. Sehingga menghemat waktu tanpa harus berkeliling rumah untuk menekan saklar dan dapat dikendalikan saat tidak berada dirumah atau bepergian. Sehingga dapat memberikan efektifitas dan mempermudah pengguna dalam pengontrolan. Sebagai pengontrolnya peneliti menggunakan *Mikrokontroller NodeMCU ESP8266*. Modul wifi pada *NodeMCU ESP8266* berfungsi sebagai konektor dengan *Smartphone Android*. *Smart Home* yang dibangun diharapkan juga dapat diimplementasikan dan digunakan untuk orang sakit yang berada di kursi roda/tempat tidur atau orang disabilitas tetapi dapat berbicara atau orang lanjut usia yang tidak dapat mencapai saklar agar dapat menghidupkan atau mematikan perangkat rumah.

Dari penjabaran diatas, peneliti merancang sistem *Smart Home* berbasis *Voice recognition* menggunakan *Mikrokontroler NodeMCU ESP8266*. Dengan adanya sistem ini pengontrolan alat-alat elektronik rumah tangga dapat dikontrol atau dikendalikan dari manapun dan kapanpun selagi pengguna masih terhubung didalam jaringan internet tanpa perlu bergerak mendekati peralatan elektronik tersebut hanya sekedar untuk menghidupkan ataupun memamatkannya, serta mengimplementasikan *Smart Home* berbasis *IOT* menggunakan perintah suara via *Google* Peneliti juga berharap dapat menarik minat masyarakat Indonesia untuk mengembangkan *Internet of things (IOT)* menuju Indonesia maju 2045 yang

merupakan harapan bangsa ini, Indonesia di usia yang memasuki 100 tahun kemerdekaan itu telah mapan baik dari segi pengembangan teknologi dan juga penciptaan energi terbarukan yang tidak tergantung lagi dari negara lain. Dengan harapan ini semoga kelak cita-cita bangsa ini bisa terwujud dengan menjadi Negara yang mandiri serta mampu mengembangkan dan mengelola teknologi sendiri tanpa perlu menjadi bangsa konsumen yang terus menerus menggunakan jasa dari bangsa lain untuk pengembangan teknologi dan segala sumber dayanya.

B. Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka Identifikasi masalah yang dikaji dalam tugas akhir ini yaitu:

1. Penggunaan saklar konvensional harus mendekati saklar perangkat untuk mengontrol perangkat.
2. Saklar konvensional tidak dapat dikendalikan oleh penyandang disabilitas.
3. Terbatasnya Penggunaan saklar konvensional dimana pengguna tidak dapat memonitoring setatus perangkat.
4. Kebiasaan masyarakat Indonesia meninggalkan rumah dalam waktu lama (*mudik*).
5. Belum dikembangkannya *Smart Home* berbasis *voice recognition* menggunakan *Mikrokontroler*.

C. Batasan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, maka dibuat batasan masalah agar penyajian lebih terarah dan mencapai sasaran yang ditentukan. Adapun batasan masalah yang dibuat adalah sebagai berikut:

1. Merancang *Trainer Smart Home* menggunakan *Mikrokontroler NodeMCU ESP8266*.
2. Mengaplikasikan *Mikrokontroler NodeMCU ESP8266* sebagai pengendali sistem *door lock*, Stop kontak, kipas angin (*speed 1, speed 2*), lampu 1, lampu 2 dan pemanas air minum.
3. Mengaplikasikan Bahasa pemrograman *html, php* dan *java* sebagai bagian dari sistem kontrol antarmuka.

D. Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang dan Batasan masalah sebelumnya, maka rumusan masalah yang didapatkan adalah:

1. Bagaimana cara merancang *Trainer Smart Home* menggunakan *Mikrokontroler NodeMCU ESP8266*?
2. Bagaimana mengaplikasikan *Mikrokontroler NodeMCU ESP8266* sebagai pengendali sistem *door lock*, Stop kontak, kipas angin (*speed 1, speed 2*), lampu 1, lampu 2 dan pemanas air minum?
3. Bagaimana Mengaplikasikan Bahasa pemrograman *html, php* dan *java* sebagai bagian dari sistem kontrol antar muka?

E. Tujuan Tugas Akhir

Berdasarkan rumusan masalah yang telah dipaparkan sebelumnya, adapun tujuan tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan teori *Internet of things* dalam kehidupan secara nyata.
2. Merancang dan membangun *Trainer* sistem *Smart Home* menggunakan *Mikrokontroler NodeMCU ESP8266*.

3. Mengembangkan sistem kontrol pada *Smart Home* menggunakan *Mikrokontroler NodeMCU ESP8266* berbasis konsep *Internet of things (IOT)*.
4. Menciptakan sebuah *Trainer Smart Home* dengan sistem kontrol yang dibangun dari Bahasa pemrograman *html, php* dan *java*.

F. Manfaat Tugas Akhir

Adapun manfaat yang diperoleh dalam penelitian tugas akhir ini, diantaranya :

1. Bagi Peneliti :
 - a. Dapat menambah ilmu dan keterampilan dalam mempelajari perancangan *Trainer Smart Home* berbasis mikrontoler.
 - b. Menghasilkan sebuah sistem kontrol peralatan elektronik yang terintegrasi dengan *Smartphone* menggunakan jaringan internet.
2. Bagi Pendidikan :
 - a. Dapat menjadi alat pembelajaran bagi laboratorium listrik mekatronika.
 - b. Menjadikan inspirasi bagi mahasiswa dalam pengembangan *Internet of things* secara berkelanjutan.
3. Bagi User :
 - a. Memudahkan akses *user* dalam mengontrol peralatan elektronik, menghidupkan dan mematikan peralatan elektronik dari mana saja dan kapan saja.
 - b. Memudahkan akses *user* dalam memonitoring peralatan elektronik dari mana saja dan kapan saja.
 - c. Bisa diterapkannya *Smart Home* ini pada rumah yang sering ditinggal pemiliknya.

- d. Meningkatkan efisiensi waktu dan pekerjaan rumah tangga.
- e. Menghemat biaya listrik bulanan.

.

BAB II LANDASAN TEORI

A. *Internet of Things (IOT)*

(Internet of Things) IOT dapat didefinisikan kemampuan berbagai *device* yang bisa saling terhubung dan saling bertukar data melalui jaringan internet. *IOT* merupakan sebuah teknologi yang memungkinkan adanya sebuah pengendalian, komunikasi dan kerjasama dengan berbagai perangkat keras melalui jaringan internet. Sehingga bisa dikatakan bahwa *Internet of things (IOT)* adalah ketika kita menyambungkan sesuatu (*things*) yang tidak dioperasikan oleh manusia ke internet (Hardyanto, 2017).

Dalam istilah lain juga dapat disebutkan bahwa *(Internet of things) IOT* adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar, dimana berbagai perangkat di sekitar selain bisa terhubung ke internet, juga dapat berkomunikasi dengan tablet, komputer/laptop, dan *smartphone* kemudian menggabungkannya ke dalam aktifitas sehari-hari. Istilah *IOT (Internet of things)* mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, *Cofounder and executive director of the Auto-ID Center* di *MIT*. Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contohnya dapat berupa: mesin produksi, mobil, peralatan elektronik, peralatan yang dapat dikenakan manusia (*wearables*), dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal

dan global menggunakan sensor dan atau aktuator yang tertanam. Kini banyak perusahaan besar mulai mendalami *Internet of things* sebut saja *Intel*, *Microsoft*, *Oracle*, *Tesla* dan banyak lainnya. Banyak yang memprediksi bahwa pengaruh *Internet of things* adalah “*the next big thing*” di dunia teknologi informasi, hal ini karena *IOT* menawarkan banyak potensi yang bisa digali. Contoh sederhana manfaat dan implementasi dari *Internet of things* misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus di stok lagi (Yoyon Efendi, 2018).

B. *Smart Home*

Sistem rumah cerdas (*Smart Home*) adalah sistem aplikasi yang merupakan gabungan antara teknologi dan pelayanan yang dikhususkan pada lingkungan rumah dengan fungsi tertentu yang bertujuan meningkatkan efisiensi, kenyamanan dan keamanan penghuninya. Sistem rumah cerdas biasanya terdiri atas perangkat kontrol, monitoring dan otomatisasi beberapa perangkat atau peralatan rumah yang dapat diakses melalui sebuah komputer (Yurnama, 2009). Rumah Cerdas (*Smart Home*) adalah aplikasi gabungan antara teknologi dan pelayanan yang dikhususkan pada lingkungan rumah dengan fungsi tertentu yang bertujuan meningkatkan keamanan, efisiensi dan kenyamanan penghuninya. Sistem *Smart Home* (*Smart Home*) biasanya terdiri dari perangkat monitoring, perangkat kontrol dan otomatis ada beberapa perangkat yang dapat diakses menggunakan komputer (Tri Fajar, 2009).

Smart Home adalah rumah yang dilengkapi dengan teknologi dan fitur-fitur yang memungkinkan setiap perangkat dapat berkomunikasi dengan yang lainnya dan

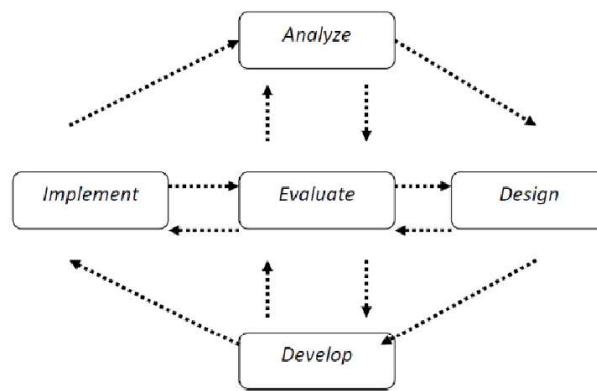
dapat dikendalikan dari mana saja menggunakan jaringan internet. *Smart Home* terdiri dari beberapa sistem dan perangkat seperti *smart tv*, Stop kontak, *AC*, kipas angin dan saklar lampu yang menjadi bagian dari sistem *Smart Home*. Sistem *Smart Home* beroperasi menggunakan *Mikrokontroler* untuk memberikan segala pelayanan, keselamatan, keamanan dan penghematan energi yang berlangsung secara otomatis dan terprogram melalui komputer pada tempat tinggal. *Smart Home* telah dikembangkan untuk dapat digunakan mengendalikan hampir semua perlengkapan dan peralatan di rumah, mulai dari mengatur kecepatan kipas hingga mengendalikan berbagai alat-alat rumah tangga lainnya.

C. Model Perancangan *ADDIE*

Pada rumusan masalah yang sudah dikemukakan di atas, jenis penelitian yang digunakan pada penelitian ini yaitu penelitian pengembangan yang menghasilkan produk berupa *Trainer Smart Home*. Metode penelitian pengembangan adalah metode penelitian yang digunakan untuk menghasilkan suatu produk tertentu dan menguji keefektifan produk tersebut (Jampel, 2014). Penelitian ini menggunakan model *ADDIE* karena model pengembangan ini lebih dinamis, efektif dan mendukung kinerja program itu sendiri. Model ini memiliki lima langkah atau tahapan yang mudah dipahami dan diimplementasikan untuk mengembangkan produk pengembangan seperti buku ajar, modul pembelajaran, video pembelajaran, multimedia dan lain sebagainya. Model *ADDIE* memberi peluang untuk melakukan evaluasi terhadap aktivitas pengembangan pada setiap tahap, hal ini berdampak positif terhadap kualitas produk pengembangan. Model pengembangan *ADDIE* ini terdiri dari 5 komponen yang saling berkaitan dan memiliki struktur yang sistematis

yang berarti pada tahapan yang pertama sampai dengan tahapan yang terakhir atau kelima dalam pengaplikasiannya harus secara sistematis dan tidak bisa diurutkan secara acak.

Pada kelima tahapan ini sangat sederhana jika dibandingkan dengan model desain yang lainnya, karena sifatnya yang sederhana dan terstruktur dengan sistematis maka model desain ini mudah dipahami dan juga mudah untuk diaplikasikan (Kirna, 2013). Model pengembangan *ADDIE* terdiri dari lima tahapan yang meliputi analisis (*analysis*), desain (*design*), pengembangan (*development*), implementasi (*implementation*) dan evaluasi (*evaluation*) (Sugiyono, 2016).



Gambar 1. Alur Pengembangan Model *ADDIE*

D. Perangkat Keras

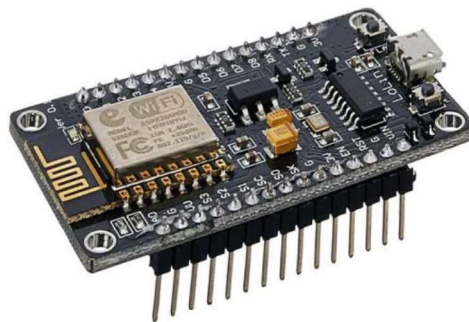
1. *NodeMCU ESP 8266*

NodeMCU ESP 8266 merupakan sebuah board dengan sistem *open source platform IOT* dan pengembangan kit yang menggunakan bahasa pemrograman *Lua* untuk membantu dalam membuat *prototype* produk *IOT* yang bisa diprogram dengan memakai sketch dengan *Arduino IDE*. Pengembangan kit ini didasarkan pada *Modul ESP8266*, yang mengintegrasikan *GPIO*, *PWM*

(Pulse Width *Modulation*), IIC, 1-Wire dan ADC (Analog to Digital Converter) semua dalam satu board (Christiyono, Y, 2019). *NodeMCU ESP8266* berukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. Board ini sudah dilengkapi dengan fitur WiFi dan Firmwarena yang bersifat open source.

