

**RANCANG BANGUN PROTOTIPE  
OTOMATISASI PROTEKSI DAN MONITORING LISTRIK  
RUMAH TANGGA DENGAN IoT ESP 32**

**SKRIPSI**

**OLEH:**

**MANGARA MUAL GUNAWAN LUBIS  
17.812.0026**



**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2022**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 26/12/22

Access From (repository.uma.ac.id)26/12/22

**RANCANG BANGUN PROTOTIPE  
OTOMATISASI PROTEKSI DAN MONITORING LISTRIK  
RUMAH TANGGA DENGAN IoT ESP 32**

**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Di fakultas Teknik  
Universitas Medan Area



**OLEH:  
MANGARA MUAL GUNAWAN LUBIS  
17.812.0026**

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2022**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 26/12/22

Access From (repository.uma.ac.id)26/12/22

## HALAMAN PENGESAHAN

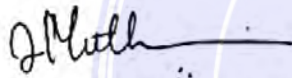
Judul Skripsi : Rancang Bangun Prototipe Otomatisasi Proteksi Dan  
Monitoring Listrik Rumah Tangga Dengan IoT Esp 32

Nama : Mangara Mual Gunawan Lubis

NPM : 17.812.0026

Fakultas : Teknik

Disetujui Oleh  
Komisi Pembimbing



Syarifah Muthia Putri, ST.MT

Pembimbing I



Habib Satria, S.Pd, MT

Pembimbing II



Dr. Rahmad Syah, S.Kom, M.Kom

Dekan



Habib Satria, S.Pd, MT

Ket. Prodi

Tanggal Lulus : 29 Juli 2022

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 26/12/22

Dipindai dengan CamScanner

Access From (repository.uma.ac.id)26/12/22

## HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam skripsi ini yang saya kutip dari karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila dikemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 29 Juli 2022



Mangara Mual Gunawan Lubis  
17.812.0026

## LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan dibawah ini :

Nama : Mangara Muall Gunawan Lubis  
NPM : 17.812.0026  
Program Studi : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Tugas Akhir/Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneklusif (Non-Exclusiv Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

“Rancang Bangun Prototipe Otomatisasi Proteksi Dan Monitoring Listrik Rumah Tangga Dengan IoT Esp 32”. Dengan Hak Bebas Royalti Noneksklusif ini, Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir/skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Medan, 29 Juli 2022



Mangara Muall Gunawan Lubis  
17.812.0026

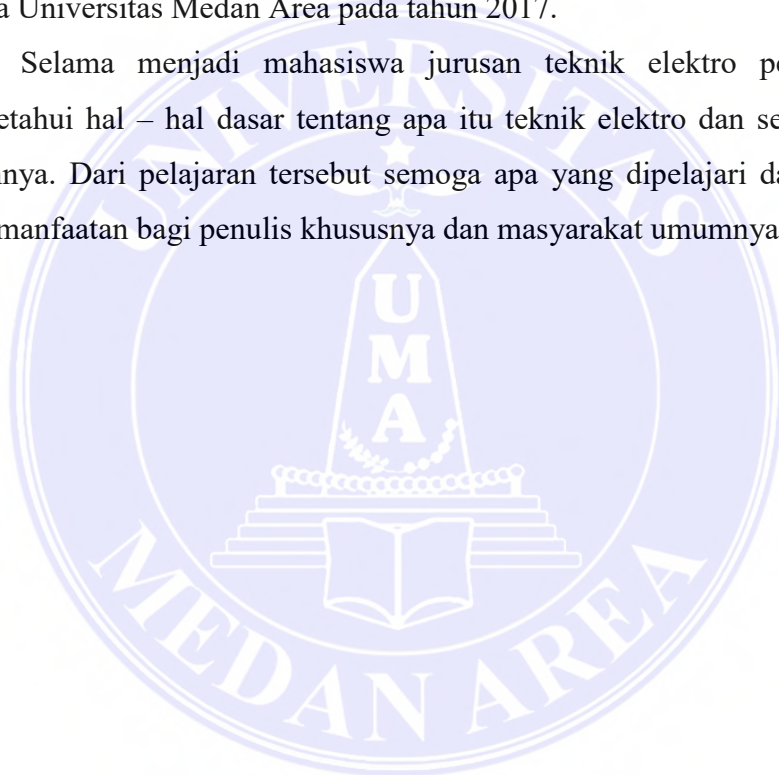


## RIWAYAT HIDUP

Penulis dilahirkan di Tembung pada tanggal 07 November 1998 dari ayah yang bernama Ramli Lubis dan Ibu Lamia Nasution. Penulis ialah putra bungsu dari 3 bersaudara.

Menyelesaikan sekolah dasar di SDN.106164 Deliserdang pada tahun 2011, SMP. Swasta sabilina pada tahun 2014 dan SMK. Negeri 1 Tanjung Morawa pada tahun 2017. Serta masuk dan terdaftar pada perguruan tinggi swasta Universitas Medan Area pada tahun 2017.

Selama menjadi mahasiswa jurusan teknik elektro penulis dapat mengetahui hal – hal dasar tentang apa itu teknik elektro dan seluk beluk di dalamnya. Dari pelajaran tersebut semoga apa yang dipelajari dapat menjadi kebermanfaatan bagi penulis khususnya dan masyarakat umumnya.



## KATA PENGANTAR

Alhamdulillah rabbilalamin segala puji syukur selalu kita lafaskan kehadiran Allah SWT yang telah memberikan kita nikmat iman, ilmu dan ihsan sehingga kita dapat menjalani kehidupan duniawi ini dengan sebaik – baiknya guna mengharap ridha dari Allah SWT guna di kehidupan ukhrawi yang kekal dan abadi kelak. Dan atas nikmat demikianlah kita masih dapat menjalankan aktivitas dengan sebaik – baiknya.

Adapun judul yang saya angkat dalam memenuhi tugas akhir ini yakni “Rancang Bangun Prototipe Otomatisasi Proteksi Dan Monitoring Listrik Rumah Tangga Dengan IoT Esp 32 “skripsi ini disusun guna menjadi salah satu syarat menyelesaikan pendidikan Strata 1 Program Studi Teknik Elektro Universitas Medan Area.

Pada proses penulisan skripsi ini terdapat banyak sekali bantuan dan dukungan yang penulis dapatkan, baik berupa dukungan moral, moril dan do’a untuk itu pada kesempatan ini penulis mengucapkan trimakasi banyak kepada:

1. Ibu saya Lamia Nasution yang telah menguliahkan saya hingga selesai, kepada Kakak Saya Rahmi Wardah Lubis, Rizka Fadillah Lubis Dan Imam Nahu Wahyudi Lubis yang turut ikhlas membantu biaya Perkuliahan saya. Teman teman saya yang telah memberikan motivasi, semangat dan do’a kepada saya.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc selaku Rektor Universitas Medan Area
3. Bapak Dr. Rahmad Syah, S.Kom, M.Kom selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Bapak Habib Satria, S.Pd, MT selaku Ketua Program Studi Teknik Elektro Universitas Medan Area dan selaku pembimbing 2 saya dalam menyelesaikan tugas akhir ini yang telah membimbing serta memberikan ide dan motivasi.
5. Ibu Syarifah Muthia Putri, ST, MT. selaku pembimbing 1 saya dalam menyelesaikan tugas akhir ini yang telah membimbing serta memberikan ide – idenya kepada saya.

6. Kepada seluruh pihak yang membantu dalam proses penulisan skripsi ini hingga selesai.

Akhir kata semoga apa yang terkandung dalam skripsi ini dapat bermanfaat serta dapat menjadi referensi dalam lingkup pengembangan ilmu kedepan.

Semoga Allah SWT selalu mencurahkan kebaikan serta kasih sayang-Nya kepada seluruh pihak yang turut membantu penulis dalam menyelesaikan skripsi dan pendidikan ini.



Medan, 29 Juli 2022

Mangara Mual Gunawan Lubis

17.812.0026



## ABSTRAK

IoT merujuk pada jaringan perangkat fisik, kendaraan, peralatan listrik rumah tangga, dan barang-barang lainnya yang ditanami perangkat elektronik, perangkat lunak, sensor, aktuator, dan konektivitas. Memungkinkan untuk perkembangan jaman semakin maju menghadirkan peralatan listrik semakin beragam untuk membantu kebutuhan manusia. Hal itu tidak menutup kemungkinan membutuhkan ketersediaan listrik selalu memadai, dengan indikator listrik stabil, dan terdapat sistem proteksi otomatis untuk meminimalisir bahaya yang disebabkan oleh tegangan yang tidak stabil meliputi tegangan lebih dan tegangan kurang. Maka dari itu dikembangkanlah sebuah alat dengan judul “Rancang bangun prototipe otomatisasi proteksi dan monitoring peralatan listrik rumah tangga dengan IoT NodeMCU ESP 32”. Alat ini dapat dikategorikan sebagai smart home, dimana penghuni rumah bisa mengontrol perabotan rumah seperti lampu lewat ponsel yang terhubung di jaringan internet, lebih cepat menggantikan sistem kontrol saklar yang masih manual. Alat ini dibuat dengan tujuan untuk melakukan proteksi tegangan dan monitoring listrik dengan tegangan normal 220 VAC yang ada di rumah tangga. Sistem yang dirancang bertujuan untuk menambahkan fungsi proteksi yang ada di rumah tangga, bukan untuk menggantikan fungsi MCB. Sistem juga telah dapat memonitor daya listrik dengan cara menampilkan data daya pada LCD dan smartphone, tingkat pembacaan sensor mampu bekerja dengan tingkat keberhasilan pengukuran tegangan 99,6%, arus 97,5% dan pengukuran daya 97,8%. Pengujian alat menggunakan beban induktif berupa lampu LED 45 watt, 35 watt, 26 watt dan beban resistif berupa setrika listrik 400 watt yang diset pada panas maksimum. Dalam fungsi proteksinya relay mampu melakukan proteksi saat tegangan berada di atas 240 VAC dan di bawah 210 VAC.

Kata kunci:

*DHT11, Firebase, IoT, NodeMCU ESP8266, Otomatisasi, PZEM004T, Proteksi.*

## ABSTRACT

IoT refers to a network of physical devices, vehicles, household electrical appliances, and other items embedded with electronic devices, software, sensors, actuators, and connectivity. This does not rule out the possibility that the availability of electricity is always adequate, with a stable electricity indicator, and there is an automatic protection system to minimize the dangers caused by unstable voltages including over voltage and under voltage. Therefore, a tool was developed with the title "Design prototype automation protection and monitoring of household electrical equipment with IoT NodeMCU ESP 32". This tool can be categorized as a smart home, where the occupants of the house can control home furnishings such as lights via a cellphone connected to the internet, replacing the switch control system that is still manual. This tool is made with the aim of carrying out voltage protection and electricity monitoring with a normal voltage of 220 V AC in the household. The designed system aims to add the existing protection function in the household, not to replace the MCB function. The system has also been able to monitor electrical power by displaying power data on the LCD and Smartphone, the sensor reading level is able to work with a success rate of 97.4% current measurement and 97.7% power measurement. Testing the tool using an inductive load in the form of a 45 watt 35, watt, 26 watt LED lamp and a resistive load in the form of a 400 watt electric iron and a 40 watt soldering iron set at the maximum hot point. In its protection function, the relay is able to perform a protective function when the voltage is above 240 VAC and under 210 VAC.

Keywords:

DHT11, Firebase, IoT, NodeMCU ESP8266, Automation, PZEM004T, Protection.

## DAFTAR ISI

HALAMAN PERNYATAAN.....	i
LEMBAR PERNYATAAN .....	ii
RIWAYAT HIDUP.....	iii
KATA PENGANTAR .....	iv
ABSTRAK .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xvi
BAB I	
PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan Penelitian .....	2
1.5. Manfaat Penelitian .....	3
1.6. Sistematik penulisan .....	3
BAB II	
TINJAUAN PUSTAKA	
2.1. Sistem Proteksi .....	4
2.2. Monitoring .....	4
2.3. IoT Internet of Things.....	5
2.4. NodeMCU ESP32ROOM.....	5
2.4.1. Spesifikasi Yang Dimiliki Oleh NodeMCU .....	6
2.4.2. Pin GPIO ESP32 WROOM DevKit V1.....	7
2.5. Arduino Nano .....	8
2.5.1. Arduino Nano Pinout .....	9
2.6. Modul Relay .....	10
2.6.1. Spesifikasi Modul Relay Yang Digunakan.....	11

2.7. Sensor PZEM-004t .....	12
2.8. LCD 16x2 .....	13
2.8.1. Modul I2C LCD .....	15
2.8.2. Spesifikasi modul LCD I2C PCF8574.....	15
2.9. Konektor .....	16
2.10. Sensor DHT11 .....	18
2.10.1. Spesifikasi DHT11 .....	18
2.11. MIT APP Inventor .....	19

### BAB III

#### METODOLOGI PENELITIAN

3.1. Tempat Dan Waktu .....	21
3.1.1 Tempat Penelitian .....	21
3.1.2. waktu penelitian .....	21
3.2. Kerangka Berpikir .....	22
3.3. Mengidentifikasi Kebutuhan Alat.....	23
3.4. Perancangan Pembuatan Alat. ....	25
3.5. Pemotongan Dan Pengeboran Kotak.....	25
3.6. Instalasi Arduino Nano .....	26
3.7. Pemasangan Sensor DHT1 .....	27
3.8. Pemasangan LCD.....	27
3.9. Pemasangan Buzzer .....	28
3.10. Pemasangan Pushbutton.....	28
3.11. Pemasangan Potensiometer .....	29
3.12. Pemasangan Modul Relay.....	29
3.13. Instalasi NoneMCU ESP32.....	30
3.14. Pemasangan Dioda Zener Dan DipSwitch .....	30
3.15. Koneksi Firebase Dengan smart phone dan perangkat.....	31
3.16. Pemasangan Module sensor PZEM OO4T .....	31
3.17. Pemasangan Peroteksi Mcb .....	32
3.18. Pemasangan Power Suplay 5 volt.....	32
3.19. Pengujian Alat .....	32

3.20. Diagram Blok.....	33
3.20.1. Fungsi Diagram Blok.....	33
3.21. Rangkaian Keseluruhan .....	34
3.22. Rancangan Anggaran Biaya .....	36
3.23. Pembuatan Firebase .....	37
3.24. Pembuatan Aplikasi Smartphone Monitoring .....	46
3.25. Penempatan Alat Dalam Rumah Tangga.....	58

## BAB IV

### PENGUJIAN DAN ANALISA

4.1. Pengujian input sensor DHT 11 .....	59
4.2. Pengujian DHT11 Menyalakan Kipas Otomatis .....	62
4.3. Pengujian Input Sensor PZEM-004T-100A .....	66
4.4. Hasil Pengujian Tanpa Beban.....	68
4.5. Hasil Pengujian Beban Kipas angin 50 Watt.....	70
4.6. Hasil Pengujian Menggunakan Lampu 141 Watt.....	71
4.7. Daya Lampu 1.....	74
4.8. Daya Lampu 2.....	76
4.9. Daya Lampu 3.....	78
4.10. Daya Lampu 4.....	80
4.11. Pengujian Setrika Maspion 400 Watt.....	81
4.12. Pengujian Menggunakan Keseluruhan Beban .....	84
4.13. Pengujian Sistem Proteksi Arus keseluruhan Beban.....	86
4.14. Analisis Sistem .....	89

## BAB V

### KESIMPULAN DAN SARAN

Kesimpulan .....	96
Saran .....	96
Daftar Pustaka.....	97
Lampiran.....	100



## DAFTAR GAMBAR

Gambar 2.4 NodeMCU Esp 32.....	6
Gambar 2.4.1 Pinout Modul Esp 32 .....	7
Gambar 2.5 Arduino Nano .....	9
Gambar 2.5.1 Arduino Nano Pinout .....	9
Gambar 2.6 Modul Relay .....	11
Gambar 2.7 Sensor Pzem-004t .....	12
Gambar 2.8 Lcd 16X2 .....	14
Gambar 2.8.2 Modul I2c.....	16
Gambar 2.8.3. Rangkaian Lcd Dan Esp .....	16
Gambar 2.9 Power Supplay .....	17
Gambar 2.10 Sensor DHT 11 .....	18
Gambar 2.11 Tampilan Pertama Mit App .....	19
Gambar 2.11.1 Tampilan kedua Mit App.....	20
Gambar 3.2 Flowchart Pembuatan Alat.....	22
Gambar 3.4 Rangkaian dengan Colerdraw .....	25
Gambar 3.6 Rangkaian Instalasi Arduino Nano .....	26
Gambar 3.7 Rangkaian Sensor DHT 11 .....	27
Gambar 3.8 Rangkaian Lcd Display.....	27
Gambar 3.9 Rangkaian Pemasangan Buzzer .....	28
Gambar 3.10 Rangkaian Pemasangan Pushbutton .....	28
Gambar 3.11 Rangkaian Pemasangan Potensiometer .....	29
Gambar 3.12 Rangkaian Pemasangan Module Relay.....	29
Gambar 3.13 Instalasi Nodemcu Esp 32.....	30
Gambar 3.14 Rangkaian Pemasangan Dioda Zener .....	30
Gambar 3.14.1 Rangkaian Pemasangan Dipswitch .....	31
Gambar 3.16 Rangkaian Sensor Pzem-004t .....	31
Gambar 3.17 Pemasangan MCB.....	32
Gambar 3.18 Pemasangan Power Supplay .....	32
Gambar 3.20 Diagram Blok.....	33
Gambar 3.21 Rangkaian Keseluruhan Alat .....	35

Gambar 3.23 Tampilan Awal Fribebase .....	38
Gambar 3.23.1 New Perojek Firebase .....	38
Gambar 3.23.2 Create Projek Name .....	38
Gambar 3.23.3 Yout Firebase projek.....	39
Gambar 3.23.4 Configure Googel .....	39
Gambar 3.23.5 Create Projek.....	40
Gambar 3.23.6 Creating yout Projek .....	40
Gambar 3.23.7 Finishing .....	40
Gambar 3.23.8 Continuiue.....	41
Gambar 3.23.9 Finishing Create .....	41
Gambar 3.23.10 Tampilan Realtame Data Base.....	42
Gambar 3.23.11 Location Data Base .....	42
Gambar 3.23.12 Lokednode database.....	42
Gambar 3.23.13 Peroses Pembuatan Data Base .....	43
Gambar 3.23.14 Hasil Data Base.....	43
Gambar 3.23.15 Projek Seting.....	44
Gambar 3.23.16 Tampilan Perojek Seting .....	44
Gambar 3.23.17 Servise Account Firebase.....	44
Gambar 3.23.18 Data Base Secret .....	45
Gambar 3.23.19 Token Program .....	45
Gambar 3.23.20 Pengujian data base.....	45
Gambar 3.24 Mitt App Inventor .....	47
Gambar 3.24.1 Create New App.....	47
Gambar 3.24.2 Tampilan Design App.....	47
Gambar 3.24.3 Hasil Design App.....	48
Gambar 3.24.4 Block App .....	48
Gambar 3.24.5 Block Tabel Data .....	49
Gambar 3.24.6 Kode Aumentikasi .....	49
Gambar 3.24.7 Block Diagram firebase .....	49
Gambar 3.24.8 Block Diagram Nilai Power.....	50
Gambar 3.24.9 Tmpilan APP Power .....	50
Gambar 3.24.10 Block Program Voltase .....	50

