

**RANCANG BANGUN SISTEM KONTROL DAN *MONITORING* BEBAN
PADA PANEL PLTS MENGGUNAKAN ESP8266 BERBASIS IoT**



TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
pendidikan diploma 3 (D-3) Program Studi Teknik Listrik

Jurusan Teknik Elektro

Politeknik Negeri Ujung Pandang

ANDIKA JAYA SAPUTRA 32121031

M. AL KAUTSAR 32121039

MUH. FAUDZAN PANDANGAI 32121042

PROGRAM STUDI D-3 TEKNIK LISTRIK

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI UJUNG PANDANG

MAKASSAR

2024

HALAMAN PENGESAHAN

Laporan tugas akhir ini dengan judul “Rancang bangun sistem kontrol dan *monitoring* beban pada panel PLTS menggunakan ESP8266 berbasis IoT” oleh Andika Jaya Saputra NIM 321 21 031, M. Al Kautsar NIM 321 21 039, dan Muh. Faudzan Pandangai NIM 321 21 042 dinyatakan layak untuk diseminarkan.

Makassar, September 2024

Menyetujui,

Dosen Pengarah Utama



Bakhtiar, S. T., M. T.
NIP 197000323 199603

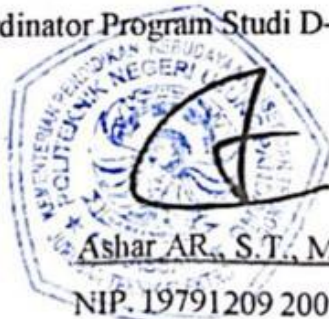
Dosen Pengarah Pendamping



Dr. Muhira Dza Faraby, S. T., M. T.
NIP 199002182022031005

Mengetahui,

Koordinator Program Studi D-3 Teknik Listrik



Ashar AR., S.T., M.T.
NIP. 19791209 200812 1 001

HALAMAN PENERIMAAN

Pada hari ini, tanggal September 2024 tim penguji ujian sidang tugas akhir telah menerima dengan baik hasil tugas akhir oleh :

ANDIKA JAYA SAPUTRA	321 210 31
M. AL KAUTSAR	321 21 039
MUH. FAUDZAN PANDANGAI	321 21 042

Judul Tugas Akhir

“Rancang bangun sistem kontrol dan *monitoring* beban pada panel PLTS menggunakan ESP8266 berbasis IoT”

Makassar, September 2024

Tim Penguji Laporan Tugas Akhir :

1) Ashar AR, S.T., M.T.	Ketua	()
2) Alamsyah Achmad, S.pd., M.T.	Sekretaris	()
3) Aksan, S.T., M.T.	Anggota 1	()
4) Desi Widyaningsih, S.ST., M.T.	Anggota 2	()
5) Bakhtiar, S.T., M.T.	Pengarah 1	()
6) Muhira Dza Faraby, S.T., M.T.	Pengarah 2	()

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT karena berkat rahmat dan hidayah-Nya penulisan tugas akhir ini dengan judul **“Rancang bangun sistem kontrol dan *monitoring* beban pada panel PLTS menggunakan ESP8266 berbasis IoT”** dapat diselesaikan dengan baik.

Dalam penulisan Tugas Akhir, penulis menyampaikan banyak terima kasih kepada :

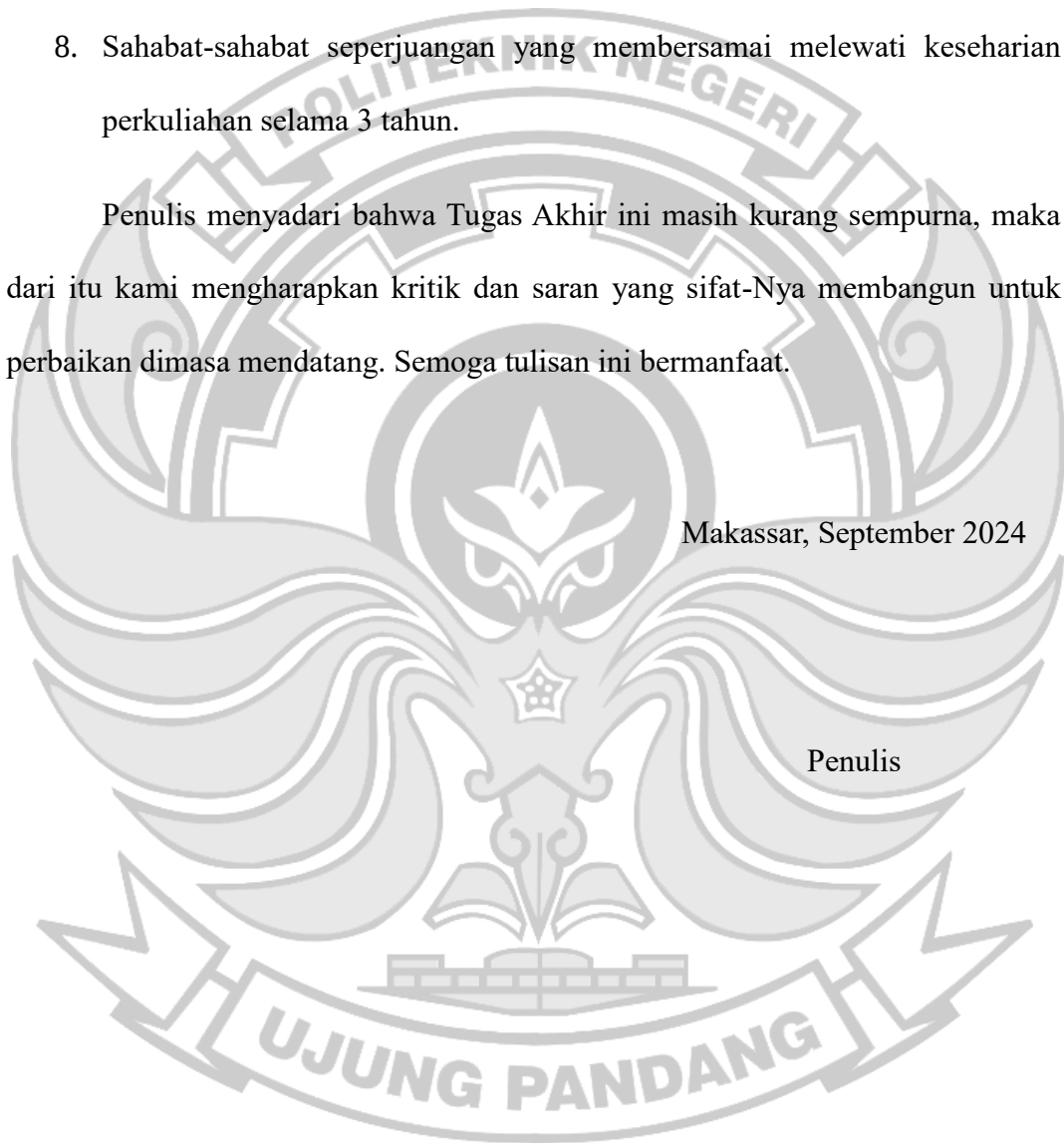
1. Keluarga tercinta yaitu kedua orang tua kami, yang telah memberikan banyak bantuan baik itu berupa dorongan moral maupun materi.
2. Bapak Ir. Ilyas Mansur, M.T. Selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Prof. Ahmad Rizal Sultan, S. T., M. T., Ph.D. Selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Ashar. AR., S.T., M.T. Selaku Ketua Program Studi D3 Teknik Listrik Politeknik Negeri Ujung Pandang.
5. Bapak Prof. Marwan, S. T., M.Eng., Sc, Ph.D. Selaku pembimbing Tata Tulis Laporan.
6. Bapak Bakhtiar, S. T., M. T. Dan Bapak Dr. Muhira Dza Farabi, S. T., M. T. Selaku pembimbing 1 dan pembimbing 2 yang telah meluangkan waktunya memberikan bimbingan dan pengarahan dalam menyelesaikan Tugas Akhir ini.

7. Seluruh dosen dan Staf Program Studi Teknik Listrik yang telah memberikan ilmunya kepada penulis selama melaksanakan perkuliahan, dan telah menyediakan fasilitas dan sarana dalam mengerjakan Tugas Akhir ini.
8. Sahabat-sahabat seperjuangan yang kebersamaan melewati keseharian perkuliahan selama 3 tahun.

Penulis menyadari bahwa Tugas Akhir ini masih kurang sempurna, maka dari itu kami mengharapkan kritik dan saran yang sifat-Nya membangun untuk perbaikan dimasa mendatang. Semoga tulisan ini bermanfaat.

Makassar, September 2024

Penulis



DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xiii
RINGKASAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup Penelitian	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 Study Literature	4
2.2 Pembangkit Listrik Tenaga Surya	6
2.2.1 Prinsip Kerja PLTS	7
2.2.2 Jenis-Jenis Solar Cell	9
2.2.3 Komponen-Komponen Utama PLTS	12
2.3 Potensi Energi Surya di Indonesia	19
2.4 Komponen Panel Monitoring dan Kontrol Beban PLTS	20

2.5	Alat dan Bahan Pelindung	26
2.6	Internet Of Things (IOT)	29
2.7	Blynk.....	30
2.8	Software Arduino IDE	30
2.9	Menentukan Kebutuhan Komponen	31
2.10	Pemrograman Bahasa C.....	32
BAB III METODOLOGI.....		61
3.1	Waktu dan Tempat	61
3.2	Alat dan Bahan.....	61
3.3	Tahapan Penelitian.....	63
3.3.1	Studi Literatur	64
3.3.2	Observasi Lapangan.....	64
3.3.3	Perancangan Desain Alat	65
3.3.4	Rancang Bangun Alat	69
3.3.5	Alat Bekerja	71
3.3.6	Pengambilan Data	72
BAB IV HASIL DAN PEMBAHASAN.....		74
4.1	Hasil	74
4.1.1	Menentukan Kebutuhan Komponen	74
4.1.2	Pengujian Solar Cell, Accu dan SCC.....	76
4.1.3	Hasil Perancangan Software	77
4.2	Pembahasan	86
BAB V PENUTUP		95

5.1 Kesimpulan	95
5.2 Saran.....	96
DAFTAR PUSTAKA	97
LAMPIRAN	100



DAFTAR GAMBAR

Gambar 2.1 Cara kerja fotovoltaiik.....	8
Gambar 2.2 <i>Gallium arsenide cell</i>	9
Gambar 2.3 <i>Cadmium Sulfide Cell</i>	10
Gambar 2.4 <i>Polycrystalline cell</i>	10
Gambar 2.5 <i>Amorphous Silikon Cell</i>	11
Gambar 2.6 <i>Copper indium diselenide cells</i>	11
Gambar 2.7 <i>Cadmium telluride cells</i>	12
Gambar 2.8 <i>Dye sensitized cells</i>	12
Gambar 2.9 Sistem PLTS	13
Gambar 2.10 MPPT	18
Gambar 2.11 Lampu LED.....	18
Gambar 2.12 Baterai/aki	19
Gambar 2.13 Box-X6.....	21
Gambar 2.14 ESP8266	22
Gambar 2.15 Pzem 017T.....	23
Gambar 2.16 Relai	25
Gambar 2.17 Kabel NYAF	26
Gambar 2.18 Mesin las	26
Gambar 2.19 Besi siku	27
Gambar 2.20 Multimeter digital.....	27
Gambar 2.21 Solder	28
Gambar 2.22 Kabel USB.....	29
Gambar 2.23 Struktur dasar dari pemrograman Arduino	34
Gambar 3.1 <i>Flowchart</i>	63
Gambar 3.2 Perancangan mekanik alat tampak depan & samping.....	65
Gambar 3.3 Perancangan <i>wiring diagram</i> sistem ESP8266	65
Gambar 3.4 Perancangan elektrik Alat.....	65
Gambar 3.5 Tampak atas & tampak bawah PCB	67

Gambar 3.6 Schematic Sistem monitoring & kontrol ESP8266.....	68
Gambar 3.7 Pemasangan panel box	69
Gambar 3.8 Pemasangan sumber PLTS ke SCC.....	70
Gambar 3.9 Pemasangan hubungan antara panel mikrokontroler, SCC, dan accu..	70
Gambar 3.10 Pemasangan beban dengan sitem monitoring & kontrol ESP8266....	71
Gambar 3.11 Skema Monitoring.....	71
Gambar 3.12 Metode <i>black box</i>	71
Gambar 4.1 Tampilan preferences	78
Gambar 4.2 tampilan menu Alat	79
Gambar 4.3 Tampilan manage library.....	79
Gambar 4.4 Notifikasi selesai kompilasi	80
Gambar 4.5 Tampilan menu Software Arduino IDE.....	80
Gambar 4.6 Tampilan Blynk IoT	81
Gambar 4.7 Tampilan Registrasi Blynk IoT	81
Gambar 4.8 Proses pembuatan akun Blynk IoT.....	82
Gambar 4.9 Tampilan beranda dan penamaan proyek	83
Gambar 4.10 Proses Setting Button	83
Gambar 4.11 Menginput kode program ke template button	84
Gambar 4.12 Membuat Template parameter Indikator	85
Gambar 4.13 Menginput kode program ke template Parameter Indikator Arus	86
Gambar 4.14 Menginput kode program ke template Parameter Indikator Tegangan	86
Gambar 4.15 Hasil pengukuran lampu 1	90
Gambar 4.16 Hasil pengukuran lampu 2	91
Gambar 4.17 Hasil pengukuran lampu 3	91

DAFTAR TABEL

Tabel 2.1 Keuntungan PWM dan MPPT	16
Tabel 2.2 Kekurangan PWM dan MPPT	17
Tabel 2.3 Potensi EBT di Indonesia Tahun 2018	20
Tabel 2.4 Jenis Type Data Pada Arduino Yang Sering digunakan	38
Tabel 2.5 Operator Yang Sering digunakan	53
Tabel 2.6 Operator Perbandingan Yang Sering digunakan	54
Tabel 2.7 Operator Boolean Yang Sering digunakan	55
Tabel 2.8 Macam-Macam Operator Bitwise yang umum digunakan	55
Tabel 2.9 Beberapa contoh berita gabungan	56
Tabel 2.10 Jumlah Pin Interrupt pada beberapa jenis arduino berbeda.....	58
Tabel 3.1 Alat yang digunakan dalam perancangan	61
Tabel 3.2 Bahan yang digunakan dalam perancangan	62
Tabel 3.3 Spesifikasi Lampu Sorot DC.....	64
Tabel 4.1 Pengujian Solar Cell, Accu, SCC.....	77
Tabel 4.2 <i>Blackbox testing</i> pada <i>Ethernet Shield</i> ESP8266.....	86
Tabel 4.3 <i>Blackbox testing</i> menu kontrol.....	87
Tabel 4.4 <i>Blackbox testing</i> menu <i>monitoring</i> pada <i>interface blynk</i>	89
Tabel 4.5 <i>Blackbox testing</i> menu <i>monitoring</i> pada LCD.....	89
Tabel 4.6 Hasil Pengukuran pada aplikasi blynk IoT, LCD, Multimeter	92

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Lengkap : Andika Jaya Saputra
NIM : 321 21 031
Program Studi : D3 Teknik Listrik
Tempat / Tgl. Lahir : Makassar / 15 Agustus 2001
Alamat : Jl. Kutacane Selatan No.52

Dengan ini menyatakan :

A. Tugas Akhir / Skripsi yang berjudul :

**"Rancang Bangun Sistem Kontrol & Monitori Beban pada Panel PLTS
Menggunakan ESP8266 Berbasis IoT"**

Adalah benar disusun / dibuat oleh saya sendiri dan jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata Tugas Akhir / Skripsi tersebut dibuatkan oleh orang lain atau diketahui bahwa Tugas Akhir / Skripsi tersebut merupakan plagiat/mencontek/menjiplak hasil karya ilmiah orang lain, maka dengan ini saya siap menerima segala yang ditimbulkan berupa pembatalan/pencabutan Gelar Akademik dan siap mengulang kembali dari awal.

B. Bahwa seluruh dokumen (copy ijazah, copy transkrip nilai) dan lain-lain sebagai persyaratan sidang adalah asli milik saya pribadi dan dapat saya pertanggung jawabkan keasliannya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Makassar, September 2024

Hormat Saya,


(Andika Jaya Saputra)
NIM 32121031

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Lengkap : M. AL KAUTSAR
NIM : 32121039
Program Studi : D3 TEKNIK LISTRIK
Tempat / Tgl. Lahir : CILALLANG/21 APRIL 2002
Alamat : KOMPLEKS SMP NEGERI 02 BELOPA DI CILALLANG

Dengan ini menyatakan :

A. Tugas Akhir / Skripsi yang berjudul :

"Rancang Bangun Sistem Kontrol Dan Monitoring Beban Pada Panel PLTS Menggunakan ESP8266 Berbasis Iot."

Adalah benar disusun / dibuat oleh saya sendiri dan jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata Tugas Akhir / Skripsi tersebut dibuatkan oleh orang lain atau diketahui bahwa Tugas Akhir / Skripsi tersebut merupakan plagiat/mencontek/menjiplak hasil karya ilmiah orang lain, maka dengan ini saya siap menerima segala yang ditimbulkan berupa pembatalan/pencabutan Gelar Akademik dan siap mengulang kembali dari awal.

B. Bahwa seluruh dokumen (copy ijazah, copy transkrip nilai) dan lain-lain sebagai persyaratan sidang adalah asli milik saya pribadi dan dapat saya pertanggung jawabkan keasliannya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Makassar, 17 Agustus 2024

Hormat Saya



(M. AL KAUTSAR)

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Lengkap	: MUH FAUDZAN PANDANGAI
NIM	: 32121042
Program Studi	: D3 TEKNIK LISTRIK
Tempat / Tgl. Lahir	: PANGKEP/16 OKTOBER 2003
Alamat	: JL.POROS MACINNA RAYA

Dengan ini menyatakan :

A. Tugas Akhir / Skripsi yang berjudul :

"Rancang Bangun Sistem Kontrol Dan Monitoring Beban Pada Panel PLTS Menggunakan ESP8266 Berbasis Iot."

Adalah benar disusun / dibuat oleh saya sendiri dan jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata Tugas Akhir / Skripsi tersebut dibuatkan oleh orang lain atau diketahui bahwa Tugas Akhir / Skripsi tersebut merupakan plagiat/mencontek/menjiplak hasil karya ilmiah orang lain, maka dengan ini saya siap menerima segala yang ditimbulkan berupa pembatalan/pencabutan Gelar Akademik dan siap mengulang kembali dari awal.

B. Bahwa seluruh dokumen (copy ijazah, copy transkrip nilai) dan lain-lain sebagai persyaratan sidang adalah asli milik saya pribadi dan dapat saya pertanggung jawabkan keasliannya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Makassar, 17 Agustus 2024

Hormat Saya,



(MUH FAUDZAN PANDANGAI)

RANCANG BANGUN SISTEM KONTROL DAN *MONITORING* BEBAN PADA PENEL PLTS MENGGUNAKAN ESP8266 BERBASIS IoT

RINGKASAN

Dalam beberapa tahun terakhir, penggunaan pembangkit listrik tenaga surya (PLTS) sebagai sumber energi terbarukan telah mengalami peningkatan signifikan yaitu berada di urutan ke empat sebesar 207,8 GWp pada tahun 2018. PLTS menjadi pilihan utama untuk mengurangi ketergantungan pada sumber energi fosil dan mengurangi dampak negatif terhadap lingkungan. ESP8266 adalah modul WiFi yang murah dan dapat diandalkan, ideal untuk proyek IoT. Keberadaannya memungkinkan pengembang untuk dengan mudah mengintegrasikan kemampuan jaringan nirkabel ke dalam berbagai perangkat elektronik khususnya penggunaan untuk kontrol dan *monitoring* beban pada panel PLTS.

Penelitian ini bertujuan untuk merancang alat kontrol dan *monitoring* beban pada panel PLTS berupa lampu berbasis IoT dimana pengontrolan dan *monitoring*-Nya secara manual menggunakan *handphone* melalui aplikasi *blynk* pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan. Penelitian ini diharapkan mampu menerapkan penggunaan teknologi energi baru terbarukan, Sehingga memberikan kemudahan oleh pengguna dalam mengontrol dan memonitoring konsumsi energi listrik secara efektif dan efisien sekaligus diharapkan dapat bermanfaat kepada masyarakat tekhusus jama'ah Masjid Uwais Al-Qorni untuk mempermudah akses menuju masjid terutama pada malam hari dengan adanya penerangan di gerbang dan halaman masjid. Setelah melakukan uji coba, sistem kontrol dan *monitoring* beban pada panel PLTS menggunakan ESP8266 berbasis IoT didapatkan hasil kontrol dan *monitoring*-Nya dimana di dapatkan nilai tegangan 13,1 V – 13,7 V dan nilai arus 0,3 A – 1,6 A. Sehingga di harapkan dari penelitian ini dapat memudahkan pengontrolan dan *monitoring* beban lampu dari jarak jauh melalui *handphone* selama terkoneksi dengan internet.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, penggunaan pembangkit listrik tenaga surya (PLTS) sebagai sumber energi terbarukan telah mengalami peningkatan signifikan yaitu berada di urutan ke empat sebesar 207,8 GWp pada tahun 2018. PLTS menjadi pilihan utama untuk mengurangi ketergantungan pada sumber energi fosil dan mengurangi dampak negatif terhadap lingkungan.

Internet of Things (IoT) telah membawa revolusi dalam cara kita mengumpulkan, mentransmisikan, dan menganalisis data. Penggunaan IoT dalam PLTS memungkinkan pengawasan dan pengendalian jarak jauh, efisiensi yang lebih baik, dan pemantauan *real-time*.

ESP8266 adalah modul WiFi yang murah dan dapat diandalkan, ideal untuk proyek IoT. Keberadaannya memungkinkan pengembang untuk dengan mudah mengintegrasikan kemampuan jaringan nirkabel ke dalam berbagai perangkat elektronik khususnya penggunaan untuk kontrol dan *monitoring* beban pada panel PLTS.

Pada penelitian (Hasan, 2020) mengenai “Rancang bangun dan *monitoring* miniatur sinyal (sign) menggunakan tenaga *solar cell* berbasis IoT di bandar” memberikan hasil yang optimal dalam mengontrol dan memonitoring beban lampu terkait besar energi yang dikonsumsi beserta meng on/off -kan lampu berbasis sistem IoT.

Penelitian ini mencoba memadukan aspek tersebut, sehingga dirancanglah sebuah panel yang mampu mengontrol dan memonitoring beban pada PLTS menggunakan mikrokontroler ESP8266 berbasis IoT pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan. Penelitian ini diharapkan mampu menerapkan penggunaan teknologi energi baru terbarukan. Sehingga memberikan kemudahan oleh pengguna dalam mengontrol dan memonitoring konsumsi energi listrik secara efektif dan efisien sekaligus diharapkan dapat bermanfaat kepada masyarakat terkhusus jama'ah Masjid Uwais Al-Qorni untuk mempermudah akses menuju masjid terutama pada malam hari dengan adanya penerangan di gerbang dan halaman masjid.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat di buat suatu rumusan permasalahan sebagai berikut :

1. Bagaimana cara merancang bangun sistem kontrol dan *monitoring* beban pada panel PLTS pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan?.
2. Bagaimana cara mengevaluasi sistem kontrol dan *monitoring* pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan?.

1.3 Ruang Lingkup Penelitian

Berdasarkan rumusan masalah di atas, maka pada ruang lingkup penelitian ini akan dibuat batasan-batasan masalah untuk membahas hal-hal yang lebih

spesifik tentang yang telah dirumuskan. Hal ini dilakukan agar pembahasan pada penulisan skripsi sesuai dengan judul yang telah dibuat dan ditetapkan. Batasan-batasan masalah tersebut yaitu sebagai berikut:

1. Memilih komponen perangkat keras yang sesuai untuk sistem, termasuk ESP8266 sebagai modul IoT, sensor-sensor (seperti, arus, tegangan dan daya), serta perangkat keras lainnya.
2. Merancang dan mengembangkan sirkuit elektronik untuk menghubungkan ESP8266 dengan sensor-sensor dan perangkat kontrol PLTS, termasuk mekanisme kontrol dan *monitoring* beban pada panel PLTS.
3. Mendesain tampilan aplikasi perangkat lunak yang menjalankan fungsi kontrol dan *monitoring* pada ESP8266. Adapun aplikasi yang digunakan yaitu *Blynk*.
4. Menangani aspek konektivitas, termasuk pengelolaan koneksi Wi-Fi pada ESP8266 dan protokol komunikasi yang digunakan untuk mentransfer data antara perangkat panel dan platform IoT.
5. Rancang bangun sistem kontrol dan *monitoring* akan di implementasikan pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan.