Spesifikasi yang dimiliki oleh *NodeMCU ESP8266* sebagai berikut:

- a. Tegangan antarmuka komunikasi: 3.3V.
- b. Jenis antena: Tersedia antenna PCB internal.
- c. Standar nirkabel 802.11 b / g / n.
- d. Wi-Fi di 2.4GHz, mendukung mode keamanan WPA / WPA2.
- e. Mendukung tiga mode operasi STA / AP / STA + AP.
- f. koneksi Klein TCP (5 MAX).
- g. D0 ~ D8, SD1 ~ SD3: digunakan sebagai GPIO, *PWM*, IIC,
- h. Kemampuan driver port 15mA.
- i. AD0: 1 saluran ADC.
- j. Input daya: 4.5V ~ 9V (10VMAX), bertenaga USB.
- k. Kecepatan transfer: 110-460800 bps



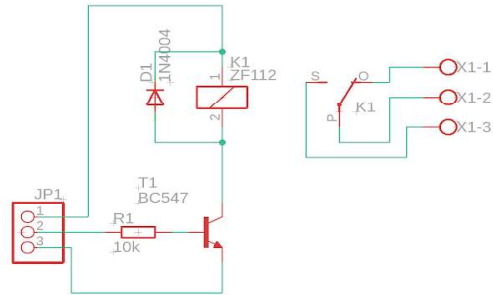
Gambar 2. Board NodeMCU ESP8266

2. *Relay Module*

Relay adalah sebuah komponen elektronika yang termasuk salah satu jenis saklar elektronik yang berfungsi sebagai penghubung dan pemutus aliran arus listrik yang dikontrol melalui *board Mikrokontroler* dengan memberikan tegangan dan arus tertentu pada koilnya. Dengan prinsip kerjanya yang menggunakan prinsip elektromagnetik membuat *Relay* dapat dikendalikan oleh rangkaian elektronik lainnya (Elga Aris Prasetyo, 2018). *Relay* pada umumnya hanya memiliki satu kumparan tetapi memiliki beberapa kontak.

Menurut Bishop didalam bukunya (2004: 54-55), Dalam menghubungkan atau memutuskan kontak digerakkan oleh fluksi yang ditimbulkan medan magnet listrik yang dihasilkan oleh kumparan yang melilit pada besi lunak. Prinsip kerja *Relay* adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetik. Jika sebuah penghantar dialiri oleh arus listrik, maka di sekitar penghantar tersebut timbul medan magnet. Medan magnet yang dihasilkan oleh arus listrik tersebut selanjutnya diinduksikan ke logam feromagnetik. Logam feromagnetik adalah logam yang mudah terinduksi medan elektromagnetik. Ketika ada induksi magnet dari lilitan yang membelit logam, logam tersebut akan menjadi magnet buatan yang sifatnya sementara yang akan memutus aliran listrik pada kakinya.

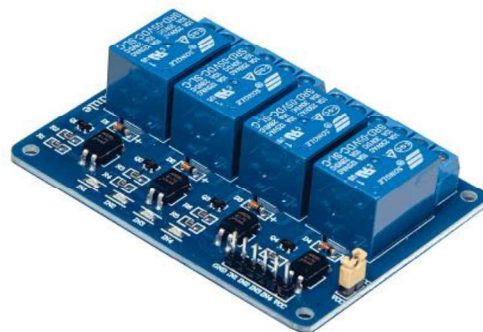
Sifat kemagnetan pada logam feromagnetik akan tetap ada selama pada kumparan yang melilitinya teraliri arus listrik. Sebaliknya, sifat kemagnetannya akan hilang jika suplai arus listrik ke lilitan diputuskan.



Gambar 3. Skema rangkaian *Relay Module*

Perbedaan prinsip kerja dari jenis konstruksi *Relay normally open*, *normally close*, dan *change over*:

- a. *NO (Normally Open)*: saklar dari *Relay* yang dalam keadaan normal (*Relay* tidak diberi tegangan) tidak terhubung dengan common.
- b. *NC (Normally Closed)*: saklar dari *Relay* yang dalam keadaan normal (*Relay* tidak diberi tegangan) terhubung dengan common.
- c. *Change Over*: saklar dari *Relay* hanya akan berpindah secara manual melalui sakelar atau dari jarak jauh melalui rangkaian



Gambar 4. *Relay Module*

3. *Catu Daya Switching*

Catu Daya Switching adalah sebuah rangkaian elektronika yang berfungsi untuk merubah arus AC menjadi arus DC dengan besar tegangan tertentu sesuai yang dibutuhkan. Sebuah catu daya Swiching pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah Transformer, Rectifier, Filter dan Voltage IC. Catu daya Swiching memiliki tegangan input 220 VAC dan tegangan output 12 Vdc 3A.



Gambar 5. Model catu daya

4. *Router (Modem)*

Router adalah salah satu peralatan jaringan yang dikhususkan untuk menghubungkan beberapa jaringan, baik itu jaringan yang sama maupun jaringan yang berbeda dari sisi teknologinya (Prawiro, 2018). Router bekerja dengan melihat alamat asal dan alamat tujuan dari paket yang melewatinya dan memutuskan rute yang akan dilewati paket tersebut untuk sampai ketujuan



Gambar 6. Router

5. Saklar Tukar

Saklar tukar atau biasa dikenal saklar hotel merupakan suatu saklar yang digunakan untuk mengendalikan (mematikan dan menghidupkan) sebuah lampu di dua tempat yang berbeda. Misalkan pada suatu ruangan bertingkat (lantai 1 dan lantai 2) dimana lampu berada di lantai 2, maka penempatan saklar hotel ini biasanya diletakkan di lantai 1 dan lantai 2. Jadi lampu tersebut dapat dimatikan dan dinyalakan dari lantai 1 dan lantai 2.



Gambar 7. Saklar tukar

Karena sangat sering sekali dipakai di perhotelan maka saklar jenis ini disebut sebagai saklar hotel. Selain itu nama lain dari saklar jenis ini adalah saklar tangga dikarenakan sangat sering dipasang di dekat tangga.

6. Stop kontak

Stop kontak merupakan instalasi listrik yang berfungsi sebagai penghubung antara arus listrik dengan peralatan listrik yang akan di gunakan. Untuk komponen yang terlihat pada Stop kontak dibagi menjadi 2 bagian utama yaitu bagian dasar dan bagian penutup.



Gambar 8. Stop kontak

Bagian dasar pada Stop kontak mempunyai fungsi sebagai dudukan colokkan dari peralatan listrik yang akan disambungkan ke bagian Stop kontak. Sedangkan bagian penutup berfungsi untuk melindungi bagian dasar dari Stop kontak. Selain itu bagian penutup ini juga memiliki fungsi estetika atau memperindah tampilan dari Stop kontak. Untuk komponen yang tidak terlihat secara langsung, kita baru bisa akan melihat bagian dalam Stop kontak ini setelah membuka bagian penutup dan membuka sekrup yang ada di bagian dasar.

7. *Smartphone Android*

Smartphone android merupakan telepon genggam yang menggunakan sistem operasi berbasis *android* yang mana Menurut (Hermawan, 2011), *Android* merupakan OS (Sistem Operasi) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini di tengah berkembangnya OS lainnya

seperti *Windows Mobile*, *OS iPhone*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini menjalankannya dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat dari potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, ada keterbatasan dari aplikasi pihak ketiga untuk mendapatkan Data asli ponsel atau *Smartphone*, percakapan antar proses serta distribusi dari aplikasi pihak ketiga untuk platform mereka.



Gambar 9. Logo Android

E. Perangkat Lunak

1. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA . Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, seperti:

- a. Sistem build berbasis Gradle yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android

- d. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- e. Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- f. Framework dan alat pengujian yang lengkap Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- g. Dukungan C++ dan NDK
- h. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.



Gambar 10. Logo Android Studio

2. Arduino IDE

IDE (Integrated Development Environment) yang diperuntukan untuk membuat perintah atau *source code*, melakukan pengecekan kesalahan, kompilasi, upload program, dan menguji hasil kerja *Arduino* melalui serial monitor.



Gambar 11. Tampilan Menu Arduino IDE

Pada Gambar 11, *Arduino IDE* memiliki toolbars *IDE* yang memberikan akses instan ke fungsi-fungsi yang penting yaitu :

- a. Tombol *Verify*, untuk mengkompilasi program yang saat ini dikerjakan.
- b. Tombol *Upload*, untuk mengkompilasi program dan mengupload keboard *Arduino* atau di *NodeMCU ESP8266*.
- c. Tombol *News*, menciptakan lembar kerja baru.
- d. Tombol *Open*, untuk membuka program yang ada di file sistem.
- e. Tombol *Save*, untuk menyimpan program yang dikerjakan.
- f. Tombol *Stop*, untuk menghentikan serial monitor yang sedang dijalankan.

3. Firebase

Firebase merupakan suatu layanan dari Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya *Firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. *Firebase* memiliki produk utama, yaitu menyediakan *Database* realtime dan backend sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi *API* yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan

disimpan di cloud *Firebase* ini. *Firebase* menyediakan library untuk berbagai client platform yang memungkinkan integrasi dengan *Android*, *iOS*, *JavaScript*, *Java*, *Objective-C* dan Node aplikasi Js dan dapat juga disebut sebagai layanan *DbaaS (Database as a Service)* dengan konsep realtime.

Firebase bekerja melalui Internet. Ini berarti hardware yang kita pilih harus bisa terhubung ke internet. Beberapa board, seperti *Arduino Uno* memerlukan *Ethernet* atau *Wifi Shield* untuk berkomunikasi, sedangkan board yang lain sudah mengaktifkan Internet-nya; seperti *ESP8266*, *Raspberry Pi* dengan dongle WiFi, *Particle Photon* atau *SparkFun Firebase Board*. Tetapi bahkan jika Anda tidak memiliki shield, kita dapat menghubungkannya dengan USB ke laptop atau desktop. Aplikasi *Firebase* dirancang dengan program antarmuka yang baik, dapat bekerja pada *iOS* dan *Android*



Gambar 12. Logo Firebase

4. *Google Assistant*

Google Assistant adalah evolusi fitur *Android* yang dikenal sebagai *Google Now*, yang memberi informasi tanpa bertanya. *Google Assistant* diluncurkan pada Mei 2016 sebagai bagian dari aplikasi perpesanan cerdas *Google Allo*. Tidak lama setelah itu, *Google Assistant* melakukan lompatan ke ponsel *Google Pixel*. *Google Assistant* hanya tersedia di lini *Pixel* untuk sementara waktu,

tetapi sekarang gratis untuk diunduh bagi siapa pun yang menggunakan *Android* 5 Lollipop atau lebih tinggi. Selain dukungan *Android Wear*, ada aplikasi *Google Assistant* untuk iOS. Dan jajaran speaker pintar google Home yang didukung oleh *Google Assistant* juga (Google,2016).

5. *Html (Hypertext Markup Language)*

Hypertext Markup Language adalah bahasa markah standar untuk dokumen yang dirancang untuk ditampilkan di peramban internet. Ini dapat dibantu oleh teknologi seperti *Cascading Style Sheets* dan bahasa skrip lainnya seperti *JavaScript*, *VBScript*, dan *PHP* (Ardhana, 2012).

6. *PHP (Hypertext Preprocessor)*

PHP atau Hypertext Preprocessor adalah Bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web. Propocessor yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server (Yudhanto & Prasetyo 2019).

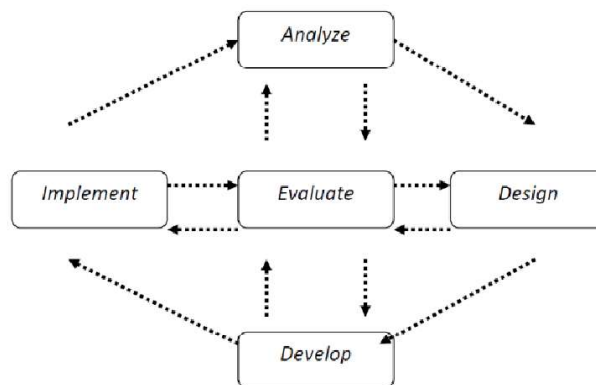
7. *Java*

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer standalone ataupun pada lingkungan jaringan (DeCoster, 2012).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Penelitian ini menggunakan model *ADDIE* (*Analyze, Design, Development, Implementation, Evaluation*) karena model pengembangan ini lebih dinamis, efektif dan mendukung kinerja program itu sendiri. Model ini memiliki lima langkah atau tahapan yang mudah dipahami dan diimplementasikan untuk mengembangkan produk pengembangan seperti buku ajar, modul pembelajaran, video pembelajaran, multimedia dan lain sebagainya. Model *ADDIE* memberi peluang untuk melakukan evaluasi terhadap aktivitas pengembangan pada setiap tahap, hal ini berdampak positif terhadap kualitas produk pengembangan. Model pengembangan *ADDIE* ini terdiri dari 5 komponen yang saling berkaitan dan memiliki struktur yang sistematis yang berarti pada tahapan yang pertama sampai dengan tahapan yang terakhir atau kelima dalam pengaplikasiannya harus secara sistematis dan tidak bisa diurutkan secara acak.



Gambar 13. Alur Pengembangan Model *ADDIE*

A. Analisis Sistem

Pada perancangan sebuah sistem, perlu diperhatikan bagaimana analisis sistem yang akan dibuat, diantaranya:

1. Analisis Sistem Berjalan

Analisis sistem yang sedang berjalan bertujuan untuk memberikan Gambaran tentang sistem yang sedang berjalan, yang bertujuan untuk mengetahui lebih jelas cara kerja sistem sehingga dapat dipahami kelebihan dan kekurangan sistem. Analisis sistem yang sedang berjalan adalah sebagai berikut:

a. Analisis Proses Bisnis

Analisis proses bisnis adalah kumpulan dari aktivitas atau tugas yang saling terkait untuk memecahkan masalah tertentu atau menghasilkan produk atau layanan. Untuk informasi yang lebih detail, berikut proses bisnis yang saat ini berjalan pada tabel di bawah ini :

Tabel 1. Analisis proses bisnis

No	Proses Bisnis	Aktivitas Rinci	User Terkait
1.	Kontrol lampu	a. Pemilik rumah mematikan atau menghidupkan lampu dengan menekan saklar lampu secara manual dengan menghampiri saklar	Pemilik Rumah
2.	Kontrol kipas angin	a. Pemilik rumah mematikan atau menghidupkan kipas angin dengan menekan saklar lampu secara manual dengan menghampiri saklar. b. Pemilik rumah mengatur kecepatan kipas angin dengan menekan saklar kipas secara manual dengan menghampiri saklar	Pemilik Rumah

3.	Kontrol kunci pintu	a. Pemilik rumah mengunci atau membuka kunci pintu menggunakan anak kunci dengan menghampiri pintu	Pemilik Rumah
4.	Kontrol Stop kontak	a. Pemilik rumah mematikan dan menghidupkan Stop kontak dengan menghampiri Stop kontak lalu mencabut Stop kontak dari sumber listrik	Pemilik Rumah
5.	Kontrol pemanas air minum	a. Pemilik rumah memanaskan air secara manual dengan menStop kontak perangkat pemanas ke Stop kontak dan mencabut Stop kontak ketika air sudah matang	Pemilik Rumah

b. Analisis Pelaku Sistem

Analisis pelaku sistem merupakan analisis untuk menentukan orang yang terlibat dalam sistem beserta fungsi dan tugasnya masing-masing. Untuk lebih jelasnya dapat dilihat dari tabel berikut:

Tabel 2. Analisis pelaku bisnis

No	Pelaku Sistem	Keterangan
1.	Pemilik Rumah	Mengontrol Lampu, kipas angin, pemanas air minum, Stop kontak dan kunci pintu secara manual

c. Analisis Aturan Bisnis

Analisis aturan bisnis merupakan penjelasan aturan bisnis yang sedang berjalan pada sistem saat ini.

