

BAB IV

PEMBUATAN ALAT

4.1 Alat dan Bahan

Adapun alat dan bahan yang digunakan pada penelitian ini yaitu terdiri dari perangkat keras dan perangkat lunak.

1. Perangkat *Hardware*

Dibawah ini merupakan Bahan yang diperlukan bisa dilihat pada tabel 4.1

Tabel 4. 1 Bahan yang diperlukan

Komponen	Jumlah
NodeMCU ESP32	1 Buah
DHT 11	1 Buah
LCD I2C	1 Buah
TVS Diode	1 Buah
Resistor Wirewound	1 Buah
Rectifier Dioda	1 Buah
Diode Zener	2 Buah
Kapasitor Polyester	1 Buah
Resistor	2 Buah
Step Down	1 Buah
LM311	1 Buah
Modul MOSFET	1 Buah
Power Supply 12V 5A	1 Buah
Jack DC Male dan Female	2 Buah
Baterai	2 Buah

2. Perangkat *Software*

- a. Arduino IDE
- b. Easyeda
- c. Fusion 360
- d. Proteus 8.13

4.2 Pembuatan *Hardware*

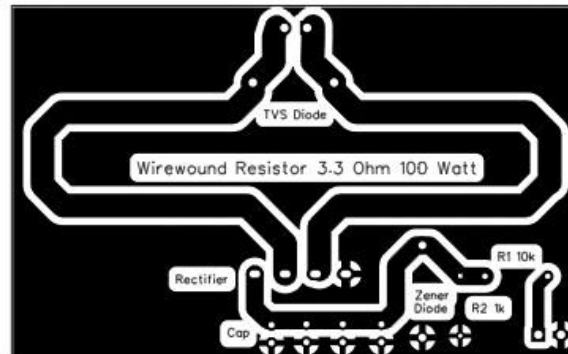
Pembuatan perangkat keras dilakukan setelah proses perancangan dan pemilihan komponen selesai pada tahap sebelumnya. Tujuan utama dari pembuatan perangkat keras ini adalah untuk mewujudkan rancangan sistem pemantauan sambaran petir berbasis Internet of Things (IoT) yang telah direncanakan, dengan mengintegrasikan komponen seperti NodeMCU ESP32, sensor CT Clamp, DHT11, LCD I2C 16x2, dan rangkaian proteksi (TVS Diode, resistor wirewound, rectifier diode, dioda Zener, dan kapasitor polyester) untuk mendeteksi sambaran petir secara real time, melindungi komponen dari lonjakan tegangan, serta memungkinkan pemantauan lokal dan jarak jauh melalui platform Blynk.

4.2.1 Pembuatan Rangkaian PCB

Perancangan PCB dimulai dengan pembuatan skema wiring diagram yang mengilustrasikan hubungan antar komponen dalam sistem pemantauan sambaran petir berbasis IoT, seperti NodeMCU ESP32, sensor CT Clamp, DHT11, LCD I2C, dan rangkaian proteksi (TVS Diode, resistor wirewound, rectifier diode, dioda Zener, dan kapasitor polyester). Skema ini menjadi acuan untuk menentukan penempatan dan koneksi komponen pada PCB. Selanjutnya, dilakukan desain layout PCB dengan mempertimbangkan faktor-faktor seperti optimalisasi ruang, efisiensi jalur koneksi, dan keandalan sistem. Posisi komponen, jalur konduktor, dan struktur lapisan PCB dirancang pada tahap ini untuk memastikan kinerja sistem yang optimal dan aman.

