

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan Perangkat Keras

Pembuatan perangkat keras merupakan tahapan awal dalam proses perancangan sistem *3D scanner* berbasis Arduino Nano. Pada tahap ini, dilakukan perakitan seluruh komponen fisik yang mendukung kinerja sistem, baik dari sisi kelistrikan maupun mekanikal. Perangkat keras yang dirancang terdiri dari beberapa komponen utama, yaitu mikrokontroler Arduino Nano sebagai pusat pengendali sistem, sensor VL53L0X sebagai pengukur dimensi objek dengan presisi tinggi, motor stepper yang digerakkan melalui driver motor untuk memutar meja putar *turntable*, serta modul Bluetooth atau USB sebagai antarmuka komunikasi data dengan komputer.

Arduino Nano berfungsi sebagai pengendali utama untuk mengatur pergerakan motor dan membaca data dari sensor. Sensor VL53L0X diposisikan secara tetap untuk mengukur jarak permukaan objek dari titik referensi selama proses pemindaian dengan akurasi tinggi dan rentang pengukuran yang stabil. Motor stepper digunakan untuk memutar objek dengan presisi sudut tertentu, sehingga seluruh permukaan objek dapat dipindai secara menyeluruh. Data hasil pemindaian kemudian dikirimkan ke komputer untuk diproses lebih lanjut dan divisualisasikan dalam bentuk model 3D. Sistem ini juga dapat dikembangkan lebih lanjut dengan menambahkan modul SD card untuk pencatatan data lokal atau modul WiFi untuk komunikasi nirkabel.

4.1.1 Alat dan Bahan

Proses pertama dalam pembuatan sistem ini adalah mempersiapkan alat dan bahan yang digunakan. Daftar alat yang digunakan untuk membangun rancang bangun *3D scanner* berbasis Arduino Nano dapat dilihat pada tabel 4-1.

Tabel 4- 1 Alat yang digunakan

No	Nama Alat	Jumlah
1	Multimeter	1 buah
2	Obeng Set	1 buah
3	Gunting	1 buah
4	Tang Potong	1 buah
5	Tang Kupas	1 buah
6	Solder	1 buah
7	Bor tangan set	1 buah
8	Jangka Sorong	1 buah
9	Alat Tulis	1 set
10	3D Print	1 buah

Pada Tabel 4-2 merupakan bahan yang digunakan untuk membangun rancang bangun 3D *scanner* berbasis Arduino Nano.

Tabel 4- 2 Bahan yang digunakan

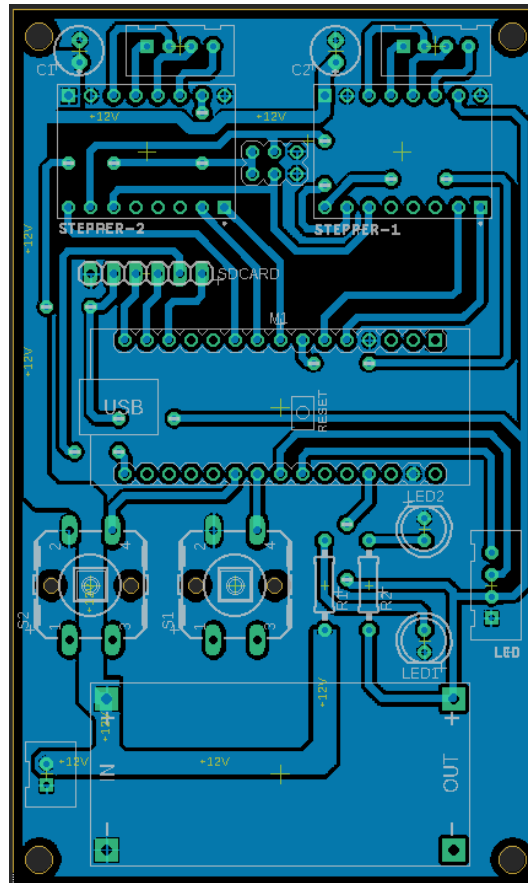
No	Nama Bahan	Jumlah
1	DC Converter step down	1 buah
2	Motor stepper nema 17	2 buah
3	ToF sensor	1 buah
4	Linear motion block bearing	2 buah
5	Batang Shaft ROD	1 buah
6	Bracket Holder Shaft	2 buah
7	Adaptor 5A 12 V	1 buah
8	LCD 16x2	1 buah
9	Sd card modul reader	1 set
10	Baut leadscrew	1 buah
11	shaft coupler	1 buah
12	nut leadcrew	1 buah
13	Arduino Nano	1 buah

4.1.2 Pembuatan Perangkat Elektrik

Pembuatan Perangkat elektrik merupakan tahap penting dalam proses perakitan rancang bangun 3D *scanner*. Pada tahap ini, dilakukan perancangan dan perakitan seluruh komponen elektronik yang bertugas mengendalikan sistem,

membaca, menggerakkan, dan mendeteksi keberhasilan pengeluaran yang diinginkan. Seluruh komponen elektronik ini dirangkai dalam satu sistem berbasis Arduino Nano. Adapun langkah-langkah pembuatan perangkat elektrik adalah sebagai berikut :