Tabel 3 . Analisis Aturan Bisnis

No	Pelaku Sistem	Keterangan
1.	Pemilik rumah	a. Pemilik rumah harus mematikan dan menghidupkan lampu secara manual b. Pemilik rumah harus mematikan dan menghidupkan kipas secara manual c. Pemilik rumah harus mematikan dan menghidupkan Stop kontak secara manual d. Pemilik rumah harus mematikan dan menghidup pemanas air secara manual e. Pemilik rumah harus membuka dan mengunci pintu secara manual

d. Analisis Masalah dan Solusi

Analisis permasalahan dan solusi adalah menganalisa masalah-masalah apa saja yang terjadi dilapangan dan solusi yang diberikan untuk menyelesaikan masalah-masalah tersebut. Pada sistem ini, permasalahan dan solusinya adalah sebagai berikut:

Tabel 4. Analisis Masalah dan Solusi

No	Masalah	Solusi
1.	<i>User</i> harus mendekat ke saklar saat ingin menghidupkan dan mematikan perangkat listrik rumah tangga	Pada sistem dapat menghidupkan dan mematikan perangkat tanpa harus mendekati saklar manual
2.	Saklar konvensional tidak dapat dikendalikan oleh penyandang disabilitas	Pada sistem dapat dikendalikan oleh penyandang disabilitas
3.	Terbatasnya Penggunaan saklar konvensional dimana pengguna tidak dapat memonitoring setatus perangkat.	Pada sistem dapat mengontrol dari jarak jauh selama terhubung dengan internet

4.	Terbatasnya jumlah perangkat yang bisa dikendalikan dengan sebuah <i>kontroler</i> berbasis <i>Bluetooth</i> .	Pada sistem dapat mengontrol banyak perangkat
5.	Terbatasnya pilihan pengontrolan perangkat konvensional.	Pada sistem dibangun aplikasi <i>Smart Home</i> include dengan fitur <i>Voice recognition</i>
6.	Belum dikembangkannya <i>Smart Home</i> berbasis <i>voice recognition</i> menggunakan <i>Mikrokontroler</i> .	Pada sistem dibangun menggunakan <i>mikrokontroler</i> dengan fitur tambahan <i>voice recognition</i>

2. Analisis Sistem yang Diusulkan

Setelah menganalisis sistem berjalan, maka analisis sistem yang diusulkan sebagai berikut :

a. Analisis *User*

Analisis *user* bertujuan untuk mengetahui siapa saja *user* yang terlibat dalam sistem. Setiap *user* memiliki fungsi dan tugas yang berbeda, sehingga dapat diketahui tingkat pengalaman dan pemahaman *user* terhadap sistem dan untuk sistem ini melibatkan satu *user*, antara lain:

Tabel 5. Analisis *User*

No	Nama <i>User</i>	Aktivitas
1	<i>User</i>	<ul style="list-style-type: none"> a. Menghidupkan dan mematikan lampu b. Menghidupkan dan mematikan kipas angin c. Mengatur kecepatan kipas angin d. Membuka dan mengunci pintu e. Menghidupkan dan mematikan pemanas air f. Menghidupkan dan mematikan Stop kontak g. Memantau aktivitas dan histori perangkat

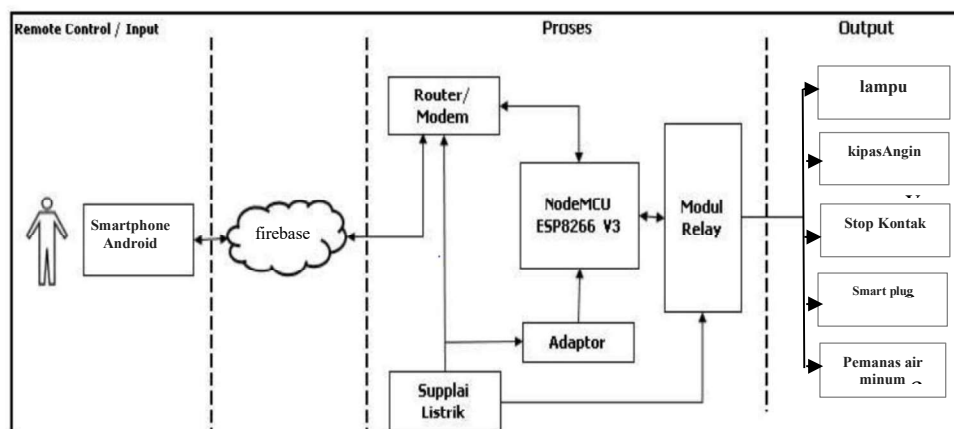
b. Analisis Prosedur

Analisis proses sistem memberikan Gambaran tentang sistem yang akan diusulkan. Prosedur sistem ini bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui.

Tabel 6. Analisis Prosedur

No	Aktivitas	User Terkait
1.	Menginstal aplikasi	User
2.	Memastikan perangkat terhubung jaringan internet	
3.	Membuka aplikasi pada perangkat <i>Smartphone</i>	
4.	Memilih menu	
5.	Mengontrol perangkat elektronik yang terdapat dalam aplikasi	
6.	Keluar aplikasi	
7.	Mengucapkan perintah melalui <i>Google Assistant</i>	

c. Flowmap yang diusulkan



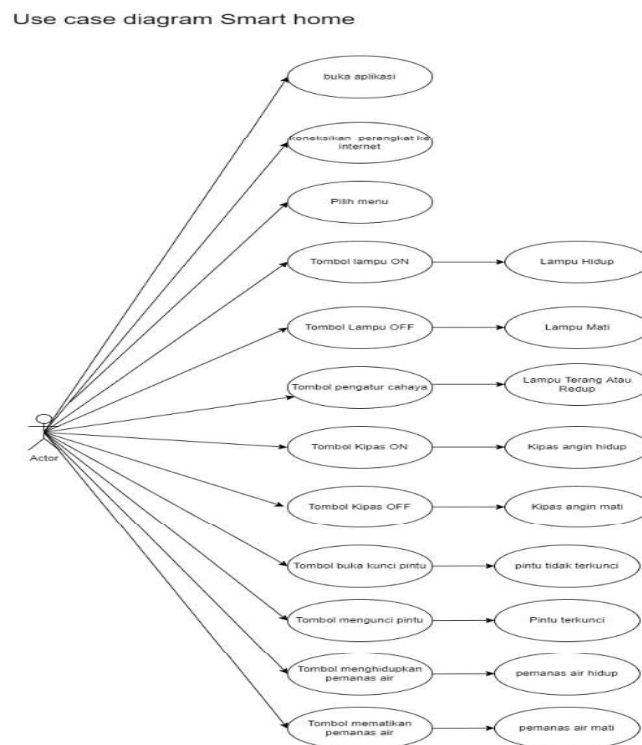
Gambar 14. flowmap Yang Diusulkan

B. Perancangan Sistem

Tahap selanjutnya adalah melakukan perancangan sistem yaitu untuk membuat pemodelan terhadap aplikasi atau sistem sehingga dapat mengatasi masalah yang terdapat pada sistem yang berjalan saat ini. Perancangan sistem ini menggunakan diagram *Unified Modelling Language (UML)*.

1. Use Case Diagram

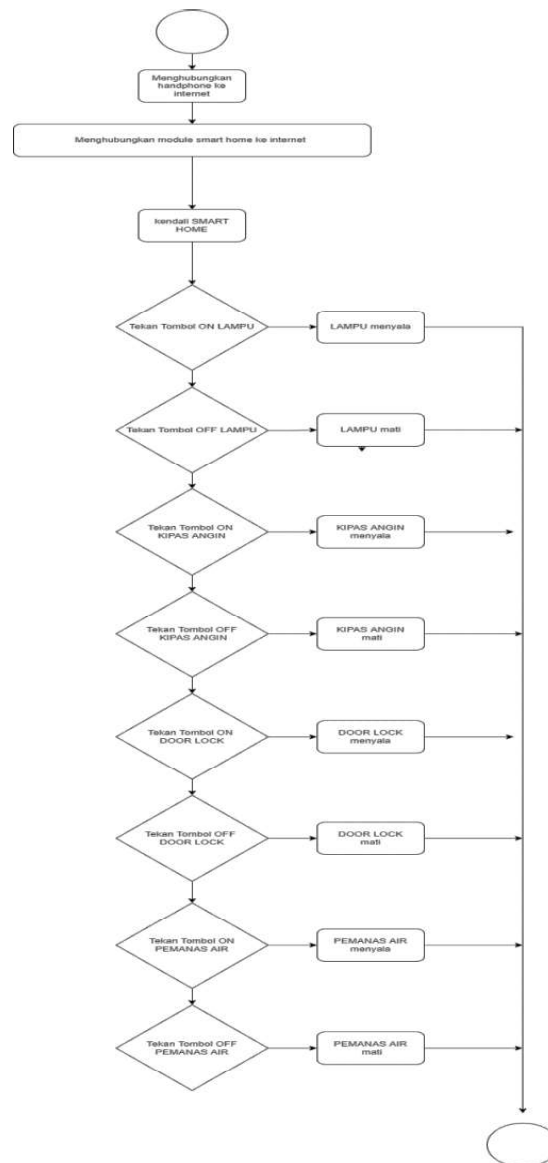
Use Case Diagram merupakan abstraksi dari interaksi antara sistem dan user. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara *user* dalam sebuah sistem dengan sistemnya sendiri melalui sebuah diagram bagaimana sistem itu dipakai. Perancangan *Use Case Diagram Smart Home* seperti pada Gambar berikut:



Gambar 15. Use Case Diagram

2. Activity Diagram

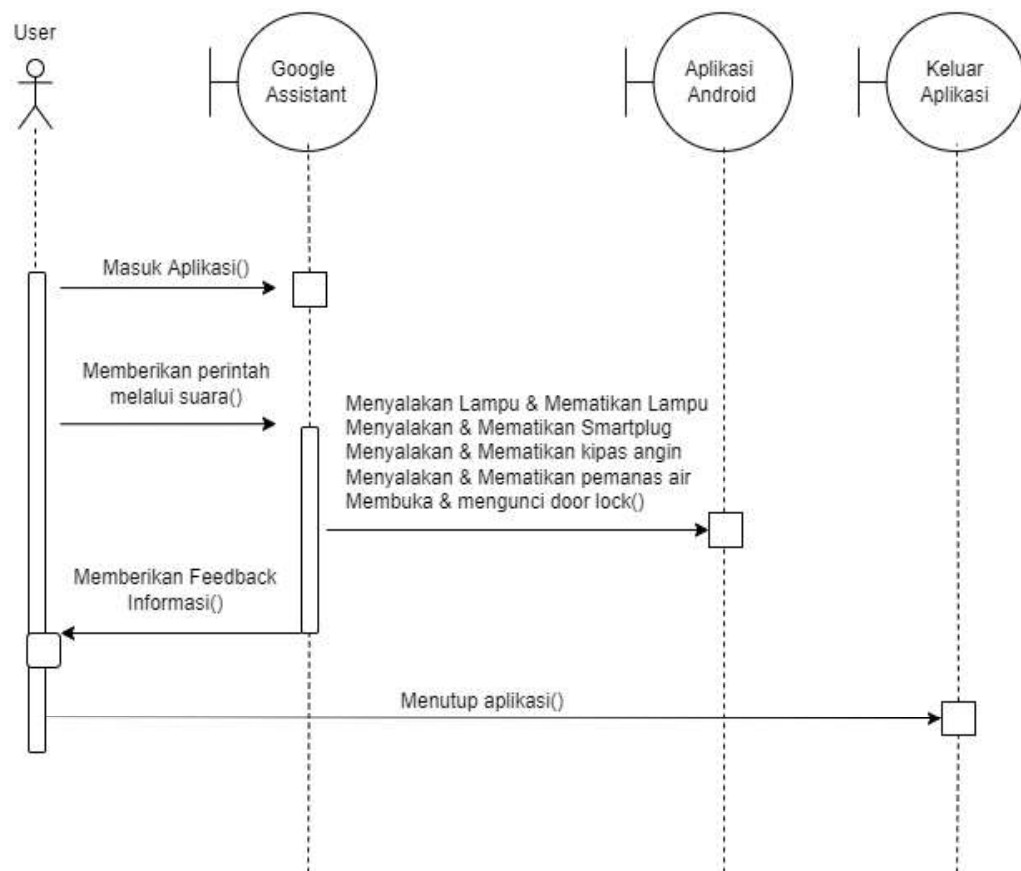
Activity Diagram menggambarkan aliran kerja (*workflow*) atau aktifitas dari sebuah sistem yang berjalan. Berikut adalah diagram aktivitas dari sistem ini:



Gambar 16. Activity Diagram

3. Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Sequence diagram berguna untuk menunjukkan rangkaian data yang dikirim antara object serta interaksi antara object. Berikut adalah diagram Sequence dari sistem ini :



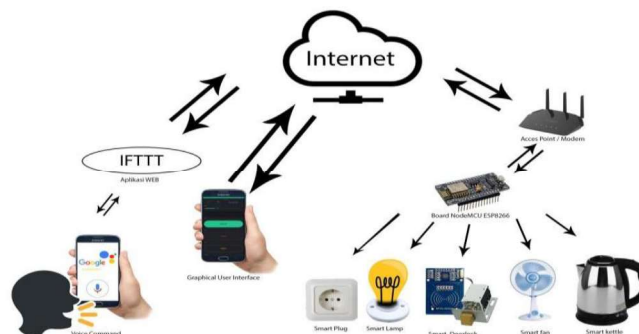
Gambar 17. Sequence Diagram

Berdasarkan *Sequence Diagram* di atas maka dibuat langkah-langkah penjelasan scenario dibawah ini:

- a. *User* memberikan perintah kondisi ke *Google Assistant*.
- b. *Google Assistant* mengirim perintah yang sudah di diberikan oleh *User* ke Sistem perangkat dengan contoh *Turn on the light* untuk menyalakan lampu.
- c. Kemudian *Google Assistant* memberikan *feedback* informasi yaitu *light turned on* atau lampu sudah menyala.
- d. Kemudian sistem kendali menerima kondisi untuk mengendalikan alat sesuai perintah.