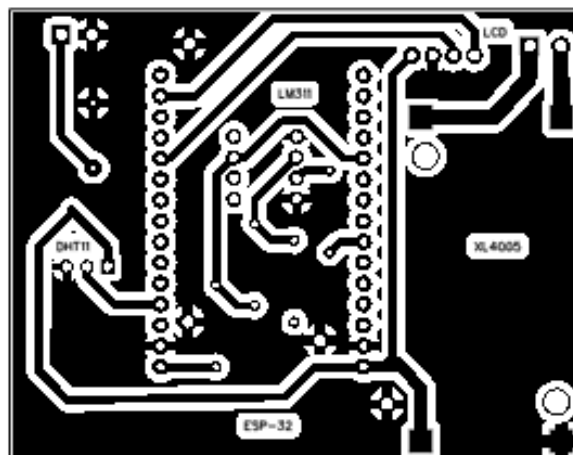
a. Membuat Rangkaian PCB

1. Layout Sistem Proteksi



Gambar 4. 1 Layout sistem Proteksi

2. Layout ESP32 dan Sensor



Gambar 4. 2 Layout ESP32

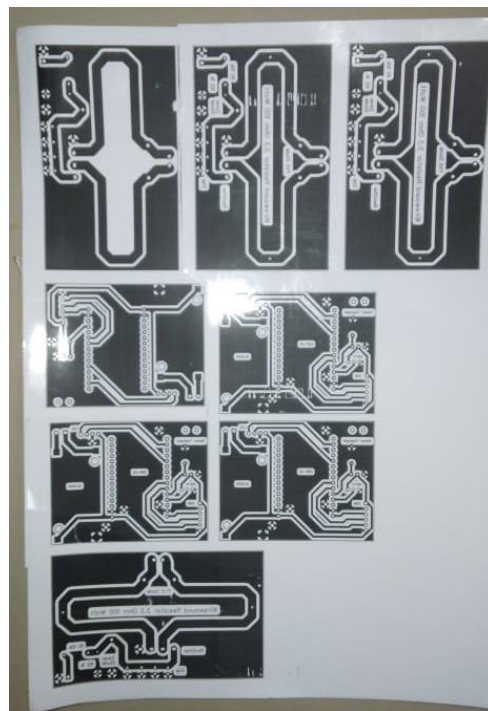
Besar ukuran PCB yang digunakan pada penelitian ini untuk rangkaian proteksi 9.7 cm x 5 cm dan Rangkaian ESP32, 7 cm x 5 cm yang telah memenuhi standar pembuatan PCB yang melibatkan berbagai parameter dan spesifikasi yang harus diperhatikan untuk memastikan kualitas dan keandalan PCB. Beberapa standar umum pembuatan PCB adalah sebagai berikut.

1. Standar lebar jalur yang bervariasi tergantung pada arus yang diperlukan dan tingkat tegangan yang digunakan dalam aplikasi tertentu. Biasanya, lebar jalur dapat ditentukan oleh kapasitas arus yang diinginkan, dalam beberapa kasus, standar umum adalah sekitar 0,2 mm hingga 0,3 mm. Pada implementasi desain PCB rangkaian ini digunakan ukuran lebar sebesar 1,0 mm untuk jalur ESP32, dan ukuran lebar jalur Proteksi 4,0 mm dan 2,0 mm. Besaran ini telah memenuhi standar umum yang telah ditentukan.
2. Jarak antar jalur digunakan untuk mencegah potensial arus bocor dan gangguan anantara jalur yang berdekatan. Standar umum untuk jarak antar jalur dalam rangkaian adalah sekitar 0,2 mm hingga 0,3 mm. Sedangkan jarak antar jalur yang diterapkan adalah Rangkaian ESP32 dan Rangkaian Proteksi 4,0 mm 2,0 mm.
3. Jarak antar komponen atau komponen ke tepi PCB juga merupakan parameter penting. Hal ini memastikan bahwa tidak ada tabrakan fisik atau interferensi antara komponen-komponen saat PCB dipasang di perangkat akhir. Standar umum untuk jarak antar komponen adalah sekitar 0,2 mm hingga 0,3 mm. Besaran jarak minimal yang diterapkan adalah jarak ESP32, sensor 1,0 mm dan jarak Rangkaian Proteksi 4,0 mm 2,0 mm.
4. Diameter lubang juga harus mematuhi standar yang ditetapkan. Diameter lubang biasanya ditentukan oleh ukuran kaki komponen yang akan dimasukkan ke dalam lubang tersebut. Standar umum untuk diameter lubang berkisar antara 0,3 mm hingga 0,4 mm. Besaran diameter lubang terkecil yang diterapkan adalah sebesar 1,0 mm.