Gambar 3.24.11 Tampilan App Voltase .....	50
Gambar 3.24.12 Program Arus .....	51
Gambar 3.24.13 Tampilan APP Arus .....	51
Gambar 3.24.14 Blog Program Frequency .....	51
Gambar 3.24.15 Tampilan App Frequency .....	51
Gambar 3.24.16 Firebase Data Change .....	52
Gambar 3.24.17 Blok Program Data Change .....	52
Gambar 3.24.18 Blok Program Menyalaka Lampu 1 .....	52
Gambar 3.24.19 Blok Program Mematikan Lampu 1 .....	53
Gambar 3.24.20 Tampilan App Nyalakan Lampu 1 .....	53
Gambar 3.24.21 Blog Program Nyalakan Lampu 2 .....	53
Gambar 3.24.22 Blog Program Mematikan Lampu 2 .....	54
Gambar 3.24.23 Tampilan App Nyalakan Lampu 2 .....	54
Gambar 3.24.24 Blog Program Nyalakan Lampu 3 .....	54
Gambar 3.24.25 Blog Program Mematikan Lampu 3 .....	55
Gambar 3.24.26 Tampilan App Nyalakan Lampu 3 .....	55
Gambar 3.24.27 Blog Program Nyalakan Lampu 4 .....	55
Gambar 3.24.28 Blog Program Mematikan Lampu 4 .....	56
Gambar 3.24.29 Tampilan App Nyalakan Lampu 4 .....	56
Gambar 3.24.30 Input Firebase .....	56
Gambar 3.24.31 Program Esp Baca S1 .....	57
Gambar 3.24.32 Program Esp Baca S2 .....	57
Gambar 3.24.33 Program Esp Baca S3 .....	57
Gambar 3.24.34 Program Esp Baca S4 .....	57
Gambar 4.1 Rangkaian Input Sensor DHT 11 .....	59
Gambar 4.1.1 Grafik Pembacaan Sensor DHT 11 .....	60
Gambar 4.1.2 Program DHT 11 .....	60
Gambar 4.1.3 Program Deklarasi DHT 11 .....	61
Gambar 4.1.4 Program Perintah Baca Temperatur .....	61
Gambar 4.1.5 Program Tampilan Nila Suhu .....	61
Gambar 4.1.6 Tampilan Lcd Nilai Suhu .....	62
Gambar 4.1.7 Pengujiann Sensor DHT 11 .....	62

Gambar 4.2 Rangkaian Komponen Listrik .....	63
Gambar 4.2.1 Program Deklarasi DHT 11 .....	63
Gambar 4.2.2 Program Deklarasi Relay 1 dan 2 .....	63
Gambar 4.2.3 Program Deklarasi Relay Sebagai Output .....	63
Gambar 4.2.4 Program Deklarasi Penggunaan DHT 11 .....	64
Gambar 3.2.5 Program Perintah Baca .....	64
Gambar 4.2.6 Menyalakan Relay Suhu .....	64
Gambar 4.2.7 Menyalakan Relay Kelembapan .....	64
Gambar 4.2.8 Grafik Pengujian Sensor DHT 11 .....	65
Gambar 4.2.9 Tampilan Lcd Suhu Dan Humidity .....	65
Gambar 4.2.10 Posisi Relay Kipas .....	66
Gambar 4.3 Rangkaian Pengujian Sensor Pzem-004t .....	66
Gambar 4.3.1 Program Liblari Pzem-004t .....	67
Gambar 4.3.2 Program Pin Komunikasi Esp .....	67
Gambar 4.3.3 Program Variabel Tipe Float .....	67
Gambar 4.3.4 Program Pembacaan Sensor Pzeem .....	68
Gambar 4.4 Tampilan Lcd Tanpa Beban .....	69
Gambar 4.4.1 Multimeter Mengukur Tegangan .....	69
Gambar 4.4.2 Pengujian Tanpa Beban .....	69
Gambar 4.5 Pengujian Kipas 50 Watt .....	70
Gambar 4.5.1 Tampilan App Beban Kipas .....	71
Gambar 4.5.2 Beban Kipas 50 Watt .....	71
Gambar 46. Rangkaian Lampu IoT .....	72
Gambar 4.6.1 Grafik 4 Lampu IoT .....	73
Gambar 4.6.2 Tampilan Lcd Semua Lampu IoT .....	73
Gambar 4.6.3 Pengujian Daya Semua Lampu IoT .....	74
Gambar 4.7 Tampilan Lcd Daya Lampu 1 .....	75
Gambar 4.7.1 Tampilan App Daya Lampu 1 .....	75
Gambar 4.7.2 Tampilan Lcd Tegangan Lampu 1 .....	75
Gambar 4.7.3 Beban Lampu 35 Watt .....	76
Gambar 4.8 Tampilan Lcd Daya Lampu 2 .....	77
Gambar 4.8.1 Tampilan App Daya Lampu 2 .....	77

Gambar 4.8.2 Tegangan Kerja Lampu 2.....	77
Gambar 4.8.3 Beban Lampu 45 Watt .....	78
Gambar 4.9 Tampilan Lcd Daya Lampu 3 .....	79
Gambar 4.9.1 Tampilan App Daya Lampu 3 .....	79
Gambar 4.9.2 Beban Lampu 35 Watt .....	79
Gambar 4.10 Tampilan Lcd Daya Lampu 4 .....	80
Gambar 4.10.1 Tampilan App Daya Lampu 4 .....	81
Gambar 4.10.2 Beban Lampu 26 Watt .....	81
Gambar 4.11 Grafik Sebelum Dan Sesudah Diberi Beban.....	83
Gambar 4.11.1 Tampilan Lcd Daya Setrika .....	83
Gambar 4.11.2 Tampilan App Daya Setrika .....	83
Gambar 4.11.3 Beban Setrika 400 Watt .....	84
Gambar 4.12 Rangkaian Pengujian Beban Keseluruhan.....	84
Gambar 4.12.1 Grafik Beban Keseluruhan.....	85
Gambar 4.12.2 Tampilan App Beban Keseluruhan.....	86
Gambar 4.13 Grafik Proteksi Arus .....	87
Gambar 4.13.1 Hasil Pengujian Semua Beban.....	87
Gambar 4.13.2 Tampilan Lcd Batas Arus .....	88
Gambar 4.13.3 Tampilan Lcd Status Relay Proteksi.....	88
Gambar 4.16 Grafik Pengukuran Error Arus .....	91
Gambar 4.17 Grafik Pengukuran Error Daya .....	92
Gambar 4.18 Grafik Simulasi UnderVoltase.....	94
Gambar 4.19 Grafik Simulasi OverVoltase.....	95



## DAFTAR TABEL

Tabel 3.1. Waktu Pelaksana.....	21
Tabel 3.3. Mengidentifikasi Kebutuhan Alat .....	23
Tabel 3.22. Spesifikasi Alat Dan Bahan .....	36
Tabel 4.1 Pembacaan Sensor DHT 11 .....	60
Tabel 4.2. Pembacaan Data Sensor DHT 11 .....	65
Tabel 4.4. Hasil Pengujian Tanpa Beban.....	68
Tabel 4.5. Pengujian Kipas 50 Watt .....	70
Tabel 4.6. Pengujian 4 Lampu IoT .....	72
Tabel 4.7. Pengujian Daya Lampu 1 .....	74
Tabel 4.8. Pengujian Daya Lampu 2 .....	76
Tabel 4.9. Pengujian Daya Lampu 3 .....	78
Tabel 4.10. Pengujian Daya Lampu 4 .....	80
Tabel 4.11. Pengujian Daya Setrika 400 Watt.....	82
Tabel 4.11.1. Pengujian Beban Setrika 400 watt.....	82
Tabel 4.12. Pengujian Beban Secara Keseluruhan .....	85
Tabel 4.13. Proteksi Arus Lebih .....	86
Tabel 4.15. Hasil Pengukuran Error Tegangan.....	90
Tabel 4.16. Hasil Pengukuran Error Arus.....	90
Tabel 4.17. Hasil Pengukuran Error Daya .....	92
Tabel 4.18. Simulasi Proteksi UnderVoltase.....	93
Tabel 4.19. Simulasi Proteksi OverVoltase .....	94

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Sistem proteksi dan monitoring adalah suatu kerangka keamanan untuk peralatan listrik rumah tangga. Pengamanan peralatan akan bekerja jika ditemukan adanya gangguan teknis, gangguan khusus, kesalahan dalam pengoperasian dan penyebab lainnya. Pada sebuah sistem proteksi harus memiliki kemampuan untuk bekerja sesuai tujuan dan kapasitas serta kemampuannya. Jika terdapat adanya sebuah gangguan dan sistem proteksi tidak dapat bekerja, akan menimbulkan kerugian yang signifikan. Kerugian tersebut meliputi kerusakan peralatan instalasi secara keseluruhan dan peralatan listrik yang digunakan.

Adanya sebuah sistem proteksi dan memonitoring, perangkat yang terhubung dengan listrik akan terlindungi dan dapat terhindar dari bahaya, perangkat keamanan yang selama ini dipakai disistem kelistrikan rumah tangga sejauh ini adalah sebuah *Miniature Circuit Breaker* atau di sebut MCB yang memiliki kemampuan untuk membatasi aliran arus.

Namun menggunakan MCB saja tidak cukup untuk memberikan jaminan, karena hanya melindungi dari beban berlebih. Sistem yang dirancang bermaksud untuk menambahkan fungsi proteksi yang berada di rumah tangga, bukan untuk menggantikan fungsi dari MCB.

Sensor yang digunakan adalah PZEM-004t yang mampu membaca nilai besaran tegangan, Arus, Daya aktif dan Energi yang digunakan serta dilengkapi dengan modul relay sebagai aktuator. Agar terhubung ke internet alat ini menggunakan ESP32, mengubah informasi menjadi nilai yang dapat dibaca oleh manusia dan kemudian akan dikirimkan ke media penyimpanan database menggunakan layanan internet wifi, dan melakukan sistem proteksi secara otomatis memberikan proteksi dengan memutuskan aliran arus hasil pembacaan diluar batas listrik ketika ada beban berlebih. Sistem ini juga dapat memonitoring dengan menggunakan smartphone.

Berdasarkan permasalahan yang ada maka dibuatlah sebuah alat untuk menambah fungsi sistem proteksi yang ada dilistrik rumah tangga, tidak untuk menggantikan fungsi MCB. Untuk mengamankan dan mempermudah kita untuk memonitorig keadaan listrik kita secara internet of things. Berdasarkan penjelasan latar belakang diatas, penulis mengajukan judul “Rancang Bangun Prototipe Otomatisasi Proteksi Dan Monitoring Listrik Rumah Tangga Dengan IoT NodeMCU ESP 32”.

## 1.2. Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah

1. Bagaimana pembuatan alat otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT NodeMCU ESP-32.
2. Bagaimana tingkat kelayakan alat otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT NodeMCU ESP-32.

## 1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah :

1. Alat ini bekerja untuk mengontrol prototipe perangkat rumah dengan menggunakan IoT NodeMCU ESP-32.
2. Sistem ini dapat dikendalikan dengan internet of things menggunakan smartphone.
3. Perangkat yang akan dikontrol yaitu lampu dan kipas angin.
4. Menggunakan Relay Module.
5. Mikrokontroler yang dipakai adalah NodeMCU ESP-32.

## 1.4. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Membuat alat sistem otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT NodeMCU ESP-32 berbasis internet of things.
2. Menguji kinerja alat sistem otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT NodeMCU ESP-32.

### 1.5. Manfaat Penelitian

Manfaat yang diharapkan dari pembuatan alat ini antara lain :

1. Membantu pemilik rumah mengotomatiskan dan pemantauan perangkat rumah kapan saja dan dari lokasi manapun dengan menjadikan inovasi baru.
2. Memberikan wawasan dan pengetahuan bagi penulis dan pembaca khususnya tentang penerapan teknologi internet of things dalam kehidupan sehari hari.
3. Sebagai sumber informasi bagi orang yang ingin mengembangkan proyek rumah pintar asli berbasis internet of things menggunakan smartphone dan NodeMCU ESP32.

### 1.6. Sistematika Penulisan

Berikut ini adalah sistematika penulisan untuk setiap bab

#### 1. BAB I PENDAHULUAN

Tujuan penulisan, manfaat penulisan, dan sistematika penulisan dibahas dalam bab ini, serta latar belakang pembuatan laporan, rumusan masalah, dan definisi masalah.

#### 2. BAB II TEORI PENUNJANG

Bab ini berisi landasan teori berupa konsep dasar dalam penyusunan alat dan laporan sehingga menghasilkan karya yang bernilai ilmiah dan memiliki daya guna.

#### 3. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang alat-alat metode penelitian, termasuk cara pengumpulan data.

#### 4. BAB IV HASIL DAN PEMBAHASAN

Dalam bab ini berisi hasil pengujian alat serta pembahasan

#### 5. BAB V KESIMPULAN DAN SARAN

Dalam bab ini berisikan tentang simpulan dan saran dari pembuatan alat dan laporan sebagai upaya untuk perbaikan kedepan.

#### 6. DAFTAR PUSTAKA

#### 7. LAMPIRAN

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Sistem Proteksi**

Sistem proteksi adalah suatu sistem keamanan untuk peralatan listrik, sistem proteksi dipicu oleh berbagai faktor, antara lain kesalahan kerja, gangguan khusus dan penyebab lainnya. Sistem proteksi digunakan untuk pengamanan listrik pada sistem tenaga listrik yang terpasang pada sistem distribusi tenaga listrik, transformator tenaga, transmisi tenaga listrik dan generator listrik yang dipergunakan untuk mengamankan sistem tenaga listrik dari gangguan atau kelebihan beban sehingga sistem kelistrikan yang tidak terputus (aliran arus ke beban atau konsumen) dapat terus berfungsi. Oleh karena itu, keselamatan sistem tenaga listrik pada dasarnya memerlukan pengamanan seluruh sistem tenaga listrik supaya kehandalan tetap terjaga (Azis and Febrianti 2019).

#### **2.2. Sistem Monitoring Melalui Internet of Things (IoT)**

Konsep sistem pemantauan melalui internet memungkinkan pengguna untuk menghubungkan, mengontrol, memperoses dan pemantauan sistem secara langsung melalui internet atau online. Konsep SMART (Specific, Measurable, Attainable, Relevant, Time - bound) spesifik, terukur, dapat dicapai, relevan, dalam rentang waktu. Banyak yang memanfaatkan realtime monitoring ini secara wireline ini seperti LCD dan tidak sedikit juga yang memanfaatkannya secara wireless seperti bluetooth, text message, dan juga web. Internet of things (IoT) adalah proses menghubungkan perangkat seperti komputer, smartphone, sensor dan aktuator melalui internet. Perangkat yang terhubung ini dapat menghasilkan data-data yang dapat digunakan oleh manusia atau sistem lain. (Amaro 2017).



### 2.3. IoT (Internet of Things)

Konsep Internet of Things lebih sering disebut IoT adalah sebuah pemikiran dimana semua item pada kenyataannya dapat berkomunikasi satu sama lain sebagai bagian dalam satu sistem terkordinasi dengan manfaat internet sebagai titik kontak. Misalnya, ruang kendali, yang jaraknya bisa beberapa kilo meter adalah tempat CCTV yang dipasang disepanjang jalan terhubung dengan koneksi internet atau sebaliknya, rumah pintar yang bisa dilihat online dari ponsel dengan bantuan koneksi internet. Pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul informasi, sambungan internet sebagai media komunikasi dan server yang pengumpul dan menganalisis data yang dikumpulkan oleh sensor (Efendi 2018).

### 2.4. NodeMCU ESP32 WROOM 32

Mikrokontroler ESP32 adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik yang dibekali dengan perangkat Wi-Fi dan bluetooth di dalamnya (Zain et al. 2022). ESP32 memiliki beberapa pin I/O sehingga sangat baik dapat dibentuk menjadi aplikasi monitoring maupun kontrol. Modul ESP32 dapat menghubungkan mikrokontroler apapun ke jaringan wifi. ESP32 mampu meng-hosting aplikasi atau mendelegasikan semua fungsi jaringan WiFi ke berbagai aplikasi.

Internet of things (IoT) terhubung ke pengguna NodeMCU ESP-32, yang memungkinkan pemantauan dan kontrol jarak jauh nirkabel melalui jaringan, yang dapat memungkinkan mekanisme kendali jarak jauh yang aman bagi pengguna. Sebuah jaringan yang disiapkan bisa kita atur sesuai dengan kebutuhan. ESP32 dapat kita lihat pada Gambar 2.4.



Gambar 2.4 NodeMCU ESP32

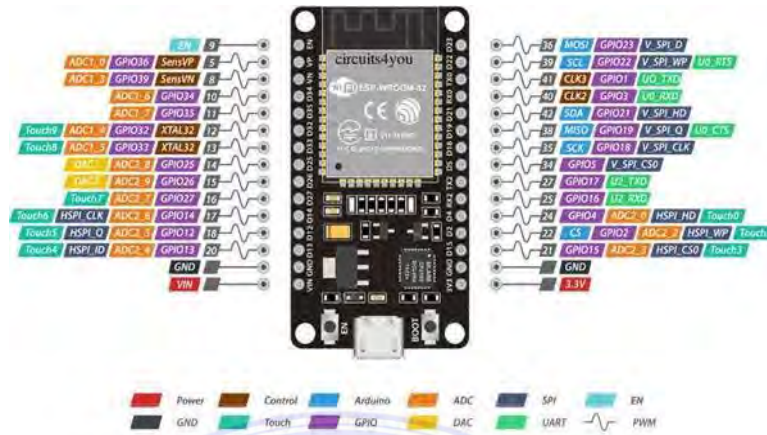
(sumber: <https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>)

#### 2.4.1. Spesifikasi Yang Dimiliki Oleh NodeMCU

Spesifikasi yang dimiliki oleh NodeMCU sebagai berikut ini :

No	Spesifikasi NodeMCU ESP32
1	Jumlah pin : 30 pin termasuk GPIO dan pin tegangan
2	Analog to Digital Converter (ADC) dengan 15 pin
3	UART Interface
4	SPI Interface
5	2 I2C Interface
6	Pulse Width Modulation (PWM) 16 pin
7	Digital to Analog Converter (DAC) 2 pin
8	Digital to Analog Converter (DAC) 2 pin
9	Flash Memory 4 MB
10	SRAM : 520 KB
11	Clock Speed : 240 Mhz
12	Wi Fi : IEEE 802.11 b/g/n/e/i
13	Mode supported : AP, STA, AP+ST

Berikutnya adalah Pin Out Module ESP32



ESP32 Dev. Board Pinout

Gambar2.5.1 Pin Out Module ESP32

(sumber: <https://www.edukasiElektronika.com/2019/07/arsitektur-dan-fitur-esp32-module-esp32.html>)

#### 2.4.2. Pin GPIO ESP32 WROOM DevKit V1.

Pada board ESP32 DevKit terdapat 25 pin GPIO (*General Purpose Input Output*) dengan masing-masing pin memiliki karakteristik tersendiri. Pin hanya sebagai INPUT :

a. GPIO 34
b. GPIO 35
c. GPIO 36
d. GPIO 39

Pin dengan internal pull up, dapat di atur melalui program :

a. GPIO14
b. GPIO16
c. GPIO17
d. GPIO18
e. GPIO19
f. GPIO21
g. GPIO22
h. GPIO23

Pin tanpa internal pull up (anda dapat menambahkan pull up luar sendiri) :

a. GPIO13
b. GPIO25
c. GPIO26
d. GPIO27
e. GPIO32
f. GPIO33

## 2.5. Arduino Nano

Definisi arduino nano adalah papan sirkuit berukuran kecil dimana didalamnya telah tersedia mikrokontroler serta sudah dapat didukung oleh penggunaan breadboard. Arduino Nano diciptakan dengan basis mikrokontroler Atmega 328.

Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B (Alvionita, Vella 2019).



Gambar 2.5 Arduino Nano

(Sumber: <https://www.aldyrazor.com/2020/08/arduino-nano.html>)

### 2.5.1. Arduino Nano Pinout

Seperti yang kita ketahui, Arduino Nano merupakan produk papan sirkuit mikrokontroler kecil yang memiliki beberapa pin. Berikut ini pinout Arduino Nano.



Gambar 2.5.1 Arduino Nano Pinout

(Sumber: <https://www.aldyrazor.com/2020/08/arduino-nano.html>)



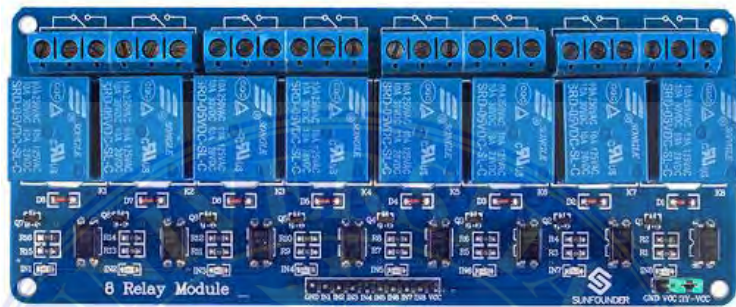
Berikut ini ringkasan spesifikasi arduino nano dapat dilihat pada tabel di bawah ini:

No	Spesifikasi Arduino Nano	
1	Jenis Mikrokontroler	Atmega328
2	Tegangan Operasi	5 Volt
3	Tegangan Disarankan	7 - 12 Volt
4	Batas Tegangan	6 - 20 volt
5	Pin Input/Output Digital	14
6	Pin PWM	6
7	Pin Input Analog	8
8	Arus Per Pin	40 Ma
9	Memori Flash	32 KB
10	SRAM	2 KB
11	EEPROM	1 KB
12	Clock Speed	16 MHz
13	Panjang	4,3 cm
14	Lebar	1,8 cm
15	Berat	5 gram

## 2.6. Modul Relay

Modul Relay adalah bagian elektronik yang dapat digunakan sebagai alat yang bekerja seperti saklar. Relay juga bekerja dengan prinsip yang sama seperti saklar mekanisme yang dapat digerakan oleh listrik. Relay dapat melakukan proses pembukaan dengan memanfaatkan gaya elektro magnetik. (Yoga Widiana, Raka Agung, and Rahardjo 2019).

Pada umumnya, *relay* 5 volt DC digunakan untuk pembuatan project yang membutuhkan tegangan tinggi atau yang sifatnya AC (*Alternating Current*). Perbedaan yang paling mendasar antara *relay* dan saklar adalah pada saat pemindahan dari posisi on ke off. Relay melakukan pemindahannya secara otomatis dengan arus listrik, sedangkan saklar dilakukan dengan cara manual.



Gambar 2.5. Modul Relay

(Sumber: <https://www.aldyrazor.com/2020/05/modul-relay-arduino.html>)

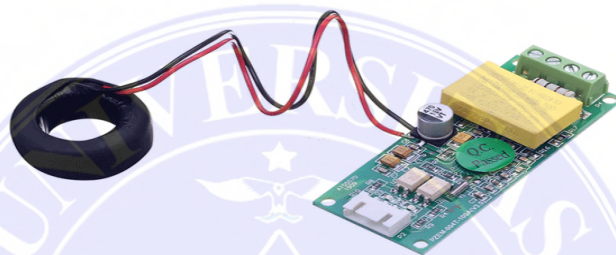
### 2.6.1. Spesifikasi Modul Relay Yang Digunakan:

Adapun Spesifikasi Yang Dimiliki Modul Relay Ini Adalah

No	Spesifikasi Modul Relay
1	Dapat menjalankan fungsi logika 5 volt dari Arduino / ESP32
2	Sarana untuk mengendalikan tegangan tinggi 240VAC hanya menggunakan tegangan rendah 5 VDC
3	Membatasi jika terjadinya penurunan tegangan
4	Memungkinkan penggunaan kemampuan penundaan waktu atau fungsi <i>time delay function</i>
5	Melindungi bagian komponen lainnya dari penyebab <i>korsleting</i> karena dilengkapi proteksi <i>optocouples</i>
6	Menyederhanakan rangkaian agar lebih ringkas.

## 2.7. Sensor PZEM-004T

PZEM-004T adalah sensor yang dapat digunakan untuk mengukur tegangan rms, arus rms dan daya aktif yang dapat dihubungkan melalui arduino ataupun platform opensource lainnya. Dimensi fisik dari papan PZEM-004T adalah  $3,1 \times 7,4$  cm, dibundel dengan kumparan trafo arus diameter 3mm yang dapat digunakan untuk mengukur arus maksimal sebesar 100A. (Anwar et al. 2019).



Gambar 2.7 Modul Sensor PAEM-004T

(Sumber: <https://roboticx.ps/product/pzem-004t-current-voltage-module-ac-80-260v-100a/>)

PZEM-004T berfungsi untuk mengukur : Voltage / Tegangan, Arus, Daya, Frekuensi, Energi dan Power Faktor. Dengan kelengkapan fungsi / feature ini, maka modul PZEM-004T sangat ideal untuk digunakan sebagai project maupun eksperimen alat pengukur daya pada sebuah jaringan listrik seperti rumah atau gedung.

Pengkabelan modul ini memiliki 2 bagian, khususnya dari kabel terminal informasi untuk tegangan dan arus, serta kabel komunikasi serial. Berdasarkan pada kebutuhan, modul ini memiliki papan pin TTL untuk mendukung komunikasi data serial antar hardware. Jika pengguna ingin mengkomunikasikan PZEM-004T ini dengan perangkat yang memiliki port USB atau RS-232 (seperti komputer), diperlukan lagi kabel konverter (TTL

ke USB, TTL ke RS232). Spesifikasi sensor PZEM-004t yang dipakai dalam pembuatan alat ini adalah :

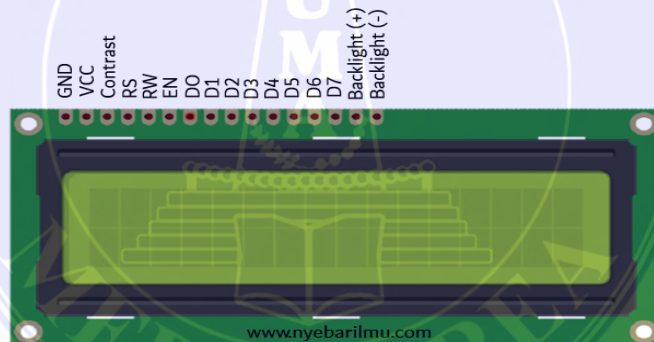
No	Spesifikasi PZEM-004t
1	Brand Name : diymore
2	Model Number : PZEM-004T Module
3	Type : Voltage Regulator
4	Dissipation Power : 1
5	Operating Temperature : 1
6	Application : Computer
7	Supply Voltage : 80-260V
8	Match 1 : pzem-004t module
9	Match 2 : pzem-004t test module
10	Match 3 : AC digital meter
11	Max Operating Current : 100A – 119A
12	Rated Voltage : 220V
13	Display Type : Digital Only
14	Power Supply : AC

## 2.8. LCD 16X2

LCD 16×2 (*Liquid Crystal Display*) adalah modul penampil informasi yang menggunakan kristal cair sebagai bahan untuk penampil informasi melalui tulisan maupun gambar. Pengaplikasian pada kehidupan sehari – hari yang mudah dijumpai antara lain pada kalkulator, gamebot, televisi, atau pun layar komputer. Modul kontrol tampilan I2C digunakan untuk menghubungkan LCD ke komputer. Sebagai hasilnya, Vcc, ground, SDA dan SCL adalah port penggunaan papan. Modul kontrol tampilan I2C dapat digunakan untuk menyesuaikan kontras karakter yang ditampilkan. (Alipudin and et. al 2019).

No	Spesifikasi LCD 16×2
1	Memiliki dua baris dan 16 karakter
2	Memiliki 192 karakter yang disimpan
3	Sebuah generator karakter yang dapat diprogram
4	Dapat ditangani dalam mode 4-bit dan 8-bit
5	Memiliki lampu latar

Pedoman cara menggunakan LCD. jalur LiquidCrystal (2, 3, 4, 5, 6, 12,7), adalah cara paling umum untuk menghubungkan pin Arduino ke pin LCD RS,Empower,D4,D5,D6, dan D7. Pada baris ini lcd adalah variabel yang dipanggil setiap saat.



Gambar 2.8 LCD (Liquid Crystal Display)

(Sumber: <https://www.indiamart.com/proddetail/16x2-lcd-green-display-9425552233.html>)

Keterangan :

**GND** : catu daya 0Vdc

**VCC** : catu daya positif

**Constrate** : untuk kontras tulisan pada LCD

**RS** : Register Select :

**R/W** atau Read/Write

**High** : mengirim data



Low : mengirim instruksi

Disambungkan dengan LOW untuk pengiriman data ke layar

E (enable) : untuk mengontrol ke LCD ketika bernilai LOW, LCD tidak dapat diakses.

**D0 – D7** = Data Bus 0 – 7

**Backlight +** : disambungkan ke VCC untuk menyalakan lampu latar

**Backlight –** : disambungkan ke GND untuk menyalakan lampu latar

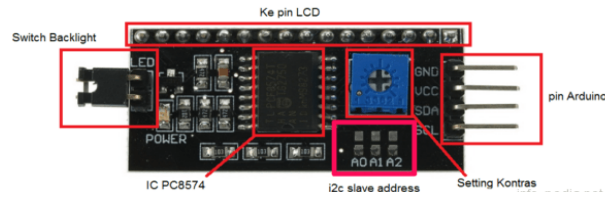
### 2.8.1. Modul I2C LCD

Inter Integrated Circuit atau sering I2C adalah standar untuk komunikasi serial dua arah dengan dua saluran yang dibuat untuk kontrol IC. saluran SCL (Serial Clock) dan SDA (Serial Data) dalam sistem I2C mentransfer data antara I2C dan pengontrolan. Untuk menghemat pin-pin pada Arduino ada 1 modul LCD yang bisa memanfaatkan untuk alternatif mengakses LCD yaitu modul LCD PCF8574. Pada modul tersebut menggunakan antarmuka atau interface I2C, sehingga hanya membutuhkan 2 pin saja yaitu SDA dan SCL.

### 2.8.2. Spesifikasi Modul LCD I2C

No	Spesifikasi Modul LCD I2C
1	Tegangan beroperasi antara 2-5 Vdc
2	Kontrol pin Sda dan Scl
3	Kompatibel dengan berbagai jenis mikrokontroler
4	Kendali 8 bit menggunakan antarmuka i2c
5	Open-drain interrupt output

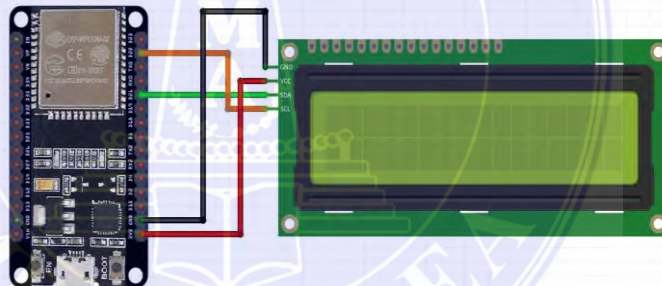
Dirver LCD ini dikemas ke dalam modul LCD i2c, seperti gambar dibawah ini.



Gambar 2.8.2 Modul I2C

(sumber; <http://digilib.polban.ac.id/files/disk1/161/jbptppolban-gdl-noviyansup-8021-3-bab2--3.pdf>)

Untuk menyambungkan LCD dengan board ESP-32 memerlukan 2 pin untuk mengendalikan sebuah modul LCD. Dengan menggunakan modul I2C ini dapat mengurangi penggunaan pin pada board ESP-32 yang hanya menggunakan 2 pin yaitu pin D22 dan D21 yang dihubungkan dengan SCL dan SDA untuk menghubungkan LCD dengan board ESP-32.



Gambar 2.8.3 LCD Dengan Board ESP32

## 2.9. Konektor

Konektor adalah sebuah alat sebagai rangkaian elektronika untuk mengubah tegangan listrik yang besar menjadi tegangan yang lebih sederhana, atau rangkaian untuk mengubah arus putar (aliran AC) menjadi arus searah (aliran DC). Konektor/catu daya merupakan bagian tengah dari perangkat keras elektronika. Konektor tersebut digunakan untuk menurunkan tegangan AC 220 Volt menjadi sedikit antara 3 volt hingga 12 volt sesuai kebutuhan perangkat elektronik.

Konektor dapat dibagi menjadi empat macam diantaranya adalah :

1. Konektor DC Converter, adalah konektor yang dapat mengubah tegangan DC yang sangat besar menjadi tegangan DC yang kecil. Misalnya: Dari tegangan 12v ke tegangan 6v.
2. Adaptor Step Up dan Step Down. Adaptor Step Up adalah konektor yang dapat mengubah tegangan AC yang kecil menjadi tegangan AC yang besar. Misalnya: Dari tegangan 110v ke tegangan 220v. Sedangkan Konektor Step Down adalah konektor yang dapat merubah tegangan AC yang besar menjadi tegangan AC yang kecil. Misalnya: Dari tegangan 220v ke tegangan 110v.
3. Konektor Inverter, adalah konektor yang dapat mengubah tegangan DC yang sedikit menjadi tegangan AC yang sangat besar. Misalnya: Dari tegangan 12v DC ke 220v AC.
4. Konektor Power Supply, adalah konektor yang dapat merubah tegangan AC yang besar menjadi tegangan DC yang kecil. Misalnya: Dari tegangan AC 220v ke tegangan 6v, 9v, atau 12v DC.



Gambar 2.9 konektor Power Supply

(Sumber: [https://shopee.co.id/Adaptor-Power-Supply-DC-5-Volt-2-Ampere-\(5V-2A\)-Switching-i.3468034.1370269156](https://shopee.co.id/Adaptor-Power-Supply-DC-5-Volt-2-Ampere-(5V-2A)-Switching-i.3468034.1370269156))

## 2.10. Sensor DHT11

Sensor DHT11 merupakan modul sensor yang mampu mendeteksi objek untuk mengukur suhu dan kelembaban di dalam ruangan yang memiliki tegangan keluaran sederhana yang juga dapat ditangani menggunakan mikrokontroler (Alipudin dkk 2019). Sensor DHT11 secara keseluruhan memiliki penyesuaian termasuk untuk pembacaan suhu dan kelembaban yang sangat presisi. Penyimpanan data kalibrasi tersebut terdapat pada memori program OTP yang disebut juga dengan nama koefisien kalibrasi.



Gambar 2.10 Sensor DHT11

(Sumber: <https://iotkece.com/cara-mudah-mengakses-sensor-dht-11-dengan-arduino/>)

### 2.10.1. Spesifikasi DHT11

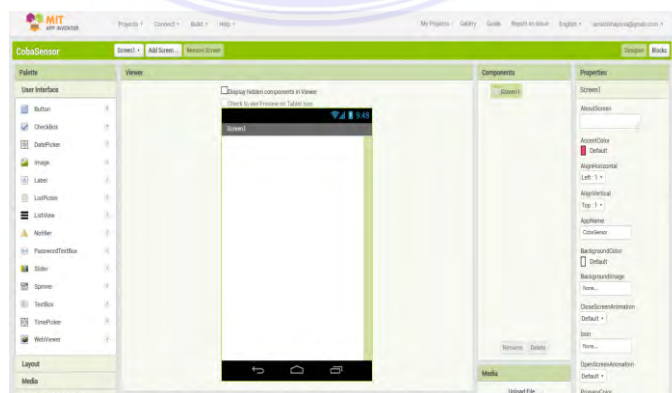
No	Spesifikasi DHT11
1	Tegangan kerja = 3.3V-5V.
2	Arus maksimum = 2.5mA
3	Range pengukuran kelembaban = 20%-80%
4	Akurasi pengukuran kelembaban = 5%
5	Range pengukuran suhu = 0°C-50°C
6	Akurasi pengukuran suhu = 2°C
7	Kecepatan pengambilan sampel tidak lebih dari 1 Hz (setiap detik)

## 2.11. MIT APP Inventor

App Inventor adalah platform untuk memudahkan dalam pembuatan aplikasi sederhana tanpa harus mempelajari atau menggunakan bahasa pemrograman yang terlalu banyak. Kita dapat mendesain aplikasi android sesuai keinginan dengan menggunakan bermacam layout dan komponen yang tersedia, karena APP Inventor ini memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi Android. Dengan app inventor, Sistem operasi berbasis android memungkinkan pengguna untuk membuat aplikasi perangkat lunak melalui pemrograman komputer.

Aplikasi MIT adalah alat pemrograman berbasis blok memungkinkan semua orang, bahkan pemula, untuk mulai memprogram dan membuat aplikasi yang sepenuhnya berguna untuk gadget Android. MIT Application Creator dibuat oleh Google. MIT untuk mengenal dan mengembangka pemograman android. (Syafitri K, Salama I, and Solihin 2020).

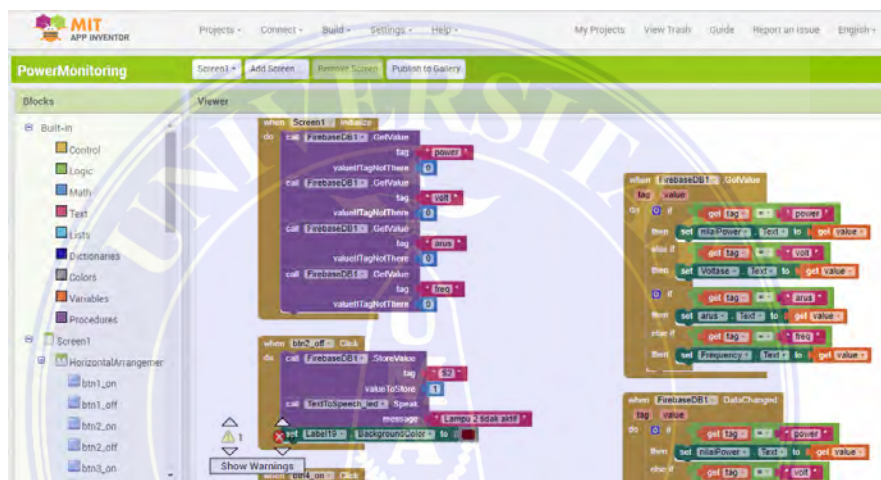
MIT Application Creator, ada dua halaman dasar, khususnya halaman dsigner dan halaman blok. Halaman designer digunakan untuk merencanakan desain aplikasi dengan berbagai bagian dan format yang diberikan sesuai keinginan. Sedangkan halaman blok digunakan untuk memprogram aplikasi android sesuai dengan keinginan .Gambar 2.11 di bawah ini merupakan tampilan jendela utama MIT Inventor untuk membuat aplikasi monitoring android pada pengembangan alat.