C. Perancangan Alat

Sistem *Smart Home* merupakan sistem yang dirancang untuk memudahkan *user* dalam mengontrol peralatan elektronik rumah tangga melalui jaringan internet, menghidupkan dan mematikan peralatan elektronik dari mana saja dan kapan saja tanpa perlu mendekati saklar atau peralatan hanya untuk sekedar menghidupkan atau mematikan. Berikut ini adalah Gambar ilustrasi dari bentuk sistem *Smart Home*.



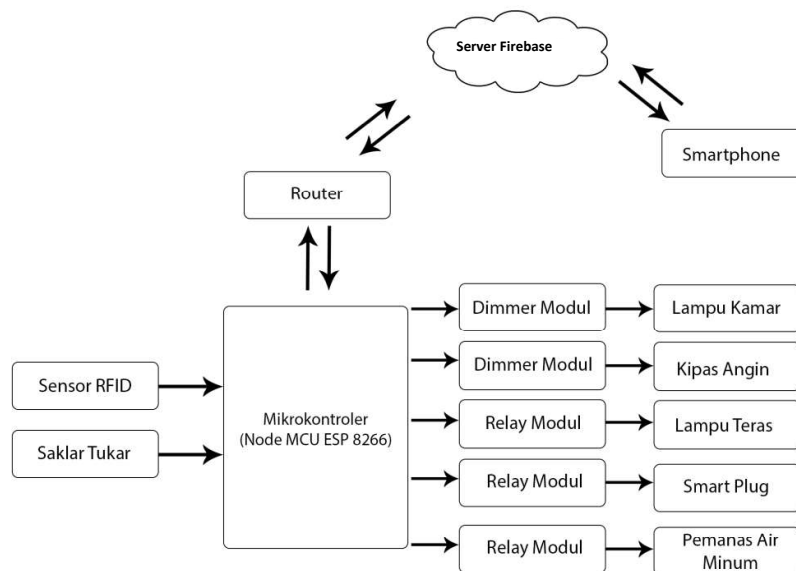
Gambar 18. Skema *Smart Home*

Seperti yang telah dijelaskan pada bab sebelumnya, pada bagian ini akan menjelaskan tentang perencanaan dan langkah pembuatan dari hardware dan software, Sistem ini menggunakan *NodeMCU ESP8266* sebagai *kontroler* dari sistem perangkat *Smart Home*. penggunaan *Mikrokontroler* ini berfungsi sebagai mediator untuk rangkaian monitoring, kontroling jarak jauh, dan untuk mendapatkan informasi data pada perangkat.

Dalam perancangan sistem alat ada yang dinamakan dengan blok input, proses dan juga output. Masing-masing dari blok ini saling bersinergi sehingga terjadi suatu sistem otomatisasi yang baik dan benar.

1. Diagram Blok Rangkaian

Dalam perancangan dan pembuatan sistem diperlukan sebuah blok diagram sebagai berikut:



Gambar 19. Blok diagram rancangan alat

Berdasarkan diagram blok diatas dapat diketahui terdapat beberapa blok yang fungsi masing-masingnya yaitu:

a. *Mikrokontroler ESP8266*

Mikrokontroler ESP8266 pada alat ini berfungsi sebagai pusat pengendali yang akan mengontrol keseluruhan sistem dan juga *Modul WiFi* pada *ESP8266* berfungsi sebagai platform pengolah data dan komunikasi antara alat dengan *User*. Pada *Smart Home* sebagai pengendali sensor *RFID*, kipas angin, lampu, stop kontak dan pemanas air minum.

b. *Sensor RFID*

Sensor RFID pada alat ini berfungsi sebagai *security* yang akan membaca *user* melalui sidik jari yang nantinya akan dipasangkan dengan sistem *door lock*, kemudian data informasi yang terukur dikirimkan ke *Mikrokontroler*.

c. *Relay Module*

Relay Modul pada alat ini berfungsi sebagai driver untuk mengendalikan beberapa aktuator seperti *door lock*, kipas angin, lampu, dan pemanas air minimum.

d. *Kebutuhan Software*

Software yang digunakan dalam pembuatan program *sistem Trainer Smart Home* dengan 3 bahasa pemograman adalah *Arduino IDE*.

2. Flowchart Sistem Kontrol

Flowchart yang digunakan dalam menjelaskan alur sistem kontrol dari aplikasi *Smart Home* ditunjukkan oleh Gambar berikut.



Gambar 20. *Flowchart sistem*

D. Prinsip Kerja Alat

Berdasarkan perancangan sistem blok diagram dan flowchart dapat dideskripsikan bahwa perancangan dan pembuatan *Trainer Smart Home* berpusat pada *Mikrokontroler NodeMCU ESP8266* sebagai pengendali utama. Prinsip kerja sistem ini adalah pada awalnya kita harus memastikan seluruh perangkat dalam keadaan menyala dan mengkoneksikan board *NodeMCU ESP8266* kedalam jaringan wifi untuk terhubung dalam jaringan internet, jika sudah mengkoneksikan dengan jaringan internet maka akan ditampilkan status *connected* pada *graphical user interface* di aplikasi *Android* yang sudah diinstal, setelah itu kita bisa

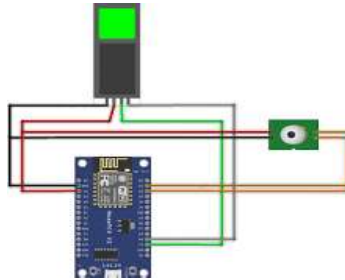
menggunakan sistem ini melalui *GUI* dan juga menggunakan perintah suara dalam pengoperasiannya. Untuk mengendalikan sistem *Smart Home* menggunakan perintah suara kita bisa mulai dengan menekan *button* pada *GUI* dan mengucapkan perintah sesuai dengan perintah yang telah didaftarkan, setelah selesai maka Google akan memproses perintah tersebut dan mengirimkan ke server *Firebase* dan melanjutkan ke *Mikrokontroler* untuk menyalakan atau mematikan perangkat

Pada salah satu bagian alat ini yaitu *Smart door lock* dirancang menggunakan 1 buah sensor yaitu sensor *RFID*. Sensor *RFID* berfungsi sebagai *security* dan akses kontrol *user* pada sistem *door lock* selanjutnya data yang terukur oleh sensor *RFID* akan dikirimkan kepada *Mikrokontroler NodeMCU ESP8266*. Data tersebut kemudian diolah sesuai dengan set point yang telah ditentukan untuk membuka kunci pada sistem *door lock*. Semua aktivitas pengontrolan pada sistem *Smart Home berbasis Voice Command* dapat di monitoring dan juga dikontrol langsung melalui *Smartphone*. Alat ini dilengkapi dengan *Modul Wifi ESP8266*, sehingga alat dapat di monitoring dan dikontrol dimana dan kapanpun asalkan terhubung dengan jaringan internet

E. Perancangan Perangkat Keras

1. Skema Rangkaian Sensor *RFID*

Skema rangkaian ini berfungsi untuk menggambarkan koneksi pin *RFID* pada *Mikrokontroler* sehingga *RFID* mampu bekerja sesuai dengan fungsinya.

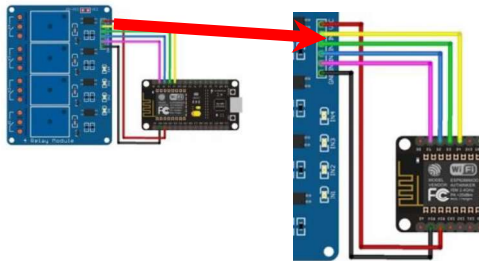


Gambar 21. Skema Rangkaian Sensor *RFID*

2. Skema Rangkaian *Relay Module*

Rangkaian driver *Relay* pada alat ini bekerja berdasarkan nilai output dari pembacaan sensor *RFID* dan Input dari *user* yang kemudian digunakan untuk mengontrol aktuator yang ada pada *Smart Home*.

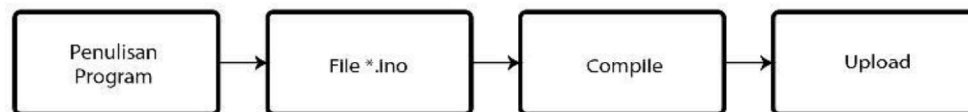
Gambar skema rangkaian driver *Relay* seperti berikut:



Gambar 22. Skema Rangkaian *Relay*

F. Perancangan Perangkat Lunak

Pembuatan program untuk *Mikrokontroler NodeMCU ESP8266* ini yaitu menggunakan bahasa C/C++ pada ketentuan *Arduino*. Proses perancangan program pada Gambar 23.

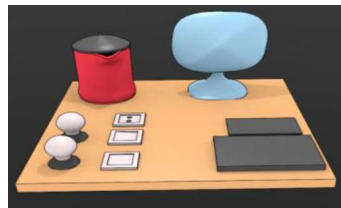


Gambar 23. Proses Perancangan Perangkat Lunak Pada *Mikrokontroler*

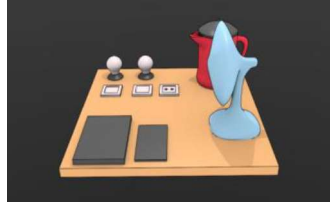
Penelitian program bahasa C/C++ dilakukan menggunakan aplikasi *Arduino IDE*. Listing program yang telah buat nantinya akan tersimpan dalam format *.ino. Selanjutnya lakukan proses kompilasi program untuk mengecek kesalahan pada baris program. Kemudian apabila tidak mendapati kesalahan dalam penelitian program maka program dapat di upload sesuai board *Mikrokontroler* yang digunakan. Maka setelah proses upload selesai *Mikrokontroler* dapat digunakan sesuai dengan program yang telah dimasukan.

G. Rancangan Fisik Alat

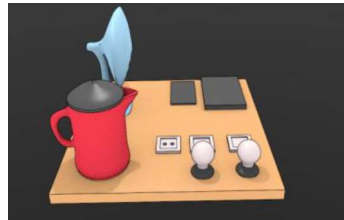
Rancangan fisik alat bertujuan agar memudahkan dalam memberikan Gambaran bentuk sistem yang akan dirancang. Dalam pembuatan fisik alat peneliti menggunakan bahan-bahan yang mudah didapat untuk pembuatan mekanik.



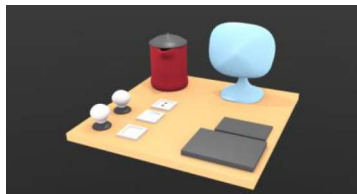
Gambar 24. Desain tampak dari depan



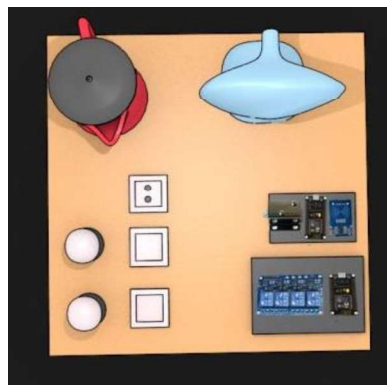
Gambar 25. Desain tampak dari kiri



Gambar 26. Desain Tampak dari kanan



Gambar 27. Desain tampak keseluruhan



Gambar 28. Gambar tampak atas

Trainer Smart Home ini memiliki spesifikasi sebagai berikut:

1. Ukuran Alat : 100 cm x 70 cm x 45 cm (p x l)
2. Tegangan Kerja : 12 VDC / 110-220 VAC / 50Hz / 50 Watt
3. *Mikrokontroler* : *NodeMCU*
4. Sensor : *RFID*
5. Monitoring : Aplikasi *Android*
6. Controlling : Aplikasi *Android* dan *Google Assistant*
7. *Modul WiFi* : *ESP8266*
8. Jenis perangkat : Kipas angin, pemanas air, 2 buah lampu dan smart *door lock*
9. Bahan : Triplek 5mm, Kayu 1 M

H. Perancangan Sistem Antarmuka

Sistem kontrol dan monitoring dapat dilakukan melalui aplikasi *Android*. Aplikasi ini menggunakan *base API Firebase app*. Aplikasi dihubungkan dengan server *Firebase.co* untuk mengambil data yang telah dikirim oleh *NodeMCU ESP8266* ke server *Firebase*. Data akan hanya akan tampil jika terhubung dengan koneksi internet.



Gambar 29. Rancangan sistem antarmuka

Pada Gambar 29 menunjukkan tampilan aplikasi saat terhubung dengan jaringan internet sehingga data dari server dapat ditampilkan. Jika aplikasi tidak terhubung dengan jaringan internet, maka data tidak akan ditampilkan dan menandakan adanya gangguan pada jaringan internet. Pada aplikasi tersedia dua tombol yaitu tombol *refresh* untuk menyegarkan data dari *server* dan tombol info untuk melihat informasi terkait aplikasi.