Desain PCB yang terdapat pada Gambar 4.1 dan 4.2 telah memenuhi semua standar umum pembuatan PCB, pemenuhan standar umum sangatlah penting untuk merancang PCB untuk memastikan keandalan dan kualitas sistem secara keseluruhan. Standar ini membantu mencegah masalah seperti interferensi sinyal, atau kerusakan mekanis yang dapat terjadi jika parameter tersebut tidak dipenuhi. Selain itu, mengikuti standar pembuatan PCB juga memudahkan proses pembuatan dan perakitan PCB.

b. Mencetak rangkaian PCB

Setelah selesai merancang layout PCB, langkah selanjutnya adalah mencetaknya. Ada beberapa metode pencetakan PCB yang dapat digunakan, seperti pencetakan laser pada kertas transfer atau pencetakan langsung menggunakan printer PCB. Jika menggunakan metode pencetakan laser pada kertas transfer, pastikan untuk mencetak dengan resolusi yang cukup tinggi agar transfer dapat berjalan dengan baik dan menggunting rangkaian tersebut dengan teliti. Yang mana dapat dilihat pada Gambar 4.3



Gambar 4. 3 Menyetak rangkaian PCB

c. Menyetrika rangkaian PCB

Setelah layout PCB dicetak, cara membuat PCB selanjutnya adalah dengan mentransfernya ke papan PCB tembaga. Letakkan kertas transfer di atas tembaga dan menjepitnya dengan kuat, kemudian Anda perlu mengaplikasikan panas dengan cara di setrika pada transfer menggunakan setrika untuk mentransfer gambar ke tembaga. Dapat dilihat pada Gambar 4.4



Gambar 4. 4 Menyetrika rangkaian PCB

Pastikan transfer terjadi dengan sempurna dan bersih, agar jalur tembaga dan komponen pada PCB tidak rusak, dengan cara di cuci rangkaian PCB dan tembaga agar menghasilkan jalur rangkaian yang diinginkan, yang mana dalam hal ini bisa dicuci dengan air bersih dan dipisahkan dengan kertas rangkaian PCB. Dapat dilihat pada Gambar 4.5



Gambar 4. 5 Membersihkan rangkaian PCB

d. Merendam PCB dengan cairan

Setelah mentransfer gambar ke tembaga, Agar rangkaian melekat di tembaga maka perlu merendam PCB dengan cairan yang mana berfungsi agar rangkaian dapat menempel dengan maksimal perlu mengikis bagian tembaga yang tidak diperlukan. Hal ini dilakukan dengan menggunakan cairan etsa

seperti *ferric chloride* atau *persulfate*. Pastikan untuk mengikuti instruksi penggunaan dengan hati-hati dan menggunakan alat pelindung diri yang tepat. Setelah proses etsa selesai, bilas PCB dengan air bersih dan keringkan. Dalam hal ini memerlukan waktu sekitar 30-60 menit untuk mendapatkan hasil yang maksimal. Dapat dilihat pada Gambar 4.6



Gambar 4. 6 Merendam PCB dengan cairan

e. Melakukan pengeboran PCB

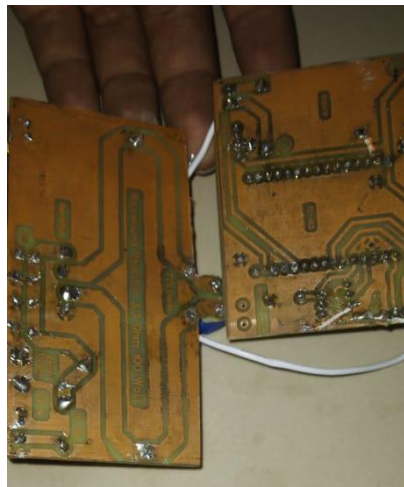
Setelah PCB selesai, perlu dilakukan pengeboran lubang pada tempat komponen akan dipasang. Dapat menggunakan mesin bor mini untuk mengebor lubang. Pastikan untuk mengebor dengan hati-hati agar tidak merusak jalur tembaga. Jangan lupa juga untuk memasang komponen elektronik ke PCB, dan pastikan untuk memasang komponen pada posisi yang benar dan sesuai dengan layout PCB yang telah dibuat. Pastikan juga untuk meninjau kembali layout PCB untuk memastikan bahwa semua komponen telah sesuai dengan kebutuhan. Dapat dilihat pada Gambar 4.7 sebagai berikut:



Gambar 4. 7 Pengeboran PCB

f. Menyolder rangkaian PCB

Setelah melakukan pengeboran, lalu proses selanjutnya yaitu mensolder rangkaian, yang mana gunakan solder dengan ukuran yang tepat, dan pastikan bahwa solder tidak berlebihan. Pastikan juga bahwa semua sambungan solder telah terlihat bersih dan kuat agar rangkaian PCB berhasil. Yang mana dapat dilihat pada dokumentasi Gambar 4.8 sebagai berikut:



Gambar 4. 8 Menyolder rangkaian PCB

g. Finishing

Setelah selesai pembuatan PCB, langkah terakhir adalah memberikan finishing pada PCB. perlu mengecat jalur tembaga dengan solder agar komponen terhubung dengan baik dan memastikan semua rangkaian PCB bisa digunakan dan tidak ada rangkaian yang terputus.