1.4 Tujuan Penelitian

Tugas Akhir ini bertujuan untuk :

1. Merancang bangun sistem kontrol dan *monitoring* beban pada panel PLTS pada Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan.

2. Mengevaluasi sistem kontrol dan *monitoring* pada Masjid Uwais Al-Qorni
Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan.

1.5 Manfaat Penelitian

Tugas akhir ini diharapkan dapat memenuhi beberapa manfaat antara lain:

1. Pemantauan *Real-Time*

Memberikan kemampuan pemantauan *output* beban pada PLTS berupa lampu secara *real-time*, memungkinkan pengguna untuk mengamati produksi dan konsumsi energi secara langsung.

2. Kontrol Jarak Jauh

Memungkinkan pengguna untuk mengontrol beban pada panel PLTS berupa lampu dari jarak jauh melalui koneksi internet, memberikan fleksibilitas dalam mengelola operasi dan pengaturan sistem.

3. Penerangan gerbang dan halaman masjid

Penempatan lampu di bagi menjadi dua tempat, yaitu sebagai penerangan gerbang masuk masjid dengan lampu sorot LED 50 watt DC. Dan juga 2 lampu sorot LED, yaitu lampu sorot LED 20 watt DC dan lampu sorot LED 30 watt DC diperuntukkan untuk penerangan di halaman masjid. Penempatan posisi lampu tersebut di harapkan bermanfaat bagi jama'ah setempat dalam perjalanan menuju ke tempat ibadah.

BAB II

TINJAUAN PUSTAKA

3.1 *Study literature*

Adapun tim kami terinspirasi dari beberapa judul hasil penelitian terdahulu yaitu sebagai berikut :

1. Rancang Bangun dan *Monitoring* Miniatur Sinyal (*Sign*) Menggunakan Tenaga *Solar Cell* Berbasis IoT di Bandar(Hasan, 2022).
2. Sistem Kontrol Beban dan *Monitoring* Daya Baterai pada Panel Surya 50WP untuk Aplikasi Penerangan Berbasis IoT(Putri, 2022).
3. Rancang Bangun Sistem Kontrol Pembangkit *Hybrid* (PLN-*Solar Cell*) Berbasis Wemos D1 Mini Esp8266(Suriana, 2022).
4. Implementasi IoT dalam Rancang Bangun Sistem *Monitoring* pada *Solar Cell* Berbasis WEB(Setiawan, 2021).
5. Rancang Bangun Sistem *Smart Power* untuk Mengontrol dan Memonitor Energi Listrik Berbasis IoT(Kurniawan, 2020).

3.2 Pembangkit Listrik Tenaga Surya

Sinar dan panas dari matahari merupakan energi surya. Fotosintesis buatan, arsitektur surya, listrik panas surya, pemanas surya, fotovoltaiik surya, merupakan pemanfaatan dari energi surya. Energi Surya dapat dimanfaatkan pada berbagai tingkatan di seluruh dunia, yang utamanya bergantung pada jarak dari khatulistiwa(Kurniawan, 2020).

Tegangan pada PLTS pembangkit selalu dinaikkan sampai dengan 5%. Hal ini dimaksudkan agar dapat mengantisipasi terjadinya drop tegangan pada saluran dengan rincian sbb:

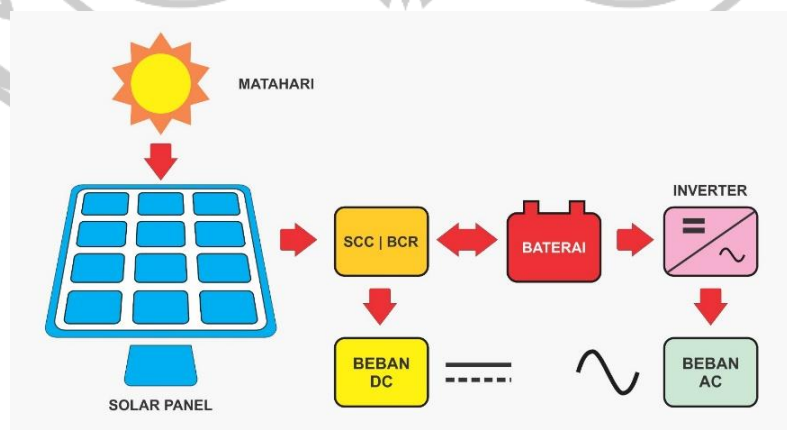
1. Maksimum 3% hilang pada saluran antara pembangkit (dalam hal ini PLTS distribusi) sampai dengan sambungan rumah.
2. Maksimum 1% hilang pada saluran antara sambungan rumah sampai dengan KWh meter.
3. Maksimum 1% hilang pada saluran KWh meter - panel, pembagi - alat listrik terjauh.

Semakin besar rugi daya dalam persen, berarti semakin besar kerugian energi yang terjadi. Penyebab Gangguan PLTS : Tegangan Lebih Akibat Petir, *Overload* dan Beban Tidak Seimbang, *Loss Contact* Pada Terminal *Bushing*, Isolator Bocor/*Bushing* Pecah, Kegagalan Isolasi Minyak PLTS/*Packing* Bocor(Darma, 2017).

3.2.1 Prinsip Kerja PLTS

Fotovoltaik (biasanya disebut juga sel surya) adalah piranti semikonduktor yang dapat merubah cahaya secara langsung menjadi menjadi arus listrik searah (DC) dengan menggunakan kristal silikon (Si) yang tipis. Sebuah kristal silindris Si diperoleh dengan cara memanaskan Si itu dengan tekanan yang diatur sehingga Si itu berubah menjadi penghantar. Bila kristal silindris itu dipotong setebal 0,3 mm, akan terbentuklah sel-sel silikon yang tipis atau yang disebut juga dengan sel surya (fotovoltaik). Sel-sel silikon itu dipasang

dengan posisi sejajar/seri dalam sebuah panel yang terbuat dari alumunium atau baja anti karat dan dilindungi oleh kaca atau plastik. Kemudian pada tiap-tiap sambungan sel itu diberi sambungan listrik. Bila sel-sel itu terkena sinar matahari maka pada sambungan itu akan mengalir arus listrik. Besarnya arus/tenaga listrik itu tergantung pada jumlah energi cahaya yang mencapai silikon itu dan luas permukaan sel itu. Pada dasarnya sel surya fotovoltaik merupakan suatu dioda semikonduktor yang berkerja dalam proses tak seimbang dan berdasarkan efek fotovoltaik. Dalam proses itu sel surya menghasilkan tegangan 0,5-1 volt tergantung intensitas cahaya dan jenis zat semikonduktor yang dipakai. Sementara itu intensitas energi yang terkandung dalam sinar matahari yang sampai ke permukaan bumi besarnya sekitar 1000 Watt. Tapi karena daya guna konversi energi radiasi menjadi energi listrik berdasarkan efek fotovol-taik baru mencapai 25%, maka produksi listrik maksimal yang dihasilkan sel surya baru mencapai 250 Watt per m². Cara kerja fotovoltaik dapat dilihat pada gambar 2.1



Gambar 2.1 Cara kerja fotovoltaik
(Sumber : {<http://jurnal.untad.ac.id>})

3.2.2 Jenis-jenis solar cell

1. *Single crystalline*

Yaitu kristal yang mempunyai satu jenis macamnya, tipe ini dalam perkembangannya mampu menghasilkan efisiensi yang sangat tinggi. Jenis *single crystalline* antara lain:

a. *Gallium Arsenide Cell.*

Gallium arsenide cell sangat efisien dari semua sel, tetapi harganya sangat mahal. Efisiensi dari sel ini mampu mencapai 25 persen(Bachtiar, 2006). *Gallium arsenide cell* dapat dilihat pada gambar 2.2



Gambar 2.2 *Gallium arsenide cell*
(Sumber : {<https://www.universitywafer.com>})

b. *Cadmium Sulfide Cell.*

Cadmium sulfide cell ini merupakan suatu bahan yang dapat dipertimbangkan dalam pembuatan sel surya, karena harga yang murah dan mudah dalam proses pembuatannya(Bachtiar, 2006). *Cadmium Sulfide Cell* dapat dilihat pada gambar 2.3

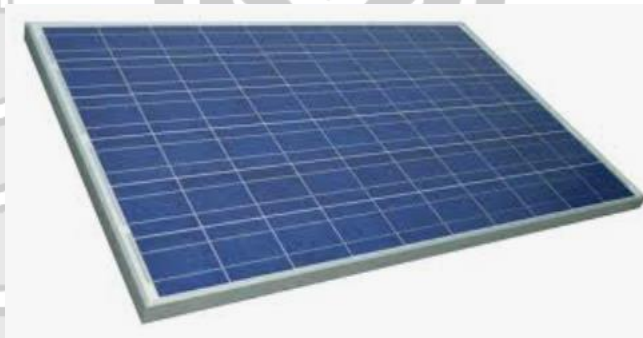


Gambar 2.3 *Cadmium Sulfide Cell*
(Sumber : {<https://www.ebay.com>})

2. *Polycrystalline cell.*

Polycrystalline cell merupakan kristal yang banyak macamnya, terbuat dari kristal silikon dengan efisiensi 10-12 persen (Bachtiar, 2006).

Polycrystalline cell dapat dilihat pada gambar 2.4



Gambar 2.4 *Polycrystalline cell*
(Sumber : {<https://www.len.co.id>})

3. Amorphous Silikon Cell.

Amorphous berarti tidak memakai kristal struktur atau non kristal, bahan yang digunakan berupa proses film yang tipis dengan efisiensi sekitar 4-6 persen (Bachtiar, 2006). *Amorphous Silikon Cell* dapat dilihat pada gambar 2.5



Gambar 2.5 *Amorphous Silikon Cell*
(Sumber : {<https://www.cheersonic.com>})

4. Copper indium diselenide (CIS) cells

Bahan semikonduktor yang aktif dalam sel surya CIS adalah *copper indium diselenide*. Senyawa CIS sering juga merupakan paduan dengan gallium dan / atau belerang, Efisiensi 9 persen sampai 11 persen (Bachtiar, 2006). *Copper indium diselenide cells* dapat dilihat pada gambar 2.6

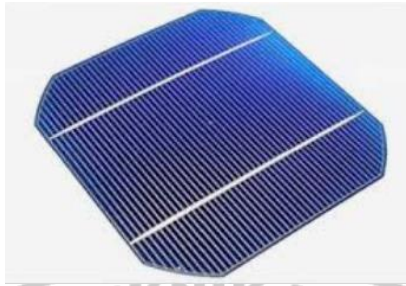


Gambar 2.6 *Copper indium diselenide cells*
(Sumber : {<https://www.solar.sheffield.ac.uk>})

5. *Cadmium telluride (CdTe) cells*

Sel surya CdTe diproduksi pada substrat kaca dengan lapisan konduktor TCO transparan biasanya terbuat dari *indium tin oxide* (ITO) sebagai kontak depan. Efisiensi 1 persen hingga 8,5 persen per efisiensi modul (Bachtiar, 2006). *Cadmium telluride cells* dapat dilihat pada gambar

2.7



Gambar 2.7 *Cadmium telluride cells*

(Sumber : {<https://www.justdial.com>})

6. *Dye sensitized*

Prinsip kerja *Dye sensitized* yaitu menyerap cahaya dalam pewarna organik mirip dengan cara di mana tanaman menggunakan klorofil untuk menangkap energi dari sinar matahari dengan fotosintesis (Bachtiar, 2006).

Dye sensitized cells dapat dilihat pada gambar 2.8



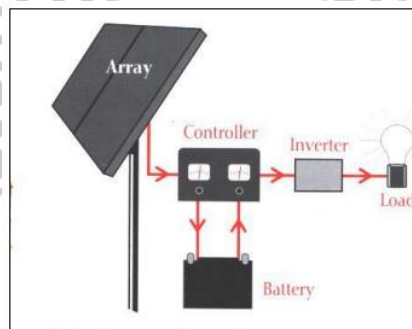
Gambar 2.8 *Dye sensitized cells*

(Sumber : {<https://www.solarquotes.com.au>})

3.2.3 **Komponen-komponen utama PLTS**

Komponen utama sistem surya fotovoltaik adalah modul yang merupakan unit rakitan beberapa sel surya fotovoltaik. Modul fotovoltaik tersusun dari beberapa sel fotovoltaik yang dihubungkan secara seri dan paralel. Teknologi ini cukup canggih dan keuntungannya adalah harganya murah, bersih, mudah dipasang dan dioperasikan dan mudah dirawat. Sedangkan kendala utama yang dihadapi dalam pengembangan energi surya fotovoltaik adalah investasi

awal yang besar dan harga per kWh listrik yang dibangkitkan relatif tinggi, karena memerlukan subsistem yang terdiri atas baterai, unit pengatur dan inverter sesuai dengan kebutuhannya. Sistem PLTS dapat dilihat pada gambar 2.9



Gambar 2.9 Sistem PLTS

(Sumber : {<http://jurnal.untad.ac.id>})

1. *solar cells*

Sel surya biasanya dinamai semikonduktor bahan dari mana mereka dibuat. Bahan-bahan tersebut harus mempunyai sifat tertentu karakteristiknya untuk menyerap sinar matahari. Beberapa sel memang demikian dirancang untuk menangani sinar matahari yang mencapai permukaan bumi, sementara yang lain dioptimalkan untuk digunakan di luar angkasa (Pido, 2019).

Sel surya bisa hanya terbuat dari satu lapisan bahan penyerap cahaya (persimpangan tunggal) atau menggunakan beberapa konfigurasi fisik (multi-persimpangan) untuk memanfaatkan berbagai penyerapan dan mekanisme pemisahan muatan. Sel surya dapat diklasifikasikan menjadi sel generasi pertama, kedua, dan ketiga. Pertama sel generasi—juga

disebut konvensional, tradisional atau sel berbasis wafer—terbuat dari silikon kristal, itu teknologi PV yang dominan secara komersial, termasuk bahan seperti polisilikon dan silikon monokristalin. Sel generasi kedua adalah sel surya film tipis, yang meliputi silikon amorf, sel CdTe dan CIGS dan signifikan secara komersial dalam pembangkit listrik fotovoltaik skala utilitas stasiun, membangun fotovoltaik terintegrasi atau dalam sistem tenaga kecil yang berdiri sendiri (Pido, 2019).

Sel surya generasi ketiga mencakup sejumlah teknologi film tipis yang sering dijelaskan sebagai fotovoltaik yang baru muncul—sebagian besar belum muncul diterapkan secara komersial dan masih dalam tahap penelitian atau fase pengembangan. Banyak yang sering menggunakan bahan organik senyawa organologam serta zat anorganik. Terlepas dari kenyataan bahwa efisiensinya rendah dan stabilitas bahan penyerap seringkali terlalu pendek aplikasi komersial, ada banyak penelitian yang diinvestasikan ke dalam teknologi ini karena mereka berjanji untuk mencapai tujuan memproduksi sel surya berbiaya rendah dan berefisiensi tinggi. Panel “generasi pertama” mencakup sel surya silikon. Mereka terbuat dari kristal silikon tunggal (mono-kristal), atau dipotong dari blok silikon yang terdiri dari banyak kristal (multi-kristal - ditunjukkan di sebelah kanan). Sel surya film tipis “generasi kedua” lebih sedikit mahal untuk diproduksi dibandingkan sel surya silikon tradisional mereka membutuhkan lebih sedikit bahan untuk konstruksi.

Sel PV film tipis, sesuai dengan namanya, adalah teknologi tipis secara fisik yang telah diterapkan fotovoltaiik. Mereka hanya sedikit kurang efisien dibandingkan yang lain jenis tetapi membutuhkan lebih banyak luas permukaan untuk menghasilkan yang sama jumlah kekuatan(Pido, 2019).

2. MPPT *Charge Controller*

Untuk meningkatkan efisiensi panel surya, penggunaan MPPT yang merupakan perangkat elektronika daya hadir ikut bermain. Dengan menggunakan MPPT, sistem akan dimulai beroperasi pada Titik Daya Maksimum (MPP) dan menghasilkan keluaran daya maksimumnya dengan mendeteksi radiasi maksimum matahari yang jatuh ke PV modul. Dengan demikian, menghasilkan biaya sistem secara keseluruhan(Bagher, 2015).

Teknik pelacakan titik daya maksimum (MPPT). digunakan untuk meningkatkan efisiensi panel surya. Demikianlah tujuan MPPT sistemnya adalah mengambil sampel sel PV keluaran dan menerapkannya resistensi yang tepat untuk mendapatkan daya maksimum untuk apa pun keadaan lingkungan. MPP adalah produk dari Tegangan MPP (V_{mpp}) dan arus MPP (I_{mpp})(Bagher, 2015). Keuntungan dan kekurangan PWM dan MPPT dapat dilihat pada tabel 2.1 dan tabel 2.2

Tabel 2.1 Keuntungan PWM dan MPPT

PWM	MPPT
1. Pengontrol PWM dibangun pada suatu teknologi yang teruji 2. Pengendali ini tidak mahal. 3. Pengontrol PWM tersedia dalam ukuran hingga 60amp. 4. Dapat mengontrol arus dan tegangan tinggi 5. Diharapkan lebih lama masa hidup-Nya	1. Muatan surya MPPT penawaran pengontrol peningkatan dalam efisiensi hingga 30%. 2. Pengendali ini juga menawarkan kemampuan potensial untuk memiliki array dengan masukan yang lebih tinggi tegangan dari bank baterai. 3. Memaksa modul PV untuk beroperasi pada tegangan mendekati maksimum titik daya untuk menggambar ketersediaan kekuatan maksimal kekuatan. 4. Mengurangi kompleksitas dari sistem, sementara keluaran sistem sangat efisien

Tabel 2.2 Kekurangan PWM dan MPPT

PWM	MPPT
1. Masukan surya tegangan nominal harus cocok bank baterai tegangan nominal jika kita akan menggunakan PWM. 2. Tidak ada satu pun berukuran pengontrol lebih dari 60 amp DC sampai saat ini. 3. Pengontrol PWM memiliki keterbatasan kapasitas untuk sistem pertumbuhan.	1. Pengontrol MPPT lebih mahal. 2. Unit MPPT umumnya lebih besar di ukuran fisik. 3. Menggunakan MPPT pengontrol memaksa susunan surya menjadi terdiri dari sejenisnya modul fotovoltaik dalam string yang sama.

Prinsip kerja pengontrol muatan PWM adalah mencocokkan tegangan panel ke tegangan baterai dan menurunkan tegangan keluaran panel. Sedangkan MPPT merupakan teknologi terbaru yang dimaksudkan untuk mengekstrak daya maksimal dari panel surya. Beroperasi sesuai dengan tegangan panel dan mengkonversi ekstrak tegangan panel menjadi arus yang meningkatkan keluaran dari *solar cells*(Bagher, 2015).

Pengontrol MPPT setidaknya 30% lebih banyak efisien daripada pengontrol PWM yaitu dengan MPPT kita mendapatkan 30% lebih banyak output dari sistem tenaga surya(Bagher, 2015). MPPT dapat dilihat pada gambar 2.10



Gambar 2.10 MPPT

(Sumber : {<https://www.jakartanotebook.com>})

3. Lampu LED

Prinsip kerja lampu LED adalah lampu ini sebenarnya dapat disuplai dengan tegangan rendah DC 12V, karena LED tetap menyala dengan suplai tegangan DC rendah. Karena instalasi rumah umumnya menggunakan tegangan AC 220 V, maka pada lampu LED yang dijual di pasaran ditambahkan peralatan penurun tegangan AC dari 220 V ke 12 V dan penyearah dari tegangan AC 12 V menjadi DC 12 V. LED-LED ini berbentuk persegi pipih, disusun seri agar dapat disuplai dengan tegangan yang lebih tinggi dari 12 V (Setiawan, 2019). Lampu LED dapat dilihat pada gambar 2.11

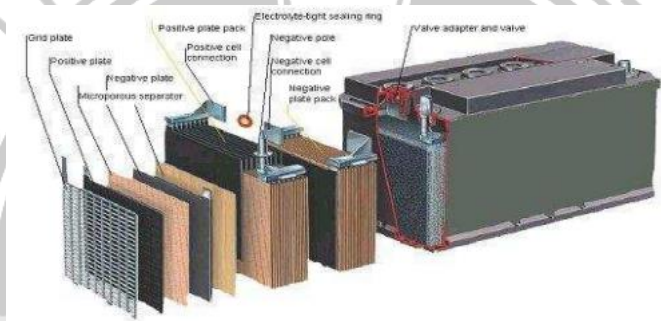


Gambar 2.11 Lampu LED

(Sumber : { <https://baleomol.com> })

4. Baterai

Baterai atau aki adalah penyimpan energi listrik pada saat matahari tidak ada. Baterai yang cocok digunakan untuk PV adalah baterai *deep cycle lead acid* yang mampu menampung kapasitas 100 Ah, 12 V, dengan efisiensi sekitar 80%. Waktu pengisian baterai/aki selama 12 jam - 16 jam (Pynkyawati, 2021). Baterai/aki dapat dilihat pada gambar 2.12



Gambar 2.12 Baterai/aki

(Sumber : {<https://d1wqtxts1xzle7.cloudfront.net>})

2.3 Potensi Energi Surya di Indonesia

Laju pertumbuhan penduduk beriringan dengan meningkatnya konsumsi terhadap energi. Energi terbarukan dan konservasi energi perlu dikembangkan sebagai alternatif jika energi konvensional semakin langka. Untuk menjaga ketahanan dan kemandirian energi, masyarakat global dan pemerintah secara terus-menerus meningkatkan peran energi baru terbarukan (EBT) sebagai solusi dalam pengurangan emisi gas rumah kaca dan solusi atas penurunan produksi bahan bakar minyak bumi. Merujuk pada PP No. 79 tahun 2014 tentang Kebijakan Energi Nasional, target bauran energi baru dan terbarukan pada tahun 2025 paling sedikit 23% dan 31% di tahun 2050. Berdasarkan

sumber Dirjen EBTKE, tahun 2018 : Indonesia memiliki potensi berbagai EBT yang cukup besar untuk mencapai target bauran energi primer tersebut, hal ini ditunjukkan pada Tabel 2.3 dimana Energi Surya mencapai 207,8 GWp dan total potensi EBT untuk pembangkit listrik tahun 2018 sebesar 8,8 GW atau 14% dari total kapasitas pembangkit listrik (fosil dan non fosil) yaitu sebesar 64,5 GW. Minimnya pemanfaatan EBT untuk ketenagalistrikan disebabkan masih relatif tingginya harga produksi dan investasi pembangkit berbasis EBT, sehingga sulit bersaing dengan pembangkit listrik energi konvensional terutama batu bara (Hasan, 2012). Potensi EBT di Indonesia tahun 2018 dapat dilihat pada tabel 2.3

Tabel 2.3 Potensi EBT di Indonesia tahun 2018

No.	Jenis Potensi Energi	Potensi Pembangkitan
1	Tenaga air	94,3 GW
2	Energi Panas Bumi	28,5 GW
3	Bioenergi	PLT Bio : 32,6 GW
4	Energi Surya	207,8 GWp
5	Energi Angin	60,6 Gw
6	Energi Laut	17,9 Gw

2.4 Komponen panel *monitoring* dan kontrol beban PLTS

Panel *monitoring* dan kontrol beban PLTS merupakan suatu produk penerapan dari perkembangan ilmu pengetahuan dewasa ini. Produk ini merupakan inovasi suatu sistem PLTS berupa *monitoring* dan kontrol beban PLTS dari jarak jauh dengan menggunakan ESP8266 terkoneksi WIFI berbasis IOT dihubungkan ke *smartphone* via *Blynk*. Berikut merupakan komponen-komponen yang dibutuhkan :

1. Box-X6

Box X-6 merupakan produk panel kecil yang sangat baik digunakan untuk menempatkan rangkaian komponen mikrokontroler. Panel kecil ini terbuat dari bahan PVC (plastik), dengan dimensi 18,5 cm x 11,7 cm x 6,5 cm. Box-X6 dapat dilihat pada gambar 2.13



Gambar 2.13 Box-X6

(Sumber : { <https://biggo.id> })

2. ESP8266

ESP8266 adalah sebuah komponen chip terintegrasi yang didesain untuk keperluan dunia masa kini yang serba tersambung. Chip ini menawarkan solusi *networking* Wi-Fi yang lengkap dan menyatu, yang dapat digunakan sebagai penyedia aplikasi atau untuk memisahkan semua fungsi *networking* Wi-Fi ke pemroses aplikasi lainnya. ESP8266 memiliki kemampuan *on-board procesing* dan *storage* yang memungkinkan chip tersebut untuk diintegrasikan dengan sensor-sensor atau dengan aplikasi alat tertentu melalui pin input output hanya dengan pemrograman singkat(Iskandar, 2020).