BAB IV HASIL DAN PEMBAHASAN

A. Hasil Pembuatan Alat

1. Bentuk Fisik Alat Keseluruhan

Setelah dilakukan pembuatan alat berdasarkan perancangan sebelumnya maka dapatlah hasil Bentuk fisik dari trainer secara keseluruhan dapat dilihat pada Gambar 30. Yang mana dengan ukuran alas 50 CM x 60 CM dapat menampung semua *modul* trainer dan juga disusun dengan posisi selayaknya alat rumah elektronik pada umumnya. Nantinya diharapkan dengan posisi alat seperti ini dapat mempermudah siswa menggunakannya untuk latihan. Serta tidak membuat siswa kebingungan dengan posisi dan jalur perkabelan yang dapat dilihat dan dipahami dengan mudah.



Gambar 30. Tampak keseluruhan



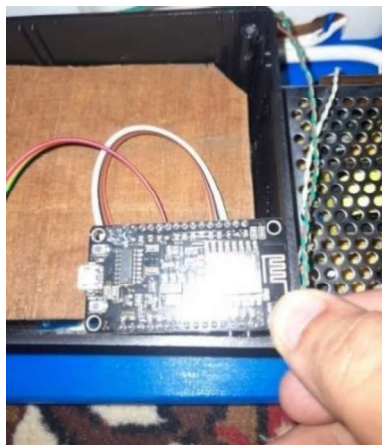
Gambar 31. Posisi Doorlock dan Posisi Kipas Angin



Gambar 32. Posisi Lampu dan Posisi Pemanas Air Minum

2. Posisi Mikrokontroler Node MCU

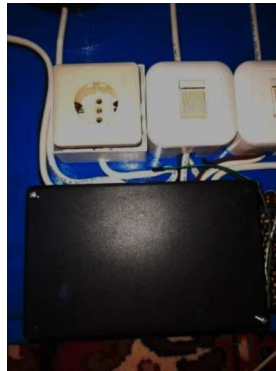
Posisi *Mikrokontroler Node MCU* di letakan pada sebuah *safety box* yang mana nantinya alat kelengkapan lainnya seperti *relay modul* juga diletakan didalam ini dengan tujuan untuk mempermudah akses pemograman dan *servis* ketika digunakan. Ketika nantinya terjadi kendala dalam pengoperasian modul ini teknisi tinggal mencari bagian yang bermasalah tanpa harus membakar susunan modul ini. Menggunakan *safety box* yang memiliki tutup sehingga dapat dibuka dan ditutup dengan mudah, agar nantinya tidak membuat user kesulitan dalam penggunaan dan membuat tampilan alat menjadi lebih rapi .



Gambar 33. Posisi Node MCU

3. Posisi Saklar

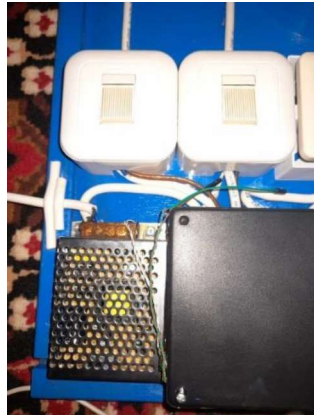
Dipasang dibagian sebelah kiri trainer tepat disebelah *power supply* bertujuan untuk menghemat kabel dan mempermudah perakitan yang mana saklar digunakan untuk menghidupkan lampu secara manual. Dalam sistem telah dibuat untuk menghidupkan dan mematikan lampu namun adakalanya sistem tidak dapat digunakan atau ingin menggunakan metode manual maka dengan tanpa menghilangkan metode manualnya maka peneliti menambahkan saklar untuk mengoperasikan peralatan secara manual.



Gambar 34. Posisi Saklar

4. Posisi *Power Supply*

Posisi *power supply* dipasang dibagian kiri *trainer* bertujuan untuk mempermudah penyusunan dan menghemat kabel. *Power supply* ini nantinya digunakan untuk membagi arus masuk yang digunakan untuk *modul* seperti. 220 v ac, 12 v dc, 9 v dc, 5 v dc dan 3 v dc. Untuk power supply ini sendiri peneliti memilih keluaran 12 V yang mana ini dinilai lebih efisien dalam membagi tegangan nantinya.



Gambar 35. Posisi *Power Supply*

B. Persiapan Pengoperasian Trainer *Smart Home*

Pengguna perlu untuk memperhatikan dan mempersiapkan beberapa hal berikut ini untuk menjalankan sistem:

1. Memastikan dan Mengukur Sumber Tegangan Pada Sistem

Sistem menggunakan sumber tegangan 220 Vac yang terhubung pada catu daya switching 12 Vdc. Agar sistem berjalan dengan baik user melakukan pengecekan terhadap tegangan keluaran dari catu daya swiching 12 Vdc serta tegangan keluaran dari *modul* step down IC 7890 dan IC 7850.

2. Menyediakan Akses Internet Untuk Sistem

Sistem memerlukan akses internet agar data dari sensor dapat dikirimkan ke firebase sehingga dapat diakses oleh aplikasi di *smartphone*. Pada system ini peneliti menggunakan modem yang dilengkapi kartu SIM. Pastikan kartu SIM sudah diaktifasi dan memiliki kuota internet. Kemudian juga pastikan SSID dan kata sandi dari WiFi disamakan dengan program.

C. Pengujian Perangkat Keras

Pengujian ini dilakukan untuk melihat apakah perangkat keras yang dibangun dapat berjalan dengan baik sesuai dengan perancangan. Pengujian yang dilakukan adalah dengan mengukur besaran arus yang dibutuhkan oleh perangkat untuk bekerja ataupun *standby* dalam menerima *input* dari *interface*.

1. Pengukuran *Power Supply* dan *IC* Tegangan

Pengukuran pada *Power Supply* dan *IC* tegangan dilakukan pada empat titik pengujian yaitu pada catu daya dan tiga step down switching *IC*. Step down switching *IC* diatur output tegangannya menjadi tiga variasi yang berbeda yaitu tegangan 9V, 5V dan 3,3V. Pada proses pengukuran peneliti menggunakan alat ukur sebuah multimeter Analog yang harus mengatur skala ukuran secara manual. Selanjutnya hasil pengukuran akan ditampilkan pada tabel 7.

Tabel 7. Pengukuran Tegangan

Titik Pengukuran	Tegangan (Vdc)
Keluaran catu daya	11.8
Keluaran regulator 9V	9.0
Keluaran regulator 5V	5.0
Keluaran regulator 3.3V	3.4

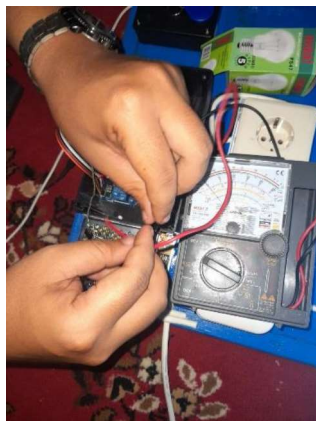
a. Pengukuran pada catu daya



Gambar 36. Pengukuran pada catu daya

Gambar 36 menunjukkan proses pengukuran pada keluaran catu daya. Dapat dilihat pada hasil pengukuran multimeter bahwa tegangan keluaran catu daya adalah 11.8V. Nilai ini sudah mendekati dengan nilai tegangan yang seharusnya yaitu 12V. Kemudian tegangan keluaran catu daya ini disalurkan pada ketiga step down switching *IC* untuk menurunkan tegangan agar tidak terjadinya konsleting.

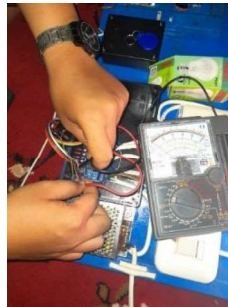
b. Pengukuran Pada Keluaran *IC* 9V



Gambar 37. Pengukuran Pada *IC* 9V

Gambar 37 menunjukkan proses pengukuran pada keluaran step down switching *IC* 9V. Dapat dilihat pada hasil pengukuran multimeter bahwa tegangan keluaran step down switching *IC* adalah 9.0Vdc. Tegangan 9Vdc ini digunakan sebagai tegangan masukan bagi nodeMCU. Apabila tegangan yang diberikan lebih dari 9V DC maka akan terjadi konsleting karena ketidak mampuannya modul *IC* bawaan board yang menerima beban lebih dari 9V dan untuk NodeMCU sendiri *relative* bekerja pada tegangan 3V sampai dengan 9V DC. Oleh karena itu jika tetap diberikan tegangan lebih misalnya 12V DC maka modul *IC* akan terjadi konsleting.

c. Pengukuran Pada Keluaran *IC* 5V



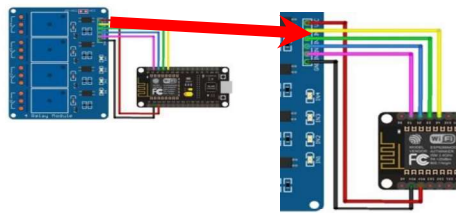
Gambar 38. Pengukuran Pada *IC* 5V

Gambar diatas menunjukkan proses pengukuran pada keluaran step down switching *IC* 5V. Dapat dilihat pada hasil pengukuran multimeter bahwa tegangan keluaran catu daya adalah 5,0Vdc. Tegangan 5Vdc ini digunakan sebagai tegangan masukan bagi sensor RFID dan *Relay Modul*.

2. Pengujian Driver *Relay*

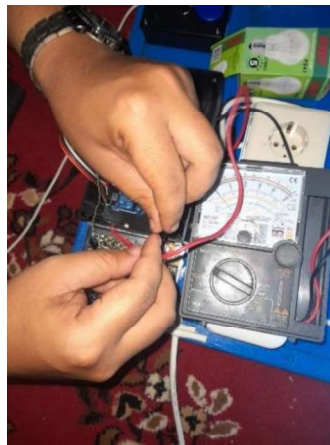
Pengujian dari *modul* ini dilakukan untuk mengetahui kondisi *relay* sudah berfungsi dengan lancar. Pengujian akan dilakukan dengan

menghubungkannya ke aplikasi arduino IDE dan kemudian melihat hasil eksekusi melalui led yang sudah terpasang pada *modul*. *Modul relay* akan diberikan program dengan masukan logika *LOW* dan *HIGH*. Berdasarkan datasheet *modul relay* bekerja pada *LOW* level, yang berarti *relay* akan aktif jika diberikan tegangan yang mendekati nilai 0V. Adapun set up rangkaian dari pengujian *modul relay* sebagai berikut:



Gambar 39. Setup Rangkaian *Relay*

Pada pengujian *modul relay*, *mikrokontroler* yang digunakan adalah nodeMCU. Untuk pin yang digunakan, yaitu pin data pada *relay* terhubung dengan pin D7, D6, D5 dan pin 10 pada *mikrokontroler*, lalu pin VCC pada *modul relay* terhubung dengan pin 5V pada *mikrokontroler* dan pin GND pada *modul relay* terhubung dengan ground pada *mikrokontroler*.



Gambar 40. pengukuran saat diberikan logika *LOW*

Pada kondisi *relay* dalam keadaan tidak aktif (*LOW*), led indikator yang terdapat pada *relay* tidak menyala. Tegangan yang terukur pada saat kondisi *relay* tidak aktif adalah 3,25 V.



Gambar 41. pengukuran saat diberikan logika *HIGH*

Pada kondisi *relay* dalam keadaan aktif (*HIGH*), led indikator merah akan menyala. Tegangan yang terukur pada saat kondisi *relay* aktif adalah 0,06 V. Hasil pengujian driver *relay* dapat dilihat pada table 8.

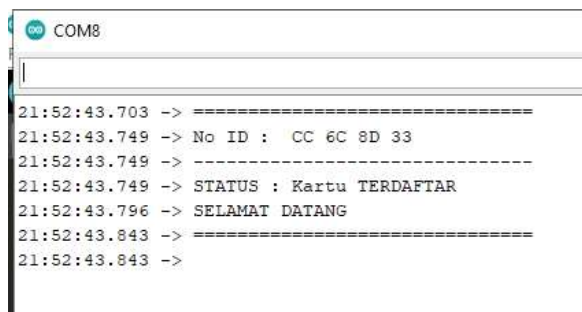
Tabel 8. Hasil pengukuran tegangan *relay*

<i>Logic Level</i>	<i>Kondisi Driver Relay</i>	
	Indikator LED	Tegangan Terukur (V)
HIGH	Mati	3,25
LOW	Hidup	0,06

Berdasarkan tabel hasil pengujian tersebut, dapat diketahui bahwa driver *relay* dapat bekerja dengan baik sesuai dengan datasheet. Oleh karena itu driver *relay* dapat digunakan pada sistem yang dibuat.