4.2.2 Perakitan Alat Prototype

Setelah rangkaian telah dibuat, maka selanjutnya adalah packaging pada alat ke box. Packaging ini bertujuan untuk merapikan alat, melindungi rangkaian. Merakit alat juga berguna dalam membenarkan alat agar semakin rapi dan bisa di gunakan sebagai prototype.

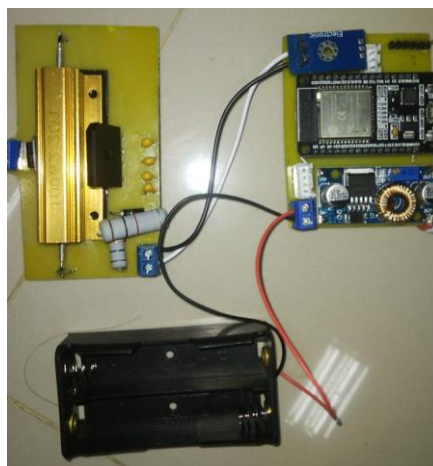
- a. Implementasi Rangkaian Elektronik sistem proteksi Pada Prototipe
Sebelum packaging. Yang mana dapat dilihat seperti Gambar 4.9



Gambar 4. 9 Rangkaian sistem Proteksi

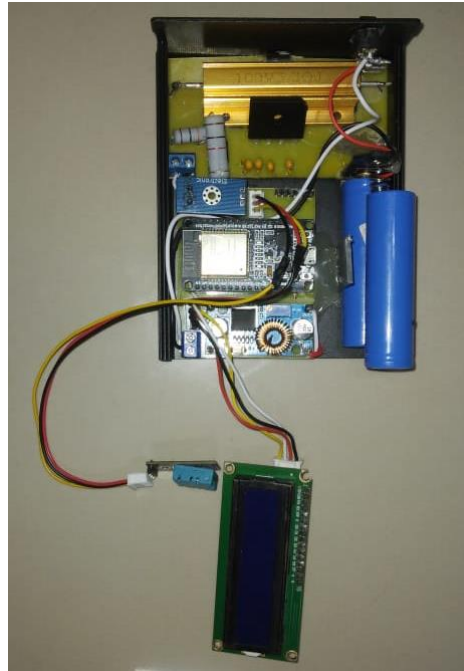
Gambar diatas merupakan gambar implementasi dari rangkaian prototipe sistem proteksi sebelum dimasukkan kedalam frame

- b. Merangkai rangkaian elektronik dan semua komponen sesuai dengan wiring diagram yang dibuat, dimana merangkai alat dari komponen satu dengan komponen satunya dapat dilihat pada Gambar 4.10



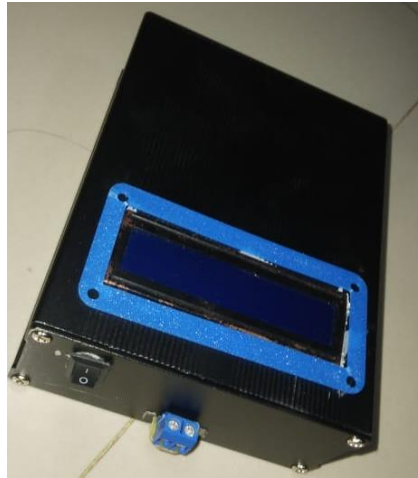
Gambar 4. 10 Merangkai komponen

- c. Melakukan penataan dalam rangkaian elektronik yang ada di dalam frame, mengenai komponen yang digunakan, mulai menyusun sesuai dengan wiring alat yang digunakan, menyusun komponen hardware. Yang mana dapat dilihat pada Gambar 4.11



Gambar 4. 11 Merangkai rangkaian Elektronik dalam bingkai (frame)

- d. Melakukan pemasangan bagian depan pada LCD di depan box serta melakukan finishing dalam pembuatan alat di bagian depan, yang mana nantinya akan terdapat LCD di depannya. Yang mana dapat dilihat seperti pada Gambar 4.12