Dengan level yang tinggi berupa *on-chip* yang terintegrasi memungkinkan external sirkuit yang ramping dan semua solusi, termasuk

modul sisi depan, didesain untuk menempati area PCB yang sempit(Iskandar, 2020).

Perlu diperhatikan bahwa modul ESP8266 bekerja dengan tegangan maksimal 3,6V. Hubungkan Vcc modul WiFi ke pin 3.3V pada Arduino. (Jangan yang ke 5V). Jika sudah mendapat tegangan, modul WiFi akan menyala merah, dan sekali-kali akan berkedip warna biru(Iskandar, 2020). ESP8266 dapat dilihat pada gambar 2.14



Gambar 2.14 ESP8266

(Sumber : {<https://www.arduino.biz.id>})

3. PZEM 017 T

PZEM-017 adalah modul komunikasi DC yang mampu mengukur daya DC hingga 300 VDC dan pengukuran arus pada rentang pemasangan shunt eksternal 50 A hingga 300 A. Modul ini juga dapat mengukur tegangan, arus, daya dan energi. Semua seri PZEM Energy Meter memiliki antarmuka komunikasi RS485 bawaan yang menggunakan protokol Modbus-RTU yang menyerupai kebanyakan perangkat industri lainnya (Siallagan 2020).

PZEM-017 adalah modul komunikasi DC yang dapat mengukur daya DC hingga 300 VDC dan pengukuran arus tunduk pada rentang terpasang

shunt eksternal 50 A, 100 A, 200 A, dan 300 A. Pzem 017T dapat dilihat pada gambar 2.15



Gambar 2.15 Pzem 017T

(Sumber : {<https://ejournal.poltekbangsby.ac.id>})

Berikut adalah deskripsi dan pengaplikasian dari modul PZEM-017a:

1. Untuk pengukuran tegangan modul ini memiliki rentang pengukuran 0,05 sampai 300 volt dengan resolusi tegangan 0,01 volt serta ke akurasi pengukuran sebesar 1%. (Ketika menggunakan sumber daya > 7 volt maka harus menggunakan power suplai yang independent).
2. Untuk pengukuran arus modul ini memiliki rentang pengukuran 0,02 - 300 A dengan resolusi 0,01 A serta akurasi pengukuran sebesar 1%.
3. Untuk pengukuran daya modul ini memiliki rentang pengukuran 0,2 – 90 kW dengan resolusi 0,1 Watt serta akurasi pengukuran sebesar 1%.
4. Untuk pengukuran tegangan modul ini memiliki rentang pengukuran 0,05 sampai 300 volt dengan resolusi tegangan 0,01 volt serta ke akurasi pengukuran sebesar 1%. (Ketika

menggunakan sumber daya > 7 volt maka harus menggunakan power suplai yang independent)

5. Untuk pengukuran arus modul ini memiliki rentang pengukuran 0,02 - 300 A dengan resolusi 0,01 A serta akurasi pengukuran sebesar 1%.
6. Untuk pengukuran daya modul ini memiliki rentang pengukuran 0,2 – 90 kW dengan resolusi 0,1 Watt serta akurasi pengukuran sebesar 1%.
7. Untuk pengukuran konsumsi energi modul ini memiliki rentang pengukuran 0 – 9999 kWh dengan resolusi 1 Wh serta akurasi pengukuran 1 %.
8. Dalam modul ini ambang tegangan dapat diatur menjadi ambang tegangan tinggi dan ambang tegangan rendah, saat tegangan terukur melebihi ambang batas maka alarm dari modul ini akan berbunyi. (Ambang batas tegangan tertinggi adalah 300 volt dan ambang batas tegangan rendah adalah 7 volt).

4. Relai

Relai adalah komponen elektronika yang berupa saklar atau *switch* elektrik yang dioperasikan menggunakan listrik. Relai juga biasa disebut sebagai komponen *electromechanical* atau elektromekanikal yang terdiri dari dua bagian utama yaitu *coil* atau elektromagnet dan kontak saklar atau mekanikal (Faraidh, 2022).

Komponen relai menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang

kecil atau *low power*, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi(Faraidh, 2022).

Berikut adalah beberapa fungsi komponen relai saat diaplikasikan ke dalam sebuah rangkaian elektronika :

- a. Mengendalikan sirkuit tegangan tinggi dengan menggunakan bantuan signal tegangan rendah.
- b. Menjalankan fungsi logika alias *logicfunction*.
- c. Memberikan fungsi penundaan waktu alias *time delay function*.
- d. Melindungi motor atau komponen lainnya dari kelebihan tegangan atau korsleting.

Relai dapat dilihat pada gambar 2.16



Gambar 2.16 Relai

(Sumber : { <https://www.firgelliauto.com> })

5. Kabel NYAF

Kabel ini direncanakan dan direkomendasikan untuk instalasi dalam kabel kotak distribusi pipa atau didalam *duct*. Kabel NYAF merupakan jenis kabel fleksibel dengan penghantar tembaga serabut berisolasi PVC. Digunakan untuk instalasi panel-panel yang memerlukan fleksibilitas yang

langsung(Alkadri, 2019). Kabel NYAF dapat dilihat pada gambar 2.17



Gambar 2.17 Kabel NYAF

(Sumber : {<https://poliban.ac.id>})

2.5 Alat & bahan pendukung

Berikut merupakan alat pendukung dalam perancangan panel *monitoring* dan kontrol beban PLTS :

1. Mesin las

Mesin las listrik adalah alat yang digunakan untuk melakukan proses pengelasan dengan memanfaatkan energi listrik sebagai sumber panas untuk melelehkan material dan menggabungkannya. Mesin las dapat dilihat pada gambar 2.18



25

2. Besi siku

Besi Siku yaitu baja siku sama kaki dengan profil berpenampang L yang diciptakan dari Proses *hot rolling mill*, bertepi bulat dengan ukuran lebar 20 mm s.d 200 mm. Material yang berbentuk siku ini dipakai sebagai bahan produk Industri, Otomotif, bangunan, furnitur, konstruksi yang banyak sub fungsinya digunakan oleh kontraktor dan Pabrikator seperti bengkel las dan bengkel bubut. Besi siku dapat dilihat pada gambar 2.19



Gambar 2.19 Besi siku

(Sumber : {<https://hargatokobesi.com>})

3. Multimeter digital

Multimeter digital adalah alat ukur yang dapat mengukur besaran seperti tegangan, arus, dan hambatan. Nilai terukur ditampilkan pada tampilan digital, sehingga dapat dibaca dengan mudah dan langsung, bahkan oleh pengguna pertama kali. Multimeter digital dapat dilihat pada gambar 2.20



Gambar 2.20 Multimeter digital

(Sumber : {<https://alatuji.co.id>})

4. Solder

Solder merupakan jenis alat bantu kerja yang berfungsi untuk menyambungkan sebuah komponen pada peralatan elektronik. Cara kerja atau fungsi solder yaitu dengan memanaskan atau melelehkan timah dengan tujuan untuk proses memudahkan dalam menyambungkan sebuah komponen. Dalam dunia teknik permesinan biasanya kita mengenalnya dengan teknik penyambungan logam dengan menggunakan las atau *welding*. Solder dapat dilihat pada gambar 2.21

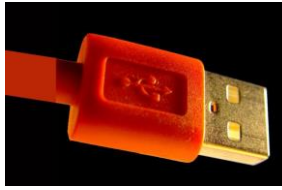


Gambar 2.21 Solder

(Sumber : {<https://lifestyle.pinhome.id>})

5. Kabel USB

USB singkatan dari *Universal Serial Bus*. Kabel USB adalah media transmisi yang memudahkan antara dua buah alat elektronik agar dapat saling terhubung dan terkoneksi. Dengan adanya USB, tidak hanya dapat membuat dua buah perangkat elektronik saling terhubung saja, akan tetapi juga dapat melakukan berbagai fungsi lainnya. Diantaranya mulai dari melakukan transfer ataupun menerima data, melakukan instalasi software, mengisi daya, backup data dan lain sebagainya. Kabel USB dapat dilihat pada gambar 2.22



Gambar 2.22 Kabel USB
(Sumber : {<https://thecityfoundry.com>})

2.6 Internet Of Things (IOT)

Internet of things atau bisa disebut juga dengan IoT adalah sebuah teknologi canggih yang memiliki konsep yang bertujuan untuk memperluas dan mengembangkan manfaat dari konektivitas internet yang tersambung terus-menerus. menghubungkan benda-benda di sekitar agar aktivitas sehari-hari menjadi lebih mudah dan efisien yang sangat membantu segala pekerjaan manusia. Pentingnya *internet of things* dapat dilihat dengan semakin banyaknya diterapkan dalam berbagai kehidupan saat ini. Menurut metode identifikasi RFID (*Radio Frequency Identification*), istilah IoT tergolong dalam metode komunikasi, meskipun IoT juga dapat mencakup teknologi sensor lainnya, teknologi nirkabel atau kode QR (*Quick Response*)(Suriansyah, 2014).

Istilah “*Internet Of Things*” terdiri dari dua bagian kata utama yaitu Internet yang menghubungkan dan mengatur sebuah konektivitas dan *Things* yang memiliki arti objek atau sebuah perangkat. Sederhananya, kamu memiliki “*Things*” yang dapat saling terhubung untuk mengumpulkan data dan mengirimkannya ke Internet. Data ini juga dapat diakses oleh “*Things*” lainnya juga. dimana sebuah “*Things*” tertentu memiliki kemampuan untuk

mengirimkan data lewat melalui jaringan dimanapun kamu berada dan tanpa adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer(Suriansyah, 2014).

2.7 *Blynk*

Blynk adalah *platform* yang mempermudah dalam pembuatan *interface* untuk melakukan *controlling* dan *monitoring* melalui Android. *Blynk* merupakan *framework* yang berupa aplikasi android dan didesain untuk *Internet of Things* yang dapat digunakan untuk melakukan *control hardware* secara *remote*, dapat menampilkan data sensor, menyimpan data, dan mengvisualisasikannya. Terdapat 3 komponen utama di *platform Blynk* yaitu *Blynk App* yang digunakan untuk membuat *interface* dengan *widget* yang disediakan, *Blynk Server* yang bertanggung jawab tentang semua komunikasi antara *smartphone* dan *hardware*, dan *Blynk Libraries* yang digunakan untuk komunikasi antara server dengan proses INPUT dan OUTPUT(Selay, 2022).

2.8 *software Arduino IDE*

Software Arduino yang akan digunakan adalah *driver* dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan Arduino. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java(Gozal, 2020). IDE Arduino terdiri dari:

- a. *Editor program*, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
- b. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah *microcontroller*

tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh *microcontroller* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.

- c. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino (Djuandi, 2011).

2.9 Menentukan Kebutuhan Komponen

Untuk menentukan berapa jumlah Panel PLTS yang digunakan, terlebih dahulu menentukan total daya beban yang digunakan x waktu pengoperasian beban. Untuk mengetahui berapa jumlah energi yang dihasilkan oleh PLTS untuk memenuhi kebutuhan beban dapat di rumuskan :

$$\text{Total beban} : 60\% \quad (1)$$

Penyinaran di Indonesia pada umumnya paling tinggi berlangsung 5 jam/hari yaitu dari pukul 10.00 pagi sampai 15.00 sore. Untuk menentukan jumlah PLTS yang akan digunakan terlebih dahulu menentukan daya puncak yang dihasilkan panel surya :

$$\text{Total daya} : \text{waktu penyinaran optimal} \quad (2)$$

Jika beban lampu DC beroperasi selama 12 jam, kita memerlukan cadangan energi di malam hari saat dimana PLTS tidak dapat menghasilkan energi listrik. Adapun solusinya adalah dengan menggunakan baterai. Adapun untuk menentukan cadangan baterai/aki sesuai kebutuhan beban yang digunakan :

$$\text{Cadangan} = \text{Daya beban} : (100\% - 5\%) \quad (3)$$

Untuk menentukan kapasitas baterai yang digunakan maka :

$$\text{Jumlah baterai} = \text{keluaran baterai} : \text{kapasitas baterai} \quad (4)$$

Karena baterai akan rusak jika di kuras secara berlebihan, maka kita perlu menyisakan 50% isi dari baterai dengan cara mengalikan 2 jumlah baterai yang diperlukan.

Untuk menentukan arus pada beban berupa lampu sorot DC dapat menggunakan rumus sebagai berikut :

$$P = I \times V$$
$$I = \frac{P}{V} \quad (5)$$

Keterangan :

P = daya (watt)

I = arus (ampere)

V = tegangan (volt)

Kemudian untuk menentukan Ampere dari SCC adalah dengan melihat spesifikasi pada panel surya (Utami, 2022).. Adapun parameter yang di butukan yaitu Isc-Nya. dapat di hitung :

$$\text{Ampere SCC} = \text{Isc} \times \text{Jumlah panel surya.} \quad (6)$$

2.10 Pemrograman Bahasa C

1. Struktur dasar program arduino

Karakteristik penulisan kode program Arduino sangatlah sederhana. Hanya terdiri atas dua fungsi utama, yaitu void setup() dan void loop(). Perlu kamu ketahui bahwa fungsi void pada Arduino itu untuk mendeklarasikan.

Struktur program Arduino :

```
Void setup()  
{  
  pernyataan;  
}  
Void loop()  
{  
  pernyataan;  
}
```

Kedua fungsi tersebut memiliki keterkaitan satu sama lain. Apabila salah satu fungsi tak terdapat di *sketch*, maka kemungkinan besar *sketch* akan *error*.

2. Perbedaan fungsi void setup() dan void loop()

Fungsi *void setup* pada Arduino adalah sebagai bentuk inisialisasi atau pengenalan dalam program Arduino dan hanya dieksekusi sekali sejak program dijalankan. Umumnya fungsi ini digunakan untuk pendefinisian mode pin atau memulai komunikasi serial. Contoh fungsi *void setup* :

```
Void setup()  
{  
  pinMode(13,OUTPUT);  
}
```

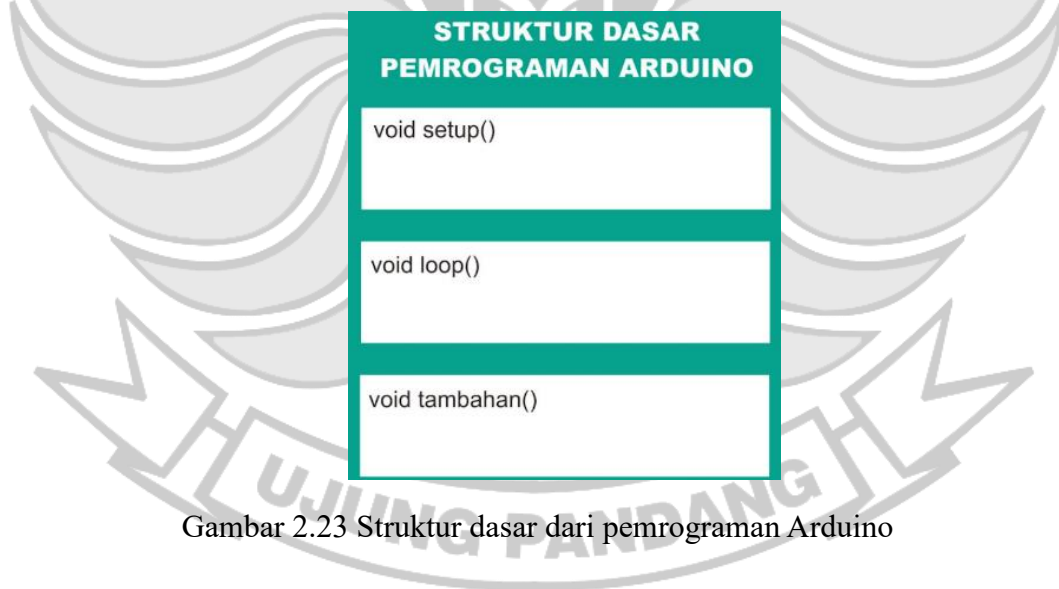
Keterangan:

Dalam perintah tersebut terdapat pernyataan atau *statement* yang berisi penginisialisasian pin 13 sebagai pin output atau keluaran. Setelah void setup() dijalankan, selanjutnya program akan menjalankan fungsi void

loop(). Fungsi ini akan dieksekusi terus-menerus secara berurutan hingga program berhenti dijalankan. Contoh fungsi void loop():

```
Void loop()
{
digitalWrite(13,HIGH);
delay(1000);
}
```

Untuk beberapa kasus pemrograman yang kompleks, terkadang programmer juga menambahkan *void* baru untuk mempermudah pemrograman. Jadi *void* baru ini nantinya bisa diletakkan setelah atau sebelum void loop(). Struktur dasar dari pemrograman Arduino dapat dilihat pada gambar 2.23



Gambar 2.23 Struktur dasar dari pemrograman Arduino

3. *Syntax* Arduino

Secara umum, *syntax* atau sintaks dikenal sebagai seperangkat aturan tata bahasa pada kata atau klausa dalam membentuk kalimat. Sedangkan dalam bahasa pemrograman, sintaks diartikan sebagai suatu

aturan dalam penulisan kode program agar kode tersebut bisa dimengerti komputer. Sintaks dalam Arduino bisa juga disebut sebagai aturan karakteristik penulisan kode program Arduino. Beberapa sintaks dasar pemrograman Arduino Uno antara lain:

a. Titik koma (*semicolon*)

Tanda titik koma adalah *syntax* wajib dalam pemrograman Arduino. Biasanya sintaks ini ditempatkan pada akhir pernyataan. Contoh titik koma :

```
Int x = 13;
```

b. Kurung kurawal (*Curly Braces*)

Kurung kurawal berfungsi untuk mendefinisikan awal dan akhir dari sebuah blok fungsi. Contoh kurung kurawal :

```
Void loop()  
{  
  Isi program  
}
```

c. #define

Fungsi define pada Arduino yaitu memungkinkan *programmer* dalam memberi nama untuk nilai konstan sebelum program dapat dikompilasi. Contoh #define :

```
#define ledpin 3
```

Keterangan:

ledPin = variabel yang digunakan

3 = nilai(value) atau pin yang sedang digunakan.

d. `#include`

Fungsi *include* pada Arduino yaitu untuk memasukkan *library* ke dalam *sketch*. Dimana fungsi *library* Arduino sendiri adalah memudahkan pengguna dalam melakukan pengodingan. Cara penggunaan sintaks sangat mudah. Yaitu menambahkan nama *library* yang ingin di-*import* ke program di belakang sintaks tersebut. Contoh `#include` :

```
#include <OneWire.h>
```

e. Komentar

Sintaks komentar pada program Arduino adalah sintaks yang berfungsi menambah keterangan pada program yang dibuat. Sehingga kata atau kalimat yang diberi sintaks komentar tak akan terbaca oleh *compiler*. Setidaknya ada dua macam sintaks komentar yang perlu kamu tahu, yaitu:

1) Sintaks komentar satu baris

Untuk membuat komentar yang hanya menggunakan satu baris saja, yaitu menggunakan sintaks garis miring dua kali `“//”`. Contoh sintaks komentar satu baris :

```
// isi komentar
```

2) Sintaks komentar lebih dari satu baris

Untuk membuat komentar yang jumlahnya barisnya lebih dari satu, yaitu menggunakan sintaks dua garis miring dan bintang.

Dimana komentar yang ingin dimasukkan berada di tengah. Contoh sintaks komentar lebih dari satu baris :

```
/* isi komentar */
```

4. Tipe data bahasa c Arduino

Jenis tipe data pada Arduino yang sering digunakan dapat dilihat pada tabel 2.4

Tabel 2.4 Jenis tipe data pada Arduino yang sering digunakan

Tipe Data	Lebar Data	Jangkauan
char	1 byte	-128 s/d 127
unsigned char	1 byte	0 s/d 255
byte	1 byte	0 s/d 255
word	2 byte	1 s/d 65535
int	2 byte	-32768 s/d 32767
unsigned int	2 byte	0 s/d 65535
long	4 byte	-2147438648 s/d 2147438647
unsigned long	4 byte	0 s/d 4294967295
float	4 byte	-3.4028235E+38 s/d 3.4028235E+38

Untuk format penulisan tipe data Arduino dalam program sangatlah mudah. Hanya perlu menempatkannya di depan variabel yang ingin di deklarasikan dengan diantara satu spasi. Misalkan mendeklarasikan suatu variabel bernama “cobaCoba” dengan tipe data integer dan menggunakan pin 5. Mendeklarasikan variabel :

```
Int cobacoba =5;
```

Jadi, fungsi int pada Arduino IDE yang letaknya di awal variabel adalah untuk mendeklarasikan bahwa variabel tersebut bertipe data integer.

5. Variabel Arduino

Variabel dalam bahasa program Arduino bisa didefinisikan sebagai suatu wadah untuk menyimpan atau menampung data. Kita dibebaskan memilih nama variabel yang diinginkan, asalkan sesuai dengan ketentuan berikut:

- Tidak menggunakan spasi
- Maksimal 32 karakter
- Tidak menggunakan istilah baku dalam bahasa C Arduino

Sebagai contoh, misalkan ingin mendefinisikan nilai pin 2 dengan variabel “ganteng” dan menggunakan tipe data integer. Maka contoh penulisan mendefinisikan nilai pin 2 :

```
Int ganteng =2;
```

6. Nilai konstan dalam pemrograman Arduino

Pada dasarnya, Arduino telah memiliki beberapa variabel yang telah memiliki nilai yang disebut sebagai konstan. Karena nilainya sudah ditentukan, jadi tak perlu lagi didefinisikan di awal. Berikut ini adalah macam-macam nilai konstan yang perlu diketahui:

a. TRUE/FALSE

TRUE dan FALSE adalah konstanta *boolean* yang digunakan untuk mendefinisikan level logika. Apakah ia bernilai benar (TRUE) atau salah (FALSE). TRUE didefinisikan sebagai 1 dan FALSE sebagai 0.