3. Pengujian Sensor Sistem Keamanan

Jenis sensor yang digunakan adalah RFID. Sensor ini diimplementasikan pada sistem keamanan *Smart Door Lock* yang mana digunakan untuk menunjang sistem keamanan *Smart Home* itu sendiri sehingga hanya yang memiliki akses saja nantinya yang dapat menggunakannya. Teknis pengujian adalah dengan menempelkan kartu dan tag kemudian melihat hasil pengujian pada *serial monitor* melalui aplikasi *Arduino IDE*. Hasil pengujian dapat kita lihat dalam Gambar 42.



```

COM8
21:52:43.703 -> =====
21:52:43.749 -> No ID : CC 6C 8D 33
21:52:43.749 -> -----
21:52:43.749 -> STATUS : Kartu TERDAFTAR
21:52:43.796 -> SELAMAT DATANG
21:52:43.843 -> =====
21:52:43.843 ->

```

Gambar 42. No ID kartu yang muncul saat *scanning*

D. Pengujian Perangkat Lunak

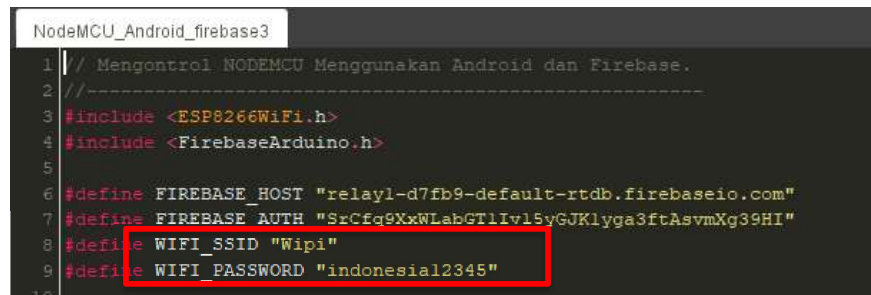
Pengujian ini bertujuan untuk melihat dan mengetahui kondisi dari sistem yang dibangun melalui dari segi perangkat lunak

1. Pengujian Pengiriman Data Ke Web Server

Pada pengujian ini, pengujian bertujuan untuk memastikan perangkat *Trainer* dapat terhubung dengan jaringan internet agar *mikrokontroler* dapat mengirimkan data ke web server.

a. Pengujian Koneksi Modul WiFi NodeMCU ESP8266

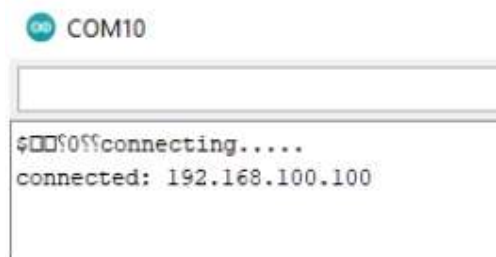
Pengujian yang dilakukan adalah untuk mengetahui keberhasilan dari modul WiFi terhubung ke akses poin yang sudah terhubung ke modem internet agar modul WiFi bisa mengirimkan data dari mikrokontroler ke web server. Pada pengujian modul WiFi ini, modul akan diatur agar bisa terhubung ke akses poin dengan SSID dan password yang terdapat pada akses poin. Untuk menghubungkan akses poin dengan modul WiFi, pertama masukan *SSID (Service Set Identifier)* dan *password* dari akses poin pada pemrograman arduino IDE seperti terlihat pada Gambar 43 dibawah ini :



```
NodeMCU_Arduino_firebase3
1 // Mengontrol NODEMCU Menggunakan Android dan Firebase.
2 //-----
3 #include <ESP8266WiFi.h>
4 #include <FirebaseArduino.h>
5
6 #define FIREBASE_HOST "relay1-d7fb9-default-rtdb.firebaseio.com"
7 #define FIREBASE_AUTH "SrCfQ9XxWLabGT1Iv15yGJKlyga3ftAsvmXg39HI"
8 #define WIFI_SSID "Wipi"
9 #define WIFI_PASSWORD "indonesial2345"
10
```

Gambar 43. SSID Dan Password Pada Pemrograman Arduino IDE

Apabila sudah memasukan SSID dan password dari akses poin, maka selanjutnya upload pemrograman ke modul WiFi dan lihat hasilnya pada serial monitor seperti pada Gambar 44 dibawah ini :



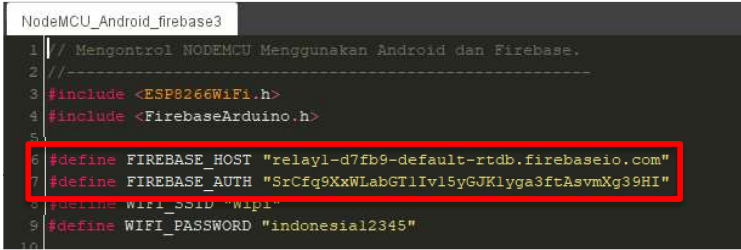
```
COM10
$$$$connecting....
connected: 192.168.100.100
```

Gambar 44. Tampilan Serial Monitor Terhubung Dengan Akses Poin

Kemudian akses poin akan terhubung ke modul WiFi apabila terdapat kata “*connected*” pada serial monitor dan terdapat juga alamat IP dari akses poin. Jadi modul WiFi sudah terhubung ke akses poin dan siap untuk ke jaringan internet. Berdasarkan hasil pengujian, modul WiFi berfungsi dengan baik dan dapat digunakan pada sistem yang dibuat.

b. Pengujian Data Terkirim Ke Web Server

Pengujian ini bertujuan untuk memastikan keberhasilan dari data yang dikirim dari mikrokontroler diterima oleh web server. Web server yang digunakan adalah web server Firebase. Cara pengujiannya yaitu pastikan modul WiFi sudah terhubung dengan jaringan internet. Setelah itu masukan alamat *FIREBASE_HOST* yang terdapat pada *Database* firebase ke dalam pemrograman arduino *IDE*. Masukan juga *FIREBASE_AUTH* sebagai ID secret dari *Database* yang dibuat pada Firebase seperti terlihat pada Gambar 45 dibawah ini :



```

NodeMCU_Android_firebase3
1 // Mengontrol NODEMCU Menggunakan Android dan Firebase.
2 //-----
3 #include <ESP8266WiFi.h>
4 #include <FirebaseArduino.h>
5
6 #define FIREBASE_HOST "relay1-d7fb9-default-rtdb.firebaseio.com"
7 #define FIREBASE_AUTH "SrCfq9XxWLabGtlIv15yGJKlyga3ftAsvmXg39HI"
8 #define WIFI_SSID "wipi"
9 #define WIFI_PASSWORD "indonesial2345"
10

```

Gambar 45. Token *FIREBASE_HOST* Dan *FIREBASE_AUTH*

Setelah memasukan kode token *FIREBASE_HOST* dan *FIREBASE_AUTH* maka setelah melakukan *compile* maka akan terlihat

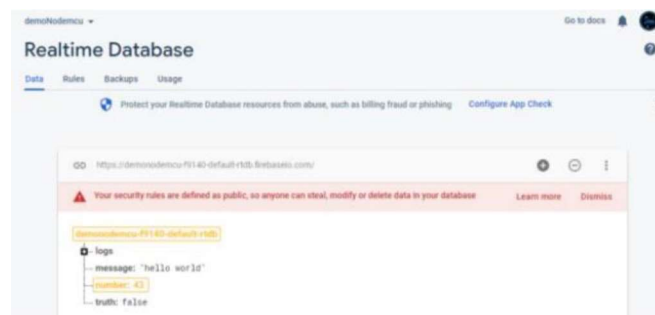
pada serial monitor data yang dikirim ke Firebase seperti terlihat pada Gambar:



```
COM10
pushed: /logs/-Mewfy9CV0e-0dDuoJor
number: 43.00
pushed: /logs/-Mewg-Jz054D0Ygth0dM
number: 43.00
pushed: /logs/-MewgIXtbBejTRw706oW
number: 43.00
```

Gambar 46. Tampilan Serial monitor Data Yang Dikirim

Data yang akan dikirim dari NodeMCU bisa dilihat pada serial monitor. Selanjutnya dilakukan pengecekan pada *Database* Firebase apakah data masuk atau tidak.

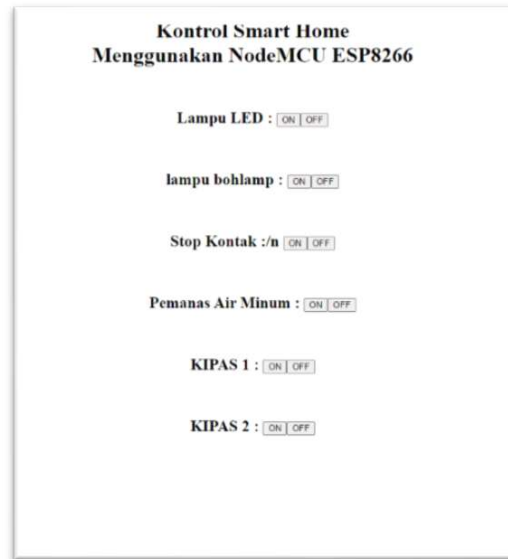


Gambar 47. Tampilan *Database* Pada Firebase

Apabila data masuk, tampilan kata dari data yang dikirim akan berubah menjadi warna jingga. Warna jingga ini mengindikasikan bahwa data telah berubah atau masuk ke database Firebase. Berdasarkan pengujian, pengiriman data ke Firebase berhasil dilakukan dan dapat diimplementasikan pada sistem yang dibuat.

2. Pengujian Perangkat melalui HTML yang telah dibangun.

Pengujian ini dilakukan dengan menekan *button* yang tersedia pada *interface* aplikasi. Kemudian pengujian ini bertujuan untuk melihat kondisi pengontrolan alat melalui tampilan web.



Gambar 48. Tampilan Button pada *user interface*

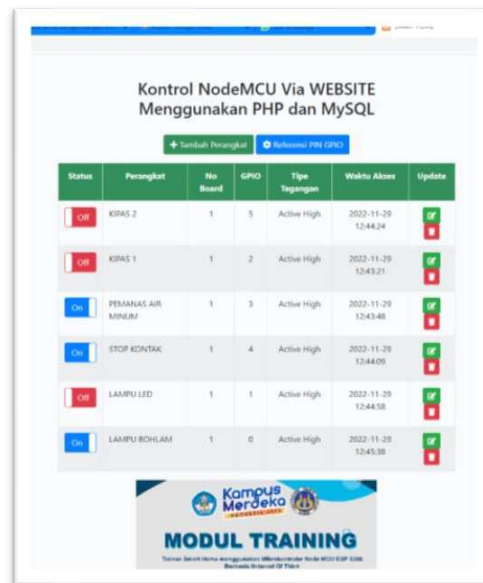
Berikut hasil pengujian ditampilkan dalam bentuk table :

Tabel 9. Hasil pengujian *user Interface* HTML

NO	<i>Button</i> Perangkat	Keterangan
1.	Lampu LED	Lampu Menyala
2.	Lampu Bohlam	Bohlam Menyala
3.	Stop kontak	Stop kontak Berfungsi
4.	Pemanas Air Minum	Pemanas air minum menyala
5.	Kipas Angin	Kipas Angin Menyala
6.	ALL ON/OFF	Button On/Off Berfungsi

3. Pengujian Perangkat melalui WEB PHP yang telah dibangun.

Pengujian ini di lakukan dengan menekan *button* yang tersedia pada *interface* aplikasi. Kemudian pengujian ini bertujuan untuk melihat kondisi pengontrolan alat melalui tampilan web yang telah dibangun.



Gambar 49. Tampilan Button pada *user interface*

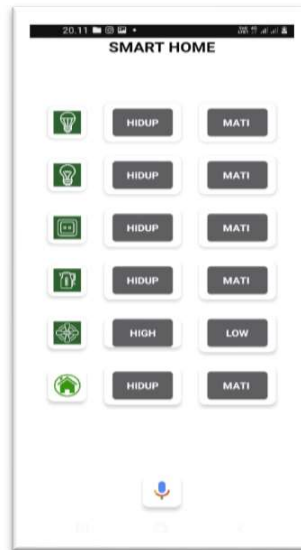
Berikut hasil pengujian ditampilkan dalam bentuk table :

Tabel 10. Hasil pengujian *user Interface* PHP

NO	<i>Button</i> Perangkat	Keterangan
1.	Lampu LED	Lampu Menyala
2.	Lampu Bohlam	Bohlam Menyala
3.	Stop kontak	Stop kontak Berfungsi
4.	Pemanas Air Minum	Pemanas air minum menyala
5.	Kipas Angin	Kipas Angin Menyala
6.	ALL ON/OFF	Button On/Off Berfungsi

4. Pengujian Perangkat melalui aplikasi android yang telah dibangun.

Pengujian ini di lakukan dengan menekan *button* yang tersedia pada *interface* aplikasi. Kemudian pengujian ini bertujuan untuk melihat kondisi pengontrolan alat melalui aplikasi android seperti yang ditampilkan pada Gambar 50.



Gambar 50. Tampilan Button pada *user interface*

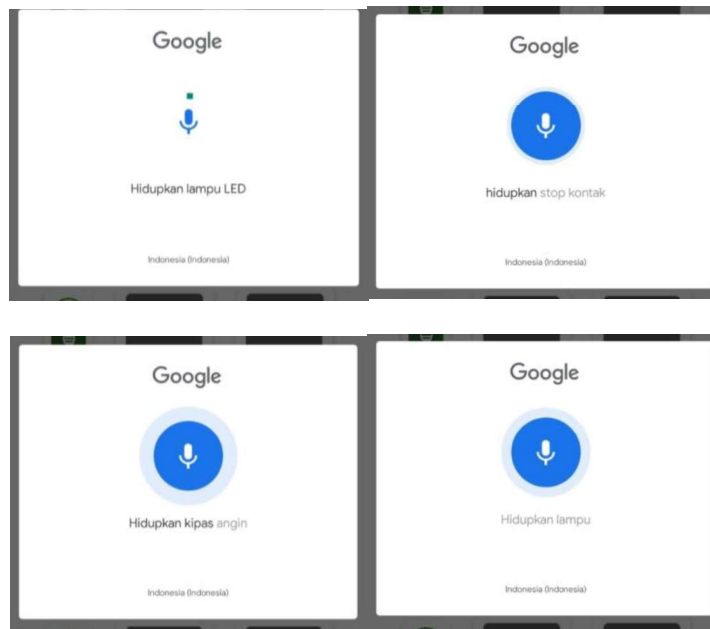
Berikut hasil pengujian ditampilkan dalam bentuk table:

Tabel 11. Hasil pengujian *user Interface* Android

NO	<i>Button</i> Perangkat	Keterangan
1.	Lampu LED	Lampu Menyala
2.	Lampu Bohlam	Bohlam Menyala
3.	Stop kontak	Stop kontak Berfungsi
4.	Pemanas Air Minum	Pemanas air minum menyala
5.	Kipas Angin	Kipas Angin Menyala
6.	ALL ON/OFF	Button On/Off Berfungsi

5. Pengujian perangkat menggunakan perintah suara

Pengujian ini di lakukan dengan menekan button *Speak* yang tersedia pada interface aplikasi. Kemudian mengucapkan perintah untuk menjalankan perangkat. Pengujian ini bertujuan untuk melihat kondisi pengontrolan alat melalui *Voice Recognition*.



Gambar 51. Tampilan *Google Voice Recognition*

Berikut ini hasil pengujian menggunakan perintah suara ditampilkan pada table 12.