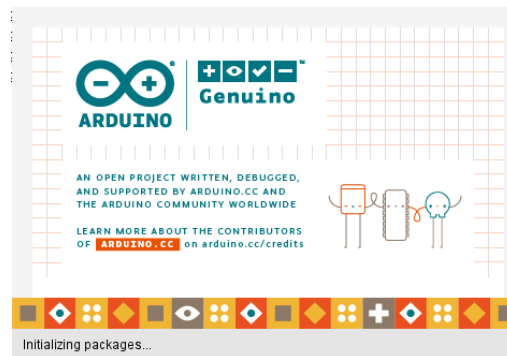
Gambar 4. 12 Melakukan pemasangan LCD

- e. Melakukan finishing alat terakhir, yang mana nantinya sudah dipasang semua perlengkapan sesuai dengan perencanaan, dimana sudah terdapat semua alat dan sudah siap diuji

4.3 Pembuatan *Software*

Pada tahap selanjutnya adalah pembuatan perangkat lunak atau software, tahap ini membuat program untuk dimasukkan ke dalam system ESP32. Proses pembuatan program ini menggunakan aplikasi Arduino IDE yang merupakan aplikasi untuk pemrogram mikrokontroler seperti Arduino dan merupakan aplikasi open source juga. Proses pembuatan program ini memiliki langkah langkah sebagai berikut :

1. Membuka aplikasi Arduino IDE yang sudah ada di laptop kita, kemudian akan terlihat tampilan seperti pada Gambar 4.13



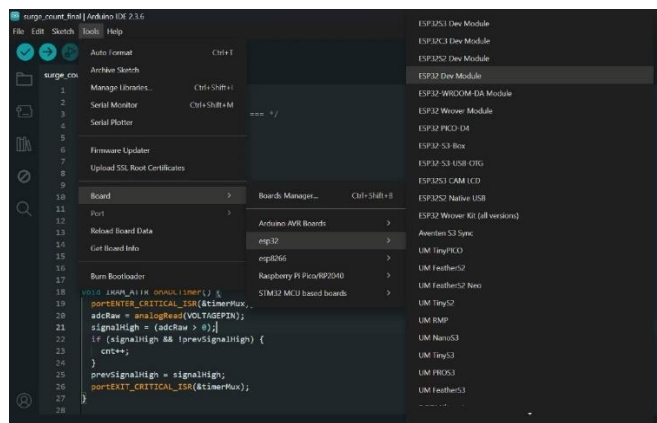
Gambar 4. 13 Arduino IDE

2. Tahap selanjutnya dengan menancapkan ESP32 ke PC yang sudah disiapkan ke port USB PC menggunakan USB pada Gambar 4.14



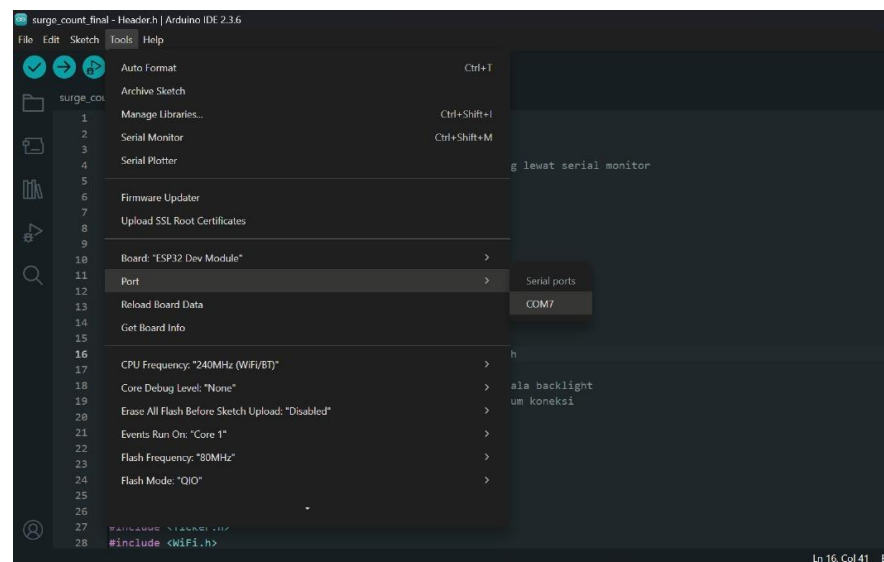
Gambar 4. 14 Penyambungan ESP32 ke Laptop