1. Perancangan Skematik dan Layout PCB



Gambar 4. 1 Skematik dan Layout PCB

Langkah awal dimulai dengan membuat skematik rangkaian elektronik menggunakan perangkat lunak EasyEDA seperti pada Gambar 4.1. Dalam skematik ini, seluruh komponen seperti modul driver stepper, slot SD Card, port USB, tombol push button, indikator LED, dan konektor daya disusun secara sistematis agar jalur koneksi antar komponen dapat diatur dengan tepat. Setelah skematik selesai dirancang, proses dilanjutkan dengan pembuatan layout PCB berdasarkan koneksi pada skematik tersebut. Tahapan ini mencakup penempatan jalur sinyal (routing), pengaturan dimensi papan, serta pemberian label pada pin-pin penting seperti jalur catu daya (+12V, -12V), pin stepper, output, dan konektor lainnya. Dengan layout

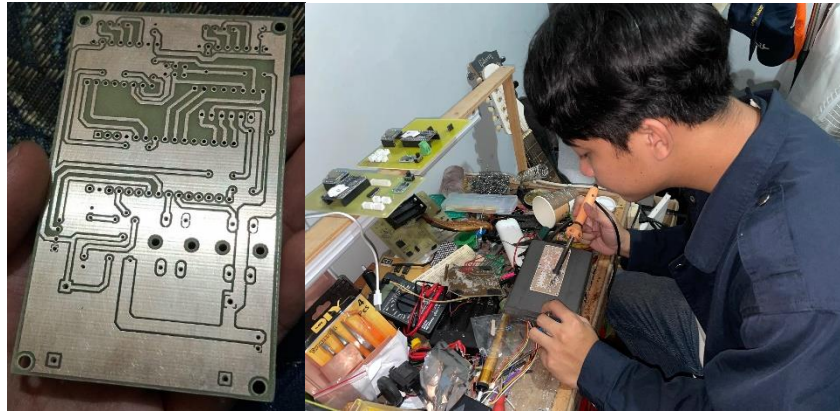
yang rapi dan padat seperti ditampilkan pada gambar, proses fabrikasi PCB dapat dilakukan secara efisien dan minim risiko kesalahan koneksi.

2. Proses Pencetakan dan Penyolderan PCB

Setelah proses perancangan skematik dan layout selesai dilakukan menggunakan perangkat lunak EasyEDA, desain PCB dicetak menggunakan printer laser pada media transparan. Proses selanjutnya adalah transfer jalur dari hasil cetakan ke papan tembaga. Proses ini dilakukan dengan menempelkan cetakan jalur ke permukaan papan tembaga dan memanaskannya menggunakan setrika atau laminator sehingga toner berpindah ke permukaan tembaga.

Setelah transfer berhasil, dilakukan proses etching dengan merendam papan dalam larutan ferric chloride (FeCl_3) untuk melarutkan bagian tembaga yang tidak tertutup toner. Hasilnya adalah jalur konduktif sesuai dengan layout desain yang sebelumnya dibuat. Setelah etching selesai, papan dibersihkan dari sisa toner dan dikeringkan.

Langkah selanjutnya adalah melakukan pelubangan (drilling) pada papan PCB menggunakan bor mini untuk membuat lubang tempat pemasangan komponen seperti pin header, terminal konektor, dan tombol. Setelah itu, dilakukan proses penyolderan komponen ke PCB. Komponen seperti modul driver stepper, slot SD card, port USB, tombol (S1, S2), serta LED indikator dipasang secara permanen melalui penyolderan. Selain itu, konektor untuk daya (+12V dan -12V) serta output ke motor stepper juga dipasang menggunakan terminal blok yang memungkinkan sambungan kabel eksternal secara aman dan fleksibel. Penempatan komponen yang rapi dan jalur yang efisien memastikan rangkaian dapat berfungsi dengan baik dan mudah dalam proses pemeliharaan.



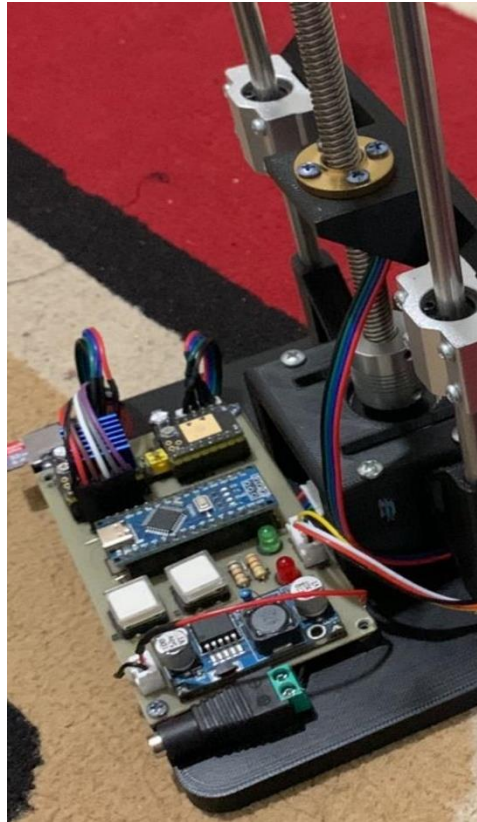
Gambar 4. 2 Pencetakan dan Penyolderan PCB