Tabel 12. Hasil pengujian *Voice Recognition*

NO	Jenis Perangkat	Perintah suara 1	Perintah suara 2	Perintah suara 3	Perintah suara 4	Keterangan
1.	Lampu LED	Nyalakan lampu LED	Hidupkan lampu LED	Matikan lampu LED	Padamkan lampu LED	Lampu Menyala
2.	Lampu Bohlam	Nyalakan Lampu Bohlam	Hidupkan Lampu Bohlam	Matikan Lampu Bohlam	Padamkan Lampu Bohlam	Bohlam Menyala
3.	Stop kontak	Nyalakan Stop kontak	Hidupkan Stop kontak	Matikan Stop kontak	Padamkan Stop kontak	Stop kontak Berfungsi
4.	Pemanas Air Minum	Nyalakan Pemanas Air Minum	Hidupkan Pemanas Air Minum	Matikan Pemanas Air Minum	Padamkan Pemanas Air Minum	Pemanas air minum menyala

5.	Kipas Angin	Nyalakan Kipas Angin	Hidupkan Kipas Angin	Matikan Kipas Angin	Padamkan Kipas Angin	Kipas Angin Menyala
6.	ALL ON/OFF	Nyalakan Semua Perangkat	Hidupkan Semua Perangkat	Matikan Semua Perangkat	Padamkan Semua Perangkat	Perintah All On/Off Berfungsi

BAB V

PENUTUP

A. Kesimpulan

Dari hasil pembahasan sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. *Mikrokontroler NodeMCU esp8266* mampu mengendalikan, *door lock*, stop kontak, kipas angin (*speed 1, speed 2*), lampu 1, lampu 2 dan pemanas air minum sesuai dengan fungsinya masing-masing.
2. Sensor *RFID, door lock*, stop kontak, kipas angin (*speed 1, speed 2*), lampu 1, lampu 2 dan pemanas air minum. berhasil diaplikasikan pada sistem *trainer Smart Home* berbasis *Internet Of Things*.
3. Sistem berhasil dikontrol menggunakan aplikasi *Smart Home* via WEB berbasis Html.
4. Sistem berhasil dikontrol menggunakan aplikasi *Smart Home* via WEB berbasis PHP.
5. Sistem berhasil dikontrol menggunakan aplikasi *Smart Home* via *smartphone android*.
6. Sistem berhasil dikontrol menggunakan perintah suara.
7. Sistem berhasil dimonitoring menggunakan aplikasi *Smart Home* via *smartphone android*.

B. Saran

Berdasarkan pengalaman yang diperoleh selama penelitian *Trainer Smart Home* tentu ada beberapa kendala yang ditemukan, untuk pengembangan dan penyempurnaan rancangan alat kedepan maka disarankan:

1. Pilihlah *modul* WiFi yang lebih stabil dan memiliki jumlah pin I/O yang mencukupi kebutuhan agar dalam proses pengolahan data, pengiriman dan penerimaan data agar dalam proses monitoring dan kontrol tidak terkendala, sebaiknya pilih *Modul* WiFi NodeMCU ESP8266.
2. Gunakanan mikrokontroler yang mampu mengolah semua input dan output dari user untuk mempermudah user dalam menjalan trainer.
3. Menggunakan aplikasi pemograman perangkat lunak yang mendukung dalam perancangan dan pembuatan perangkat
4. Pilihlah *mikrokontroler* yang sesuai dengan kebutuhan pin yang akan digunakan dalam pembuatan sebuah alat, sehingga tidak terjadi kelebihan pin atau kekurangan pin.
5. Pilihlah sensor identifikasi ID yang berkualitas untuk menunjang kenyamanan dan keamanan perangkat.

DAFTAR PUSTAKA

- Adi, B., & Herlina, A. (2019). *Smart Home With Smart Control, berbasis Bluetooth Mikrokontroler. Journal of Electrical Engineering and Computer (JEECOM), 1(1)*.
- Agus Faudin, “Apa itu Module NodeMCU ESP8266?”, Nyebarilmu. <https://www.nyebarilmu.com/apa-itu-Module-NodeMCUESP8266/>. Diakses pada 16 Mei 2021
- Asisten Google, Google milikmu.* (n.d.). Retrieved February 9, 2022, from <https://assistant.google.com/>
- Azman, N. (2009). Perancangan Software Aplikasi Pervasive *Smart Home*. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Behmann, F., & Wu, K. (2015). *Collaborative Internet of things (C-IOT): For future smart connected life and business*. John Wiley & Sons.
- Berkenalan dengan Firebase, Apa itu Firebase? - IOT Kece.* (n.d.). Retrieved February 9, 2022, from <https://IOTkece.com/berkenalan-dengan-Firebase-apa-itu-Firebase/>
- Contoh Produk *IOT* Sederhana yang Ada di Sekitar Kita <https://aqi.co.id/news/contoh-produk-IOT-sederhana> (Diakses pada 18 Juli 2021)
- Efendi, Y. (2018). *Internet of things (IOT) sistem pengendalian lampu menggunakan Raspberry PI berbasis mobile. Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar, 4(2), 21-27.*

- Hanani, A., & Hariyadi, M. A. (2020). *Smart Home* berbasis *IOT* Menggunakan Suara Pada *Google Assistant*. *Jurnal Ilmiah Teknologi Informasi Asia*, 14(1), 49-56.
- Hardyanto, R. H. (2017). Konsep *Internet of things* Pada Pembelajaran berbasis Web. *Jurnal Dinamika Informatika*, 6(1), 87-97.
- Hermawan Susanto, S. (2011). Mudah Membuat Aplikasi *Android*. Yogyakarta: CV ANDI OFFSET.
- IFTTT. (n.d.). Retrieved February 9, 2022, from <https://IFTTT.com/>
- Keuntungan SmartHome <https://www.tatarumah.com/2019/12/keuntungan-kelemahan-smart-home.html>. (Diakses pada 18 Juli 2021)
- Owen, B. (2004). Dasar-Dasar Elektronika. Penerbit Erlangga, Jakarta.
- Panduardi, F., & Haq, E. S. (2016). Wireless *Smart Home* system menggunakan raspberry pi berbasis *Android*. *Jurnal Teknologi Informasi Dan Terapan*, 3(1), 320-325.
- Prasetyo, Elga Aris. 2018. "Relay", Arduino Indonesia. <https://www.arduinoindonesia.id/2018/07/relay.html>. Diakses pada 24 Feb 2021.
- Putra, M. K. A., Akbar, S. R., & Setyawan, G. E. (2018). Perancangan Sistem Keamanan Pada *Smart Home* Menggunakan *Voice recognition* dengan Konektivitas *Bluetooth*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X*.

- Putro, M. D., & Kambey, F. D. (2016). Sistem Pengaturan Pencahayaan Ruangan berbasis *Android* Pada *Smart Home*. *Jurnal Nasional Teknik Elektro*, 5(3), 297-307
- Rachman, D., Al Azam, M. N., & Anindito, B. (2017). Sistem Pemantau & Pengendalian *Smart Home* Menggunakan Infrastruktur Internet Messaging. *Jurnal Ilmiah: Lintas Sistem Informasi dan Komputer (LINK)*, 26(1).
- Relay - Arduino Indonesia | Tutorial Lengkap Arduino Bahasa Indonesia*. (n.d.). Retrieved February 9, 2022, from <https://www.arduinoindonesia.id/2018/07/relay.html>
- Satriadi, A., Wahyudi, W., & Christyono, Y. (2019). Perancangan home automation berbasis *NodeMCU*. *Transient: Jurnal Ilmiah Teknik Elektro*, 8(1), 64-71.
- Wibowo, B., Utama, H. S., & Kusumaningrum, N. (2019). Perancangan dan Realisasi Sistem Kendali Lampu, Air Conditionerer berbasis *Android*. *Perancangan dan Realisasi Sistem Kendali Lampu, Air Conditionerer berbasis Android*.
- Smart Home | NextGen.web.id*. (n.d.). Retrieved April 25, 2022, from <https://nextgen.web.id/smart-home/>

LAMPIRAN

Lampiran 1. Source Code Mengontrol NODEMCU Menggunakan HTML.

```

// Mengontrol NODEMCU Menggunakan HTML.
//-----

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

#define WIFI_SSID "Wipi"// Nama SSID AP/Hotspot
#define WIFI_PASSWORD "indonesia12345"// Password Wifi

int relay1 = 2;//D4 Mendefinisikan PIN Out dari Nodemcu
int relay2 = 0;//D3
int relay3 = 4;//D2
int relay4 = 5;//D1
int buzzer = 16;//D0

//const char* ssid = "Wipi";
//const char* password = "indonesia12345"; // Password Wifi

ESP8266WebServer server(80); //Menyatakan Webserver pada port 80
String webpage;

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);
  pinMode(relay4, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  // Connect to WiFi network -----
  Serial.println();
  Serial.print("Konfigurasi Jaringan... \n");

  // Mengatur WiFi -----

```

```

//WiFi.mode(WIFI_AP); // Mode Station
//WiFi.begin(ssid, password); // Mencocokkan SSID dan Password
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Menghubungkan Ke Jaringan");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);}

// Print status Connect -----
Serial.println("\nIP address: ");
Serial.println(WiFi.localIP());

// Isi dari Webpage -----
webpage+= "<center><h1> Kontrol Smart Home <br> Menggunakan NodeMCU ESP8266</h1></center>";
webpage+= "<center><h2><p>Relay 1 : ";
webpage+= "<a href=\"RELAY1ON\"><button>ON</button></a><a
href=\"RELAY1OFF\"><button>OFF</button></a></p></h2><br></center>";
webpage+= "<center><h2><p>Relay 2 : ";
webpage+= "<a href=\"RELAY2ON\"><button>ON</button></a><a
href=\"RELAY2OFF\"><button>OFF</button></a></p></h2><br></center>";
webpage+= "<center><h2><p>Relay 3 : ";
webpage+= "<a href=\"RELAY3ON\"><button>ON</button></a><a
href=\"RELAY3OFF\"><button>OFF</button></a></p></h2><br></center>";
webpage+= "<center><h2><p>Relay 4 : ";
webpage+= "<a href=\"RELAY4ON\"><button>ON</button></a><a
href=\"RELAY4OFF\"><button>OFF</button></a></p></h2><br></center>";
webpage+= "<center><h2><p>Buzzer : ";
webpage+= "<a href=\"BUZZERON\"><button>ON</button></a><a
href=\"BUZZEROFF\"><button>OFF</button></a></p></h2><br></center>";
webpage+= "<center><h2><p>Semua : ";
webpage+= "<a href=\"SEMUAON\"><button>ON</button></a><a
href=\"SEMUAOFF\"><button>OFF</button></a></p></h2></center>";

// Membuat tampilan Webpage -----
server.on("/", [] () {
  server.send(200, "text/html", webpage);
  Serial.print("Terhubung dengan WEBSITE...");
});

// Bagian ini untuk merespon perintah yang masuk -----

```

```

//Kondisi Hidup -----
server.on("/RELAY1ON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay1, LOW);
  Serial.print("Lampu LED 1 Hidup...\n");
  delay(1000);
});
server.on("/RELAY2ON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay2,LOW);
  Serial.print("Lampu LED 2 Hidup...\n");
  delay(1000);
});
server.on("/RELAY3ON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay3,LOW);
  Serial.print("Lampu LED 3 Hidup...\n");
  delay(1000);
});
server.on("/RELAY4ON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay4,LOW);
  Serial.print("Lampu LED 4 Hidup...\n");
  delay(1000);
});
server.on("/BUZZERON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(buzzer,LOW);
  Serial.print("BUZZER Hidup...\n");
  delay(1000);
});
server.on("/SEMUAON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(buzzer,LOW);
  digitalWrite(relay1,LOW);
  digitalWrite(relay2,LOW);
  digitalWrite(relay3,LOW);
  digitalWrite(relay4,LOW);
  Serial.print("SEMUA HIDUP...\n");
  delay(1000);
});

```

```

//Kondisi Mati -----
server.on("/RELAY1OFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay1, HIGH);
  Serial.print("Lampu LED 1 Mati...\n");
  delay(1000);
});
server.on("/RELAY2OFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay2,HIGH);
  Serial.print("Lampu LED 2 Mati...\n");
  delay(1000);
});
server.on("/RELAY3OFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay3,HIGH);
  Serial.print("Lampu LED 3 Mati...\n");
  delay(1000);
});
server.on("/RELAY4OFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(relay4,HIGH);
  Serial.print("Lampu LED 4 Mati...\n");
  delay(1000);
});
server.on("/BUZZEROFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(buzzer,HIGH);
  Serial.print("BUZZER MATI...\n");
  delay(1000);
});
server.on("/SEMUAOFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(buzzer,HIGH);
  digitalWrite(relay1,HIGH);
  digitalWrite(relay2,HIGH);
  digitalWrite(relay3,HIGH);
  digitalWrite(relay4,HIGH);
  Serial.print("SEMUA MATI...\n");
  delay(1000);
});

```

```

server.begin();
Serial.println("Web Server NodeMCU dijalankan");
}

void loop() {
  server.handleClient();
}

```

Lampiran 2. Source Code Mengontrol NODEMCU menggunakan PHP 1

```

// Mengontrol NODEMCU Menggunakan PHP dan MySQL.
//-----

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char* ssid = "Wipi";
const char* pass = "indonesia12345";
const char* host = "192.168.18.5";

void setup() {
  // put your setup code here, to run once:

  Serial.begin (9600);

  WiFi.hostname ("NodeMCU");
  WiFi.begin(ssid, pass);

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

  Serial.println("Berhasil Terhubung . . . .");

  pinMode(D4, OUTPUT);

```



```
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
    WiFiClient client;  
    const int httpPort = 80;  
    if(!client.connect(host, httpPort))  
    {  
        Serial.println("Gagal koneksi ke server");  
        return;  
    }  
    //Serial.println("berhasil koneksi ke server");  
    String LinkRelay;  
    HTTPClient httpRelay;  
    LinkRelay = "http://" +  
String(host)+"/kontrolling/bacarelay.php";  
    httpRelay.begin(LinkRelay);  
    httpRelay.GET();  
    String responseRelay = httpRelay.getString();  
    Serial.println(responseRelay);  
    httpRelay.end();  
    digitalWrite(D4, responseRelay.toInt());  
  
}
```