Contoh *true/false* :

```
If (c==TRUE);
```

```
{
```

Perintah:

```
}
```

b. HIGH/LOW

Variabel konstan HIGH dan LOW umumnya digunakan untuk menentukan kondisi pin saat membaca dan menulis data di pin digital. HIGH didefinisikan sebagai 1, ON, 5 volt, atau menyala. Sedangkan LOW didefinisikan sebagai 0, OFF, 0 volt, atau padam. Sehingga dapat dikatakan bahwa perintah untuk menyalakan/memberikan sinyal ON pada program Arduino adalah HIGH. Sedangkan perintah untuk memberikan sinyal OFF pada program Arduino adalah FALSE. Contoh penulisannya seperti ini. Contoh *high/low* :

```
digitalWrite(13, HIGH);
```

```
digitalWrite(13, LOW);
```

c. INPUT/OUTPUT

Variabel konstan ini sering digunakan pada fungsi pinMode() untuk mendefinisikan pin digital. Apakah ia berperan sebagai INPUT atau OUTPUT. Contoh *input/output* :

```
pinMode(13, OUTPUT);
```

7. Perintah pin digital I/O Arduino

Setidaknya, ada tiga macam perintah yang biasa digunakan untuk mengatur pin digital input/output dalam pemrograman Arduino, yaitu terdiri atas:

a. pinMode

Yaitu suatu perintah yang biasa ditemukan pada *block* void setup(). Fungsi pinMode() pada Arduino adalah untuk memperjelas apakah suatu pin digunakan sebagai INPUT atau OUTPUT. Contoh pinMode :
pinMode(3, OUTPUT);

Keterangan:

3 = nomor pin yang ingin dikonfigurasi
INPUT = mode yang digunakan

b. digitalWrite(pin)

Penggunaan perintah ini bertujuan untuk membaca nilai pin digital yang dikehendaki lalu menyimpannya dalam suatu variabel. Contoh digitalWrite :

Baca = digitalWrite(8);

Keterangan:

Baca = nama variabel yang menyimpan data hasil pembacaan

8 = nomor pin digital yang digunakan

c. digitalWrite(pin, value)

Yaitu suatu perintah untuk menuliskan suatu nilai pada pin digital. Lengkapnya, fungsi digitalWrite() pada Arduino adalah untuk menentukan nilai logika awal pada suatu pin. Apakah ia LOW atau HIGH. Contoh digitalWrite(pin, value) :

digitalWrite(9, HIGH);

Keterangan:

9 = nomor pin digital yang digunakan

HIGH = kondisi yang diinginkan

8. Perintah pin analog I/O Arduino

Berbeda dengan pin digital, pin analog hanya memiliki dua jenis perintah yang sering digunakan. Ini karena pin analog tak membutuhkan perintah penginisialisasian dengan `pinMode()`. Berikut ini adalah dua macam perintah pada pin analog yang sering digunakan.

a. `analogRead(pin)`

Fungsi `analogRead()` pada Arduino yaitu untuk membaca data dari pin analog kemudian disimpan dalam suatu variabel. Dimana data hasil pembacaan tersebut berupa nilai integer dengan kisaran 0 sampai 1023.

Contoh `analogRead(pin)` :

```
Baca = AnalogRead(A0);
```

Keterangan:

Baca = nama variabel tempat menampung data hasil pembacaan

A0 = nama pin analog yang digunakan (A0 sampai A6)

b. `analogWrite(pin, value)`

Perintah ini berfungsi untuk mengirimkan nilai analog dengan metode PWM (*Pulse Width Modulation*) pada pin khusus yang berlabel PWM, yaitu pin 3, 5, 6, 9, 10, dan 11. Pin PWM biasanya ditandai dengan

adanya tanda tilde (~) di depan atau di atas nama pin. Adapun untuk nilai yang biasa digunakan dalam PWM, yaitu mulai dari 0 sampai 255. Jadi, kamu bisa mengatur frekuensi PWM dari nilai tersebut.

Contoh `analogWrite(pin, value)` :

```
AnalogWrite(5, 100);
```

Keterangan:

5 = nama pin analog yang digunakan

100 = nilai PWM yang digunakan

Dari dua perintah tersebut, dapat diidentifikasi perbedaan `analogRead()` dan `analogWrite()` dari segi fungsi. Perintah `analogRead()` untuk membaca data analog Arduino, sedangkan perintah `analogWrite()` untuk menuliskan data analog pada pin Arduino. `analogWrite` untuk menulis data analog, sedangkan `digitalWrite()` untuk menulis data digital.

9. Pengujian kondisi dalam percabangan program Arduino

Agar tak salah dalam mengeksekusi perintah, program Arduino butuh yang namanya pengujian kondisi. Ini sangat penting ketika program berada dalam kondisi percabangan yang mengharuskannya memilih salah satu diantara beberapa pilihan. Pengujian kondisi merupakan perintah yang berfungsi memilih satu pernyataan yang paling benar dari beberapa pilihan pernyataan berdasarkan kriteria penilaian tertentu. Beberapa pengujian kondisi yang umumnya sering digunakan antara lain:

a. *if*

Perintah ini digunakan apabila hanya ada satu pernyataan yang ingin dieksekusi. Jadi apabila kondisinya memenuhi maka pernyataan akan dieksekusi. Dan jika tak memenuhi maka pernyataan tak dieksekusi.

Contoh *if*:

```
If(kondisi){  
Pernyataan/perintah;  
}  
if(x==6) {  
a = a+5;  
}
```

b. *if ... else*

Berbeda dengan *if*, perintah *if ... else* digunakan apabila ada dua pilihan pernyataan yang ingin dieksekusi. Apabila kondisinya terpenuhi, maka perintah yang ada di dalam *if* akan dieksekusi. Dan jika tidak, maka pernyataan di dalam *else*-lah yang akan dieksekusi. Contoh *if ... else* :

```
If(kondisi){  
Pernyataan/perintah 1  
}  
Else{  
Pernyataan/perintah 2  
}
```

c. *if ... else if*

Perbedaan perintah *if ... else if* dengan perintah *if* lainnya yaitu dapat melakukan pengujian lebih dari satu kondisi. Itulah alasan mengapa

perintah ini sering kali disebut dengan perintah *if* bertingkat Arduino.

Contoh *if... else if* :

```
If(kondisi1){  
  Pernyataan/perintah 1  
}  
Else If(kondisi2){  
  Pernyataan/perintah 2  
}  
Else If(kondisi ke-n){  
  Pernyataan/perintah ke-n  
}
```

d. *switch case*

Perintah ini digunakan untuk memilih pernyataan yang benar berdasarkan kondisi yang ada. Berbeda dengan perintah *if*, perintah *switch case* menggunakan nilai variabel untuk pengujiannya. Apabila nilai variabel memenuhi syarat salah satu dari *case*, maka ia akan mengeksekusi pernyataan di dalam *case* tersebut. Dan apabila tak ada nilai variabel yang cocok dari salah satu *case*, maka secara otomatis yang dieksekusi adalah pernyataan/perintah *default*.

Contoh *switch case* :

```
Switch(variabel){  
  
Case 1 : pernyataan/perintah 1  
  
Break;
```

Case 2 : pernyataan/perintah 2

Break;

Case n : pernyataan/perintah n

Break;

Default : pernyataan/perintah default

}

10. Pengulangan dalam pemrograman Arduino

Dalam program Arduino juga sering dikenal ada istilah pengulangan. Yaitu suatu perintah yang memungkinkan kita untuk mengeksekusi suatu pernyataan berdasarkan acuan nilai yang mengalami pengulangan.

Beberapa contoh perintah *coding* pengulangan Arduino yang biasa digunakan antara lain:

a. *while()*

Fungsi *while* pada Arduino yakni memungkinkan terjadinya pengulangan tak terbatas selama kondisi di dalamnya terpenuhi.

Format penulisannya seperti ini:

```
While(kondisi){  
  //pernyataan/perintah  
}
```

Keterangan:

Program akan terus melakukan pengulangan hingga nilai a mencapai angka 100.

b. *do ... while()*

Perintah ini berfungsi untuk mengeksekusi suatu pernyataan atau perintah lalu melihat kondisi di dalam *while*. Apabila kondisinya sesuai maka pernyataan akan dieksekusi kembali. Dan jika tidak maka pernyataan tak akan dieksekusi. Contoh *do ... while()* :

```
Do {  
  Pernyataan/perintah  
}  
While(kondisi);
```

Keterangan:

Perulangan pertambahan nilai *a* dengan 1 akan terus berlangsung hingga nilai *a* mencapai angka 100.

c. *for()*

Fungsi *for* pada Arduino yaitu untuk melakukan perulangan yang sifatnya terbatas. Dalam perintah *for* wajib disertakan nilai awal, kondisi, dan penambahan (*increment*) atau pengurangan (*decrement*).

Untuk pengulangan yang merubah nilai awal ke angka yang lebih besar, menggunakan operator *++*. Sedangkan untuk mengubah nilainya ke angka yang lebih kecil, maka menggunakan operator *--*.

Contoh *for()* :

```
For(nilai awal; kondisis; penambahan/pengurangan){  
  Pernyataan/perintah;  
}
```

Keterangan:

Hasil program tersebut akan menampilkan nilai a dari 0 sampai 100

d. *goto*

Perintah ini biasanya digunakan untuk melompat secara langsung atau menuju perintah yang telah diberi label. Contoh *goto* :

Nama label perintah/pernyataan

Goto nama label;

Nyala: digitalWrite(pin 1, HIGH);

Goto nyala

e. *return*

Fungsi *return* pada Arduino adalah untuk memberikan nilai balik dari suatu fungsi. Contoh *return* :

```
Int data(){
```

```
  If(analogRead(A0)>100){
```

```
    Return 1;
```

```
  Else
```

```
    Return 0;
```

```
  }
```

```
}
```

f. *continue*

Perintah *continue* umumnya digunakan untuk melewati perulangan yang tersisa dari struktur *looping*. Baik itu *do*, *for*, maupun *while*.

Contoh *continue* pada perulangan *for* :

```
For(nilai awal; kondisi; penambahan/pengurangan){
```

```
Pernyataan/perintah 1
```

```
if(kondisi 2){
```

```
continue;
```

```
}
```

```
Pernyataan/perintah 2
```

```
}
```

Keterangan:

Pada saat program dijalankan, maka nilai b yang awalnya 1 akan terus bertambah hingga mencapai angka 10. Tetapi pada saat nilai b mencapai 5, maka kode program yang ada di bawah continue; akan dilewati dan akhirnya berlanjut ke perulangan 6. Begitu seterusnya.

g. *Break*

Sederhananya, fungsi *break* pada Arduino adalah memaksa sebuah perulangan berhenti sebelum waktunya. Baik itu dalam perulangan *for*, *while*, maupun *do ... while*. Sehingga dapat dikatakan bahwa penggunaan perintah *break* adalah salah satu cara menghentikan *looping* Arduino. Contoh *break* :

```
For(nilai awal; kondisi; penambahan/pengurangan){
```

```
Pernyataan/perintah
```

```
if(kondisi 2){
```

```
break;
```

}

}

Keterangan:

Saat program di atas dijalankan, maka nilai c yang awalnya 1 akan terus bertambah hingga mencapai angka 10. Tetapi pada saat nilai c mencapai 8, program akan berhenti secara otomatis.

11. Format ekspresi bilangan Arduino

Dalam pemrograman Arduino, suatu bilangan dapat diekspresikan dalam beberapa format, yaitu sebagai berikut:

a. Biner

Biasanya ditulis dengan awalan huruf '0b'. Contoh: 0b11110010

b. Desimal

Biasa ditulis tanpa awalan. Contoh: 435

c. Oktal

Umumnya ditulis dengan awalan '0'. Contoh: 0753

d. Heksadesimal

Biasanya diawali dengan '0x'. Contoh: 0x5A

12. Pengaturan waktu (*Time*) Arduino

Ada tiga perintah yang biasa digunakan dalam pemrograman Arduino untuk mengatur waktu, yaitu:

a. *millis()*

Fungsi *millis* pada Arduino yaitu menghasilkan nilai waktu dalam satuan milidetik sejak program Arduino dijalankan. Namun setelah 50 hari dijalankan, nilainya akan kembali lagi ke nol. Tipe data yang sering digunakan untuk perintah ini adalah *unsigned long*. Jadi, pastikan mendefinisikan variabel yang ingin digunakan sebagai penampung nilai *millis()* dengan tipe data *unsigned long*. Contoh *millis* :

```
Variabel = millis();  
Unsigned long waktu;  
Void setup(){  
  Serial.print("Milidetik:");  
  
  Wakyu = millis();  
  Serial.println(waktu);  
}
```

b. *micros()*

Perintah ini berfungsi untuk menghasilkan nilai waktu dalam satuan mikrodetik sejak program mulai dijalankan. Nilai akan kembali lagi ke nol apabila telah mencapai 70 menit. Contoh *micros()* :

```
Variabel = micros();  
  
Unsigned long waktu;  
  
Void setup(){  
  
  Serial.print("Mikrodetik:");  
  
  Wakyu = micros();
```

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

```
}
```

c. *delay()*

Yaitu untuk memberikan tambahan waktu atau penundaan dalam satuan milidetik sebelum mengeksekusi program pada baris selanjutnya. Adapun untuk waktu penundaannya menyesuaikan dengan nilai yang dimasukkan ke perintah. Contoh *delay* :

```
Delay(1000);
```

Keterangan:

Perintah tersebut berfungsi untuk memberikan jeda pada program selama 1000 milidetik atau setara dengan 1 detik.

d. *delayMicroseconds()*

Sama seperti poin sebelumnya, perintah ini juga berfungsi untuk memberikan jeda waktu. Hanya saja, jeda yang diberikan dalam satuan mikrodetik. Dimana 1 juta mikrodetik setara dengan 1 detik.

Jadi, sangat tidak disarankan menggunakan perintah ini bila ingin berpatokan pada satuan detik. Contoh penulisan *delaymicroseconds()*:

```
delayMicrosecond(300);
```

Keterangan:

Perintah tersebut berfungsi untuk menjeda program selama 300 mikrodetik atau setara dengan 0,0003 detik.

13. Operator aritmetika Arduino

Operator aritmetika Arduino yang sering digunakan dapat dilihat pada tabel 2.5

Tabel 2.5 Operator yang sering digunakan

Operator	Keterangan
=	Pemberian nilai
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagian

Dalam pemrograman Arduino, suatu rumus yang dituliskan harus mempunyai variabel penampung. Gunanya untuk menampung nilai hasil perhitungan rumus tadi. Contoh variabel penampung :

Variabel penampung = rumus dan variabel bernilai/nilai konstan;

Luaspersegi panjang = panjang*lebar;

Keterangan:

luasPersegiPanjang = nama variabel penampung

panjang = variabel yang memiliki nilai panjang

luas = variabel yang memiliki nilai luas

14. Akar Arduino

Untuk mencari nilai akar kuadrat dari suatu bilangan, yaitu menggunakan perintah `sqrt()`. Contoh perintah perintah `sqrt()` :

Variabel penampung hasil = `sqrt(bilangan);`

15. Pangkat di Arduino

Dalam bahasa pemrograman Arduino, bisa mencari nilai pangkat dari suatu bilangan. Dalam hal ini kamu bisa menggunakan fungsi `pow()`. Bentuk penulisannya yaitu:

Variabel penampung hasil = `pow(bilangan, pangkat);`

16. Operator perbandingan Arduino

Dalam pemrograman Arduino juga terdapat operator perbandingan. Yaitu operator yang fungsinya membandingkan dua variabel dan akan bernilai 1 atau TRUE jika kondisinya terpenuhi. Beberapa operator perbandingan yang biasa digunakan dapat dilihat pada tabel 2.6

Tabel 2.6 Operator perbandingan yang sering digunakan

Operator	Keterangan
<code>==</code>	Persamaan. Jika kedua nilai yang dibandingkan sama maka hasilnya TRUE
<code>!=</code>	Pertidaksamaan. Jika kedua nilai yang dibandingkan tidak sama maka hasilnya FALSE
<code>></code>	Lebih Besar
<code><</code>	Lebih Kecil
<code>>=</code>	Lebih Besar atau Sama Dengan
<code><=</code>	Lebih Kecil atau Sama Dengan

17. Operator Boolean di Arduino

Boolean adalah sejenis tipe data yang hanya memiliki dua nilai. Yaitu benar atau salah, 1 atau 0, serta TRUE atau FALSE. Tergantung jenis nilai mana yang digunakan. Namun kadang juga operator *boolean* ini digunakan untuk membandingkan atau memanipulasi nilai dalam suatu variabel. Beberapa operator *boolean* yang sering digunakan dapat dilihat pada tabel 2.7

Tabel 2.7 Operator *boolean* yang sering digunakan

Operator	Keterangan
&&	AND
	OR
!	NOT

18. Operator *Bitwise* di Arduino

Operator *bitwise* adalah operator yang digunakan untuk memanipulasi data bertipe bit dari tipe data *integer* atau *byte*. Macam-macam operator *bitwise* yang umum digunakan dapat dilihat pada tabel 2.8

Tabel 2.8 Macam-macam operator *bitwise* yang umum digunakan

Operator	Keterangan
<<	Geser Kiri
>>	Geser Kanan
&	AND
	OR
^	XOR
~	NOT

19. Perintah gabungan dalam pemrograman Arduino

Dalam pemrograman Arduino ada juga yang disebut perintah gabungan. Yaitu perintah yang penulisannya dipersingkat agar bisa lebih simpel dalam penulisannya. Beberapa contoh perintah gabungan dapat dilihat pada tabel 2.9

Tabel 2.9 Beberapa contoh perintah gabungan

Operator	Fungsi	Contoh Penulisan	Keterangan
++	Pertambahan 1/increment	a++	a=a+1
--	Pengurangan 1/decrement	a--	a=a-1
+=	Mempersingkat Penulisan Program	a+=b	a=a+b
-=		a-=b	a=a-b
=		a=b	a=a*b
/=		a/=b	a=a/b
=		a =b	a=a OR b
&=		a&=b	a= a AND b
^=		a^=b	a=a^b

20. Komunikasi Serial Arduino

Pada dasarnya, komunikasi serial adalah komunikasi dua arah yang melibatkan *transmitter* dan *receiver*. Pada Arduino sendiri, *transmitter* dan *receiver*-nya sudah tersedia di pin Rx dan Tx maupun pada USB. Umumnya, komunikasi serial digunakan untuk menampilkan data hasil perhitungan Arduino di serial monitor. Beberapa contoh perintah Arduino yang sering digunakan dalam komunikasi serial antara lain:

a. Serial.begin()

Sebenarnya ini adalah suatu perintah untuk mengaktifkan komunikasi serial dengan nilai *baudrate* atau kecepatan transmisi sebesar 9600 *bps*. Contoh serial.begin() :

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

b. `Serial.print()`

Yaitu suatu perintah yang berfungsi untuk menampilkan data di serial monitor untuk satu baris saja. Jadi, data akan terus bermunculan hanya pada satu baris di serial monitor. Contoh penulisan `Serial.print()` :

```
Serial.print(Tampil);
```

Keterangan:

Tampil adalah variabel penampung nilai yang ingin ditampilkan pada serial monitor.

c. `Serial.println()`

Perintah ini kira-kira fungsinya hampir sama dengan `Serial.print()`. Hanya saja, data yang ditampilkan akan terus bermunculan ke bawah. Jadi data yang muncul tak berada dalam satu baris saja. Ini karena setiap selesai menampilkan data, secara otomatis program akan memberi perintah *enter* atau pindah ke baris berikutnya. Contoh penulisan `serial.println()` :

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

21. *Eksternal Interrupt* Arduino

Salah satu jenis pin yang dimiliki oleh Arduino adalah pin *Interrupt*. Pin ini berfungsi sebagai pemicu untuk meminta perhatian dari prosesor sebagai tanggapan bahwa adanya kekeliruan di dalam program yang dijalankan. Kira-kira prinsip kerjanya sama seperti melakukan interupsi pada saat rapat berlangsung. Jadi saat pin ini bernilai HIGH, maka perintah

yang sedang berjalan akan dihentikan sementara dan perintah pada pin *Interrupt* akan dijalankan.

Tabel 2.10 Jumlah pin interrupt pada beberapa jenis Arduino berbeda

Jenis Papan	Nama Pin Untuk <i>Interrupt</i>
Uno, Nano, Mini, dan seri Atmega328 lainnya	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Mikro, Leonardo, dan seri 32u4 lainnya	0, 1, 2, 3, 7
Zero	Semua pin digital kecuali pin 4
MKR1000 Rev 1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	Semua pin digital
101	Semua pin digital

Adapun untuk perintah pin *interrupt* :

`attachInterrupt (digitalPinToInterrupt (pin), fungsi, mode);`

Keterangan:

Pin = nomor pin *interrupt* yang digunakan

Fungsi = fungsi yang akan dieksekusi

Mode = mode deteksi pin apa yang digunakan

Mode yang bisa digunakan ada empat, yaitu:

- **LOW**, yaitu *interrupt* akan dieksekusi terus menerus selama pin membaca logika LOW
- **RISING**, yaitu *interrupt* akan dieksekusi jika terjadi perubahan nilai pembacaan dari LOW ke HIGH (dieksekusi satu kali saat terjadi perubahan)

- **CHANGE**, yaitu *interrupt* akan dieksekusi bila terjadi perubahan pada nilai yang dibaca. Baik dari HIGH ke LOW maupun dari LOW ke HIGH. (dieksekusi satu kali saat terjadi perubahan).
- **FALLING**, yaitu *interrupt* akan dieksekusi jika terjadi perubahan nilai pembacaan dari HIGH ke LOW. (dieksekusi satu kali saat terjadi perubahan).

Contoh *source code* Arduino tentang interupsi eksternal di bawah ini:

```
Void setup(){  
  Serial.begin(9600);  
  pinMode(2, INPUT_PULLUP);  
  attachInterrupt(digitalPinToInterrupt(2), sisip,FALLING);  
}  
  
Void sisip(){  
  Serial.print("Tombol Ditekan");  
  Delay(1000);  
}  
  
Void loop(){  
  Serial.println("Tes kode 1");  
  Delay(1000);  
}
```

Keterangan:

Jika kode di atas dijalankan, maka di serial monitor akan muncul tulisan "Tes Kode 1" sampai "Tes Kode 4" secara berurutan dengan jeda 1 detik.

Namun ketika button yang terhubung dengan pin 2 ditekan maka di serial monitor akan muncul kata “Tombol Ditekan”.

22. Nilai Maksimum dan Minimum pada Arduino

a. $\min(x, y)$

Fungsi perintah ini adalah untuk membandingkan dua variabel dan akan mengembalikan nilai yang paling kecil lalu menyimpannya dalam suatu variabel. Penulisan $\min(x, y)$:

Baca = $\min(x, y)$

Keterangan:

baca = variabel penampung

x = variabel 1

y = variabel 2

b. $\max(x, y)$

Perintah ini berfungsi untuk membandingkan dua nilai variabel lalu mengembalikan nilai terbesar (Utami, 2022). Bisa dikatakan bahwa perintah ini adalah kebalikan dari perintah $\min(x, y)$. Penulisannya :

Baca = $\max(x, y)$;

Keterangan:

baca = variabel penampung

x = variabel 1

y = variabel 2.

BAB III

METODOLOGI

3.1 Waktu dan tempat

Perancangan ini dilaksanakan di Masjid Uwais Al-Qorni Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan yang berlangsung dari tanggal 1 Januari 2024 sampai dengan 28 September 2024.