Gambar 2.11 Tampilan Halaman Utama MIT APP Inventor



App Inventor menggunakan antarmuka grafis, serupa dengan antarmuka pengguna pada Scratch dan StarLogo TNG, yang memungkinkan pengguna untuk men-drag-and-drop objek visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat Android. Dalam menciptakan App Inventor, Google telah melakukan riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembangan online Google. Hasil dari aplikasi yang telah dibangun pada penelitian ini berfungsi sebagai media untuk memonitor data.(Hilmansyah et al. 2020)



Gambar 2.11.1 Tampilan Halaman Kedua MIT APP Inventor

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Tempat Dan Waktu Pelaksanaan

##### 3.1.1. Tempat Penelitian

Kegiatan penelitian ini dilakukan di CV.Angkasa Mobie Tech JL.sultan serdang Dusun II Sena Gg.ikhlas kec.Batangkuis kab deli serdang, Sumatera utara.

##### 3.1.2. Waktu Penelitian

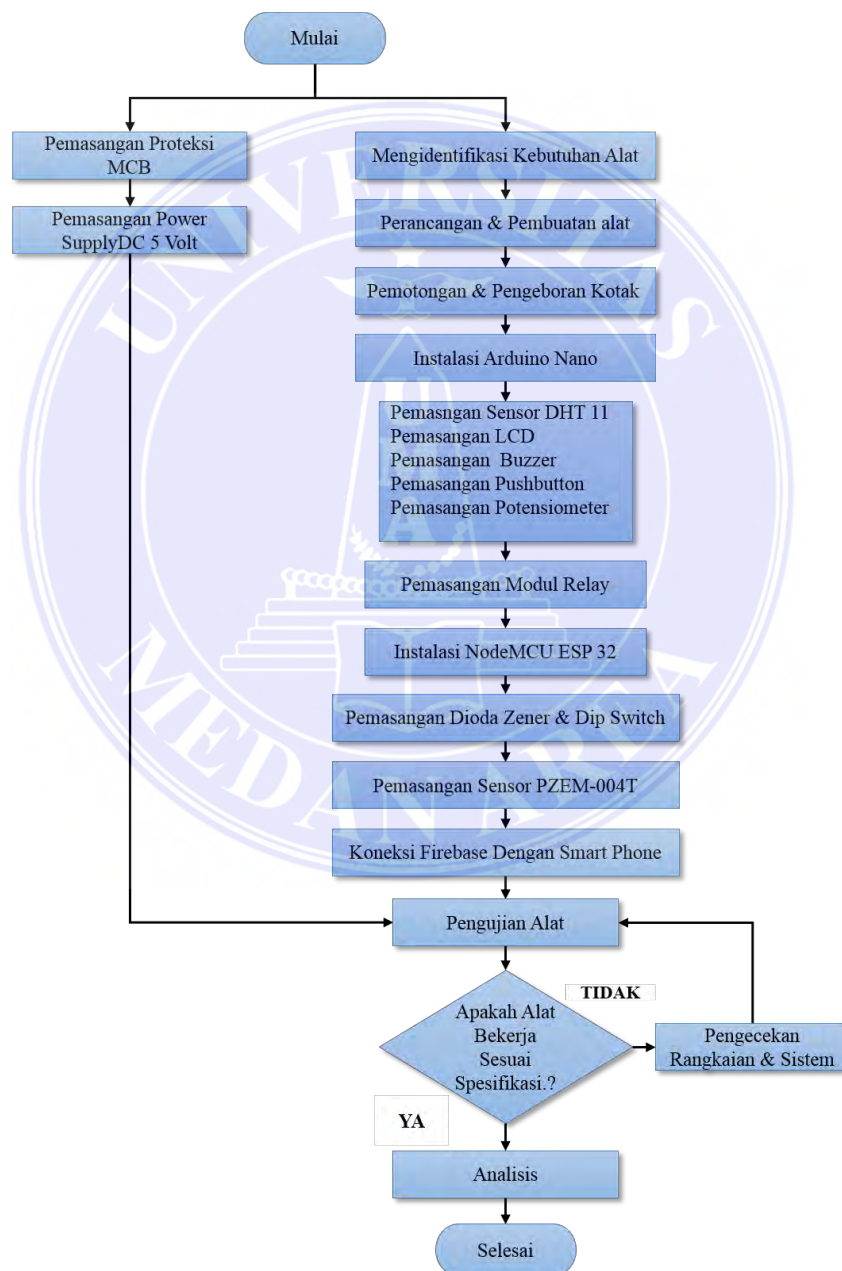
Waktu penelitian yang dibutuhkan dalam penelitian ini kurang lebih 6 bulan, berikut jadwal penelitian:

**Tabel 3.1. Waktu Pelaksanaan**

No.	Kegiatan	Bulan					
		Juni	Juli	agustus	september	juni	Juli
1	Studi Literatur						
2	Proposal						
3	Menyiapkan Alat dan Bahan						
4	Pembuatan Alat						
5	Pengumpulan Data						
6	Analisa Data						
7	Seminar Hasil						
8	Sidang						

### 3.2. Kerangka Berpikir

Untuk mempermudah proses dan memudahkan dalam memahami arah penelitian, maka penelitian ini dilakukan secara bertahap. Bagan alur kerangka penelitian dapat dilihat pada Gambar 3.2 di bawah ini. Peneliti melakukan proses perancangan atas dasar diagram alur ini, rancang bangun prototipe otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT ESP 32.



Gambar 3.2 Flowchart Pembuatan Alat

### 3.3. Mengidentifikasi Kebutuhan Alat

Tabel 3.3 Mengidentifikasi Kebutuhan Alat

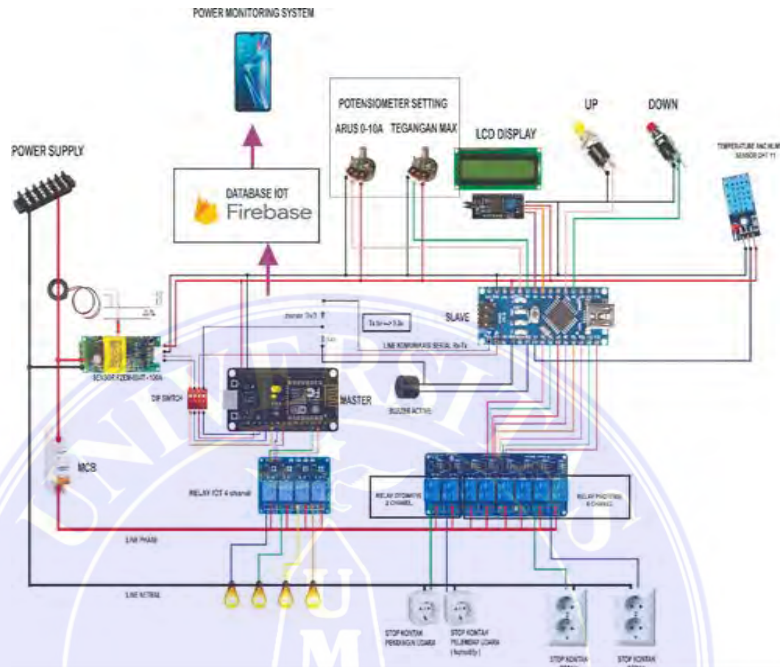
No.	NAMA	SPEKIFIKASI	JUMLAH
1	ESP32 WROOM-32	Microprosesor Xtensa Dual-Core 32 Bit LX6 Freq Clock up to 240 MHz SRAM 520 kB Flash memori 4 MB 11 b/g/n WiFi transceiver Bluetooth 4.2/BLE 48 pin GPIO 15 pin channel ADC 25 pin PWM 2 pin channel DAC	1
2	Power Supply	Input 200 ~ 240 VAC, output 5VDC, 2A	1
3	Arduino Nano 328P	Flash memori : 32 Kbyte kbyte (ATmega328) EEPROM : 512 byte (Atmega328) Kecepatan clock : 16 MHz Ukuran board : 4,5 mm x 18 mm Berat : 5 gram	
4	LCD 16x2	-	1
5	Module I2C LCD	Power : DC 5V. Support LCD 1602 dan 2004 (LCD 16x2, LCD 16x4) Kontrol pin : SDA dan SC	1

6	Sensor PZEM-004t	Working voltage: 80 ~ 260VAC Rated power: 100A / 22000W Working Frequency: 45-65Hz Measurement accuracy: 1.0	1
7	MCB 6A	240 Vac, 6A	1
8	Module Relay 5v 8 chanel	Input 5Vdc, kontak 220 Vac, 10A	4
9	Sensor DHT 11	-	1
10	Stop kontak 4 lubang	10 A, 240 Volt	1
11	Stop kontak 1 lubang	6 A, 240 Volt	4
12	Stop kontak 2 lubang	6A, 240 Volt	1
13	Lampu hemat energi	40w, 220 V <sub>AC</sub>	4
14	Fiting lampu	2A, 240 Volt	4
15	Fuse	2A, 1A	3
16	Buzzer lamp	Red 220Vac	56
17	Box alat	Hitam	2
18	Terminal block	12 chanel	1 batang
19	skrup	3 mm	24
20	Kertas Stiker	A4	3
21	MCB	6 A 240 Volt	1



### 3.4. Perancangan Dan Pembuatan Alat

Perancangan rangkaian keseluruhan dengan menggunakan Coler draw X8



Gambar 3.4 Rangkaian Keseluruhan

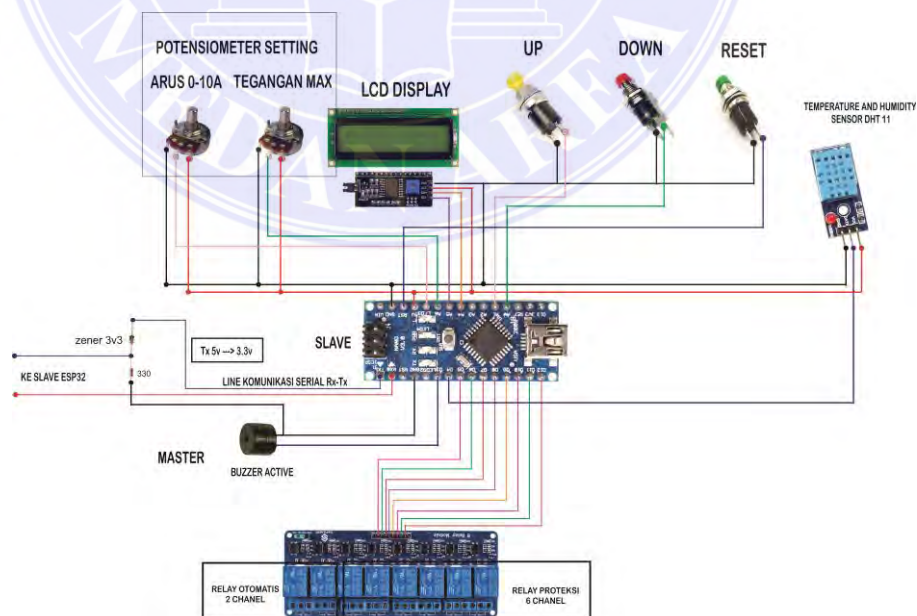
### 3.5. Pemotongan Dan Pengeboran Kotak

Pengeboran adalah merupakan kegiatan melubangi dengan mata bor bertujuan untuk membuat lubang melingkar pada benda padat. Pengeboran pada umumnya akan bekerja dengan cara diputar berkecepatan tinggi.

Pemotongan merupakan pemisahan benda padat menjadi bagian dua atau lebih melalui aplikasi gaya yang terarah melalui luas bidang permukaan yang kecil. Benda yang umum digunakan untuk memotong adalah pisau, gergaji dan gunting.

### 3.6. Instalasi Arduino Nano

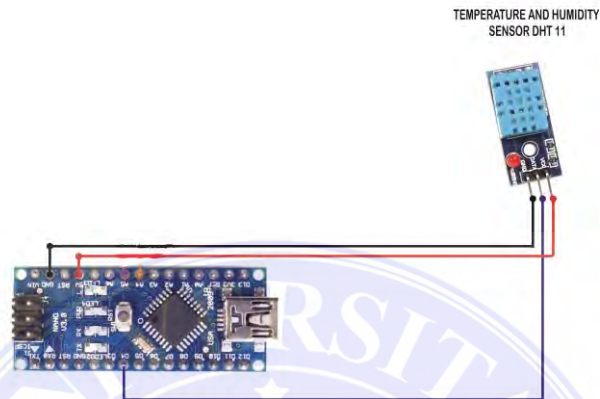
1. Pemasangan pin GND potensiometer pada pin GND arduino, pemasangan pin OUT potensio pada pin A6 dan A7 arduino, pemasangan pin 5V potensio pada pin 5V arduino.
2. Pemasangan pin SCL lcd pada pin A5 arduino, pemasangan pin SDA lcd pada pin A4 arduino, pemasangan pin VCC Lcd pada pin 5V arduino, pemasangan GND lcd pada GND arduino.
3. Pemasangan push button UP Dan Down pada pin A0 dan A1, pemasangan GND push button pada GND arduino.
4. Pemasangan pin VCC sensor DHT 11 pada pin 5V arduino, pemasangan pin DATA pada pin D4 arduino, pemasangan pin GND pada GND arduino.
5. Pemasangan pin negatif buzzer pada pin GND arduino, pemasangan pin positif buzzer pada pin D3 arduino.
6. Pemasangan pin relay 8 chanel pada pin D12 sampai D5 pada arduino.
7. Pemasangan dioda zener pada line komunikasi serial on-board antara pin Rx dan Tx arduino.



Gambar 3.6 Rangkaian Instalasi Arduino Nano

### 3.7. Pemasangan Sensor DHT 11

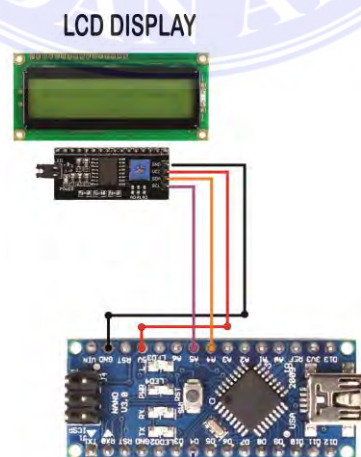
1. Pemasangan pin VCC sensor DHT 11 pada pin 5V arduino, pemasangan pin DATA pada pin D4 arduino, pemasangan pin GND DHT 11 pada pin GND Arduino.



Gambar 3.7 Ragkaian Sensor DHT 11

### 3.8. Pemasangan LCD

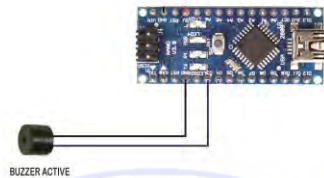
1. Pemasangan pin SCL lcd pada pin A5 arduino, pemasangan pin SDA lcd pada pin A4 arduino, pemasangan pin VCC lcd pada pin 5V arduino, pemasangan GND lcd pada GND arduino.



Gambar 3.8 Rangkaian Lcd Display

### 3.9. Pemasangan Buzzer

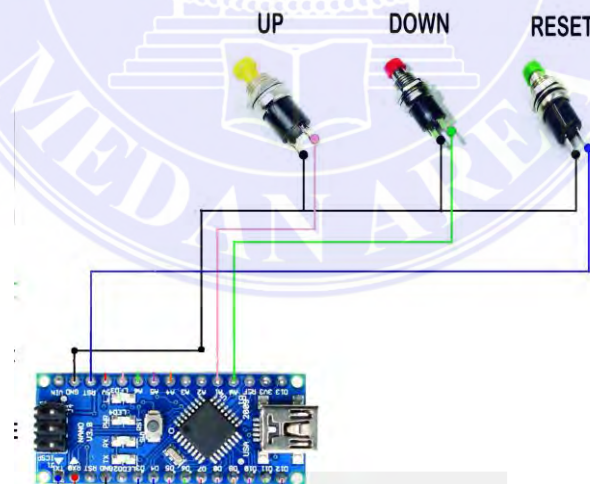
1. Pemasangan pin negatif buzzer pada pin GND arduino, pemasangan pin positif buzzer pada pin D3 arduino.



Gambar 3.9 Rangkain Pemasangan Buzzer

### 3.10. Pemasangan Push Button

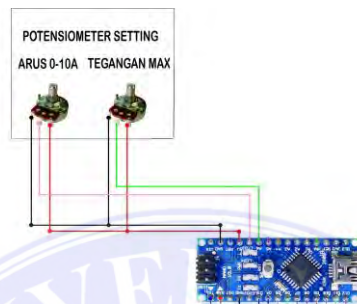
1. Pemasangan push button UP Dan DOWN pada pin A0 dan A1 arduino, pemasangan push button reset pada tombol reset arduino, pemasangan GND push button pada GND arduino.



Gambar 3.10 Rangkaian Pemasangan Push Button

### 3.11. Pemasangan Potensiometer

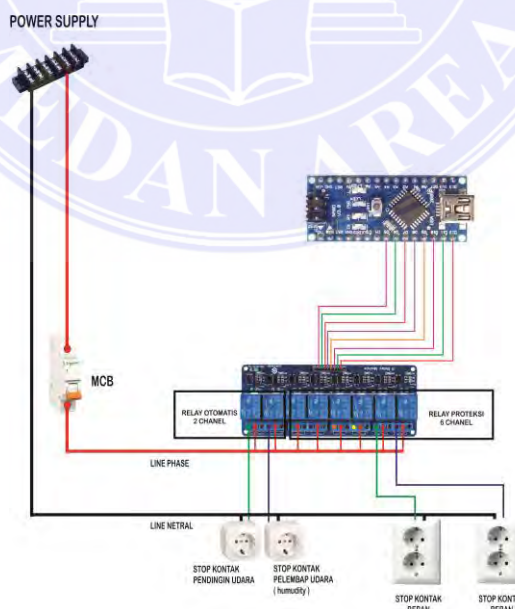
1. Memasang pin GND potensiometer ke pin GND arduino nano.
2. Memasang pin OUT potensio pada pin A6 dan A7 arduino, pin 5V potensio pada pin 5V arduino.