3. Selanjutnya yaitu melakukan setting untuk board ESP32, yang mana dengan cara klik menu tools lalu pilih ESP32 sesuai dengan kebutuhan. Tampilan dapat dilihat pada Gambar 4.15



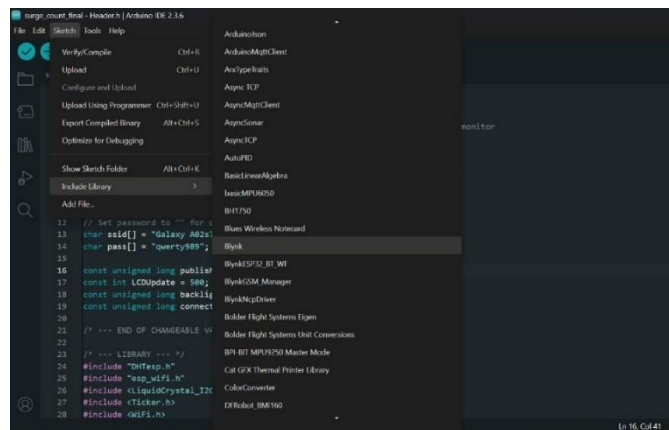
Gambar 4. 15 Memilih Board ESP32

4. Lalu, tahap selanjutnya adalah dengan klik menu tools, dan pilihlah port yang dipakai oleh ESP32 pada PC. Yang mana dapat dilihat tampilan pada Gambar 4.16



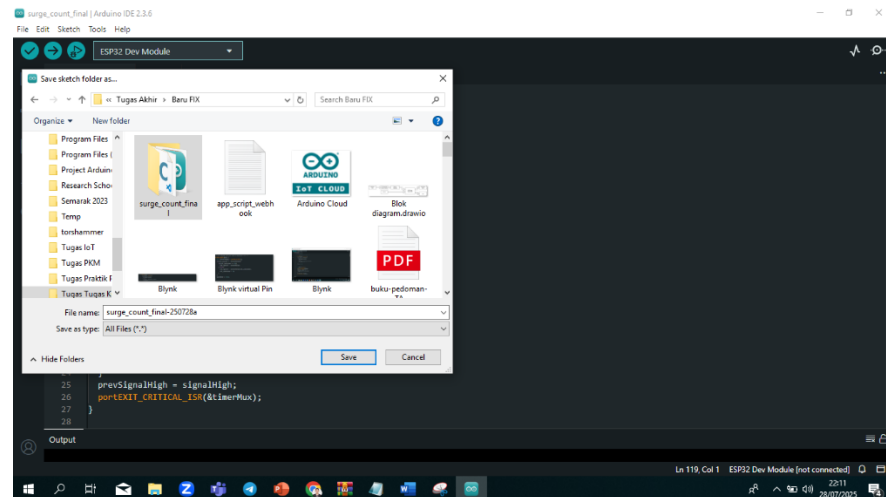
Gambar 4. 16 Melakukan pemilihan port pada Arduino IDE

5. Selanjutnya tambahkan library Serial, yang digunakan untuk melakukan komunikasi antara Arduino dengan Modul Wifi ESP32. Langkah berikutnya klik di menu bar kemudian pilih Sketch selanjutnya klik Include Library selanjutnya klik Add .ZIP Library. Proses dapat dilihat pada Gambar 4.17



Gambar 4. 17 Menambahkan Library

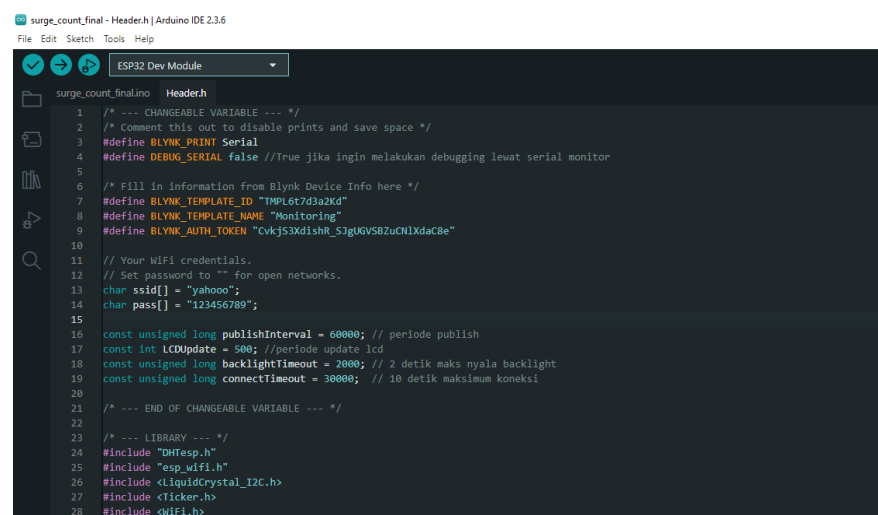
6. Tahap selanjutnya adalah pilihlah file library yang nantinya dibutuhkan, proses pemilihan ini dapat di lihat pada Gambar 4.18