3.2 Alat & bahan

Alat yang digunakan dalam perancangan ini dapat dilihat pada tabel 3.1

Tabel 3.1 Alat yang digunakan dalam perancangan

No.	Nama Alat	Jumlah	Satuan	Spesifikasi
1	Gurinda	1	Buah	Mak ita
2	Mesin las inverter	1	Buah	lakoni basic 123 ix
3	Atractor	1	Buah	ED-300
4	Solder 40W	1	Buah	Goot kx 40R
5	Tang Kombinasi	1	Buah	Goot
6	Obeng plus	1	Buah	Sanfix
7	Obeng minus	1	Buah	Sanfix
8	Multimeter Digital	1	Buah	EZREN DT-9205A
9	Pinset	1	Buah	Stainless Steel
10	Laptop	1	Buah	Lenovo
11	Cutter	1	Buah	JENIA
12	Setrika	1	Buah	MIYAKO E 1009 M

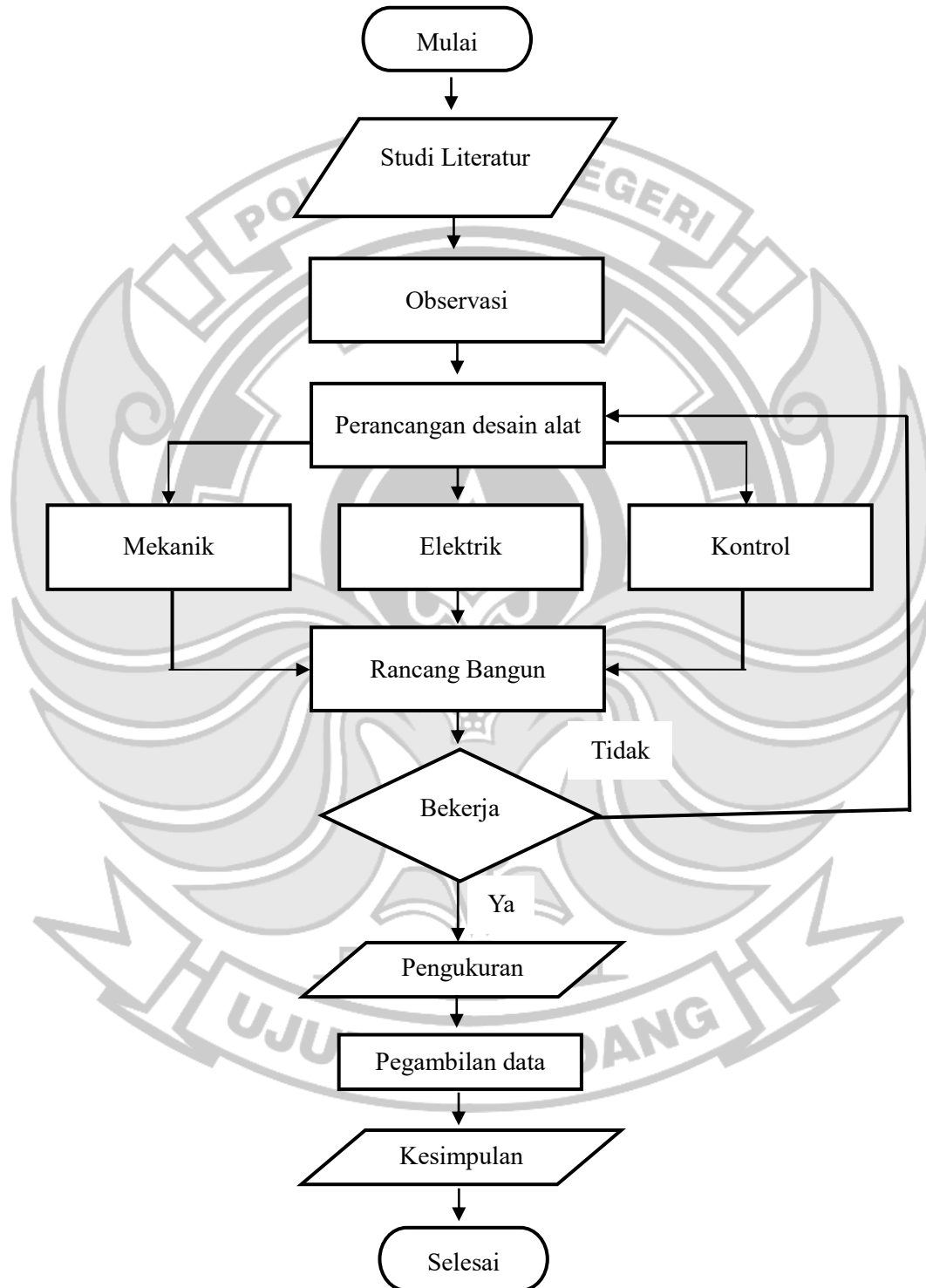
Bahan-bahan yang digunakan dalam perancangan ini dapat dilihat pada tabel 3.2

Tabel 3.2 Bahan-bahan yang digunakan dalam perancangan

No.	Nama Alat	Jumlah	Satuan	Spesifikasi
1	<i>Solar cell</i>	3	Buah	100 wp
2	MPPT	1	Buah	12 V 100 A
3	Kabel PLTS	25	Meter	Twin kabel 2 x 2,5 mm ²
4	Baterai	1	Buah	100 Ah
5	LCD	1	Buah	16 x 2
6	PCB fiber	1	Set	10 x 20 cm
7	ESP8266	1	Buah	<i>Shield</i>
8	NodeMCU ESP8266	1	Buah	V3 lolin lua WIFI CH340
9	Sensor PZEM 017 T	1	Buah	100 A 0-300 VDC
10	Kabel NYAF	10	Meter	0,8 mm
11	USB male to Type-B micro	1	Buah	10 cm
12	Relai	3	Buah	Input 5 Vdc, Output 250 Vac
13	Lampu Sorot DC	1	Buah	20W
14	Lampu Sorot DC	1	Buah	30W
15	Lampu Sorot DC	1	Buah	50W
16	Box X6	1	Buah	PVC hitam
17	Timah	1	Roll	0,8 mm
18	Mata las	¹ / ₂	Kg	Nikko RD 260 2 mm
19	Mur baut	12	Buah	14 mm
20	Sekrup	4	Buah	3 mm

3.3 Tahapan penelitian

Pelaksanaan dan tahapan penelitian ini dapat dilihat pada gambar 3.1



Gambar 3.1 *Flowchart*

3.3.1 Studi Literatur

Tahapan ini merupakan langkah awal sebelum melakukan perancangan, yaitu dengan mencari/mengumpulkan informasi terlebih dahulu mengenai perhitungan dasar serta komponen-komponen yang berkaitan dengan topik yang di bahas.

3.3.2 Observasi Lapangan

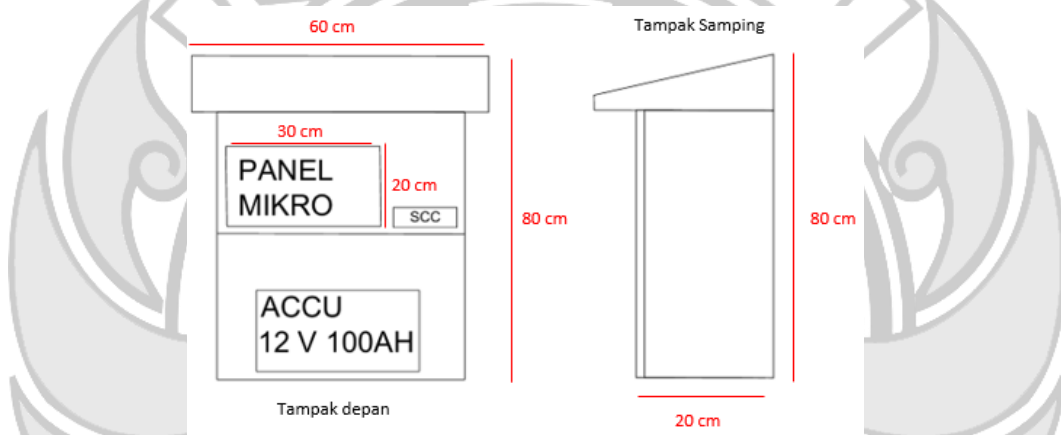
Merupakan proses pengumpulan data/informasi dengan cara melakukan pengamatan/observasi dan pencatatan secara sistematis dan terarah terhadap objek penelitian. Spesifikasi Lampu Sorot DC dapat dilihat pada tabel 3.3

Tabel 3.3 Spesifikasi Lampu Sorot DC

Tipe	PFL5012	PFL5013	PFL5013
V	12Vdc50/60Hz	12Vdc50/60Hz	12Vdc50/60Hz
W	20Watt	30Watt	50Watt
Usia	25.000 Jam	25.000 Jam	25.000 Jam
CCT	Daylight 6500k	Daylight 6500k	Daylight 6500k
Lumen	980Lm	1470Lm	2450Lm
IP	IP66	IP66	IP66

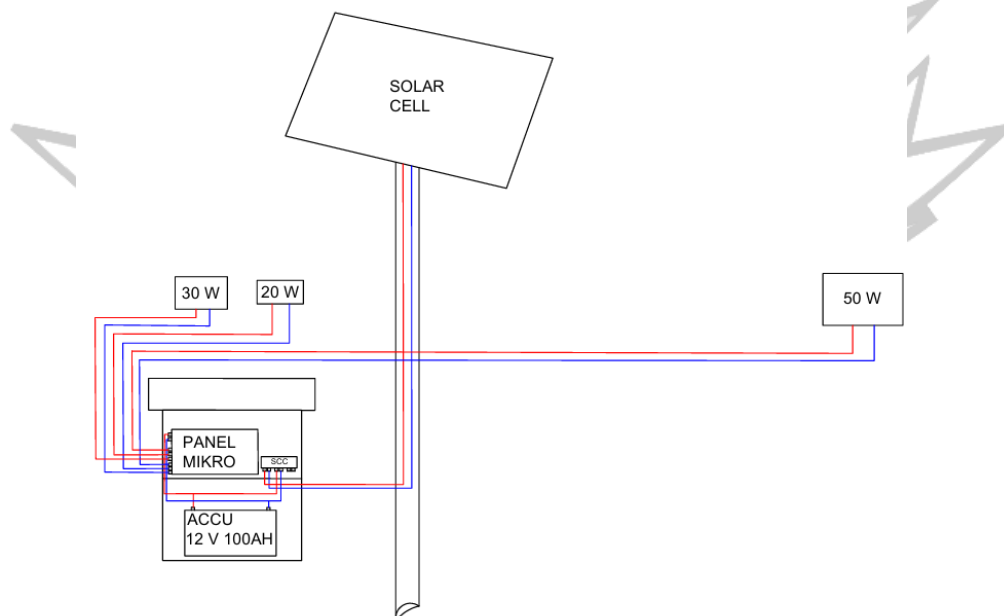
3.3.3 Perancangan Desain Alat

Pada tahap ini merupakan tahap untuk mendesain/membuat gambaran *monitoring* arus dan tegangan beban PLTS berupa lampu serta menentukan alat dan bahan yang diperlukan. Perancangan mekanik alat tampak depan & samping dapat dilihat pada gambar



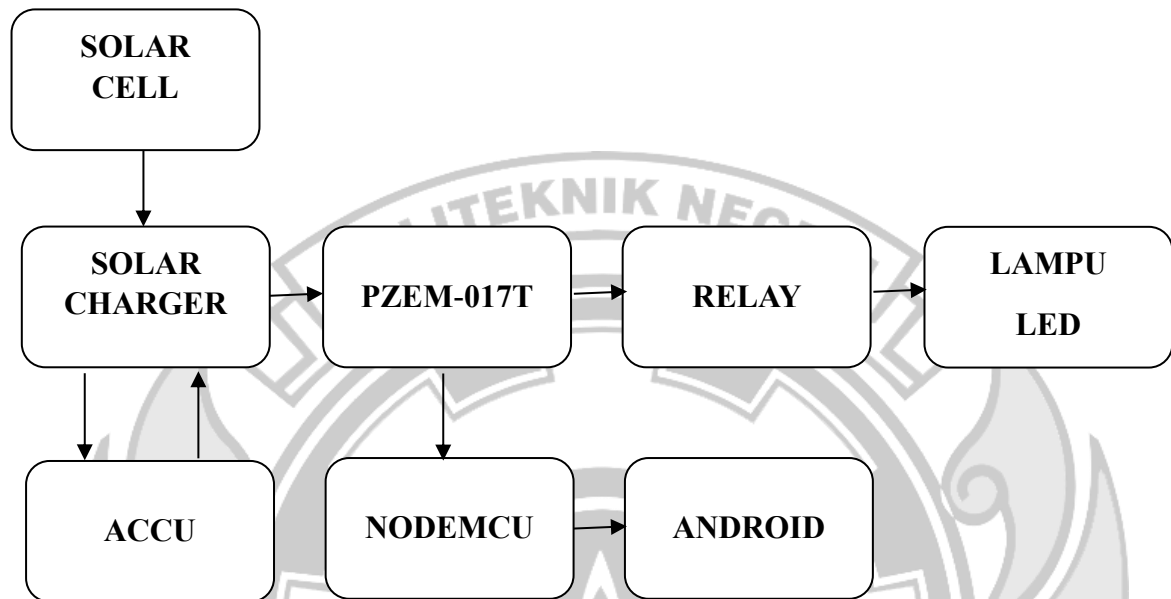
Gambar 3.2 Perancangan mekanik alat tampak depan & samping

Perancangan *wiring diagram* dapat dilihat pada gambar 3.3



Gambar 3.3 Perancangan *wiring diagram* sistem ESP8266

Perancangan elektrik Alat dapat dilihat pada gambar 3.4



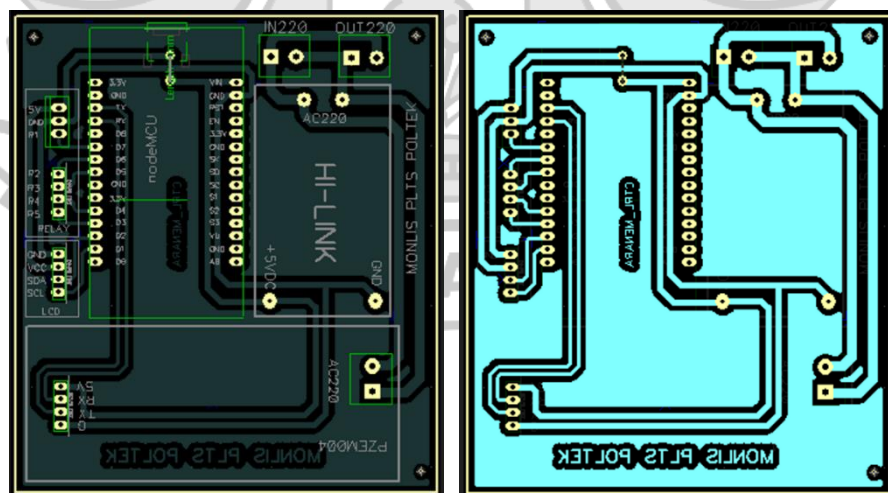
Gambar 3.4 Perancangan elektrik Alat

- 1) Panel surya mentransfer energi dari panas matahari menjadi energi listrik, digunakan sebagai sumber energi listrik.
- 2) *Solar Charger Control* digunakan untuk mengontrol daya pada accu serta meneruskannya ke beban.
- 3) Accu digunakan untuk penyimpanan energi yang telah di serap dari matahari
- 4) *Relay* digunakan untuk mengonrol nyala/mati lampu sorot LED
- 5) Lampu sorot LED digunakan sebagai beban PLTS sekaligus untuk penerangan
- 6) Pzem-017T berfungsi untuk membaca arus dan tegangan
- 7) NodeMCU digunakan untuk menjalankan fungsi mikrokontroler dan juga koneksi internet (Wifi)
- 8) Aplikasi Blynk digunakan untuk menampilkan *Button* kontrol lampu serta menampilkan data hasil *monitoring* arus dan tegangan

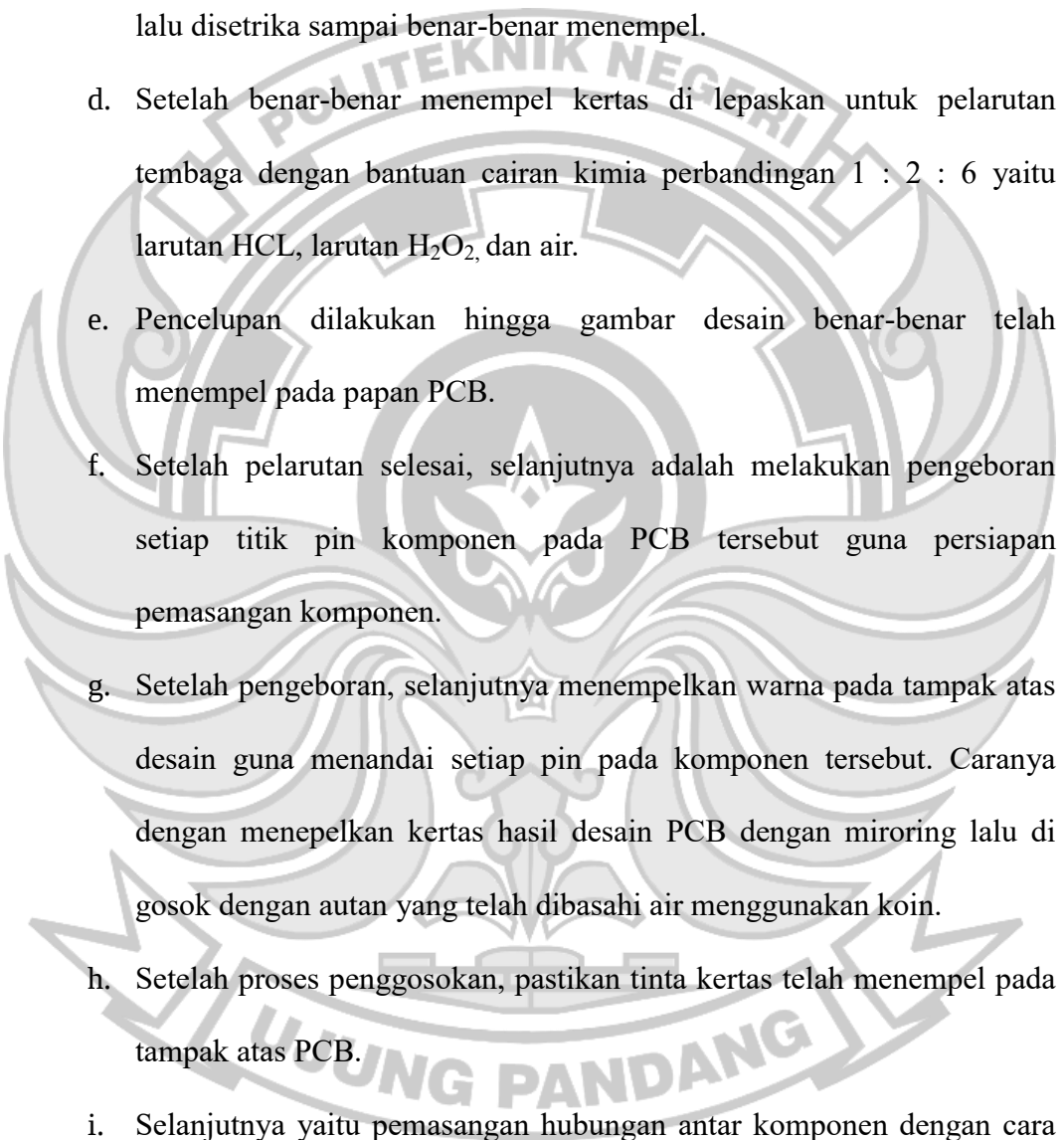
Setelah selesai membuat desain rancangan mekanik dan elektrik, selanjutnya ialah membuat suatu rancangan kontrol berupa rancangan PCB & panel sistem ESP8266.

Adapun langkah-langkah pembuatan PCB & panel sistem ESP8266 sebagai berikut :

1. Merancang desain blok diagram sistem.
2. Menentukan komponen-komponen yang dibutuhkan berdasarkan blok diagram yang telah dirancang sebelumnya.
3. Setelah komponen telah ditentukan, kemudian kita membuat rancangan PCB berupa papan rangkaian yang akan menghubungkan antara setiap pin-pin pada komponen yang digunakan. Langkah-langkahnya yaitu :
 - a. Membuat desain Layout PCB menggunakan aplikasi DipTraceLauncher. Tampak atas & tampak bawah PCB dapat dilihat pada gambar 3.5

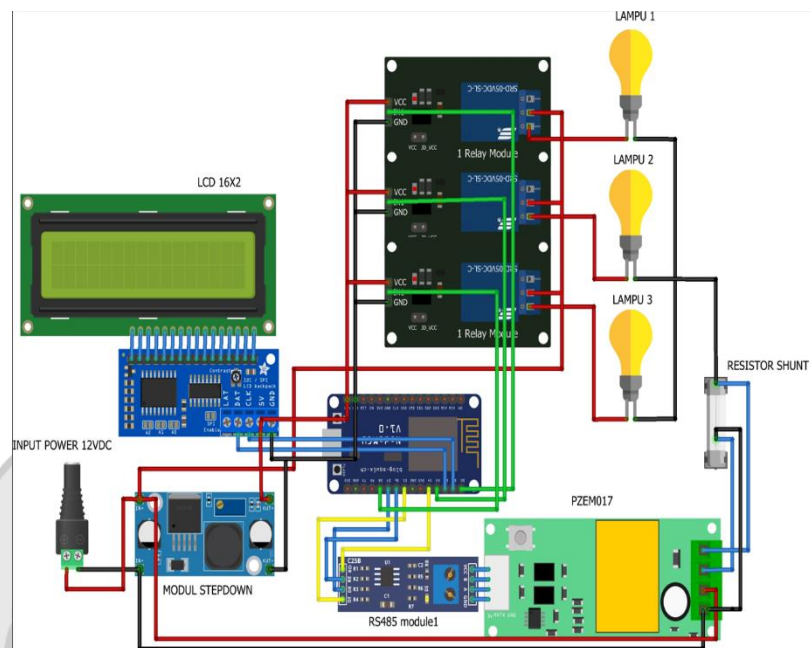


Gambar 3.5 Tampak atas & tampak bawah PCB

- 
- b. Mencetak desain PCB menggunakan printer laser menggunakan proses pemanasan dengan strika.
 - c. Kertas desain yang telah dicetak ditempelkan pada papan PCB polos lalu disetrika sampai benar-benar menempel.
 - d. Setelah benar-benar menempel kertas di lepaskan untuk pelarutan tembaga dengan bantuan cairan kimia perbandingan 1 : 2 : 6 yaitu larutan HCL, larutan H₂O₂, dan air.
 - e. Pencelupan dilakukan hingga gambar desain benar-benar telah menempel pada papan PCB.
 - f. Setelah pelarutan selesai, selanjutnya adalah melakukan pengeboran setiap titik pin komponen pada PCB tersebut guna persiapan pemasangan komponen.
 - g. Setelah pengeboran, selanjutnya menempelkan warna pada tampak atas desain guna menandai setiap pin pada komponen tersebut. Caranya dengan menempelkan kertas hasil desain PCB dengan miroring lalu digosok dengan autan yang telah dibasahi air menggunakan koin.
 - h. Setelah proses penggosokan, pastikan tinta kertas telah menempel pada tampak atas PCB.
 - i. Selanjutnya yaitu pemasangan hubungan antar komponen dengan cara di solder.

4. Selanjutnya merancang wiring diagram sistem pada aplikasi fritzing

Schematic Sistem monitoring & kontrol ESP8266 dapat dilihat pada gambar



Gambar 3.6 *Schematic Sistem monitoring & kontrol ESP8266*

5. Selanjutnya kita melakukan instalasi peralatan secara keseluruhan dengan menghubungkan antara kelistrikan, memasang LCD, sensor, ESP8266, relay, dan terminal input sumber dan terminal output sumber.
6. Setelah merakit instalasi secara keseluruhan, selanjutnya dialah menempelkan komponen ke dalam panel menggunakan lem tembak dan baut.
7. Selanjutnya ialah membuat kode program.

3.3.4 Rancang Bangun Alat

Pada tahap kali ini, setelah semua alat dan bahan sudah disiapkan maka alat akan dirakit sesuai dengan desain yang telah dibuat.