Gambar 3.11 Rangkaian Pemasangan Potensiometer

### 3.12. Pemasangan Modul Relay

1. Pemasangan pin relay 8 channel pada pin D12 sampai D5 pada arduino.

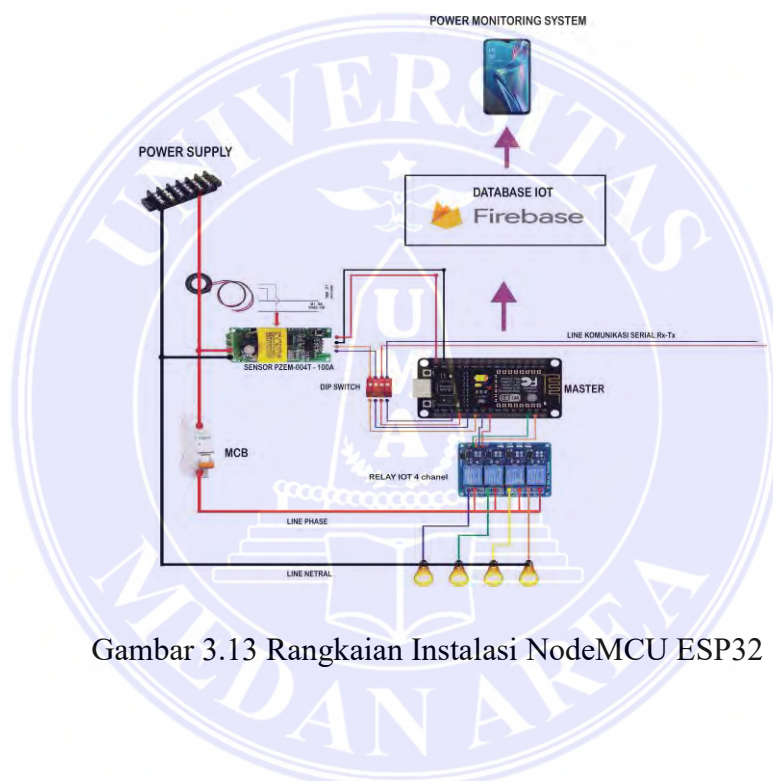


Gambar 3.12 Rangkaian Pemasangan Module Relay



### 3.13. Instalasi NodeMCU ESP32

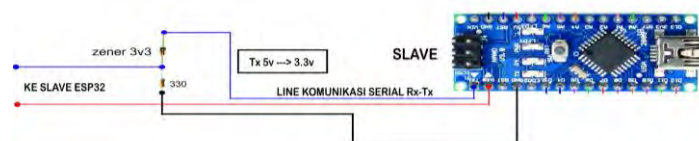
1. Pemasangan pin VCC sensor PZEMM 004T pada pin VIN NodeMCU ESP32, pemasangan pin GND PZEMM 004T pada pin GND NodeMCU ESP32, pemasangan pin Rx Tx pada pin D7 Dan D8 NodeMCU ESP32
2. Pemasangan modul relay 4 chanel pada pin D1,D2,D5,D6. NodeMCU ESP32.
3. Pemasangan dip switch pada line komunikasi serial Rx Tx Node MCU ESP32, dan Rx Tx sensor PZEMM 004T pada pin D7 dan D8 NodeMCU ESP32.



Gambar 3.13 Rangkaian Instalasi NodeMCU ESP32

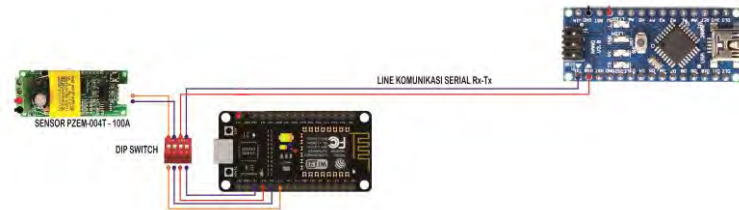
### 3.14. Pemasangan Dioda Zener Dan Dip Switch

1. Pemasangan dioda zener pada line komunikasi serial pin Rx dan Tx pada arduino. Dioda zener berfungsi untuk menurunkan tegangan 5V ke 3,3V.



Gambar 3.14 Rangkaian Pemasangan Dioda Zener

2. Pemasangan dip switch pada line komunikasi serial Rx Tx Node MCU ESP32, dan Rx Tx Sensor PZEMM 004T pada pin D7 dan D8 NodeMCU ESP32.



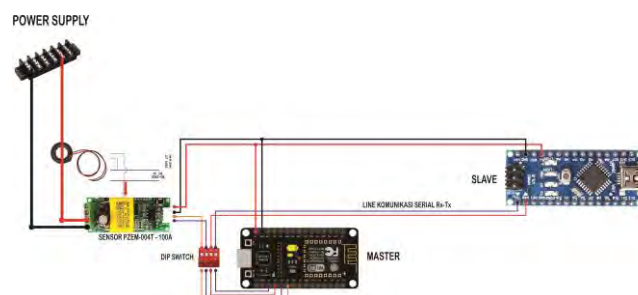
Gambar 3.14.1 Rangkaian Pemasangan Dip Switch

### 3.15. Koneksi Firebase Dengan Smart Phone Dan Perangkat

Database berfungsi untuk mengumpulkan data secara efektif menyimpan informasi yang digunakan oleh suatu aplikasi atau website. Misalnya, dalam website, database menyimpan data-data seperti jumlah visitor, konten yang diunggah, atau konten yang diedit. Firebase Realtime Database adalah produk layanan dari Google. Firebase ada di dalam host pada cloud. Realtime, artinya seluruh perangkat yang terhubung akan menerima update dalam waktu mili detik.

### 3.16. Pemasangan Module Sensor PZEM 004T

1. Pemasangan pin VCC sensor PZEMM 004T pada pin VIN NodeMCU ESP32, pemasangan pin GND PZEMM 004T pada pin GND NodeMCU ESP32, pemasangan pin Rx Tx pada pin D7 dan D8 NodeMCU ESP32.



Gambar 3.16 Rangkaian Sensor PZEM 004T

### 3.17. Pemasangan Peroteksi Mcb

Bagian ini berfungsi sebagai kerangka pengaman di instalasi listrik jika terjadi kelebihan arus listrik atau korsleting.



Gambar 3.17 Miniatur Circuit Breaker

### 3.18. Pemasangan Power Suplay 5 Volt

Konektor Power Supply, adalah konektor yang dapat mengubah tegangan AC yang sangat besar menjadi tegangan DC yang kecil. Misalnya: Dari tegangan AC 220v ke 5v, 9v, atau 12v DC



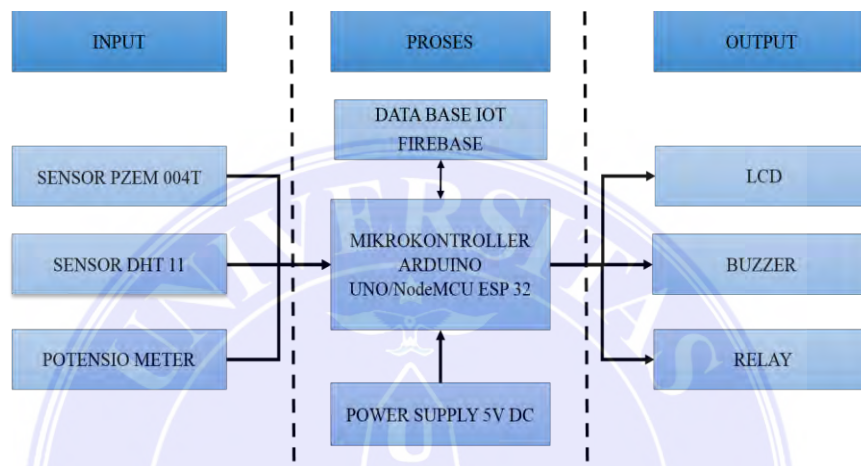
Gambar 3.18 Power Suplay 5 Volt

### 3.19. Pengujian Alat

Pengujian perangkat direncanakan untuk melihat apakah kerangka kerja yang dibuat sudah sesuai dengan yang telah direncanakan. Sistem pengujian ini menggabungkan pembacaan nilai tegangan, arus, daya dinamis, dan energi yang digunakan membaca nilai besaran tegangan, Arus, Daya aktif dan Energi yang dipakai.

### 3.20. Diagram Blok

Diagram Blok Salah satu cara termudah untuk memahami operasi kerangka kerja adalah melalui diagram blok. Dengan menggunakan diagram blok, kita dapat melihat bagaimana rangkain bekerja dan apa komponen utama untuk membuat alat ini.



Gambar 3.20 Diagram Blok

#### 3.20.1. Fungsi –fungsi Diagram Blok

1. PZEM-004T adalah sensor yang dapat dihubungkan ke arduino atau platform open source lainnya untuk mengukur tegangan rms, arus rms dan daya aktif.
2. Sensor DHT11 merupakan modul sensor yang mampu mengatur suhu dan kelembaban yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler.
3. Arduino nano dan Nodemcu ESP32 sebagai pengendali dari sensor Pzem, Suhu dan komponen lainnya
4. LCD berfungsi untuk menampilkan nilai dari pengukuran.
5. Buzzer berfungsi untuk mengubah getaran listrik menjadi getaran suara



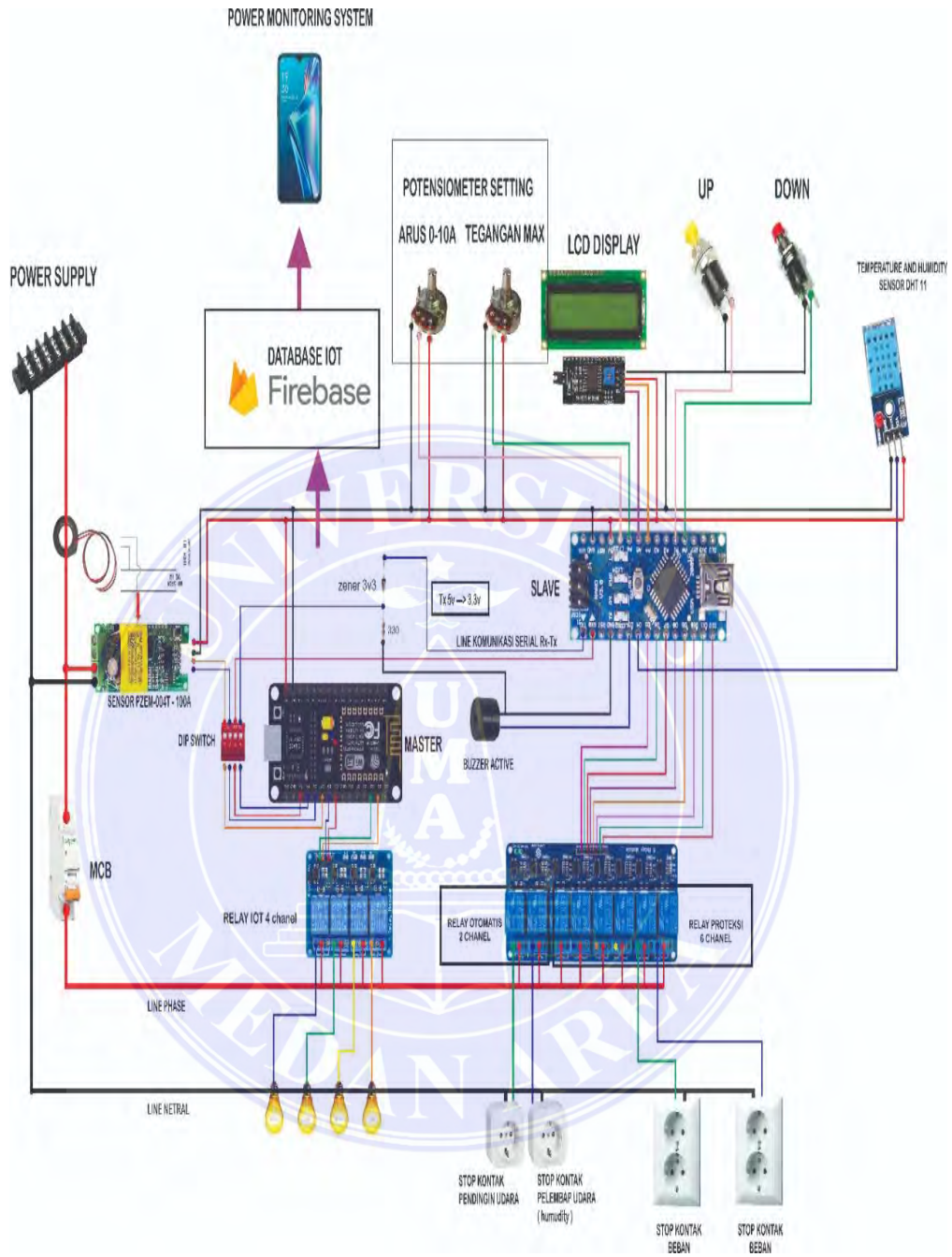
6. Database berfungsi kumpulan data menyimpan informasi yang di gunakan oleh aplikasi atau situs.
7. Power supply atau catu daya mengubah tegangan AC yang tinggi menjadi tegangan DC yang lebih rendah.
8. Relay berfungsi sebagai alat yang layaknya bekerja sebagai sebuah saklar /switch, prinsip saklar mekanik yang dapat digerakkan oleh energi listrik.

### 3.21. Rangkaian Keseluruhan

Rangkaian keseluruhan adalah alur kerja sistem secara sederhana yang bertujuan untuk menerangkan cara kerja sistem alat berupa rancang bangun prototipe otomatisasi proteksi dan monitoring listrik rumah tangga dengan IOT ESP32 secara garis besar berupa gambar dengan tujuan agar lebih mudah dimengerti dan dipahami. Pada perancangan ini prototipe otomatisasi proteksi listrik rumah menggunakan ESP32 sebagai komponen pengendali dan pengelolah data sensor utama.

Sensor PZEM-004-T digunakan untuk membaca parameter tegangan, arus, daya, dan faktor daya yang mengalir kemudian data tersebut diolah dan dikalkulasi oleh ESP32 kemudian dikirim ke database dan dimonitoring dengan aplikasi pada smart phone yang dibuat. Pada aplikasi di smart phone juga dapat dilakukan kendali pada unit instalasi penerangan rumah. Untuk lebih detailnya dapat di lihat pada gambar 3.21 di bawah ini.





Gambar 3.21 Rangkaian Keseluruhan Otomatisasi Proteksi Dan Monitoring

### 3.2.2. Rancangan Anggaran Biaya

Rancangan anggaran biaya atau biasa disebut Rab merupakan upaya dilakukan untuk menghitung biaya yang dibutuhkan untuk mengerjakan suatu proyek maupun pekerjaan sehingga dapat diperkirakan berapa total biaya yang dibutuhkan hingga selesai. Tabel 3.22 dibawah ini merupakan rancangan anggaran biaya yang dibutuhkan membuat satu unit prototipe otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT ESP 32.

Tabel 3.22 Spesifikasi Alat Dan Bahan

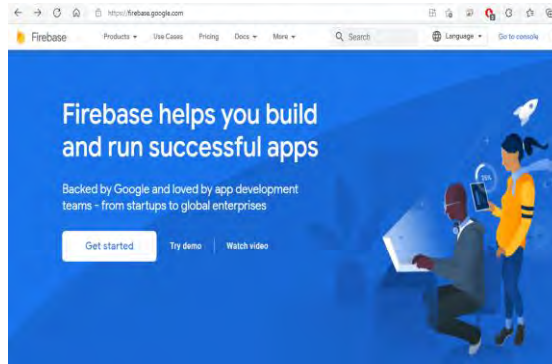
No.	NAMA	HARGA SATUAN	JUMLAH	TOTAL HARGA
1	ESP32 WROOM-32	78.000	1	78.000
2	Power Supply	65.000	1	65.000
3	LCD 16x2	23.500	1	23.500
4	Module I2C LCD	11.500	1	11.500
5	Sensor PZEM-004t	152.500	1	152.500
6	MCB 6A	37.000	1	37.000
7	Module Relay 5v 8 chanel	81.000	1	81.000
8	Sensor asap	30.000	1	30.000
9	Stop kontak 4 lubang	51.000	1	51.000
10	Lampu pijar	22.000	1	22.000
11	Fiting lampu	13.000	1	13.000
12	Fuse	1.200	3	3.600
13	Buzzer lamp	21.400	56	78.400
14	Akrilik	100.000	1 lembar	100.000
15	Terminal block	43.000	2 batang	86.000

16	Mur baut	500	24	12.000
17	Kertas Stiker	6.400	3	19.200
18	Dan lain-lain	150.000	1	150.000
TOTAL				1.087.700

### 3.23. Pembuatan Firebase RealTime Database

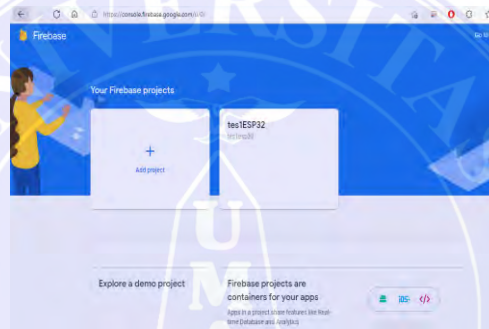
Database adalah suatu tempat di dalam komputer yang berfungsi sebagai tempat untuk mengumpulkan dan menyimpan informasi. Kapasitas ini juga dilakukan secara mendasar didalam komputer untuk mendapatkan data dari database tersebut. Saat ini ada banyak macam kumpulan data . Firebase adalah salah satu kumpulan data yang paling banyak digunakan oleh developer. Firebase sendiri merupakan layanan dari Google. Firebase atau dikenal juga dengan Firebase Google. Database berfungsi untuk menyimpan informasi yang digunakan oleh suatu aplikasi atau website. Misalnya, dalam website, database menyimpan informasi seperti jumlah visitor, konten yang diunggah, atau konten yang diubah.

Firebase Realtime Database adalah bantuan dari Google. Firebase Realtime Database berada didalam host pada cloudnya. Realtime, artinya seluruh perangkat yang terhubung akan menerima update dalam waktu milidetik. Firebase Google dapat diakses secara offline. Pengguna akan menerima notifikasi terkait hal-hal yang terlewat setelah dapat terhubung ke internet kembali. Perangkat klien dapat mengakses, serta dapat diakses langsung dari perangkat seluler. Firebase Realtime Database menjawab kebutuhan data aplikasi yang akan dibuat pada penelitian system monitoring daya listrik Berikut ini langkah-langkah membuat Realtime Database di Firebase. Pertama buka firebase di alamat <https://firebase.google.com/> akan tampil halaman seperti gambar dibawah.



Gambar 3.23 Tampilan Awal Firebase

Tekan tombol Add Project.



Gambar 3.23.1 New Project Firebase

Setelah tombol add project ditekan, akan tampil halaman dibawah. Isi nama project. Lalu tekan tombol Continue.

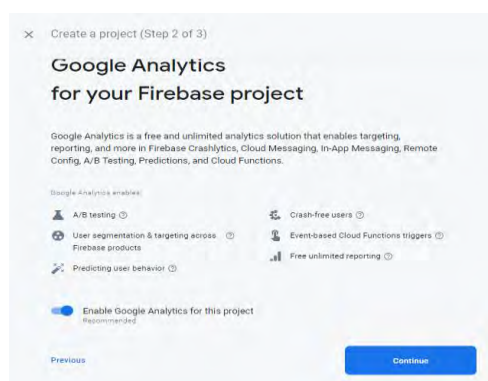


Gambar 3.23.2 Create Project Name

Setelah tombol continue ditekan, Firebase akan menanyakan Google analytics. Jika Anda ingin menganalisa traffic, silakan nyalakan tombol *Enable Google Analytics for this project*. Pada step berikutnya akan ditanyakan *Google Account* untuk dihubungkan dengan *Google Analytics*.