Gambar 4. 18 Pemilihan file Library

7. Menyusun Library dan Header

Langkah pertama adalah menyertakan file Header.h yang berisi semua library dan definisi variabel yang diperlukan. Proses ini dapat dilihat pada Gambar 4.19



Gambar 4. 19 Menyusun library dan Header.h

8. Membuat Fungsi Penanganan Input dari Blynk (Virtual Pin)

Digunakan untuk membaca nilai cnt_container dari server agar tetap tersinkron. Dapat dilihat pada Gambar 4.20

```
/* === Blynk Virtual Write Handler === */
BLYNK_WRITE(V2) {
  int serverValue = param.asInt();
  if(cnt_synced != serverValue){
    if(cnt_container == 0){
      cnt_synced = serverValue;
    } else{
      cnt_synced = serverValue+cnt_container;
      cnt_container = 0;
    }
  }
  syncDone = true;
}
```

Gambar 4. 20 Penanganan input dari Blynk

9. Membuat Algoritma Pulse Counter

Fungsi ini menangani konfigurasi pulse counter pada pin GPIO untuk mendeteksi dan menghitung perubahan sinyal sebagai indikasi lonjakan arus (surge). Dapat dilihat pada Gambar 4.21

```
// ===== PCNT setup =====
void setupPCNT() {
  pcnt_config_t pcnt_config = {};
  pcnt_config.pulse_gpio_num = PCNT_INPUT_SIG_IO;
  pcnt_config.ctrl_gpio_num = PCNT_PIN_NOT_USED;
  pcnt_config.unit = PCNT_UNIT_USED;
  pcnt_config.channel = PCNT_CHANNEL_USED;
  pcnt_config.pos_mode = PCNT_COUNT_INC; // Count rising edge
  pcnt_config.neg_mode = PCNT_COUNT_DIS; // Abaikan falling edge
  pcnt_config.lctrl_mode = PCNT_MODE_KEEP;
  pcnt_config.hctrl_mode = PCNT_MODE_KEEP;
  pcnt_config.counter_h_lim = 32767;
  pcnt_config.counter_l_lim = 0;

  pcnt_unit_config(&pcnt_config);

  // Filter untuk noise < 2us
  uint16_t filter_ticks = (uint16_t)(MIN_PULSE_US * 80.0 + 0.5); // 80 MHz base clk
  if (filter_ticks < 1) filter_ticks = 1;
  pcnt_set_filter_value(PCNT_UNIT_USED, filter_ticks);
  pcnt_filter_enable(PCNT_UNIT_USED);

  pcnt_counter_pause(PCNT_UNIT_USED);
  pcnt_counter_clear(PCNT_UNIT_USED);
  pcnt_counter_resume(PCNT_UNIT_USED);
}
```

Gambar 4. 21 Konfigurasi Pulse Counter untuk Deteksi Perubahan Sinyal

10. Membuat Fungsi Pembacaan Sensor DHT

Fungsi ini membaca suhu dan kelembaban dari sensor DHT11, dan menyimpannya ke variabel temp dan hum. Dapat dilihat pada Gambar 4.22

```
/* === Baca DHT === */
void readDHT() {
    TempAndHumidity data = dht.getTempAndHumidity();
    if (dht.getStatus() == 0) {
        hum = data.humidity;
        temp = data.temperature;
    }
}
```

Gambar 4. 22 Pembacaan sensor DHT11

11. Membuat Fungsi Tampilan ke LCD

Digunakan untuk menampilkan informasi suhu, kelembaban, dan jumlah lonjakan arus ke LCD. Dapat dilihat pada Gambar 4.23

```
/* === Tampilkan LCD === */
void printStatusLCD(const String& line1, const String& line2 = "") {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(line1.substring(0, 16));
    lcd.setCursor(0, 1);
    lcd.print(line2.substring(0, 16));
}
```