Berikut langkah-langkah perakitan alat *monitoring & kontrol* beban PLTS berupa lampu sorot LED :

1. Pemasangan panel box

Pemasangan panel box dapat dilihat pada gambar 3.7



Gambar 3.7 Pemasangan panel box

2. Pemasangan sumber PLTS ke SCC

Pemasangan sumber PLTS ke SCC dapat dilihat pada gambar 3.8



Gambar 3.8 Pemasangan sumber PLTS ke SCC

3. Pemasangan hubungan antara panel mikrokontroler, SCC, dan accu

Pemasangan hubungan antara panel mikrokontroler, SCC, dan accu dapat dilihat pada gambar 3.9



Gambar 3.9 Pemasangan hubungan antara panel mikrokontroler, SCC, dan accu

4. Pemasangan beban dengan sitem *monitoring* & kontrol ESP8266

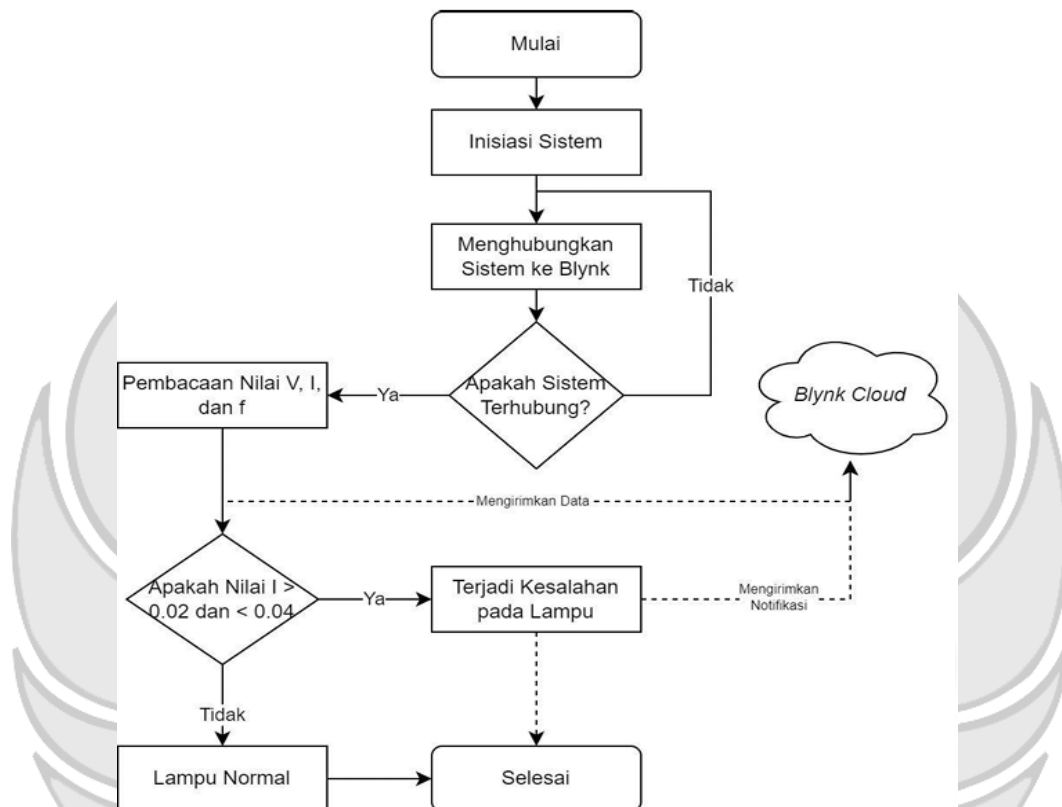
Pemasangan beban dengan sitem *monitoring* & kontrol ESP8266 dapat dilihat pada gambar 3.10



Gambar 3.10 Pemasangan beban dengan sitem *monitoring* & kontrol ESP8266

3.3.5 Alat Bekerja

Skema *Monitoring* dapat dilihat pada gambar 3.11



Gambar 3.11 Skema *Monitoring*

Setelah monitoring arus dan tegangan dirangkai, selanjutnya adalah tahap pengujian. Tahap ini dilakukan untuk memastikan alat bekerja sesuai dengan yang direncanakan. Setelah alat bekerja dengan baik, maka dapat dilakukan pengambilan data yang dibutuhkan. Akan tetapi apabila alat Error tidak sesuai dengan yang direncanakan maka harus dilakukan pengecekan ulang pada bagian instalasi alat. Hal ini dilakukan beberapa kali untuk mendapatkan hasil yang terbaik.

3.3.6 Pengambilan Data

Dalam pengambilan data, dapat dilihat apakah *monitoring* arus dan tegangan bekerja dengan baik serta lampu LED aktif. Data yang akan diambil adalah arus dan tegangan, selanjutnya dari data tersebut akan digunakan sebagai bahan untuk melakukan analisis dan juga mengambil kesimpulan dari penelitian/perancangan alat ini. Adapun metode pengujian yang digunakan ialah metode *black box* yang dimana tujuan nya ialah membandingkan nilai input dan output yang diperoleh dari pengambilan data. Penggambaran metode *black box* dapat dilihat pada gambar 3.12



Gambar 3.12 Metode *black box*

Salah satu contoh teknik metode pengujian *black box* ialah teknik *decision table*. Teknik ini digunakan karena terdapat dua input atau kondisi berbeda yang saling berhubungan, dan *output*-Nya tergantung pada kesesuaian hubungan tersebut.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Bab ini berisi tentang hasil dan pembahasan Sistem *Monitoring* dan Kontrol beban PLTS menggunakan ESP8266 berbasis IoT, yang terdiri dari menentukan kebutuhan komponen, pengujian *solar cell*; Accu; SCC, hasil perancangan *software*, dan pembahasan. Hasil pengujian berupa data yang diperoleh untuk memperlihatkan kemampuan perangkat keras dan perangkat lunak yang dirancang dapat bekerja dengan baik atau tidak. Dari data tersebut dapat dilakukan analisis kinerja perangkat secara keseluruhan berdasarkan pengujian dari masing-masing subsistem untuk mendapatkan hasil yang lebih akurat serta kemudian dapat digunakan untuk mengambil kesimpulan akhir terhadap perangkat tersebut.

4.1.1 Menentukan kebutuhan Komponen

Untuk menentukan berapa jumlah PLTS yang digunakan, terlebih dahulu menentukan total daya beban yang digunakan x waktu pengoperasian beban . Beban yang digunakan yaitu 3 lampu DC masing-masing 20 watt, 30 watt, dan 50 watt dengan total daya beban yaitu $100 \text{ watt} \times 12 \text{ jam} = 1200 \text{ wh}$. Secara teori PLTS hanya menghasilkan 40% output energi listrik. Maka dari itu, untuk mengetahui berapa jumlah energi yang dihasilkan oleh PLTS untuk memenuhi kebutuhan beban 1200 wh dapat di rumuskan :

Total beban : 60% (1)

$$1200 \text{ wh} : 60\% = 2000 \text{ wh}$$

Penyinaran di Indonesia pada umumnya paling tinggi berlangsung 5 jam/hari yaitu dari pukul 10.00 pagi sampai 15.00 sore. Untuk menentukan jumlah PLTS yang akan digunakan terlebih dahulu menentukan daya puncak yang dihasilkan panel surya :

Total daya : waktu penyinaran optimal (2)

$$2000 \text{ wh} : 5 \text{ jam} = 400 \text{ watt}$$

Dari hasil perhitungan, daya puncak yang dihasilkan PLTS untuk memenuhi kebutuhan daya 2000 wh dalam 5 jam di dapatkan nilai 400 wP. Dari hasil tersebut kita dapat menentukan jumlah panel yang dibutuhkan yaitu 4 buah PLTS 100wP.

Dari penjelasan di atas di ketahui bahwa beban lampu DC beroperasi selama 12 jam. Maka dari itu kita memerlukan cadangan energi di malam hari saat dimana PLTS tidak dapat menghasilkan energi listrik. Adapun solusinya adalah dengan menggunakan baterai. Adapun untuk menentukan cadangan baterai/aki sesuai kebutuhan beban yang digunakan :

Cadangan = Daya beban : (100% - 5%) (3)

$$= 1200 \text{ wh} : 95\%$$

$$= 1263 \text{ wh}$$

Dari hasil energi yang keluaran yang didapatkan, kita dapat menentukan jumlah baterai yang digunakan. Misalkan kapasitas baterai yang digunakan 12V x 100Ah maka :

$$\text{Jumlah baterai} = \text{keluaran baterai} : \text{kapasitas baterai} \quad (4)$$

$$= 1263 \text{ wh} : (12\text{V} \times 100\text{Ah})$$

$$= 1263 \text{ wh} : 1200 \text{ wh}$$

$$= 1 \text{ baterai}$$

Karena baterai akan rusak jika di kuras secara berlebihan, maka kita perlu menyisakan 50% isi dari baterai. Sehingga jumlah baterai yang dibutuhkan dikalikan 2. Maka jumlah baterai yang digunakan adalah 2 baterai 12 V 100Ah.

Kemudian untuk menentukan Ampere dari SCC adalah dengan melihat spesifikasi pada panel surya. Adapun parameter yang di butukan yaitu Isc-Nya. Contoh misalkan Isc-Nya 5,89A maka dapat di hitung :

$$\text{Ampere SCC} = \text{Isc} \times \text{Jumlah panel surya} \quad (6)$$

$$= 5,89\text{A} \times 4 \text{ Pcs}$$

$$= 23,56 \text{ A}$$

Dari hasil Ampere SCC yang di dapatkan maka SCC yang di butuhkan yaitu memiliki nilai arus 30 A.

4.1.2 Pengujian *Solar Cell* , Accu, dan SCC

Pengujian panel surya, aki, dan SCC bertujuan untuk mengecek apakah panel surya dapat menyerap energi matahari dengan baik sesuai dengan fungsinya, mengecek tegangan pada aki juga SCC. Dalam pengujian ini menggunakan AVO meter yang di hubungkan dengan polaritas positif *pada solar cell*, aki, SCC masing-masing ke positif AVO begitu pula dengan polaritas negatif nya. Setelah pengujian di lakukan dan memperoleh hasil pada tampilan AVO meter menunjukan tegangan yang naik apabila panel surya terkena sinar matahari maka

panel surya di pastikan dapat digunakan untuk menyuplai energi pada rangkaian yang di buat. Pengujian *Solar cell*, Accu, SCC dapat dilihat pada tabel 4.1

Tabel 4.1 Pengujian *Solar cell*, Accu, SCC

NO	HASIL	KETERANGAN
1	19,25 V	Tegangan <i>solar cell</i> – SCC
2	12,95 V	Tegangan SCC – Accu
3	12,97 V	Tegangan Accu – Beban DC

Dari hasil pengukuran diatas dapat ditarik kesimpulan bahwa tegangan yang mengalir pada rangkaian sesudah SCC mengalami penurunan dan cenderung stabil pada nilai $DC \pm 12$ V. Hal ini dikarenakan fungsi SCC adalah untuk proteksi Accu dengan menjaga tegangan yang masuk bernilai 12 V atau mendekati 12 V.

4.1.3 Hasil perancangan *Software*

Pada perancangan *software*, terlebih dahulu menginstall aplikasi yang di butuhkan. Adapun aplikasi yang digunakan untuk program komputer/aptop yaitu Arduino IDE dan aplikasi pada HP yang digunakan yaitu Blynk Iot. Berikut merupakan langkah – langkah perancangan *software* :

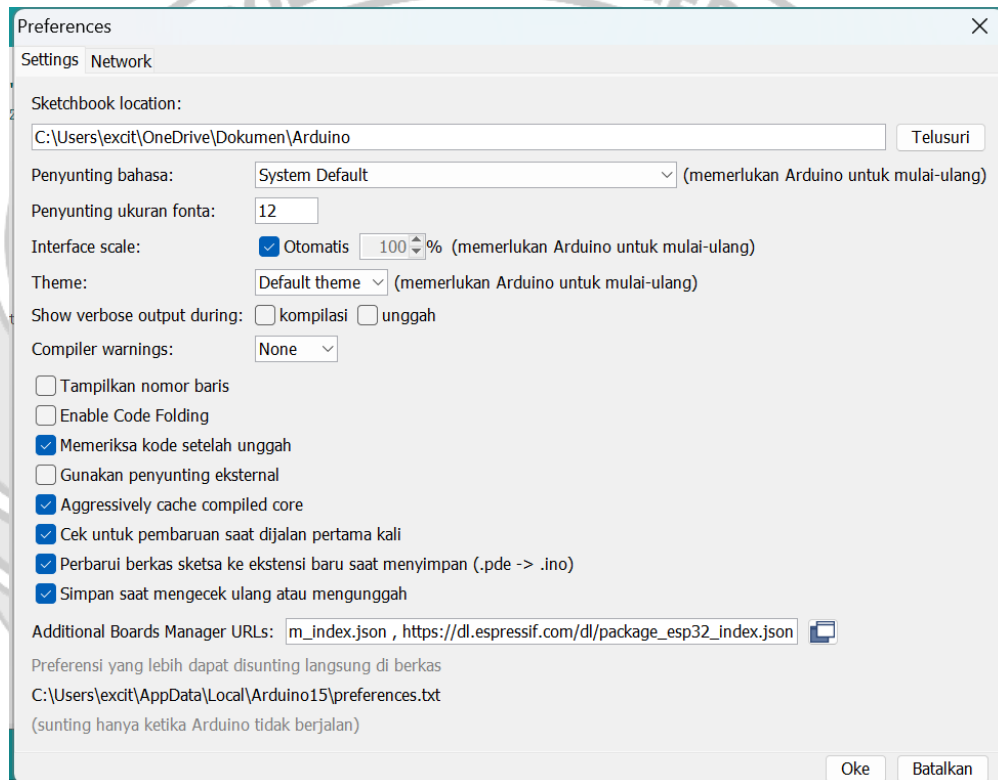
➤ Langkah-langkah koding pada software Arduino IDE

1. Menginstall software Arduino IDE. Klik link download :

<https://drive.google.com/file/d/1jehIUvI0yCeDloisoJFLEzKDDuR2Rb0o/view?usp=drivesdk>

2. Setelah menginstall software Arduino IDE, selanjutnya lakukan beberapa setting berikut :

- a. Mendownload library ESP32 dengan cara pada menu berkas lalu klik *preferences*. Tampilan *preferences* dapat dilihat pada gambar 4.1



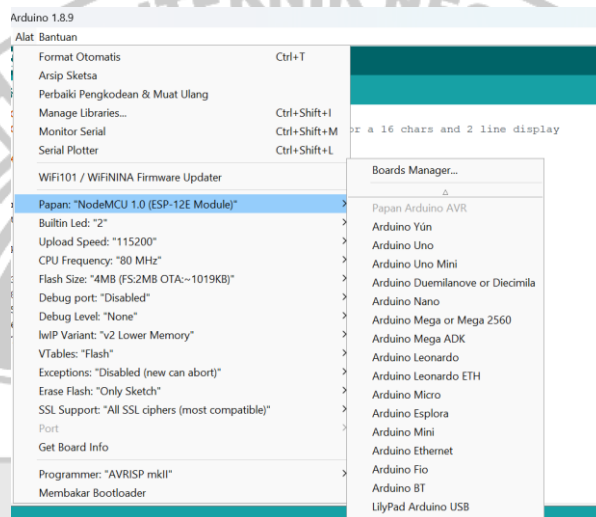
Gambar 4.1 Tampilan *preferences*

Pada bagian pengisian Additional Boards Manager URLs, masukkan link berikut :

http://arduino.esp8266.com/stable/package_esp8266com_index.json ,

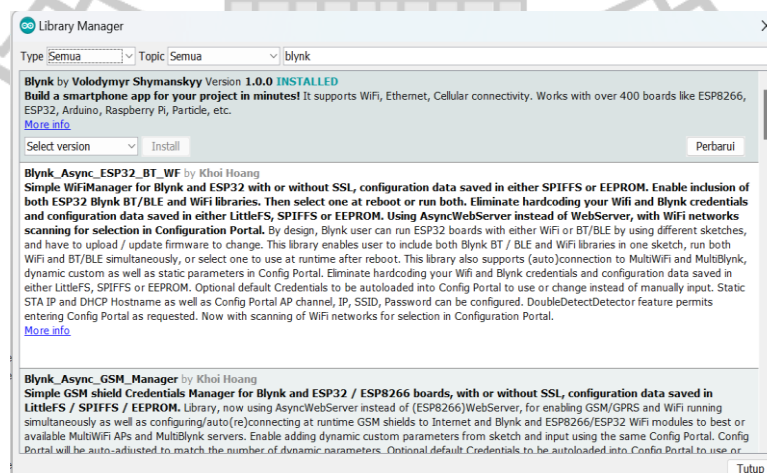
https://dl.espressif.com/dl/package_esp32_index.json

- b. Selanjutnya memilih papan yang ingin digunakan dengan cara klik menu alat lalu pada bagian papan scroll dan pilih papan. tampilan menu Alat dapat dilihat pada gambar 4.2



Gambar 4.2 tampilan menu Alat

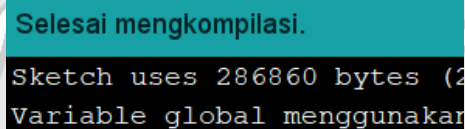
- c. Kemudian menginstal *library blynk* dengan cara klik menu *sketch* lalu pilih *include library* lalu akan muncul pilihan disebelah kanan pilih *manage library*. Tampilan *manage library* dapat dilihat pada gambar 4.3



Gambar 4.3 Tampilan *manage library*

Pada pencarian ketik *blinky* lalu enter, setelah muncul pilih blinky versi terbaru kemudian install.

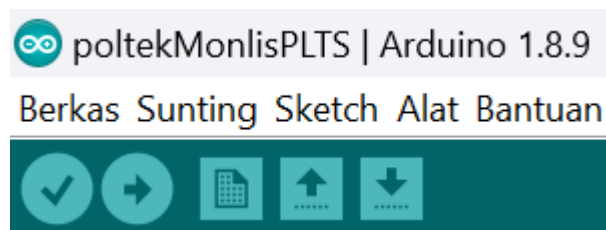
3. Setelah menginstall semua *library* yang diperlukan, selanjutnya ialah menginput coding data ke dalam *software* Arduino IDE. keterangan : pada kode blinky silahkan mengcopy kode yang diberikan oleh aplikasi blinky setelah mendesain template yang disediakan. Kemudian pada kode password disesuaikan dengan username dan password masing-masing.
4. Setelah mengcoding program selanjutnya memeriksa error program dengan cara klik tanda centang.
5. Tunggu beberapa sampai muncul pemberitahuan selesai kompilasi. Notifikasi selesai kompilasi dapat dilihat pada gambar 4.4



Selesai mengkompilasi.
Sketch uses 286860 bytes (2
Variable global menggunakan

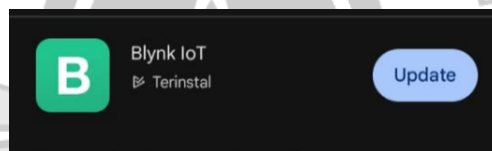
Gambar 4.4 Notifikasi selesai kompilasi

6. Setelah kompilasi sukses, hubungkan laptop dengan ESP8266 dengan perantara kabel USB. Lalu upload coding ke dalam modul wifi ESP8266 dengan cara klik tanda panah kekanan. Tampilan menu *Software* Arduino IDE dapat dilihat pada gambar 4.5



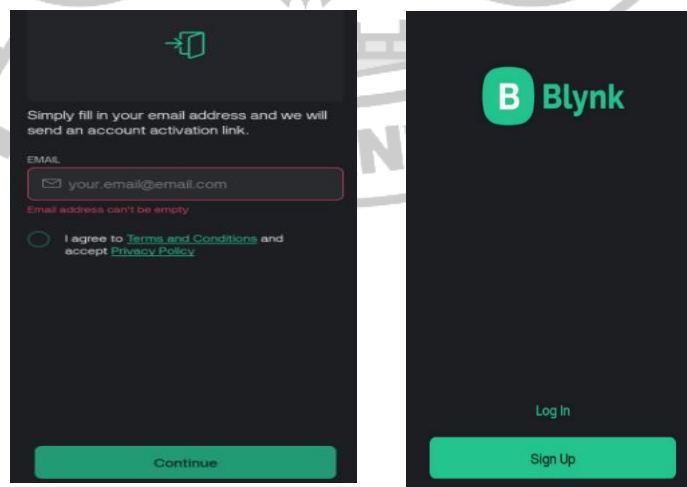
Gambar 4.5 Tampilan menu *Software* Arduino IDE

7. Tunggu sampai *upload* selesai.
 8. Setelah berhasil upload, selanjutnya buka aplikasi blynk klik desain yang telah di buat. Nyalakan tethering hotspot dan tunggu sampai lampu hijau di pojok kanan tampilan desain menyala(terhubung).
- Langkah-langkah mendesain template blynk IoT
1. Buka playstore pada HP android anda lalu pada pencarian ketik blynk
 2. Lalu pilih aplikasi blynk dan install aplikasi-Nya. Tampilan Blynk IoT dapat dilihat pada gambar 4.6



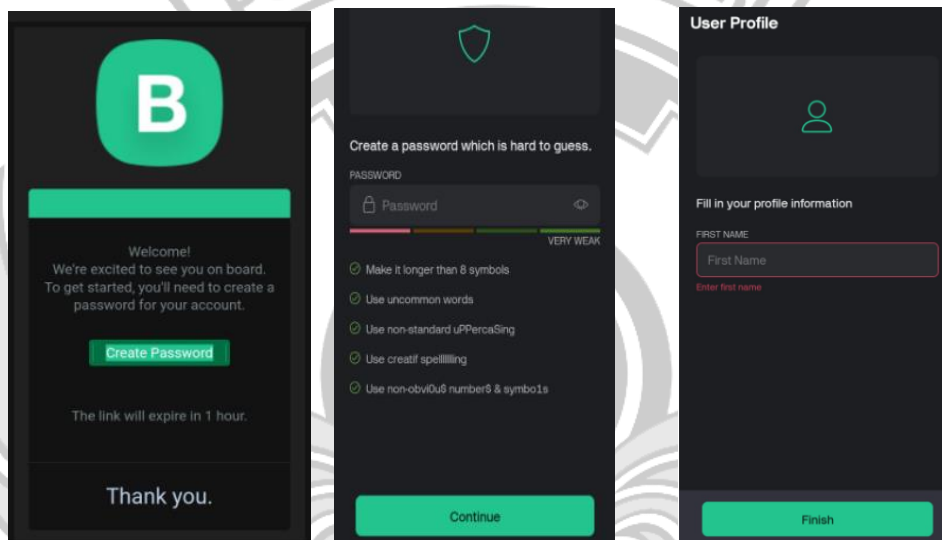
Gambar 4.6 Tampilan Blynk IoT

3. Setelah aplikasi ter-install, buka aplikasi lalu pilih menu *sign up* kemudian daftarkan email. Tampilan Registrasi Blynk IoT dapat dilihat pada gambar 4.7



Gambar 4.7 Tampilan Registrasi Blynk IoT

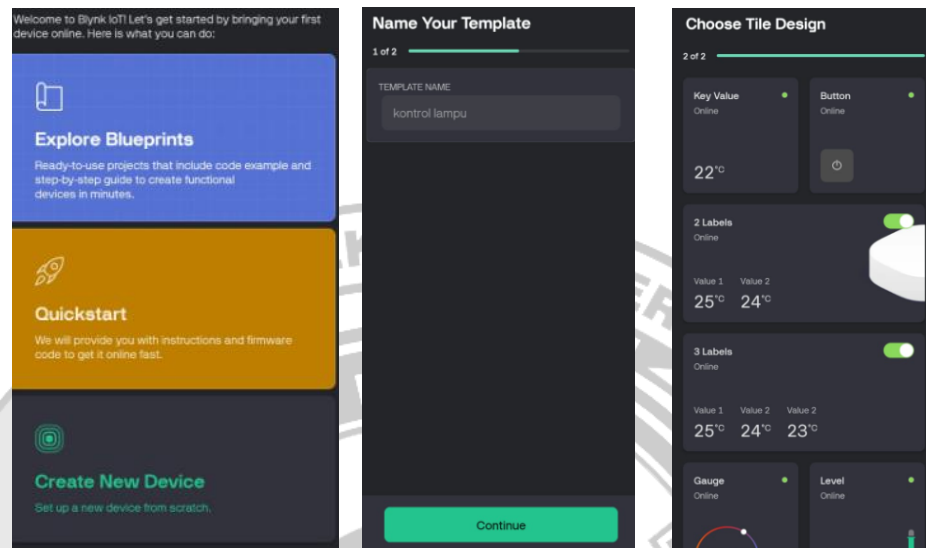
4. Setelah itu tunggu notifikasi dari blynk yang masuk ke email.
5. setelah notifikasi dari blynk masuk ke email, klik notifikasi. Proses pembuatan akun Blynk IoT dapat dilihat pada gambar 4.8



Gambar 4.8 Proses pembuatan akun Blynk IoT

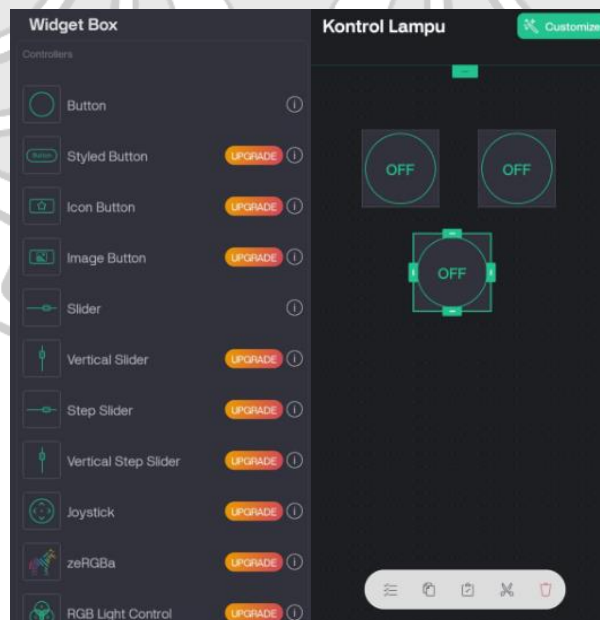
Buat *password* yang aman dan mudah di ingat. Kemudian buat nama pengguna.