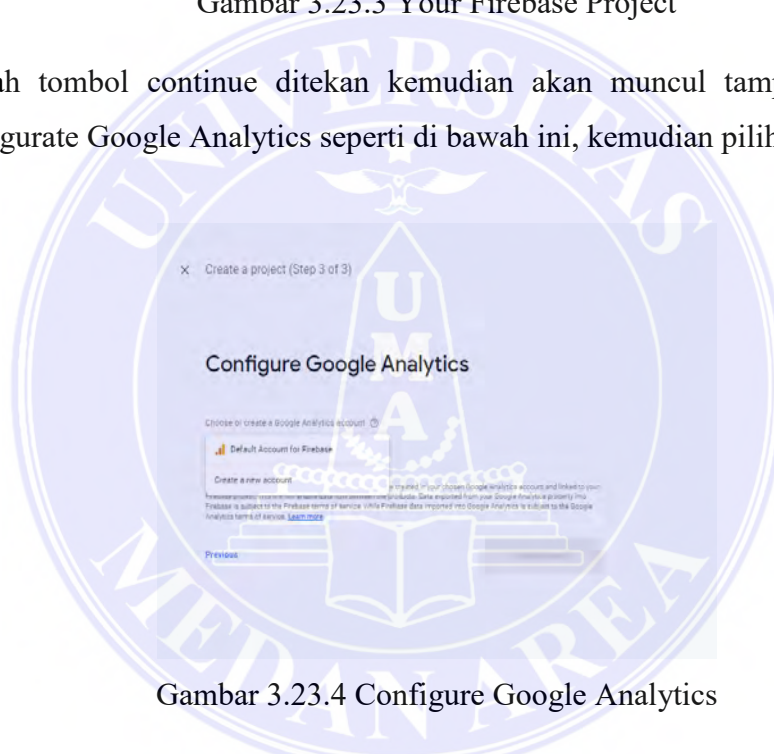


Pada panduan ini diaktifkan. Setelah itu tekan tombol Create Project.



Gambar 3.23.3 Your Firebase Project

Setelah tombol continue ditekan kemudian akan muncul tampilan jendela Configure Google Analytics seperti di bawah ini, kemudian pilih Default

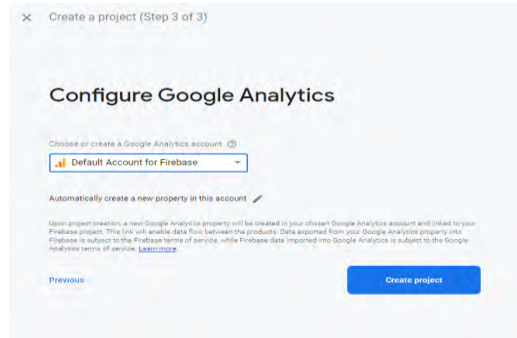


Gambar 3.23.4 Configure Google Analytics

Setelah tombol create project ditekan, silakan tunggu hingga project selesai dibuat. Setelah selesai akan tampil screen seperti dibawah. Lalu tekan tombol Continue

Berikutnya anda akan mendapatkan halaman seperti terlampir. Klik Build, lalu, pada saat itu, klik Realtime Database.





Gambar 3.23.5 Create Project Default Account For Firebase

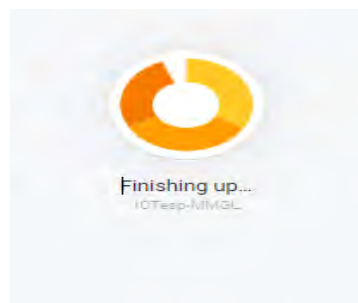
Klik Create Project



Gambar 3.23.6 Creating Your Project Firebase

Proses pembuatan project

Setelah tombol create project ditekan, silakan tunggu hingga project selesai dibuat. Setelah selesai akan tampil screen seperti dibawah. Lalu tekan tombol Continue.



Gambar 3.23.7 Finishing up

Pembuatan Project Telah Selesai

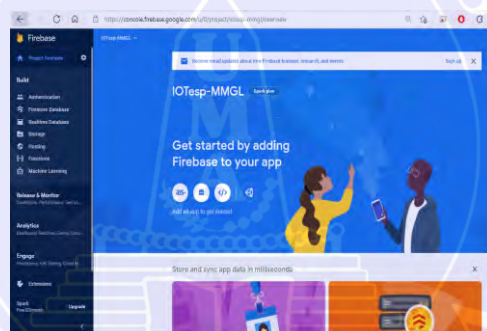
Kemudian Klik Continue



Gambar 3.23.8 Continue

Kemudian muncul jendela di bawah ini.

Halaman dibawah akan ditampilkan, pilih menu Realtime Database pada menu navigasi disebelah kiri.

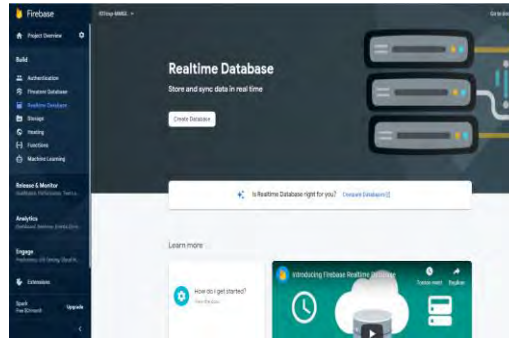


Gambar 3.23.9 Finising Create Firebase

Klik Realtime Database

Halaman Realtime Database akan tampil seperti dibawah, lalu tekan tombol

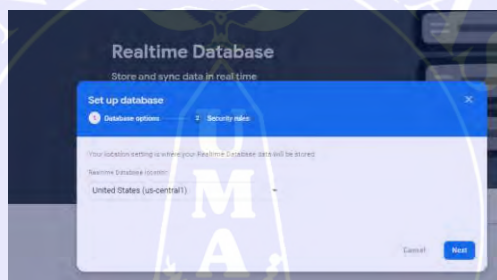
Create Database.



Gambar 3.23.10 Tampilan Realtime Database

Kemudian klik create database

Desain selanjutnya Google Analytics selesai. Kemudian, pada saat itu, ubah negara tempat database realtime yang ingin digunakan.

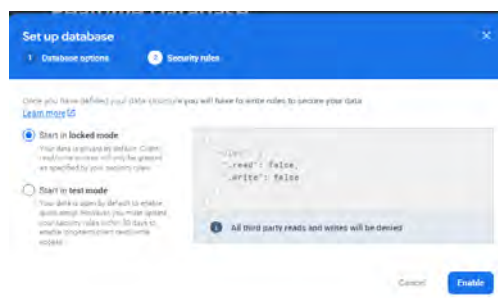


Gambar 3.23.11 Select Location Database

Kemudian pilih realtime Database Location “United State”, kemudian klik Next

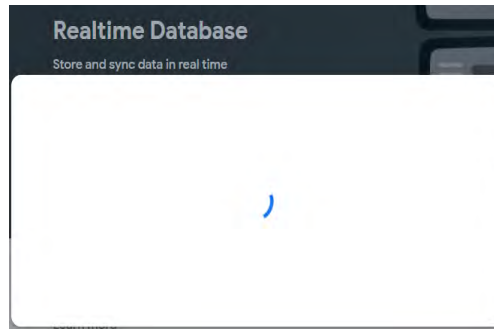
Akan ditampilkan panel popup, pilih radio button *Start in locked mode*. Lalu tekan tombol Enable.

Pilih locked mode, kemudian klik “Enable”



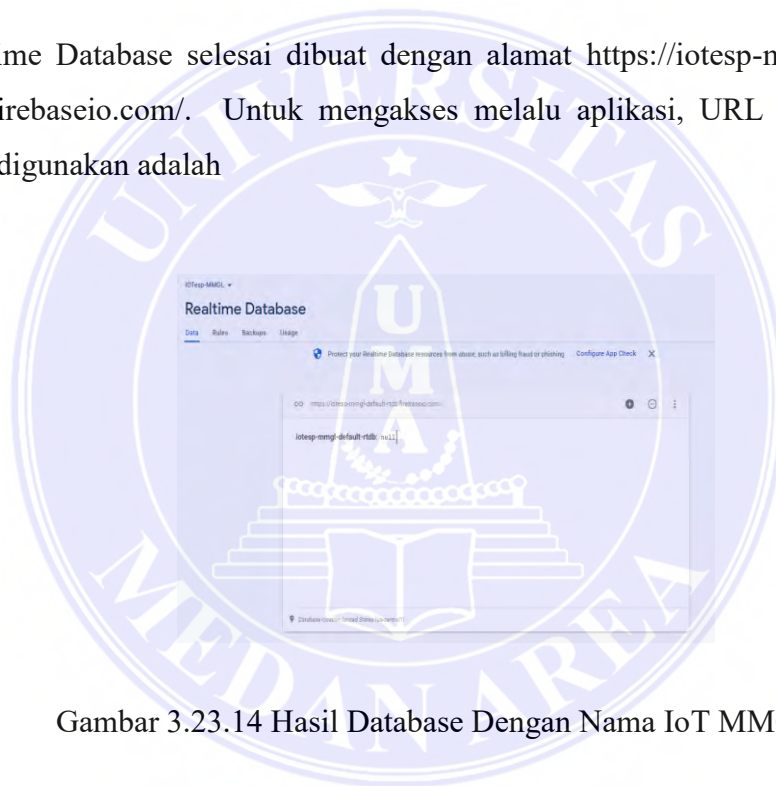
Gambar 3.23.12 Locked Mode Database

### Proses Membuat Realtime Database



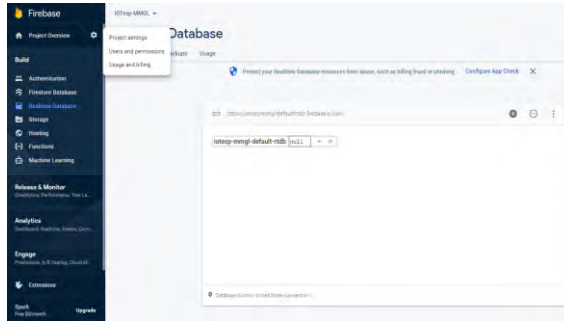
Gambar 3.23.13 Proses Pembuatan Database

Realtime Database selesai dibuat dengan alamat <https://iotesp-mmgl-default-rtbd.firebaseio.com/>. Untuk mengakses melalui aplikasi, URL pada tutorial yang digunakan adalah



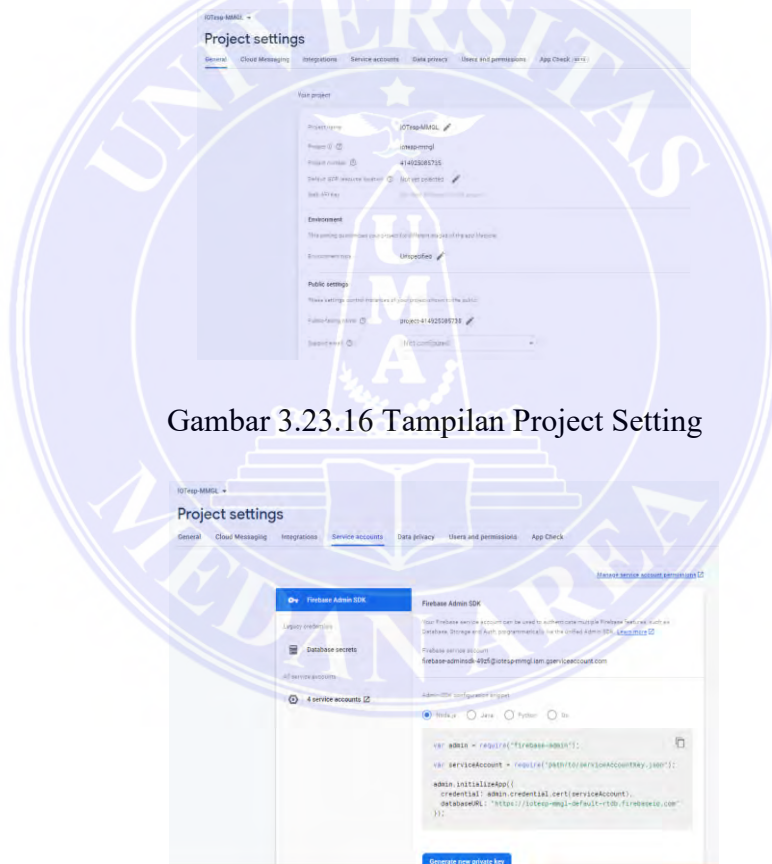
Gambar 3.23.14 Hasil Database Dengan Nama IoT MMGL

Kemudian copy link URL pada firebase host <https://iotesp-mmgl-default-rtbd.firebaseio.com/>  
Kemudian klik project setting, seperti pada gambar di bawah ini



Gambar 3.23.15 Project Setting Firebase

Tampilan project settings firebase database dengan nama IoTesp-MMGL

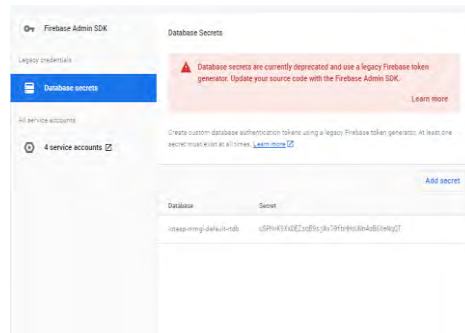


Gambar 3.23.16 Tampilan Project Setting

Gambar 3.23.17 Service Account Firebase



## Klik Database Secrets



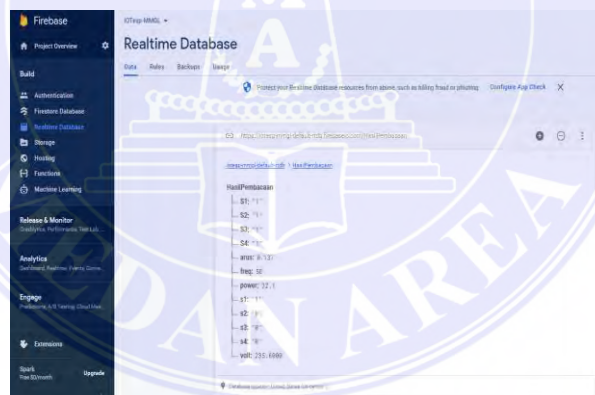
Gambar 3.23.18 Database Secrets

Copy token ke program Arduino nya seperti di bawah ini

```
#define FIREBASE_HOST "https://iotesp-mmgl-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "u5PHvK9XxDEZsqB9sjWx70fbHHoUNn4oB6XeNgQT"
```

Gambar 3.23.19 Token Program

Hasil Pengujian Firebase RealTime Database



Gambar 3.23.20 Hasil Pengujian Database

### 3.24. Pembuatan Aplikasi Smartphone Monitoring RealTime

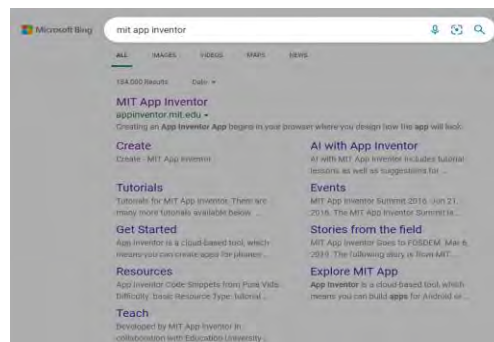
App Inventor 2 adalah *Integrated Development Environment* (IDE) atau Tools yang dapat diakses oleh semua individu untuk mengembangkan aplikasi Android tanpa perlu pengalaman di bidang pemrograman. Perancang Aplikasi awalnya dikembangkan dan diawasi oleh Google namun saat ini diawasi oleh Massachusetts Foundation of Innovation (MIT), sebuah Organisasi/Perguruan Tinggi yang berpartisipasi dalam Inovasi dan telah dikenal di seluruh dunia.

App Inventor sendiri dibuat pada tahun 2010 dan pada tanggal 15 Desember 2010. App Inventor disampaikan kepada masyarakat pada umumnya dan untuk kelompok pengembangan Application Creator yang dimotori oleh Imprint Friedman dari Google dan Prof. Hal Abelson dari MIT serta berbagai pemberi dari Google, termasuk Sharon Perl, Liz Looney dan Ellen Spertus. Aplikasi Innovator 2 dimaksudkan untuk menjadi simpel dan mudah dimengerti. Pada App Inventor kita tidak perlu bersentuhan dengan dunia pemrograman komputer yang membingungkan. Kita tidak perlu membuat kode pemrograman untuk membuat aplikasi.

Yang baru dari Application Designer 2 adalah Application Creator 2 berdasarkan Visual Block Programming yang dapat mengubah pengkodean bahasa pemrograman menjadi struktur visual sebagai kode program. Pengkodean program diselesaikan dengan Drag-Drop blok kode program ke dalam Block Manager yang kemudian, pada saat itu, mengatur blok seperti kita membuat teka-teki seperti yang ditunjukkan oleh aliran program yang kita butuhkan. Langkah - Langkah Membuat Aplikasi power monitoring Menggunakan App Inventor 2.

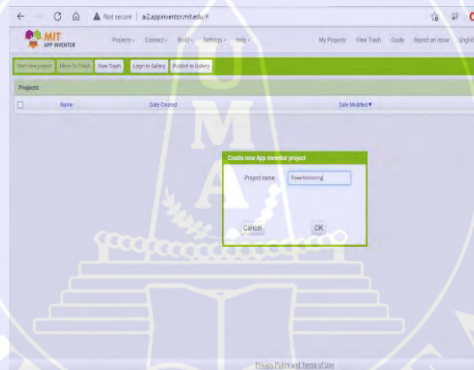
Sebelum memasuki langkah-langkah membuat aplikasi sederhana yang pada contoh kali ini akan membuat aplikasi “Rancang bangun prototipe otomatisasi proteksi dan monitoring listrik rumah tangga dengan IoT ESP”, hal pertama yang dilakukan adalah menginstall appinventor online yang tersedia di <http://appinventor.mit.edu/explore/ai2/windows.html> dan setelah melakukan install langsung saja kita masuk ke tutorial membuat aplikasi “proteksi dan monitoring listrik rumah tangga” sebagai berikut:

Buka App Inventor <http://appinventor.mit.edu/> kemudian pilih menu “Creats Apps!”



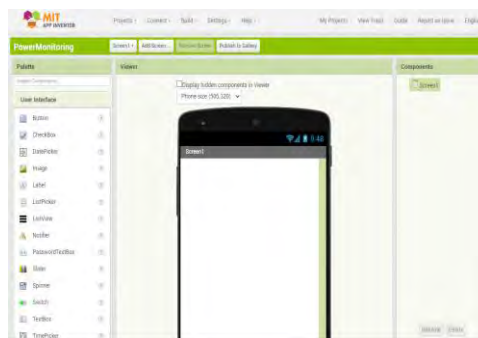
Gambar 3.24 MIT App Inventor

MIT App Inventor | Explore MIT App Inventor  
Kemudian klik MIT App Inventor dengan alamat link MIT App Inventor  
Setelah terbuka, klik Start new Project, seperti pada gambar di bawah ini



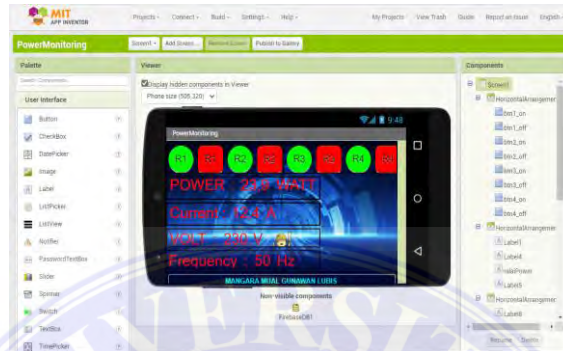
Gambar 3.24.1 Create New App

Beri nama “Powermonitoring”, kemudian klik OK  
Kemudian akan terbuka jendela seperti pada gambar di bawah ini



Gambar 3.24.2 Tampilan Design App

Setelah mengklik "Baiklah" tampilan akan berubah dan kemudian Pada layar1 pilih Ukuran Telepon dan dapat mengubah properti di layar 1 di kiri dan kanan layar dengan menambahkan desain, tombol, tanda, mengubah fondasi dan lain-lain sebagai sesuai keinginan anda, seperti model terlampir.