Gambar 4. 23 Membuat tampilan LCD

12. Inisialisasi Sistem pada Fungsi

Bagian setup() berisi inisialisasi seluruh komponen sistem, meliputi konfigurasi pin input tegangan (ADC), aktivasi DAC sebagai referensi, inisialisasi LCD untuk menampilkan status awal, pengaturan sensor DHT11 dengan timer pembacaan periodik, inisialisasi Pulse Counter (PCNT), serta pengaturan koneksi WiFi dan konfigurasi ke server Blynk. Yang mana prosesnya dapat dilihat pada Gambar 4.24


```

/* === Setup === */
void setup() {
  if (DEBUG_SERIAL) {
    Serial.begin(115200);
    delay(1500);
  }

  pinMode(VOLTAGEPIN, INPUT);

  // DAC hampir 0V (Vref op-amp)
  dac_output_enable(DAC_CHANNEL_2); // GPIO26
  dac_output_voltage(DAC_CHANNEL_2, DAC_VAL);

  lcd.init();
  lcd.backlight();
  printStatusLCD("Surge Counter", "Init...");

  dht.setup(DHTPIN, DHTesp::DHT11);
  dhtTicker.attach(5.0, readDHT);

  // Start PCNT
  setupPCNT();

  // Start network & Blynk (core 1)
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, pass);
  Blynk.config(BLYNK_AUTH_TOKEN);
  Blynk.connect(); // non-blocking

  stateStartTime = millis();
}

```

Gambar 4. 24 Inisialisasi Sistem

13. Menjalankan Fungsi Utama Program

Bagian loop() menjalankan dua hal utama: update tampilan LCD dan kirim data ke server Blynk. Proses dapat dilihat pada Gambar 4.25

```

/* === Loop (core 1) === */
void loop() {
  Blynk.run(); // keep Blynk alive
  unsigned long now = millis();

  // 1) Update LCD & accumulate pulse every LCDUpdate ms
  static unsigned long lastLCDUpdate = 0;
  if (now - lastLCDUpdate >= LCDUpdate) {
    int16_t cntNow = 0;
    pcnt_get_counter_value(PCNT_UNIT_USED, &cntNow);

    int32_t safeCNT = (int32_t)cntNow - (int32_t)lastPCNT;
    lastPCNT = cntNow;
    noInterrupts();
    safeTemp = temp;
    safeHum = hum;
    interrupts();
    portENTER_CRITICAL(&cntMux);
    cnt_container += safeCNT;
    if (syncDone) count = cnt_synced + cnt_container;
    portEXIT_CRITICAL(&cntMux);

    // tampilkan perubahan jika ada
    if (currentState != PUBLISHING) {
      if (temp != lastTemp || hum != lastHum ||
          cnt_container != lastCNT || currentState != lastState) {
        printStatusLCD(
          "T:" + String(safeTemp, 2) + " H:" + String(safeHum, 2),
          "Counter: " + String(count) // + " [" + String((int)currentState) + "]"
        );
      }
    }
  }
}

```

Gambar 4. 25 Menjalankan fungsi utama Program

14. Mengirim Data dan Masuk ke Mode Tidur (*Sleep Mode*)

Untuk menghemat daya, sistem mematikan WiFi dan masuk ke *light sleep*, kemudian akan aktif kembali setelah publishInterval. Dapat dilihat pada Gambar 4.26

```
case CONNECTING:
  if (WiFi.status() == WL_CONNECTED) {
    if (!Blynk.connected()) {
      Blynk.connect();
    } else {
      Blynk.run();
    }

    // Pastikan hanya sekali sync
    if (!syncDone) {
      Blynk.syncVirtual(V2); // Minta data dari server
      stateStartTime = now; // Catat waktu untuk timeout sync
    }

    // Tunggu BLYNK_WRITE dieksekusi (syncDone == true)
    if (syncDone || (now - stateStartTime > 2000)) { // Timeout 2 detik jika gagal
      currentState = PUBLISHING;
      stateStartTime = now;
    }
  }
  } else if (now - stateStartTime > connectTimeout) {
    currentState = SLEEPING;
    stateStartTime = now;
  }
  break;
```

Gambar 4. 26 Mengirim data dan masuk ke mode Sleep