6. setelah membuat *username* dan *password*, kemudian memasuki tampilan beranda. Pada tampilan beranda, klik *create new devise*. Kemudian berikan nama proyek lalu mendedign *button*. Tampilan beranda dan penamaan proyek dapat dilihat pada gambar 4.9



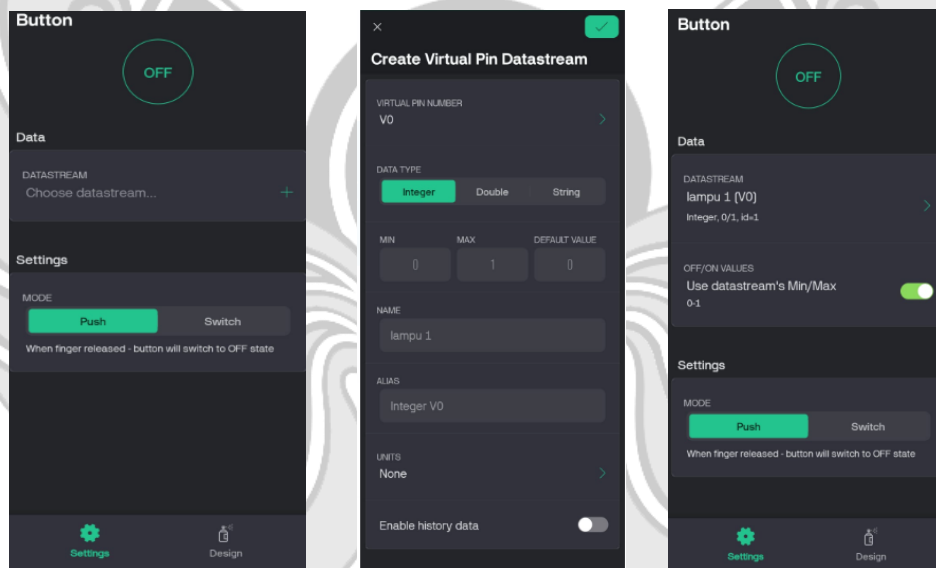
Gambar 4.9 Tampilan beranda dan penamaan proyek

7. selanjutnya akan muncul tampilan sebagai berikut. Klik tanda + berwarna hijau lalu pilih button sebanyak 3. Jika ingin mengatur posisi button klik tahan lalu geser sesuai keinginan. Proses Setting Button dapat dilihat pada gambar 4.10



Gambar 4.10 Proses Setting Button

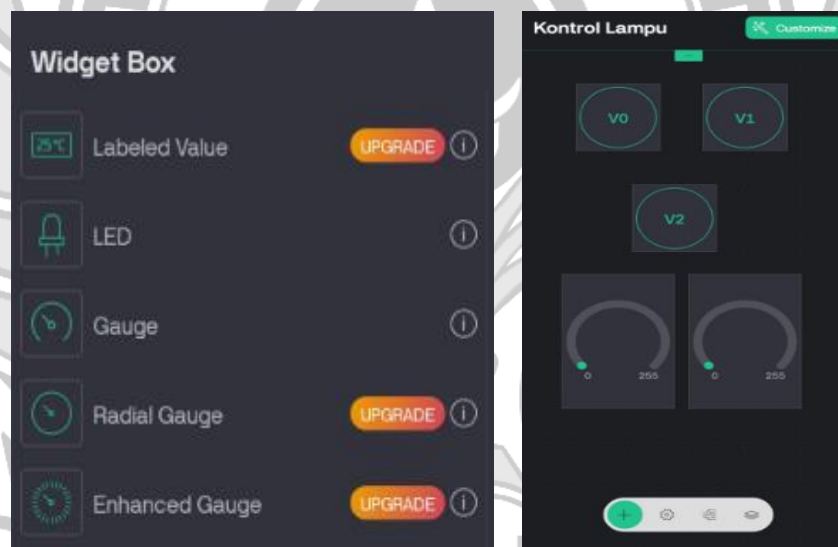
8. Selanjutnya untuk menghubungkan button dengan ESP8266, masukkan kode yang berasal dari program ke *button* dengan cara klik satu kali *button* maka akan muncul sebuah tampilan. Pilih settingan *switch* atau *button* sesuai keinginan lalu klik data *stream* kemudian klik *create virtual pin data stream*. Lalu ikuti setting gambar di bawah. Setelah selesai setting, klik tanda centang. Contoh menginput kode program ke template *button* dapat dilihat pada gambar 4.11



Gambar 4.11 Menginput kode program ke template *button*

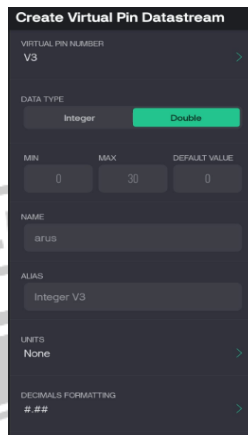
9. Selanjutnya ialah klik tanda x pada pojok kiri atas lalu ulangi hal yang sama seperti langkah 8 hanya saja yang perlu di perhatikan adalah pada *button* kedua isikan nama lampu 2, virtual pin number v1 dan pada button ketiga isikan nama lampu 3, virtual pin number v2.

10. Selanjutnya ialah membuat kreasi desain warna, besar tulisan dan lain-lain dengan cara klik satu kali button lalu pilih desain. Selanjutnya buatlah desain sesuai kreatifitas.
11. Selanjutnya ialah membuat parameter tegangan dan arus. Dengan cara klik tanda + berwarna hijau lalu *scroll* kebawah cari *template* dengan nama *gauge*. Lakukan sebanyak dua kali. Setting posisi *template* monitoring sesuai keinginan. Contoh membuat *Template* parameter Indikator dapat dilihat pada gambar 4.12



Gambar 4.12 Membuat *Template* parameter Indikator

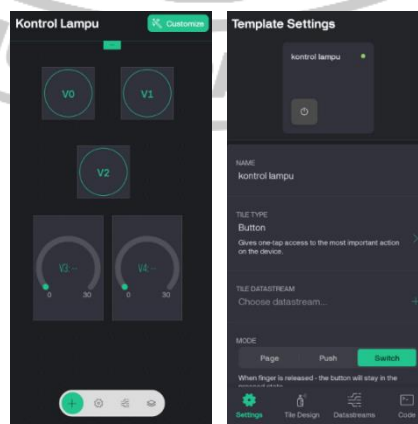
12. Selanjutnya ialah memasukkan kode program ke *template monitoring* arus. Dengan cara mengeklik satu kali *template monitoring* lalu klik *choose datastream* lalu pilih *create new* lalu setting sesuai dengan gambar berikut. Lalu klik tanda centang. Contoh menginput kode program ke *template* Parameter Indikator Arus dapat dilihat pada gambar 4.13



Gambar 4.13 Menginput kode program ke template Parameter Indikator Arus

13. Selanjutnya ialah memasukkan kode program ke *template monitoring* tegangan. Adapun caranya sama dengan langkah 12 hanya saja yang berbeda ialah *name template* nya ialah tegangan, virtual pin *number*-Nya ialah V4.

14. Selanjutnya ialah melakukan *setting* desain pada *template* arus dan tegangan dengan cara klik satu kali pada *template* lalu klik desain. Buatlah desain sesuai keinginan dan kreatifitas masing-masing. Contoh Menginput kode program ke template Parameter Indikator Tegangan dapat dilihat pada gambar 4.14



Gambar 4.14 Menginput kode program ke template Parameter Indikator Tegangan

15. Langkah terakhir ialah mengambil kode alamat program dari blynk IoT untuk dimasukkan ke dalam program Arduino IDE. Dengan cara kembali ke tampilan desain seperti pada gambar di bawah
16. Kemudian klik di samping tanda + berwarna hijau dengan simbol setting. Setelah muncul tampilan baru, klik kode yang berada di pojok kanan bawah, lalu akan muncul kode akses dari blynk yang kemudian di *copy* ataupun di *share* lalu di masukkan ke dalam kode program.

Hal yang perlu di perhatikan :

Untuk desain secara gratis developer blynk IoT membatasi jumlah slot *template* yang bisa digunakan tidak bisa lebih dari 5 buah.

- Langkah-langkah pengoperasian panel sistem *monitoring* dan kontrol beban PLTS :
1. *Upload* koding dari *software* arduino IDE ke ESP8266
 2. Hubungkan sumber listrik DC dari keluaran sistem PLTS ke panel *monitoring* dan kontrol beban PLTS
 3. Aktifkan tethering hotspot HP ataupun wifi yang tercantum pada program yang telah di buat lalu buka aplikasi blynk IoT. Perhatikan tanda titik pada desain *template* yang telah di buat, jika menyala hijau berarti hotspot/wifi telah terhubung dengan ESP8266 dan aplikasi blynk IoT.

4.2 Pembahasan

Sebelum melakukan pengambilan data, terlebih dahulu melakukan pengujian kinerja *software* terhadap sistem kerja dari alat yang dibuat. Adapun metode pengujian yang digunakan ialah metode *black box* dengan menggunakan teknik *decision table* karena terdapat dua input atau kondisi berbeda (modul wifi ESP8266 dan sensor arus, tegangan, dan daya pzem 017T) yang saling berhubungan, dan output-Nya tergantung pada kesesuaian hubungan tersebut.

1. Pengujian *Ethernet Shield* ESP8266

Ketika alat dinyalakan ESP8266 akan mencari koneksi wifi sebelum lanjut ke proses yang lain, selama belum terkoneksi ESP8266 tidak akan menerima dan mengirimkan data dari atau ke komponen mikrokontroler yang lain. Adapun skenario dan hasil dari pengujian *Ethernet Shield* ESP8266 dengan metode *blackbox testing* sesuai kebutuhan sistem disajikan oleh tabel 4.2

Tabel 4.2 *Blackbox testing* pada *Ethernet Shield* ESP8266

Butir Uji	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan
Terhubung ke internet	<i>Ethernet shield</i> ESP8266 terkoneksi dengan internet.	<i>Ethernet shield</i> ESP8266 terhubung ke internet, dapat menerima dan mengirim serta menampilkan data	☒ Berhasil ☐ Gagal

2. Pengujian menu kontrol

Pada menu kontrol terdapat *button on/off* untuk mengendalikan masing-masing Lampu sorot LED. *User* melakukan pengujian untuk menu ini

terutama untuk bagian pengendalian lampu LED sebagai salah satu poin utama dari penelitian ini yaitu *on/off* lampu LED. Hasil dari percobaan *user* nyata bahwa dapat melakukan pengendalian lampu pada setiap lampu LED dan dapat melihat hasil status dari pengendalian lampu LED tersebut. Beberapa butir uji dibawah ini akan menjabarkan mengenai menu kontrol dengan metode *blackbox testing* yang ditunjuk pada tabel 4.3

Tabel 4.3 *Blackbox testing* menu kontrol

Butir Uji	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan
Menghidupkan Lampu 1	Mengklik <i>button</i> “OFF” untuk menyalakan Lampu 1	Dapat mengklik <i>button</i> “OFF” dan Lampu 1 menyala	☒ Berhasil ☐ Gagal
Mematikan Lampu 1	Mengklik <i>button</i> “ON” untuk mematikan Lampu 1	Dapat mengklik <i>button</i> “ON” dan Lampu 1 padam	☒ Berhasil ☐ Gagal
Status Lampu 1	Menampilkan status Lampu1	Dapat melihat status Lampu 1 “ON” atau “OFF”	☒ Berhasil ☐ Gagal
Menghidupkan Lampu 2	Mengklik <i>button</i> “OFF” untuk menyalakan Lampu 2	Dapat mengklik <i>button</i> “OFF” dan Lampu 2 menyala	☒ Berhasil ☐ Gagal
Mematikan Lampu 2	Mengklik <i>button</i> “ON” untuk menyalakan Lampu 2	Dapat mengklik <i>button</i> “ON” dan Lampu 2 padam	☒ Berhasil ☐ Gagal
Status Lampu 2	Menampilkan status Lampu2	Dapat melihat status Lampu 2 “nyala” atau “padam”	☒ Berhasil ☐ Gagal

Menghidupkan Lampu 3	Mengklik <i>button</i> “OFF” untuk menyalakan Lampu 3	Dapat mengklik <i>button</i> “OFF” dan Lampu 3 menyala	☒ Berhasil ☐ Gagal
Mematikan Lampu 3	Mengklik <i>button</i> “ON” untuk menyalakan Lampu 3	Dapat mengklik <i>button</i> “ON” dan Lampu 3 padam	☒ Berhasil ☐ Gagal
Status Lampu 3	Menampilkan status Lampu 3	Dapat melihat status Lampu 3 “nyala” atau “padam”	☒ Berhasil ☐ Gagal

3. Pengujian menu monitoring

Pada menu monitoring terdapat tampilan arus dan tegangan secara *real time* di tampilan *interface* aplikasi *blynk* serta pada LCD terdapat tampilan arus, tegangan, daya dan konsumsi daya/jam dengan tujuan untuk memonitoring masing-masing Lampu sorot LED. *User* melakukan pengujian untuk menu ini terutama untuk bagian *monitoring* lampu LED sebagai salah satu poin utama dari penelitian ini yaitu dapat memonitor lampu LED. Hasil dari percobaan *user* nyata bahwa dapat melakukan *monitoring* lampu pada setiap lampu LED dan dapat melihat hasil status dari pengendalian lampu LED tersebut. Beberapa butir uji dibawah ini akan menjabarkan mengenai menu *monitoring* dengan metode *blackbox testing* pada *interface blynk* dapat dilihat pada tabel

4.4

Tabel 4.4 *Blackbox testing* menu *monitoring* pada *interface blynk*

Butir Uji	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan
Memonitor Lampu 1	Dapat menampilkan nilai arus dan tegangan Lampu 1 secara <i>real time</i>	Arus dan tegangan Lampu 1 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal
Memonitor Lampu 2	Dapat menampilkan nilai arus dan tegangan Lampu 2 secara <i>real time</i>	Arus dan tegangan Lampu 2 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal
Memonitor Lampu 3	Dapat menampilkan nilai arus dan tegangan Lampu 3 secara <i>real time</i>	Arus dan tegangan Lampu 3 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal

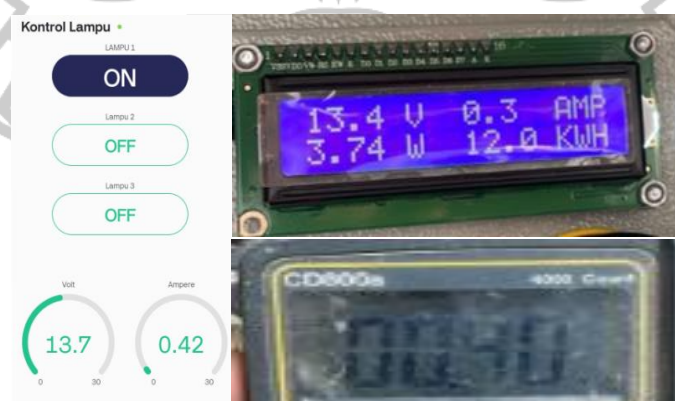
Menu *monitoring* dengan metode *blackbox testing* pada LCD dapat dilihat pada tabel 4.5

Tabel 4.5 *Blackbox testing* menu *monitoring* pada LCD

Butir Uji	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan
Memonitor Lampu 1	Dapat menampilkan nilai arus, tegangan, daya, konsumsi daya/jam Lampu 1 secara <i>real time</i>	Arus, tegangan, daya, konsumsi daya/jam Lampu 1 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal

Memonitor Lampu 2	Dapat menampilkan nilai arus, tegangan, daya, konsumsi daya/jam Lampu 2 secara <i>real time</i>	Arus, tegangan, daya, konsumsi daya/jam Lampu 2 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal
Memonitor Lampu 3	Dapat menampilkan nilai arus, tegangan, daya, konsumsi daya/jam Lampu 3 secara <i>real time</i>	Arus, tegangan, daya, konsumsi daya/jam Lampu 3 ketika alat dinyalakan terlihat	☒ Berhasil ☐ Gagal

Pengambilan data dilakukan pada tanggal 28 September 2024 di lokasi Jl. Hartaqa, Daya, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan tepatnya pada pukul 10.39 sampai 10.46 WITA dengan hasil pengukuran tegangan *solar cell*, Accu, dan SCC sesuai dengan tabel 4.1. Hasil pengukuran lampu 1 dapat dilihat pada gambar 4.15



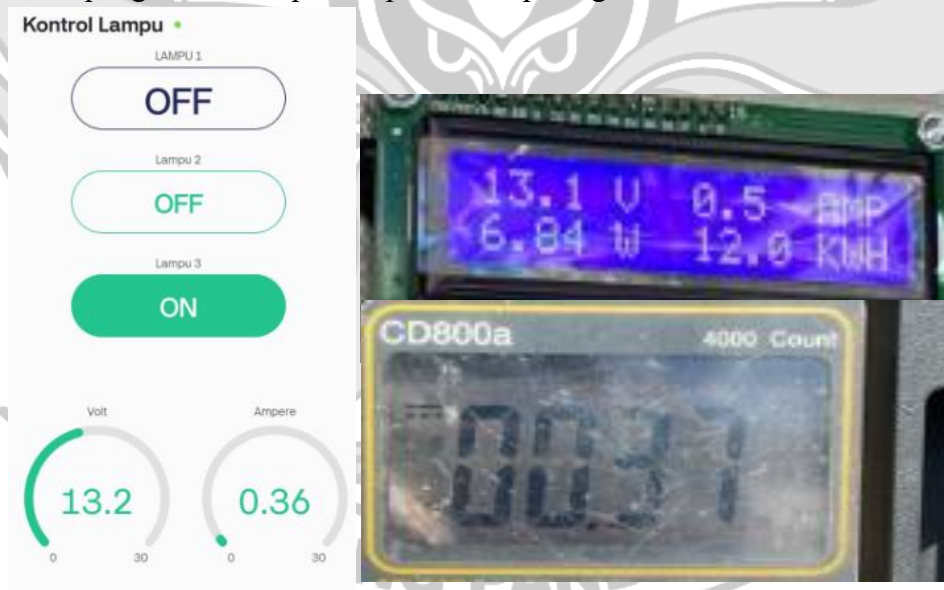
Gambar 4.15 Hasil pengukuran lampu 1

Hasil pengukuran lampu 2 dapat dilihat pada gambar 4.16



Gambar 4.16 Hasil pengukuran lampu 2

Hasil pengukuran lampu 3 dapat dilihat pada gambar 4.17



Gambar 4.17 Hasil pengukuran lampu 3

Hasil pengukuran pada aplikasi *blinky* IoT, LCD, Multimeter dapat dilihat pada tabel 4.6

Tabel 4.6 Hasil pengukuran pada aplikasi *blynk* IoT, LCD, Multimeter

LAMPU	(W)	Blynk IoT		LCD		Multimeter	
		(A)	(V)	(A)	(V)	(A)	(V)
Tanpa Beban	0	0	13,8	0	13,8	-	-
L1	20	0,42	13,7	0,3	13,4	0,4	13,6
L2	50	1,6	13,6	1,6	13,5	1,3	13,4
L3	30	0,36	13,2	0,5	13,1	0,37	13,3

Dari hasil pada tabel di atas kita dapat membandingkannya dengan teori dasar dengan rumus sebagai berikut :

$$P = I \times V$$

$$I = \frac{P}{V} \quad (5)$$

Untuk kondisi tanpa beban :

$$I = \frac{P}{V} = I = \frac{0}{12} = 0 \text{ A} \quad (5)$$

Dari hasil pada tabel 4.2 didapatkan nilai tegangan 13,8 volt dc yang dimana nilai tersebut mendekati dengan output aki yaitu sebesar 12 volt dc.

Untuk Lampu 1 secara teori dengan daya 20 Watt :

$$I = \frac{P}{V} = I = \frac{20}{12} = 1,6 \text{ A} \quad (5)$$

Dari hasil pada tabel 4.3 yang mendekati ialah 0,42 A. Dengan ini terdapat selisih yang jauh yaitu dengan nilai selisih 2,08 A. Hal ini disebabkan karena spesifikasi lampu tidak sesuai dengan yang tertulis di *nameplate* dengan nilai yang didapatkan yaitu 3,74 watt.

Perhitungan daya lampu berdasarkan hasil dari LCD

$$P = I \times V \quad (5)$$

$$P = 0,3 \times 13,4$$

$$P = 4,02 \text{ watt}$$

Untuk Lampu 2 secara teori dengan daya 50 Watt :

$$I = \frac{P}{V} = I = \frac{50}{12} = 4,1 \text{ A}$$

Dari hasil pada tabel 4.2 yang mendekati ialah 1,6 A. Dengan ini terdapat selisih yang jauh yaitu dengan nilai selisih 2,5 A. Hal ini disebabkan karena spesifikasi lampu tidak sesuai dengan yang tertulis di *nameplate* dengan nilai yang didapatkan yaitu 21,6 watt.

Perhitungan daya lampu berdasarkan hasil dari LCD

$$P = I \times V \quad (5)$$

$$P = 1,6 \times 13,5$$

$$P = 21,6 \text{ watt}$$

Untuk Lampu 3 secara teori dengan daya 30 Watt :

$$I = \frac{P}{V} = I = \frac{30}{12} = 2,5 \text{ A} \quad (5)$$

Dari hasil pada tabel 4.2 yang mendekati ialah 0,5 A. Dengan ini terdapat selisih yang jauh yaitu dengan nilai selisih 2A. Hal ini disebabkan karena spesifikasi lampu tidak sesuai dengan yang tertulis di *nameplate* dengan nilai yang didapatkan yaitu 6,84 watt.