Gambar 3.24.3 Hasil Design App

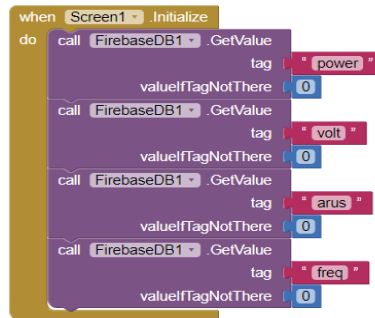
Setelah selesai design tampilan aplikasi seperti pada gambar di atas, selanjutnya beralih ke bagian blocks pada kiri atas layar untuk memberikan inputan tindakan pada aplikasi seperti tampilan berikut



Gambar 3.24.4 Block App

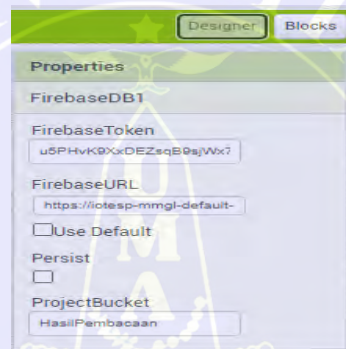
Pada blocks screen1 pilih call firebase DB1 getValue, perintah ini berguna untuk mengambil data dan mengisi nilai inisialisasi ke firebase, kemudian mengisinya pada label dengan nama masing-masing “power”, “volt”, “arus”, dan “freq”.





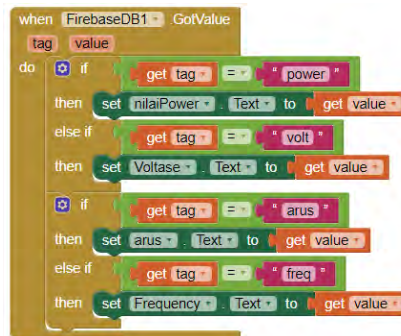
Gambar 3.24.5 Block Tabel Data

Untuk mengisi kode autentikasi dan url pada program APP inventor 2 ini yaitu dengan memasukkan nilai pada bagian seperti pada gambar di bawah ini.



Gambar 3.24.6 Kode Autentikasi

Kemudian gunakan perintah GotValue pada blok program FirebaseDB untuk mengambil data power, volt, arus dan freq kemudian mengisinya ke masing masing label nilaiPower, Voltase, arus, Frequency, seperti pada gambar di bawah ini



Gambar 3.24.7 Blok Program Firebase DB1



Blok program ini bawah ini berguna untuk mengambil nilai (get value) power pada database Firebase kemudian menyalin datanya pada label berjudul “nilaiPower”



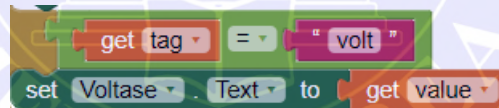
Gambar 3.24.8 Blok Program Nilai Power

Nilai power akan muncul pada tampilan App di bawah ini



Gambar 3.24.9 Tampilan App Power

Blok program ini bawah ini berguna untuk mengambil nilai (get value) volt pada database Firebase kemudian menyalin datanya pada label berjudul “Voltase”



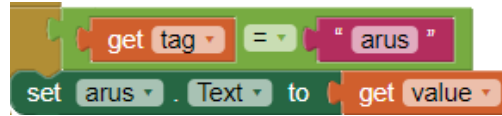
Gambar 3.24.10 Blok Program Voltase

Nilai voltase akan muncul pada tampilan App di bawah ini



Gambar 3.24.11 Tampilan App Voltase

Blok program di bawah ini berguna untuk mengambil nilai (get value) arus pada database Firebase kemudian menyalin datanya pada label berjudul “arus”



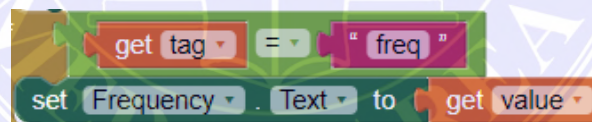
Gambar 3.24.12 Blok Program Arus

Nilai Arus akan muncul pada tampilan App di bawah ini



Gambar 3.24.13 Tampilan App Arus

Blok program ini bawah ini berguna untuk mengambil nilai (get value) freq pada database Firebase kemudian menyalin datanya pada label berjudul “Frequency”



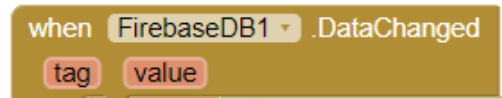
Gambar 3.24.14 Blok Program Frequency

Nilai Frequency listrik akan muncul pada tampilan App di bawah ini



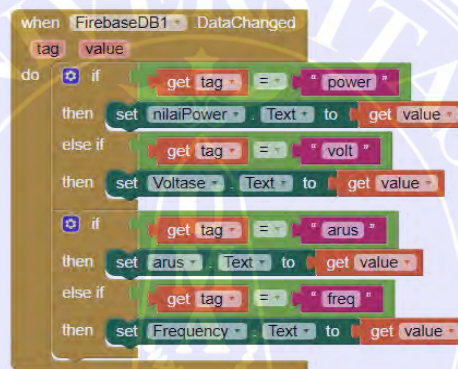
Gambar 3.24.15 Tampilan App Frequency

Karena nilai pada database Firebase realtime selalu berubah-ubah setiap detik maka perlu menambahkan kode program firebaseDB1 onDataChange, berguna untuk membaca nilai pembacaan sensor terbaru.



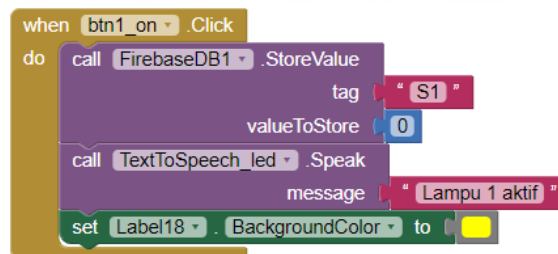
Gambar 3.24.16 FirebaseDB1 Data Change

Kode program lengkap untuk membaca perintah data terbaru dan mengisi nya ke label masing-masing nilai seperti pada gambar di bawah ini



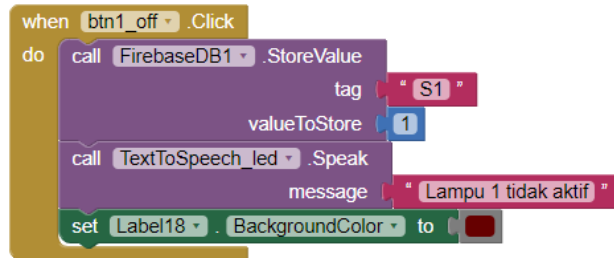
Gambar 3.24.17 Blok Program Data Change

Kemudian program blok untuk memberikan perintah menyalakan lampu / relay 1 IoT yaitu menggunakan perintah button1\_click kemudian isi nilai 0 ke firebase pada tag "S1", seperti pada gambar di bawah ini



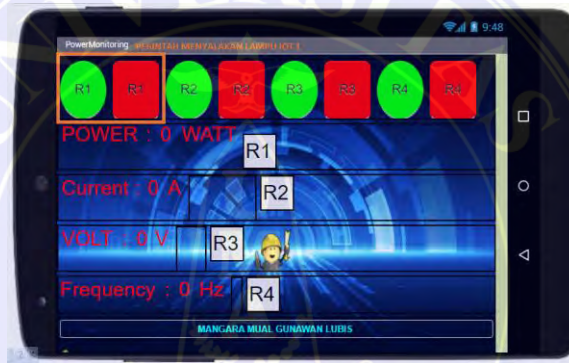
Gambar 3.24.18 Blok Program Relay ON

Kemudian program blok untuk memberikan perintah mematikan lampu / relay 1 IoT yaitu menggunakan perintah `button1_click` kemudian isi nilai 1 ke firebase pada tag "S1", seperti pada gambar di bawah



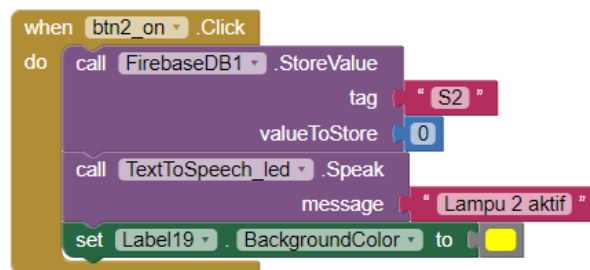
Gambar 3.24.19 Blok Program Relay OFF

Pada gambar di bawah ini merupakan perintah menyalakan lampu IoT 1



Gambar 3.24.20 Tampilan App Menyalakan Lampu IoT

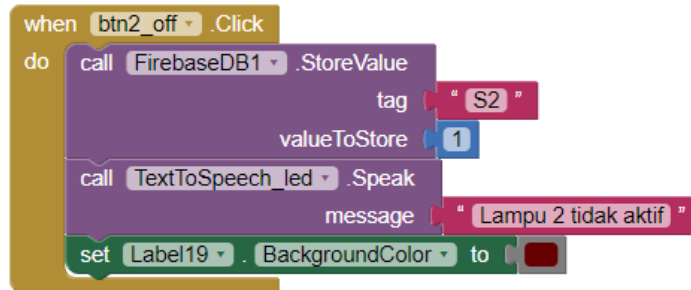
Kemudian program blok untuk memberikan perintah menyalakan lampu / relay 2 IoT yaitu menggunakan perintah `button2_click` kemudian isi nilai 0 ke firebase pada tag "S2", seperti pada gambar di bawah.



Gambar 3.24.21 Blok Program Menyalakan Lampu Relay 2 IoT



Kemudian program blok untuk memberikan perintah mematikan lampu / relay 2 IoT yaitu menggunakan perintah `button2_click` kemudian isi nilai 1 ke firebase pada tag "S2", seperti pada gambar di bawah



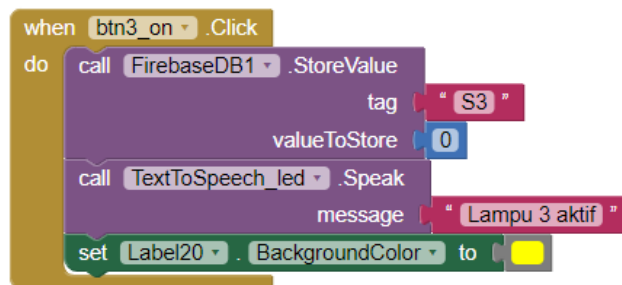
Gambar 3.24.22 Blok Program Mematikan Lampu Relay 2 IoT

Pada gambar di bawah ini merupakan perintah menyalakan lampu IoT 2



Gambar 3.24.23 Tampilan App Menyalakan Lampu IoT 2

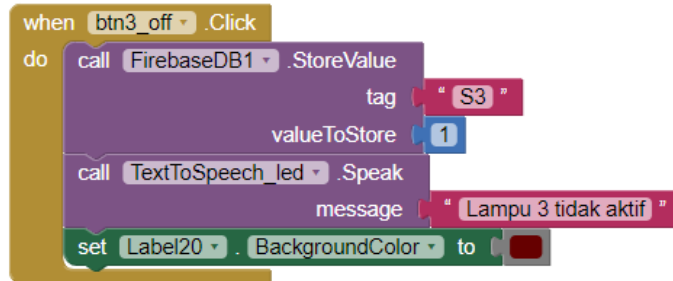
Kemudian program blok untuk memberikan perintah menyalakan lampu / relay 3 IoT yaitu menggunakan perintah `button3_click` kemudian isi nilai 0 ke firebase pada tag "S3", seperti pada gambar di bawah



Gambar 3.24.24 Blok Program Menyalakan Lampu Relay 3 IoT



Kemudian program blok untuk memberikan perintah mematikan lampu / relay 3 IoT yaitu menggunakan perintah `button3_click` kemudian isi nilai 1 ke firebase pada tag "S3", seperti pada gambar di bawah



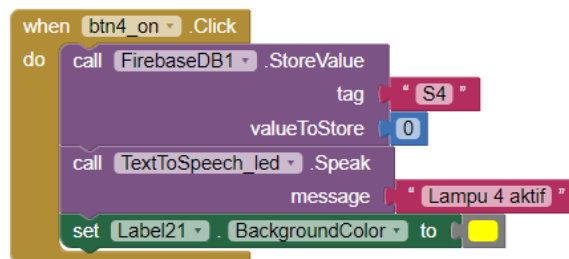
Gambar 3.24.25 Blok Program Mematikan Lampu Relay 3 IoT

Pada gambar di bawah ini merupakan perintah menyalakan lampu IoT 3



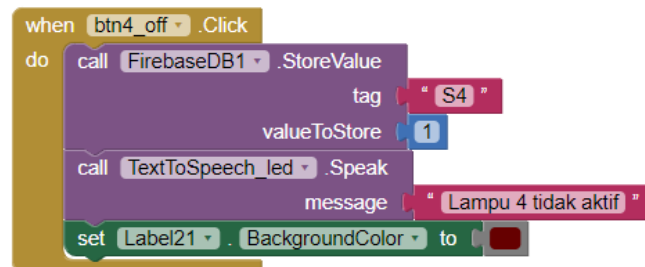
Gambar 3.24.26 Tampilan App Menyalakan Lampu IoT 3

Kemudian program blok untuk memberikan perintah menyalakan lampu / relay 4 IoT yaitu menggunakan perintah `button4_click` kemudian isi nilai 0 ke firebase pada tag "S4", seperti pada gambar di bawah



Gambar 3.24.27 Blok Program Menyalakan Lampu Relay 4 IoT

Kemudian program blok untuk memberikan perintah mematikan lampu / relay 4 IoT yaitu menggunakan perintah button4\_click kemudian isi nilai 1 ke firebase pada tag "S4", seperti pada gambar di bawah



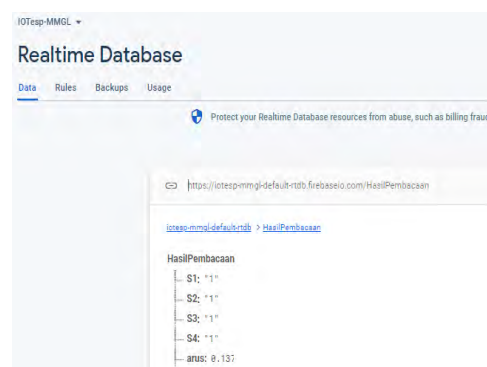
Gambar 3.24.28 Blok Program Mematikan Lampu Relay 3 IoT

Pada gambar di bawah ini merupakan perintah menyalakan lampu IoT 4



Gambar 3.24.29 Tampilan App Menyalakan Lampu IoT 4

Nilai input yang masuk ke firebase saat button ditekan akan muncul dengan nama masing-masing tag nya , bernilai 1 atau 0 sesuai button mana yang ditekan, seperti pada gambar di bawah ini



Gambar 3.24.30 Input Firebase

Sedangkan program mikrokontroller ESP untuk membaca nilai S1 menggunakan perintah `getString`, kemudian disimpan pada variable `val1` jika nilai string / karakter bernilai 0 maka relay akan aktif, dan jika nilai pada `val1` bernilai 1 maka relay off, kondisi modul relay yaitu active low seperti pada gambar di bawah ini,

```
val1 = Firebase.getString(firebaseData, "/HasilPembacaan/S1");
val1 = firebaseData.stringData();
if(val1 == "1"){ digitalWrite(relay1,HIGH); }
else if (val1 == "0"){ digitalWrite(relay1,LOW); }
```

Gambar 3.24.31 Program ESP Membaca Nilai S1

Sedangkan untuk membaca nilai S2 menggunakan perintah `getString`, kemudian disimpan pada variable `val2` jika nilai string / karakter bernilai 0 maka relay 2 akan aktif, dan jika nilai pada `val2` bernilai 1 maka relay 2 off, kondisi modul relay2 yaitu active low seperti pada gambar di bawah ini,

```
val2 = Firebase.getString(firebaseData, "/HasilPembacaan/S2");
val2 = firebaseData.stringData();
if(val2 == "1"){ digitalWrite(relay2,HIGH); }
else if (val2 == "0"){ digitalWrite(relay2,LOW); }
```

Gambar 3.24.32 Program ESP Membaca Nilai S2

untuk membaca nilai S3 menggunakan perintah `getString`, kemudian disimpan pada variable `val3` jika nilai string / karakter bernilai 0 maka relay 3 akan aktif, dan jika nilai pada `val3` bernilai 1 maka relay 4 off

```
val3 = Firebase.getString(firebaseData, "/HasilPembacaan/S3");
val3 = firebaseData.stringData();
if(val3 == "1"){ digitalWrite(relay3,HIGH); }
else if (val3 == "0"){ digitalWrite(relay3,LOW); }
```

Gambar 3.24.33 Program ESP Membaca Nilai S3

untuk membaca nilai S4 menggunakan perintah `getString`, kemudian disimpan pada variable `val4` jika nilai string / karakter bernilai 0 maka relay 4 akan aktif, dan jika nilai pada `val4` bernilai 1 maka relay 4 off

```
val4 = Firebase.getString(firebaseData, "/HasilPembacaan/S4");
val4 = firebaseData.stringData();
if(val4 == "1"){ digitalWrite(relay4,HIGH); }
else if (val4 == "0"){ digitalWrite(relay4,LOW); }
```

Gambar 3.24.34 Program ESP Membaca Nilai S4

### 3.25. Penempatan Alat Dalam Rumah Tangga

Sistem alat yang dirancang bertujuan untuk menambahkan sebuah sistem proteksi yang ada di rumah tangga, bukan untuk menggantikan fungsi MCB. Alat dipasang dengan menggunakan sebuah box panel yang dihubungkan dengan sumber tegangan listrik PLN 220 Volt. Kabel positif dari PLN dihubungkan dalam koil sensor PZEM-004T agar koil dapat membaca daya, tegangan, dan arus frequency. Output dari relay IoT dan relay proteksinya dapat disesuaikan panjang kabelnya dengan kebutuhan rumah.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian alat otomatisasi proteksi dan monitoring listrik rumah tangga dengan menggunakan IoT Node MCU ESP8266 dan menggunakan sensor PZEM-004T untuk membaca nilai besaran tegangan, arus, daya aktif dan energi yang dipakai, dilengkapi dengan modul relay sebagai aktuator. Alat ini dapat digunakan dengan baik pembacaan sensor mampu bekerja dengan tingkat keberhasilan pengukuran tegangan 99,6%, rata-rata eror arus 97,5% dan eror daya adalah 97,8%. Relay proteksi mampu melakukan proteksinya saat tegangan berada diatas 240 VAC dan dibawah 210 VAC, serta dapat berguna untuk pembelajaran di jurusan teknik elektro Universitas Medan Area.