Lampiran 3. Source Code Mengontrol NODEMCU http client

```
//Program control relay menggunakan http client
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <Arduino_JSON.h>

const char* ssid = "Wipi"; //masukkan ssid
const char* password = "indonesia12345"; //masukkan password
const char* host = "192.168.18.5";

const long interval = 10; //5000
unsigned long previousMillis = 0;

int board = 1;

void setup () {

  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(1000);
    Serial.println("Menghubungkan..");

  }

  if(WiFi.status() == WL_CONNECTED){
    Serial.println("Terhubung!!!");
  }
  else{
    Serial.println("Gagal menghubungkan!!!");
  }
}
```

```

}

void loop() {

    HTTPClient http;

    http.begin("http://" +
String(host)+"/TA_final/proses.php?board="+String(board));
    //"/kontrolling/bacarelay.php";

    int httpCode = http.GET();

    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis >= interval) {
        if (WiFi.status() == WL_CONNECTED) {
            if (httpCode > 0) {
                String payload = http.getString();
                JSONVar myObject = JSON.parse(payload);

                Serial.print("JSON object = ");
                Serial.println(myObject);

                JSONVar keys = myObject.keys();

                for (int i = 0; i < keys.length(); i++) {
                    JSONVar value = myObject[keys[i]];
                    Serial.print("GPIO: ");
                    Serial.print(keys[i]);
                    Serial.print(" - SET to: ");
                    Serial.println(value);
                    pinMode(atoi(keys[i]), OUTPUT);
                    digitalWrite(atoi(keys[i]), atoi(value));
                }
                previousMillis = currentMillis;
            }
            http.end();
        }
    }
}

```

```

}

}

```

Lampiran 4. Source Code Mengontrol NODEMCU menggunakan android

```

// Mengontrol NODEMCU Menggunakan ANDROID.
//-----
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "relay1-d7fb9-default-rtdb.firebaseio.com" // Firebase host
#define FIREBASE_AUTH "SrcFq9XxWLabGT1Iv15yGJKlyga3ftAsvmXg39HI" //Firebase Auth
code
#define WIFI_SSID "Wipi" //Enter your wifi Name
#define WIFI_PASSWORD "indonesia12345" // Enter your password
int fireStatus;

void setup() {

  Serial.begin(9600);

  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);

  // Connect to WiFi network -----
  Serial.println();
  Serial.print("Konfigurasi Jaringan... \n");

  // Mengatur WiFi -----
  //WiFi.mode(WIFI_AP); // Mode Station
  //WiFi.begin(ssid, password); // Mencocokkan SSID dan Password
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting");

```

```

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

// Print status Connect -----
Serial.println();
Serial.println("Connected.");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

}

void loop() {
    fireStatus = Firebase.getInt("STATUS_Buzzer");
    if (fireStatus == 1) {
        digitalWrite(D0, LOW);
        Serial.println("Buzzer ON");
        // delay(5000);
    }
    else if (fireStatus == 0) {
        digitalWrite(D0, HIGH);
        Serial.println("Buzzer OFF");
        //delay(5000);
    }
    else {
        Serial.println("PERINTAH SALAH, COBA MASUKKAN 0/1");
    }
    fireStatus = Firebase.getInt("STATUS_Led");
    if (fireStatus == 1) {
        digitalWrite(D4, LOW);
        Serial.println("Led ON");
        // delay(5000);
    }
    else if (fireStatus == 0) {

```

```
digitalWrite(D4, HIGH);
Serial.println("Led OFF");
//delay(5000);
}
else {
  Serial.println("PERINTAH SALAH, COBA MASUKKAN 0/1");
}

fireStatus = Firebase.getInt("STATUS_Bohlam");
if (fireStatus == 1) {
  digitalWrite(D3, LOW);
  Serial.println("Bohlam ON");
  //delay(5000);
}
else if (fireStatus == 0) {
  digitalWrite(D3, HIGH);
  Serial.println("Bohlam OFF");
  //delay(5000);
}
else {
  Serial.println("PERINTAH SALAH, COBA MASUKKAN 0/1");
}

fireStatus = Firebase.getInt("STATUS_Steker");
if (fireStatus == 1) {
  digitalWrite(D2, LOW);
  Serial.println("Steker ON");
  //delay(5000);
}
else if (fireStatus == 0) {
  digitalWrite(D2, HIGH);
  Serial.println("Steker OFF");
  //delay(5000);
}
else {
```

```

        Serial.println("PERINTAH SALAH, COBA MASUKKAN 0/1");
    }

    fireStatus = Firebase.getInt("STATUS_Kettel");
    if (fireStatus == 1) {
        digitalWrite(D1, LOW);
        Serial.println("Kettel ON");
        //delay(5000);
    }
    else if (fireStatus == 0) {
        digitalWrite(D1, HIGH);
        Serial.println("Kettel OFF");
        //delay(5000);
    }
    else {
        Serial.println("PERINTAH SALAH, COBA MASUKKAN 0/1");
    }
}

```

Lampiran 5. Source Code Mengontrol NODEMCU terhubung ke Firebase

```

// Mengontrol NODEMCU Menggunakan Android dan Firebase.
//-----
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "relay1-d7fb9-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "SrCfq9XxWLabGT1Iv15yGJKlyga3ftAsvmXg39HI"
#define WIFI_SSID "Wipi"
#define WIFI_PASSWORD "indonesia12345"

int Relay1 = 5;//D1 Mendefinisikan PIN Out dari Nodemcu
int Relay2 = 4;//D2
int Relay3 = 0;//D3
int Relay4 = 2;//D4
int Buzzer = 16;//D0

```

```
int Relay5 = 14;//D5
int Relay6 = 12;//D6

int value1;
int value2;
int value3;
int value4;
int value5;
int value6;
int value7;

void setup()
{
  Serial.begin(9600);
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  pinMode(Relay3,OUTPUT);
  pinMode(Relay4,OUTPUT);
  pinMode(Buzzer,OUTPUT);
  pinMode(Relay5,OUTPUT);
  pinMode(Relay6,OUTPUT);

  digitalWrite(Buzzer,HIGH);
  digitalWrite(Relay1,HIGH);
  digitalWrite(Relay2,HIGH);
  digitalWrite(Relay3,HIGH);
  digitalWrite(Relay4,HIGH);

  WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() !=WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
```



```
Serial.print("connected:");  
Serial.println(WiFi.localIP());  
  
Firebase.begin(FIREBASE_HOST);  
//Firebase.setInt("STATUS_Led",0);  
//Firebase.setInt("STATUS_Bohlam",0);  
//Firebase.setInt("STATUS_Steker",0);  
//Firebase.setInt("STATUS_Kettle",0);  
//Firebase.setInt("STATUS_Kipas",0);  
  
}  
  
void firebaseconnect()  
{  
  Serial.println("Trying to reconnect");  
  Firebase.begin(FIREBASE_HOST);  
}  
  
void loop()  
{  
  if (Firebase.failed())  
  {  
    Serial.print("setting number failed:");  
    Serial.println(Firebase.error());  
    firebaseconnect();  
    return;  
  }  
  
  value1=Firebase.getString("STATUS_Led").toInt();  
  
  if(value1==1)  
  {  
    digitalWrite(Relay4,LOW);
```

```
        //Serial.println("LED ON");
    }
    else if(value1==0)
    {
        digitalWrite(Relay4,HIGH);
        // Serial.println("LED OFF");
    }

value2=Firebase.getString("STATUS_Bohlam").toInt();

if(value2==1)
    {
        digitalWrite(Relay3,LOW);
        //Serial.println("BOHLAM ON");
    }
    else if(value2==0)
    {
        digitalWrite(Relay3,HIGH);
        //Serial.println("BOHLAM OFF");
    }

value3=Firebase.getString("STATUS_Steker").toInt();

if(value3==1)
    {
        digitalWrite(Relay2,LOW);
        //Serial.println("STEKER ON");
    }
    else if(value3==0)
    {
        digitalWrite(Relay2,HIGH);
        //Serial.println("STEKER OFF");
    }

value4=Firebase.getString("STATUS_Kettel").toInt();
```

```
if(value4==1)
{
    digitalWrite(Relay1,LOW);
    //Serial.println("KETTEL ON");
}
else if(value4==0)
{
    digitalWrite(Relay1,HIGH);
    //Serial.println("KETTEL OFF");
}

value5=Firebase.getString("STATUS_Kipas1").toInt();

if(value5==1)
{
    digitalWrite(Relay5,LOW);
    //Serial.println("KIPAS ON");
}
else if(value5==0)
{
    digitalWrite(Relay5,HIGH);
    //Serial.println("KIPASS OFF");
}

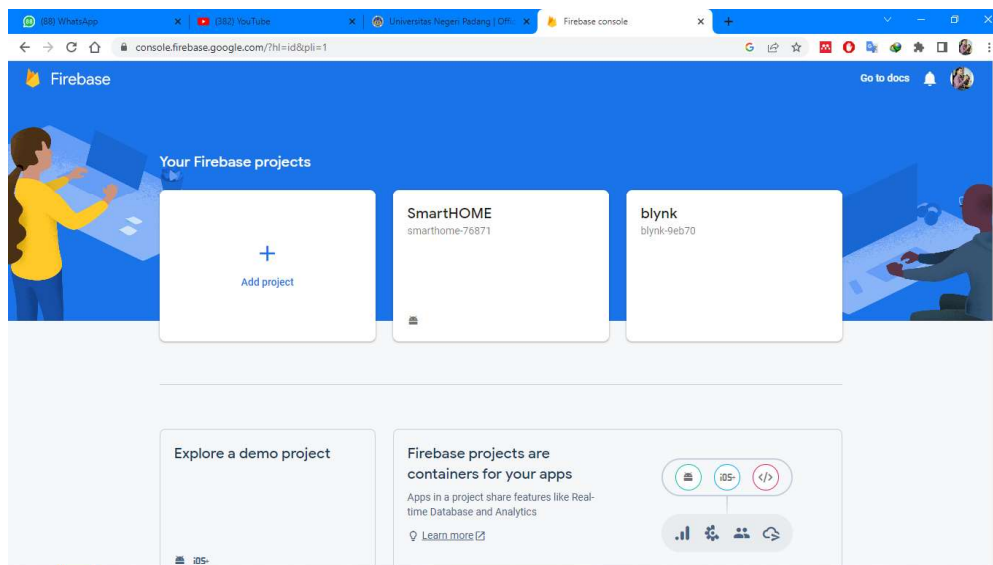
value6=Firebase.getString("STATUS_Kipas2").toInt();

if(value6==1)
{
    digitalWrite(Relay6,LOW);
    //Serial.println("KIPAS ON");
}
else if(value6==0)
{
    digitalWrite(Relay6,HIGH);
    //Serial.println("KIPASS OFF");
}

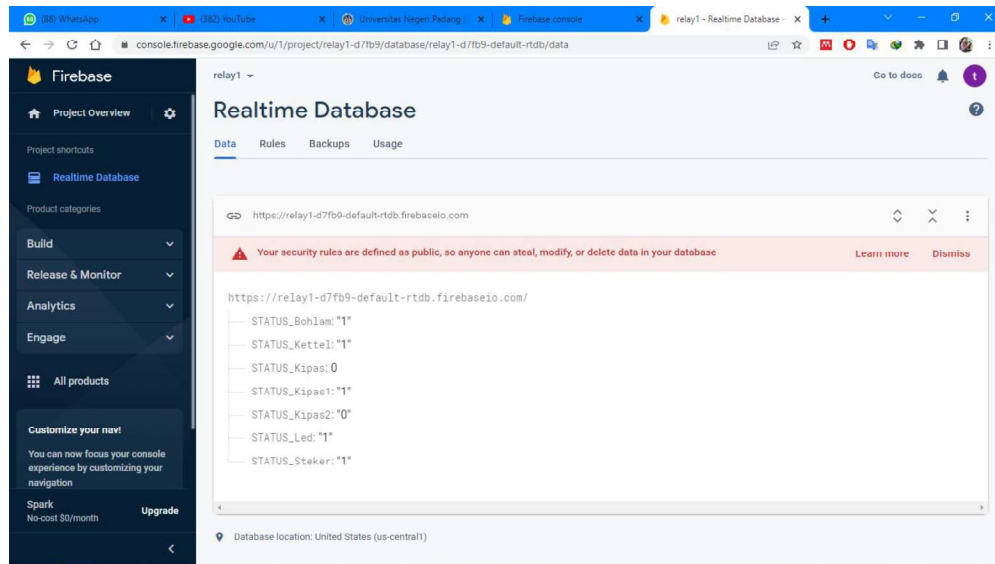
value7=Firebase.getString("STATUS_Buzzer").toInt();
```

```
if(value7==1)
{
    digitalWrite(Buzzer,HIGH);
    //Serial.println("KIPAS ON");
}
else if(value6==0)
{
    digitalWrite(Buzzer,LOW);
    //Serial.println("KIPASS OFF");
}
}
```

Lampiran 6. Tampilan WEB Firebase.



Lampiran 7. Tampilan Realtime database firebase



The screenshot displays the Firebase Realtime Database console interface. The left sidebar contains navigation options such as Project Overview, Realtime Database, Build, Release & Monitor, Analytics, Engage, and All products. The main content area shows the Realtime Database page with tabs for Data, Rules, Backups, and Usage. A prominent red warning banner states: "Your security rules are defined as public, so anyone can steal, modify, or delete data in your database." Below the warning, a list of data values is displayed for the URL `https://relay1-d7fb9-default-rtdb.firebaseio.com/`:

```
STATUS_Bohlam: "1"  
STATUS_Kettel: "1"  
STATUS_Kipas: 0  
STATUS_Kipas1: "1"  
STATUS_Kipas2: "0"  
STATUS_Led: "1"  
STATUS_Steker: "1"
```

At the bottom, the database location is identified as "United States (us-central1)".

Lampiran 8. Modul Trainer