Perhitungan daya lampu berdasarkan hasil dari LCD

$$P = I \times V \quad (5)$$

$$P = 0,5 \times 13,1$$

$$P = 6,55 \text{ watt}$$

BAB V

PENUTUP

5.1 Kesimpulan

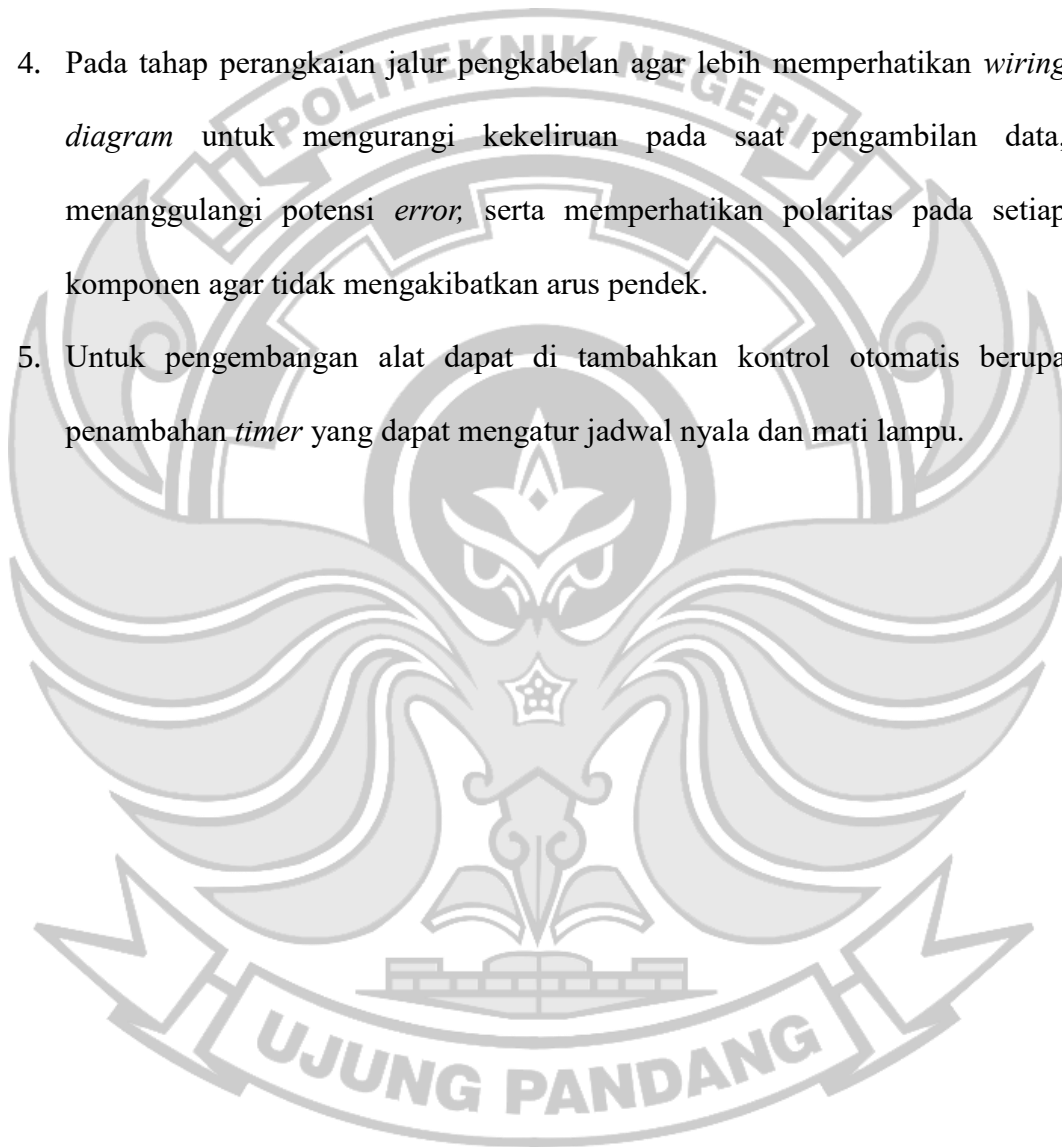
Setelah dilakukan penelitian ini, maka dapat diambil kesimpulan sebagai berikut :

1. Setelah melakukan uji coba, sistem kontrol dan *monitoring* beban pada panel PLTS menggunakan ESP8266 berbasis IoT didapatkan hasil kontrol dan *monitoring*-Nya dimana di dapatkan nilai tegangan 13,1 V – 13,7 V dan nilai arus 0,3 A – 1,6 A. Sehingga di harapkan dari penelitian ini dapat memudahkan pengontrolan dan *monitoring* beban lampu dari jarak jauh melalui *handphone* selama terkoneksi dengan internet.
2. Dari hasil *monitoring* beban lampu, didapatkan performa sensor optimal dalam membaca nilai tegangan dengan pembacaan 13,1 V – 13,7 V, dan nilai arus 0,3 A – 1,6 A, mendekati nilai keluaran aki 12 V sesuai dengan teori. Adapun dari penelitian ini didapatkan adanya ketidaksesuaian spesifikasi lampu dengan hasil yang didapatkan di lapangan dengan lampu 1 spesifikasi pada *nameplate* memiliki daya 20 watt tetapi data pengukuran di lapangan didapatkan hasil daya 3,74 watt, lampu 2 spesifikasi pada *nameplate* memiliki daya 50 watt tetapi data pengukuran di lapangan didapatkan hasil daya 21,6 watt, dan lampu 3 spesifikasi pada *nameplate* memiliki daya 30 watt tetapi data pengukuran di lapangan didapatkan hasil daya 6,84 watt.

5.2 Saran

Dari hasil tugas akhir ini masih banyak kekurangan yang perlu dilakukan pengembangan lebih lanjut, yaitu :

4. Pada tahap perangkaian jalur pengkabelan agar lebih memperhatikan *wiring diagram* untuk mengurangi kekeliruan pada saat pengambilan data, menanggulangi potensi *error*, serta memperhatikan polaritas pada setiap komponen agar tidak mengakibatkan arus pendek.
5. Untuk pengembangan alat dapat di tambahkan kontrol otomatis berupa penambahan *timer* yang dapat mengatur jadwal nyala dan mati lampu.



DAFTAR PUSTAKA

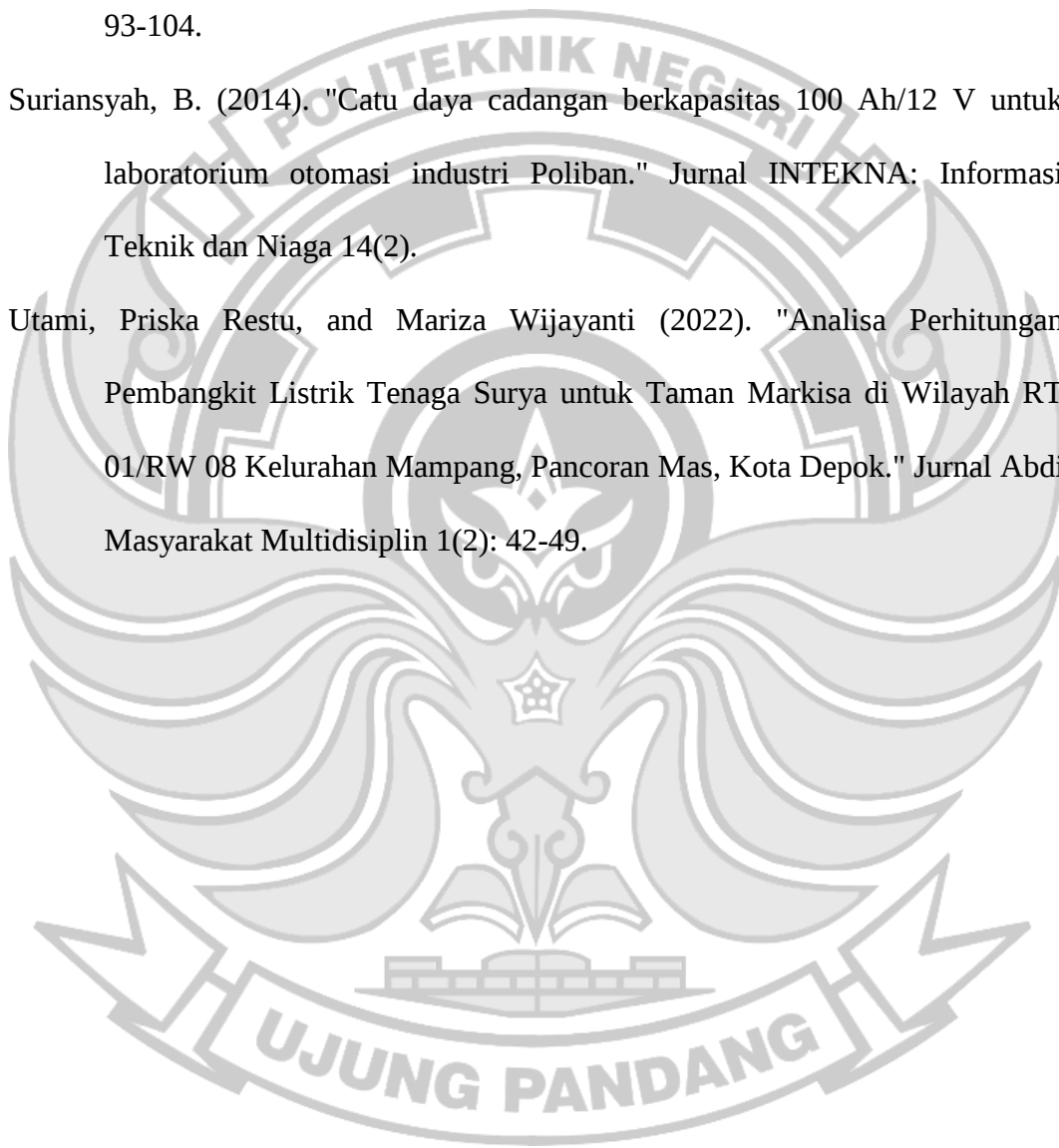
- Bachtiar, M. (2006). "Prosedur perancangan sistem pembangkit listrik tenaga surya untuk perumahan (solar home system)." *SMARTek* 4(3).
- Bagher, A. M., et al. (2015). "Types of solar cells and application." *American Journal of optics and Photonics* 3(5): 94-113.
- Darma, S. (2017). "Analisa perkiraan kemampuan daya yang dibutuhkan untuk perencanaan pembangkit listrik tenaga surya (PLTS)." *Jurnal Ampere* 2(1): 39-53.
- Djuandi, F. (2011). "Pengenalan arduino." *E-book. www.tobuku* 24.
- Faraidh, A. S. and L. Rochmawati (2022). RANCANG BANGUN ENERGI TURBIN AIR MENGGUNAKAN TURBIN CROSSFLOW SEBAGAI PLTA SKALA RUMAH TANGGA. Prosiding SNITP (Seminar Nasional Inovasi Teknologi Penerbangan).
- Gozal, R. P., et al. (2020). "Aplikasi SmartRoom Berbasis Blynk untuk Mengurangi Pemakaian Tenaga Listrik." *Jurnal Infra* 8(1): 39-45.
- Hasan, F. (2022). Rancang Bangun Dan Monitoring Miniatur Sinyal (Sign) Menggunakan Tenaga Solar Cell Berbasis Iot Di Bandar, Politeknik Negeri Ujung Pandang.
- Hasan, H. (2012). "perancangan pembangkit listrik tenaga surya di pulau Saugi." *Jurnal riset dan teknologi kelautan* 10(2): 169-180.
- Iskandar, H. R. (2020). Praktis Belajar Pembangkit Listrik Tenaga Surya, Deepublish.

- Kurniawan, B. (2020). Rancang Bangun Sistem Smart Power Untuk Mengontrol Dan Memonitor Energi Listrik Berbasis Internet Of Things (IoT), Institut Teknologi Nasional Malang.
- Pido, R., et al. (2019). "Analisa Pengaruh Kenaikan Temperatur Permukaan Solar Cell Terhadap Daya Output." *Gorontalo Journal of Infrastructure and Science Engineering* 2(2): 24-30.
- Putri, M. R. (2022). "Sistem Kontrol Beban Dan Monitoring Daya Baterai Pada Panel Surya 50wp Untuk Aplikasi Penerangan Berbasis Internet Of Things."
- Pynkyawati, I.T. and I.S. Wahadamaputera, *Utilitas Bangunan Modul Kenyamanan*. 2021: Penebar Swadaya Grup.
- Selay, A., et al. (2022). "Internet Of Things." *KARIMAH TAUHID* 1(6): 860-868.
- Setiawan, D., et al. (2019). "Desain dan analisis inverter satu fasa berbasis Arduino menggunakan metode SPWM." *Jurnal Teknik* 13(2): 128-135.
- Setiawan, M. T., et al. (2021). "Implementasi Internet Of Things Dalam Rancang Bangun Sistem Monitoring Pada Solar Cell Berbasis Web." *JEECOM J. Electr. Eng. Comput* 3(1): 34-38.
- Siallagan, T. F. and T. Tita (2020). "Di Rancang Bangun Sistem Keamanan Terhadap Kunci Ruangan Berbasis Bot Telegram Menggunakan Mikrokontroler Esp8266: Rancang Bangun Sistem Keamanan Terhadap Kunci Ruangan Berbasis Bot Telegram Menggunakan Mikrokontroler Esp8266." *Journal of Information Technology* 2(2): 45-54.

Suriana, I. W. and I. W. Sugarayasa (2022). "Rancang Bangun Sistem Kontrol Pembangkit Hybrid (PLN-Solar Cell) Berbasis Wemos D1 Mini Esp8266." Jurnal Riset Teknologi dan Inovasi Pendidikan (Jartika) 5(2): 93-104.

Suriansyah, B. (2014). "Catu daya cadangan berkapasitas 100 Ah/12 V untuk laboratorium otomasi industri Poliban." Jurnal INTEKNA: Informasi Teknik dan Niaga 14(2).

Utami, Priska Restu, and Mariza Wijayanti (2022). "Analisa Perhitungan Pembangkit Listrik Tenaga Surya untuk Taman Markisa di Wilayah RT 01/RW 08 Kelurahan Mampang, Pancoran Mas, Kota Depok." Jurnal Abdi Masyarakat Multidisiplin 1(2): 42-49.



LAMPIRAN

Lampiran 1. Bahasa program Modul ESP8266

```
#define BLYNK_TEMPLATE_ID "TMPL6GKXwq0bI"
#define BLYNK_TEMPLATE_NAME "kontrol lampu"
#define BLYNK_AUTH_TOKEN
"Yz5kULXYucZGKBCEZU64l1kmCxcwWj2g"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char ssid[] = "Iye";
char pass[] = "oooooooo";
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars
and 2 line display
#include <SoftwareSerial.h>
SoftwareSerial PZEMSerial;
#include <ModbusMaster.h>

#define MAX485_DE D7      //DO
#define MAX485_RE D6      //D1

static uint8_t pzemSlaveAddr = 0x01;

static uint16_t NewshuntAddr = 0x0002;
```

ModbusMaster node

float PZEMVoltage =0;

float PZEMCurrent =0;

float PZEMPower =0;

float PZEMEnergy=0;

unsigned long startMillisPZEM;

unsigned long currentMillisPZEM;

const unsigned long periodPZEM = 1000;

unsigned long startMillisReadData;

unsigned long currentMillisReadData;

const unsigned long periodReadData = 1000;

unsigned long startMillis1;

#define relay1 D3

#define relay2 D0

#define relay3 D8

BLYNK_WRITE(V0){

bool button1 = param.asInt();

if(button1==1) {digitalWrite(relay1,1);}

```

if(button1==0) {digitalWrite(relay1,0);}

BLYNK_WRITE(V1){

  bool button2 = param.asInt();

  if(button2==1) {digitalWrite(relay2,1);}

  if(button2==0) {digitalWrite(relay2,0);}

  BLYNK_WRITE(V2){

    bool button3 = param.asInt();

    if(button3==1) {digitalWrite(relay3,1);}

    if(button3==0) {digitalWrite(relay3,0);}

    void setup() {

      WiFi.begin (ssid, pass);

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

      Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

      startMillis1 = millis();

      Serial.begin(9600);

      PZEMSerial.begin(9600,SWSERIAL_8N2,D4,D5);

      startMillisPZEM = millis();

      pinMode(MAX485_RE, OUTPUT);

```

```
pinMode(MAX485_DE, OUTPUT);

digitalWrite(MAX485_RE, 0);

digitalWrite(MAX485_DE, 0);

node.preTransmission(preTransmission);

node.postTransmission(postTransmission);

node.begin(pzemSlaveAddr,PZEMSerial);

delay(1000);

startMillisReadData = millis();

//ResetEnergy();

pinMode(relay1,OUTPUT);

pinMode(relay2,OUTPUT);

pinMode(relay3,OUTPUT);

lcd.init();

lcd.backlight();

digitalWrite(relay1,1);

digitalWrite(relay2,1);

digitalWrite(relay3,1);

}
```

```

void loop() {

    currentMillisPZEM = millis();

    if (currentMillisPZEM - startMillisPZEM >= periodPZEM)
    {

        tempdouble = (node.getResponseBuffer(0x0003) << 16) +
        node.getResponseBuffer(0x0002);

        PZEMPower = tempdouble / 10.0;

        tempdouble = (node.getResponseBuffer(0x0005) << 16) +
        node.getResponseBuffer(0x0004);

        lcd.setCursor(0,0);lcd.print(PZEMVoltage,1);lcd.setCursor(5,0);lcd.print("V");

        lcd.setCursor(8,0);lcd.print(PZEMCurrent,1);lcd.setCursor(13,0);lcd.print("AMP"
        );

        lcd.setCursor(0,1);lcd.print(PZEMPower,1);lcd.setCursor(5,1);lcd.print("W");

        lcd.setCursor(8,1);lcd.print(PZEMEnergy,1);lcd.setCursor(13,1);lcd.print("KWH"
        );

        Blynk.run();

    }

    void preTransmission()

    {

        if(millis() - startMillis1 > 5000)

```

```

{

digitalWrite(MAX485_RE, 1);

digitalWrite(MAX485_DE, 1);

delay(1); } }

void postTransmission()

{
if(millis() - startMillis1 > 5000)
{
delay(3);

digitalWrite(MAX485_RE, 0);
digitalWrite(MAX485_DE, 0); } }

void ResetEnergy(){

uint16_t u16CRC = 0xFFFF

static uint8_t resetCommand = 0x42;

uint8_t slaveAddr = pzemSlaveAddr;

u16CRC = crc16_update(u16CRC, slaveAddr);

u16CRC = crc16_update(u16CRC, resetCommand);

preTransmission();

```

```
PZEMSerial.write(slaveAddr);
```

```
PZEMSerial.write(resetCommand);
```

```
PZEMSerial.write(lowByte(u16CRC));
```

```
PZEMSerial.write(highByte(u16CRC));
```

```
delay(10);
```

```
postTransmission();
```

```
delay(100);
```

```
}
```



Lampiran 2. Foto kegiatan proses perakitan dan pemasangan alat



Proses perakitan alat



Proses pemasangan alat



Hasil pembuatan dan pemasangan alat



QR Code video hasil pengukuran dan monitoring sistem ESP8266

Lampiran 3. RAB

NO	NAMA KOMPONEN	JUMLAH (Pcs)	HARGA SATUAN (Rp)	HARGA TOTAL(Rp)
	NodeMcu V3 Lolyn	1	65.000	65.000
	Sensor PZEM017	1	175.000	175.000
	Modul stepdown LM2596	1	20.000	20.000
	Saklar on/off	1	5.000	5.000
	LCD 16X2	1	45.000	45.000
	Modul Relay 1 Channel	1	25.000	25.000
	Modul Relay 2 Channel	1	40.000	40.000
	Lampu DC 20W	1	95.000	95.000
	Lampu DC 30W	1	145.000	145.000
	Lampu DC 50W	1	180.000	180.000
	Kabel listrik	5	8.000	40.000
	Kabel AWG24	5	3.000	15.000
	PCB Fiber	1	60.000	60.000
	Pelarut PCB	1	120.000	120.000
	Timah	1	115.000	115.000
	Box X6	1	40.000	40.000
	Baut	15	1.000	15.000
	TOTAL			1.200.000



Lampiran 4. Kertas Asistensi



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI**
POLITEKNIK NEGERI UJUNG PANDANG
Jalan Pemuda Kemendiknas Km. 10 Tarahanra, Makassar 90245
Telepon: 0411-585364, 585367, 585368, Faksimili: 0411-585943
Email: www.pnup.ac.id E-Mail: pnup@pnup.ac.id

CATATAN KONSULTASI TUGAS AKHIR
PROGRAM STUDI DI TEKNIK LUBRIK

NAMA MAHASISWA		NAMA DOSEN																																					
Muhammad Fauzan P. Andika Jala, M. Al Rahmansar		MIR (2202002102010001) 20200001																																					
Judul: Rancang bangun Panel monitoring Dan kontrol Bahan PLTS menggunakan arduino berbasis IoT		NAMA PENGARAH																																					
Tema dan subtema		1																																					
<table border="1"> <thead> <tr> <th colspan="4">Pembaca Utama</th> </tr> <tr> <th>No</th> <th>Tanggal</th> <th>Catatan/komentar</th> <th>Tanda Tangan</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> <th></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>16/07/2024</td> <td> - Pengertian bahan menjadi 5. - Pengurangan bahan menjadi 3 untuk Tarahanra monitoring. </td> <td><i>[Signature]</i></td> </tr> <tr> <td>2.</td> <td>17/07/2024</td> <td> - Tiang. - Peningkatan diintegrasikan penjelasan data ke dalam - tambahan data penjelasan kekurangan data penelitian yang lain. - Kesimpulan diintegrasikan argument. </td> <td><i>[Signature]</i></td> </tr> <tr> <td>3</td> <td>25/07/2024</td> <td> - Tujuan reguler + penelitian. - Memungkinkan di lanjut dengan menguraikan. </td> <td><i>[Signature]</i></td> </tr> <tr> <td>4</td> <td>31/07/2024</td> <td> - Kata Efisiensi diubah menjadi memudahkan. - Tema dasar Tlg PLTS dan Pans arduino. </td> <td><i>[Signature]</i></td> </tr> <tr> <td>5</td> <td>1/08/2024</td> <td> - Pengambilan referensi? - Instansi Micro </td> <td><i>[Signature]</i></td> </tr> <tr> <td>6.</td> <td>5/08/2024</td> <td> - Tiga di kuasai baik tulisan. - Program di integrasi - Sei pengujian Reguler + Lanjutan lanjutkan pengujian. </td> <td><i>[Signature]</i></td> </tr> </tbody> </table>				Pembaca Utama				No	Tanggal	Catatan/komentar	Tanda Tangan	1	2	3		1.	16/07/2024	- Pengertian bahan menjadi 5. - Pengurangan bahan menjadi 3 untuk Tarahanra monitoring.	<i>[Signature]</i>	2.	17/07/2024	- Tiang. - Peningkatan diintegrasikan penjelasan data ke dalam - tambahan data penjelasan kekurangan data penelitian yang lain. - Kesimpulan diintegrasikan argument.	<i>[Signature]</i>	3	25/07/2024	- Tujuan reguler + penelitian. - Memungkinkan di lanjut dengan menguraikan.	<i>[Signature]</i>	4	31/07/2024	- Kata Efisiensi diubah menjadi memudahkan. - Tema dasar Tlg PLTS dan Pans arduino.	<i>[Signature]</i>	5	1/08/2024	- Pengambilan referensi? - Instansi Micro	<i>[Signature]</i>	6.	5/08/2024	- Tiga di kuasai baik tulisan. - Program di integrasi - Sei pengujian Reguler + Lanjutan lanjutkan pengujian.	<i>[Signature]</i>
Pembaca Utama																																							
No	Tanggal	Catatan/komentar	Tanda Tangan																																				
1	2	3																																					
1.	16/07/2024	- Pengertian bahan menjadi 5. - Pengurangan bahan menjadi 3 untuk Tarahanra monitoring.	<i>[Signature]</i>																																				
2.	17/07/2024	- Tiang. - Peningkatan diintegrasikan penjelasan data ke dalam - tambahan data penjelasan kekurangan data penelitian yang lain. - Kesimpulan diintegrasikan argument.	<i>[Signature]</i>																																				
3	25/07/2024	- Tujuan reguler + penelitian. - Memungkinkan di lanjut dengan menguraikan.	<i>[Signature]</i>																																				
4	31/07/2024	- Kata Efisiensi diubah menjadi memudahkan. - Tema dasar Tlg PLTS dan Pans arduino.	<i>[Signature]</i>																																				
5	1/08/2024	- Pengambilan referensi? - Instansi Micro	<i>[Signature]</i>																																				
6.	5/08/2024	- Tiga di kuasai baik tulisan. - Program di integrasi - Sei pengujian Reguler + Lanjutan lanjutkan pengujian.	<i>[Signature]</i>																																				

Pembaca Utama



Mukher Dfa Faraby, S.T, M
NIP 19900218 2022031005



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI UJUNG PANDANG

Jalan Jendral Kemerdekaan Km. 10 Tamalunan, Makassar 90245
Telepon: (0411)-585365, 585367, 585368; Faksimili: (0411)-586043
Laman : www.polupg.ac.id/ E-Mail : pnupg@polupg.ac.id

CATATAN KONSULTASI TUGAS AKHIR
PROGRAM STUDI DI TEKNIK LISTRIK

NAMA MAHASISWA : <u>Muhammad P. Andra Jans M.A. Nur</u>		STB :	
JUDUL : <u>Pancang bangun panel monitoring dan kontrol PLTS menggunakan ESP8266 Berbasis IoT</u>		NAMA PENGARAH PENDAMPING <u>1.</u>	
Tanggal Penetapan Judul :			
Pengaruh Pendamping			
No.	Tanggal	Catatan/Komentar	Tanda Tangan
1	2	3	4
1	Senin 2/10/2024	Konsultasi Wi caba. <i>dit-</i>	<i>[Signature]</i> 2/10/24
2	Sabtu 10/10/2024	Pemasangan net dan Panel.	<i>[Signature]</i> 9/10/24
3	Minggu 8/10/2024	Pemasangan dan Pengaturan beban, Dan pengambilan data.	
4	Senin 9/10/2024	- Perbaikan format Laporan. - Pendalaman materi	
5	Sabtu 17/10/2024	- PPT - Laporan format.	
<i>[Signature]</i>			

Pengaruh Pendamping.

[Signature]
Muhammad P. Andra Jans M.A. Nur
NID 174060323199603