Alat ini dapat dipakai untuk kendali lampu, kipas angin, blender, setrika, dan beban motor listrik dengan arus yang rendah. Terhubungnya alat ini dengan system IoT memungkinkan alat ini dapat dikendalikan dari jarak yang jauh letaknya, dengan terhubung jaringan internet. Pengujian kinerja dilakukan dengan cara menyesuaikan input dan output alat. Alat ini terdapat 4 relay yang dapat dikendalikan secara IoT dengan Smartphone, 2 relay otomatis yang dikendalikan oleh sensor DHT untuk menyalakan kipas dan pendingin udara jika suhu yang terukur melebihi batas yang ditentukan, dan terdapat 6 relay proteksi untuk beban listrik rumah tangga lainnya.

#### **5.2. Saran**

Ada beberapa kekurangan pada alat ini yang perlu diperbaiki seperti :

1. Penambahan Emi-filter untuk mengurangi gangguan listrik secara fruktatif, seperti kejutan tegangan listrik.
2. Penambahan stabilisator / STAVOL / AVR, karena tidak semua beban listrik rumah bisa mudah di ON-OFF kan alat ini, karena jika terlalu sering ON-OFF justru memperbesar biaya listrik, seperti kulkas dan pompa air yang sedang beroperasi



## DAFTAR PUSTAKA

- Anwar, S., Artono, T., Nasrul, N., Dasrul, D., & Fadli, A. (2019). Pengukuran Energi Listrik Berbasis PZEM-004T. In *Prosiding Seminar Nasional Politeknik Negeri Lhokseumawe* (Vol. 3, No. 1, p. 272).
- Sanjaya, H., Triyanto, J., Andri, R., Yani, F., Sanjaya, P. P., & Daulay, N. K. (2021, October). Kipas Angin Otomatis Menggunakan Sensor Suhu DHT11. In *Seminar Nasional Sains dan Teknologi Informasi (SENSASI)* (Vol. 3, No. 1, pp. 187-191).
- Widiana, I. W. Y., Agung, I. G. A. P. R., & Rahardjo, P. (2019). Rancang bangun kendali otomatis lampu dan pendingin ruangan pada ruang perkuliahan berbasis mikrokontroler arduino nano. *Jurnal SPEKTRUM*, 6(2), 112-120.
- Kasim, M. A. A., Ruslan, R., & Zain, S. G. PENGEMBANGAN SMART FITTING BERBASIS IOT (INTERNET OF THINGS) DENGAN MENGGUNAKAN MIKROKONTROLER ESP 32 S. *Jurnal Media Elektrik*, 19(2), 61-71.
- Akbar, R. (2018). Rancang Bangun Alat Monitoring Tegangan, Arus, Daya, Kwh, Serta Estimasi Biaya Pemakaian Peralatan Listrik Pada Rumah Tangga.
- Kurniawan, E., & Estanto, E. (2017). SISTEM PROTEKSI DAN MONITOR TEGANGAN TERHADAP BEBAN KELISTRIKAN RUMAH TANGGA. *Jurnal Elektra*, 2(2), 1-9.

- Mario, M., Lapanporo, B. P., & Muliadi, M. (2018). Rancang Bangun Sistem Proteksi dan Monitoring Penggunaan Daya Listrik Pada Beban Skala Rumah Tangga Berbasis Mikrokontroler ATmega328P. *Prisma Fisika*, 6(1), 26-33.
- Umam, B. A., & Efenie, Y. (2019). Sistem Rumah Cerdas Berbasis Internet of Things. *Energy-Jurnal Ilmiah Ilmu-Ilmu Teknik*, 9(2).
- Amaro, N. (2017). Sistem monitoring besaran listrik dengan teknologi IoT (Internet of Things). *Fakultas Teknik Universitas Lampung. Bandar Lampung*.
- Hidayat, A., Sofyan, S., & Thaha, S. (2021). Rancang Bangun Sistem Proteksi Undervoltage dan Overvoltage pada Instalasi Bangunan Sederhana Berbasis Internet of Things. *Prosiding SNST Fakultas Teknik*, 1(1).
- Hilmansyah, H., Utomo, R. M., Saputra, A. W., & Alif, R. F. (2020). RANCANG BANGUN WIRELESS BATTERY MONITORING SYSTEM BERBASIS ESP32. *PROSIDING SNITT POLTEKBA*, 4, 194-199.
- Alipudin, A. M. (2018). Rancang bangun alat monitoring biaya listrik terpakai berbasis internet of things (IOT). *Jurnal Online Mahasiswa (JOM) Bidang Teknik Elektro*, 1(1).
- Handarly, D., & Lianda, J. (2018). Sistem Monitoring Daya Listrik Berbasis IoT (Internet of Thing). *J. Electr. Electron. Control Automot. Eng*, 3(2), 205-208.
- Salamah, I. (2020). Implementasi Smart Home Menggunakan Raspberry Pi Berbasis Android. *Jurnal Teknik Elektro dan Komputer*, 9(2), 109-116.

- 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.
- Azis, A., & Febrianti, I. K. (2019). Analisis Sistem Proteksi Arus Lebih pada Penyulang Cendana Gardu Induk Bungaran Palembang. *Jurnal Ampere*, 4(2), 332-344.
- Ivionita, Vella, Pamila Mei Ade. 2019. "Bab II Landasan Teori." *Journal of Chemical Information and Modeling* 53(9): 1689–99.
- Pangestu, A. D., Ardianto, F., & Alfaresi, B. (2019). Sistem Monitoring Beban Listrik Berbasis Arduino Nodemcu Esp8266. *Jurnal Ampere*, 4(1), 187-197.
- Ma'ruf, A., Purnama, R., & Susilo, K. E. (2021). Rancang Bangun Alat Monitoring Tegangan, Arus, Daya, dan Faktor Daya Berbasis IoT. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, 5(1), 81-86.

## LAMPIRAN

Program Arduino Uno :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,20,4);
#define pinBuz 3
#define pb1 A0
#define pb2 A1
#define pb3 A2

boolean PB1_press = LOW;
boolean PB1_release = LOW;
boolean PB2_press = LOW;
boolean PB2_release = LOW;
boolean PB3_press = LOW;
boolean PB3_release = LOW;
unsigned long previousMillis = 0;
const long interval = 300;
unsigned long previousMillis1 = 0;
const long interval1 = 10000;

#define relay1 5
#define relay2 6
#define relay3 7
#define relay4 8
#define relay5 9
```

```
#define relay6 10
#define relay7 11
#define relay8 12

void setup() {
  Serial.begin(115200);
  dht.begin();
  lcd.init();
  lcd.backlight();
  pinMode(pb1,INPUT_PULLUP);
  pinMode(pb2,INPUT_PULLUP);
  pinMode(pb3,INPUT_PULLUP);
  pinMode(pinBuz,OUTPUT);
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(relay3,OUTPUT);
  pinMode(relay4,OUTPUT);
  pinMode(relay5,OUTPUT);
  pinMode(relay6,OUTPUT);
  pinMode(relay7,OUTPUT);
  pinMode(relay8,OUTPUT);
  offAllRelay();
}
int r,y;
int i=1;
int p=1;
float arus,tega;
float ar,teg;
float curr;
float volt;
char datamasuk;
float h,t;
```



```
String datamasukan;
String data1,data2,data3,data4;
float dataVolt,dataCur,dataPow,dataFreq;
void loop() {
  PB1_press = digitalRead(pb1);
  PB2_press = digitalRead(pb2);
  PB3_press = digitalRead(pb3);
  tekanPB();
  if(Serial.available()){
    datamasukan = Serial.readStringUntil('\n');
    data1 = datamasukan.substring(0,4); //volt
    data2 = datamasukan.substring(6,10); // arus
    data3 = datamasukan.substring(10,16); // power
    data4 = datamasukan.substring(16,20); // freq

    Serial.print("data 1 : " + data1); Serial.print(" ");
    Serial.print("data 2 : " + data2); Serial.print(" ");
    Serial.print("data 3 : " + data3); Serial.print(" ");
    Serial.print("data 4 : " + data4); Serial.print("\n");
    dataVolt = data1.toFloat();
    dataCur = data2.toFloat();
    dataPow = data3.toFloat();
    dataFreq = data4.toFloat();
  }
  unsigned long currentMillis = millis();
  unsigned long currentMillis1 = millis();
  t = dht.readTemperature();
  h = dht.readHumidity();
  if(t >= 35.5){
    digitalWrite(relay1,LOW); // RELAY KHUSUS SUHU
  }
  else{
```

```

digitalWrite(relay1,HIGH);
if(h <= 52.5){      // RELAY KHUSUS UDARA KERING
  digitalWrite(relay2,LOW);
}
else{
  digitalWrite(relay2,HIGH);
}
if(currentMillis - previousMillis >= interval){
  previousMillis = currentMillis;
  r = analogRead(A6); // 278 - 1022 // batas Arus
  y = analogRead(A7); // 278 - 1022 // batas Tegangan
  arus = map(r,10,1021,0,100);
  tega = map(y,10,1021,2000,2390); // 239 VOLT MAX
  ar  = (float) arus / 10.0;
  teg  = (float) tega / 10.0;
  //=====
}
if( (dataVolt > teg) || (dataVolt <= 210) || (dataCur > ar ) ) {
  offAllRelay();
  tone(pinBuz, 200, 100);
}
else {
  if(currentMillis1 - previousMillis1 >= interval1){
    previousMillis1 = currentMillis1;
    onAllRelay();
  }
}
tampil_data();
}
void tampil_data(){
  if(i==1 ){
    lcd.setCursor(0,0); lcd.print("Volt : ");

```

```
lcd.setCursor(7,0); lcd.print(dataVolt,1);  
lcd.setCursor(13,0); lcd.print("V");  
lcd.setCursor(0,1); lcd.print("Curr : ");  
lcd.setCursor(7,1); lcd.print(dataCur,2);  
lcd.setCursor(13,1); lcd.print("A");  
}  
if(i==2){  
lcd.setCursor(0,0); lcd.print("Curr : ");  
lcd.setCursor(7,0); lcd.print(dataCur,2);  
lcd.setCursor(13,0); lcd.print("A");  
lcd.setCursor(0,1); lcd.print("pow : ");  
lcd.setCursor(7,1); lcd.print(dataPow,2);  
lcd.setCursor(13,1); lcd.print("W");  
}  
if(i==3){  
lcd.setCursor(0,0); lcd.print("pow : ");  
lcd.setCursor(7,0); lcd.print(dataPow,2);  
lcd.setCursor(13,0); lcd.print("W")  
lcd.setCursor(0,1); lcd.print("Freq : ");  
lcd.setCursor(7,1); lcd.print(dataFreq,1);  
lcd.setCursor(13,1); lcd.print("Hz");  
}  
if(i==4){  
lcd.setCursor(0,0); lcd.print("Freq : ");  
lcd.setCursor(7,0); lcd.print(dataFreq,1);  
lcd.setCursor(13,0); lcd.print("Hz");  
lcd.setCursor(0,1); lcd.print("Daya : ");  
lcd.setCursor(7,1); lcd.print(dataPow,2);  
lcd.setCursor(13,1); lcd.print("wh");  
}  
if(i==5){  
lcd.setCursor(0,0); lcd.print("Suhu : ");
```

```

lcd.setCursor(7,0); lcd.print(t,1); // digit 1 belakang koma
lcd.setCursor(13,0); lcd.print(" *C");
lcd.setCursor(0,1); lcd.print("Humd : ");
lcd.setCursor(7,1); lcd.print(h,1);
lcd.setCursor(13,1); lcd.print(" %");
}
if(i==6){
lcd.setCursor(0,0); lcd.print("Set Teg : ");
lcd.setCursor(9,0); lcd.print(teg,1); // digit 1 belakang koma

lcd.setCursor(0,1); lcd.print("Set Curr: ");
lcd.setCursor(9,1); lcd.print(ar,1);
lcd.setCursor(15,1); lcd.print("A");
}
}
void tekanPB(){
// =====
if(PB1_press != PB1_release){
if(PB1_press == LOW)
{
lcd.setCursor(0,0); lcd.print(" ");
lcd.setCursor(0,1); lcd.print(" ");
i++;
}
PB1_release = PB1_press;
}
if(PB2_press != PB2_release){
if(PB2_press == LOW)
{
lcd.setCursor(0,0); lcd.print(" ");
lcd.setCursor(0,1); lcd.print(" ");
i--;
}
}
}

```

```
    }
    PB2_release = PB2_press;
  }
  if(i<=1) i=1;
  if(i>=6) i=6;
  if(PB3_press != PB3_release){
    if(PB3_press == LOW)
      { p++; }
    PB3_release = PB3_press;
  }
  if(p==2) { lcd.backlight(); }
  if(p>=3){ lcd.noBacklight(); p=1;}
}
void offAllRelay(){
  //digitalWrite(relay1,HIGH);
  //digitalWrite(relay2,HIGH);
  digitalWrite(relay3,HIGH);
  digitalWrite(relay4,HIGH);
  digitalWrite(relay5,HIGH);
  digitalWrite(relay6,HIGH);
  digitalWrite(relay7,HIGH);
  digitalWrite(relay8,HIGH);
}
void onAllRelay(){
  //digitalWrite(relay1,LOW);
  //digitalWrite(relay2,LOW);
  digitalWrite(relay3,LOW);
  digitalWrite(relay4,LOW);
  digitalWrite(relay5,LOW);
  digitalWrite(relay6,LOW);
  digitalWrite(relay7,LOW);
  digitalWrite(relay8,LOW);
}
```



```
}
```

Program NodeMcu :

```
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
#if defined(ESP32)
  #error "Software Serial is not supported on the ESP32"
#endif
int ledState = D2;
unsigned long previousMillis3 = 0;
const long interval3 = 100;

#define FIREBASE_HOST "https://iot98-mmgl-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "GPMheLvkyzdOtuX9lCD1MsSsjmfzM7mVeQPxF4u"
#define WIFI_SSID "lubiS9898"
#define WIFI_PASSWORD "lubis9898"

FirebaseData firebaseData;
#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
#define PZEM_RX_PIN D7
#define PZEM_TX_PIN D8
#endif
SoftwareSerial pzemSWSerial(PZEM_RX_PIN, PZEM_TX_PIN);
PZEM004Tv30 pzem(pzemSWSerial);
float voltage, current, power, energy, frequency, pf;
unsigned long previousMillis = 0;
const long interval = 500;
#define relay1 12 //
String val1;
```

```
#define relay2 14 //
String val2;
#define relay3 4 //D1
String val3;
#define relay4 5 //D2
String val4;
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(relay3,OUTPUT);
  pinMode(relay4,OUTPUT);
  digitalWrite(relay1,HIGH);
  digitalWrite(relay2,HIGH);
  digitalWrite(relay3,HIGH);
  digitalWrite(relay4,HIGH);
  Serial.begin(115200);
  // Koneksi ke Wifi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  //Serial.print("connecting");
  while(WiFi.status() != WL_CONNECTED) {
    //Serial.print("connecting Failed");
    delay(500);
  }
  Serial.println();
  //Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.setInt(firebaseData,"/S1",0);
  Firebase.setInt(firebaseData,"/S2",0);
  Firebase.setInt(firebaseData,"/S3",0);
}
```

```

    Firebase.setInt(firebaseData,"/S4",0);
}
void loop(){
    unsigned long currentMillis3 = millis();
    if (currentMillis3 - previousMillis3 >= interval3) {
        previousMillis3 = currentMillis3;
        if (ledState == LOW) {
            ledState = HIGH; // Note that this switches the LED *off*
        } else {
            ledState = LOW; // Note that this switches the LED *on*
        }
        digitalWrite(LED_BUILTIN, ledState);
    }
    if(WiFi.status() != WL_CONNECTED){
        WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    }
    val1 = Firebase.getString(firebaseData,"/iot98-mmgl/S1");
    val1 = firebaseData.stringData();
    if(val1 == "1"){ digitalWrite(relay1,HIGH); }
    else if (val1 == "0"){ digitalWrite(relay1,LOW);}
    val2 = Firebase.getString(firebaseData,"/iot98-mmgl/S2");
    val2 = firebaseData.stringData();
    if(val2 == "1"){ digitalWrite(relay2,HIGH);}
    else if (val2 == "0"){ digitalWrite(relay2,LOW);}

    val3 = Firebase.getString(firebaseData,"/iot98-mmgl/S3");
    val3 = firebaseData.stringData();
    if(val3 == "1"){ digitalWrite(relay3,HIGH);}
    else if (val3 == "0"){ digitalWrite(relay3,LOW);}

    val4 = Firebase.getString(firebaseData,"/iot98-mmgl/S4");
    val4 = firebaseData.stringData();

```

```

if(val4 == "1"){ digitalWrite(relay4,HIGH);}
  else if (val4 == "0"){ digitalWrite(relay4,LOW);}
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;

  voltage = pzem.voltage();
  current = pzem.current();
  power = pzem.power();
  energy = pzem.energy();
  frequency = pzem.frequency();
  pf = pzem.pf();
  Serial.print(voltage,1);
  Serial.print(" ");
  Serial.print(current,2);
  Serial.print(" ");
  Serial.print(power,2);
  Serial.print(" ");
  Serial.print(frequency,1);
  Serial.println();
  if (Firebase.setFloat(firebaseData, "/iot98-mmgl/volt", voltage)){
    //Serial.println("volt terkirim");
  }else{
    //Serial.println("volt tidak terkirim");
    //Serial.println("Karena: " + firebaseData.errorReason());
  }
  if (Firebase.setFloat(firebaseData, "/iot98-mmgl/arus", current)){
    //Serial.println("arus terkirim"); Serial.println();
  } else{
    //Serial.println("arus tidak terkirim");
    //Serial.println("Karena: " + firebaseData.errorReason());
  }
}

```

```
if (Firebase.setFloat(firebaseData, "/iot98-mmgl/power", power)){
    //Serial.println("power terkirim"); Serial.println();
} else{
    //Serial.println("power tidak terkirim");
    //Serial.println("Karena: " + firebaseData.errorReason());
}
if (Firebase.setFloat(firebaseData, "/iot98-mmgl/freq", frequency)){
    //Serial.println("freq terkirim"); Serial.println();
} else{
    //Serial.println("freq tidak terkirim");
    //Serial.println("Karena: " + firebaseData.errorReason());
}
// tampilData();
}
}
void tampilData(){
    Serial.print("Voltage: ");    Serial.print(voltage,1);    Serial.println("V");
    Serial.print("Current: ");    Serial.print(current);    Serial.println("A");
    Serial.print("Power: ");    Serial.print(power,1);    Serial.println("W");
    Serial.print("Energy: ");    Serial.print(energy,1);    Serial.println("kWh");
    Serial.print("Frequency: ");    Serial.print(frequency, 1); Serial.println("Hz");
    Serial.print("PF: ");    Serial.println(pf);
    Serial.println();
}
```