3. Integrasi Sistem

Beberapa komponen pada sistem 3D scanner berbasis Arduino Nano, seperti sensor jarak (misalnya VL53L0X atau HC-SR04), motor stepper, driver stepper, tombol input, maupun modul SD card, tidak disolder langsung ke papan PCB. Sebagai gantinya, komponen-komponen tersebut dipasang menggunakan kabel JST atau header pin socket.

Tujuan dari pemasangan ini adalah untuk memberikan fleksibilitas dalam penempatan komponen saat integrasi ke dalam rangka mekanik scanner yang umumnya dirakit menggunakan case hasil 3D printing atau struktur akrilik/plexiglass. Hal ini sangat membantu, terutama karena posisi sensor dan motor seringkali harus disesuaikan secara presisi agar pemindaian 3D dapat berjalan optimal.

Selain fleksibilitas dalam penempatan, penggunaan kabel JST juga memudahkan proses perawatan dan penggantian komponen, misalnya jika sensor rusak atau motor tidak bekerja, pengguna dapat dengan mudah melepas dan mengganti komponen tersebut tanpa perlu melakukan penyolderan ulang. Ini sangat berguna dalam proyek pengembangan atau prototipe yang sering mengalami iterasi desain. Gambar 4.3 menunjukkan komponen yang sudah terpasang di PCB.



Gambar 4. 3 Komponen yang Sudah Terpasang lengkap

4.1.3 Pembuatan Perangkat Mekanik

Perangkat mekanik pada sistem 3D Scanner berbasis arduino nano dirancang untuk mendukung proses otomatis dan terintegrasi dengan komponen elektronik. Seluruh bagian mekanik, mulai dari wadah penyimpanan, mekanisme pengeluaran, hingga dudukan komponen elektronik, dibuat menggunakan teknologi pencetakan tiga dimensi (*3D printing*) dengan material infill 30% PLA+. Adapun langkah-langkah pembuatan perangkat mekanik adalah sebagai berikut :

1. Desain Komponen Mekanik

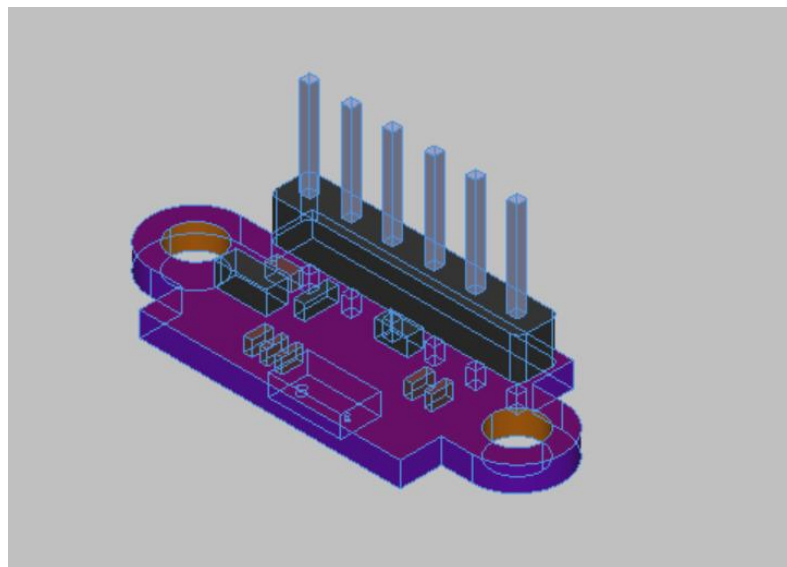
Desain komponen mekanik pada sistem 3D scanner dilakukan secara menyeluruh menggunakan perangkat lunak Autodesk Inventor. Sistem pemindaian dirancang berbasis mekanisme rotasi yang digerakkan oleh motor stepper, sehingga objek dapat diputar secara presisi selama proses pemindaian berlangsung. Posisi rotasi yang stabil sangat penting untuk menghasilkan hasil pemindaian yang akurat dan konsisten.

Selain itu, rancangan casing juga mencakup ruang khusus untuk pemasangan sensor jarak (seperti VL53L0X atau HC-SR04) yang bertugas membaca profil permukaan objek dari berbagai sudut. Penempatan sensor ini diperhitungkan agar tetap sejajar dan memiliki sudut pengukuran optimal. Desain juga mempertimbangkan ruang sirkulasi untuk mencegah penumpukan panas, terutama di sekitar driver stepper dan Arduino Nano. Secara keseluruhan

2. Proses Pencetakan 3D Print

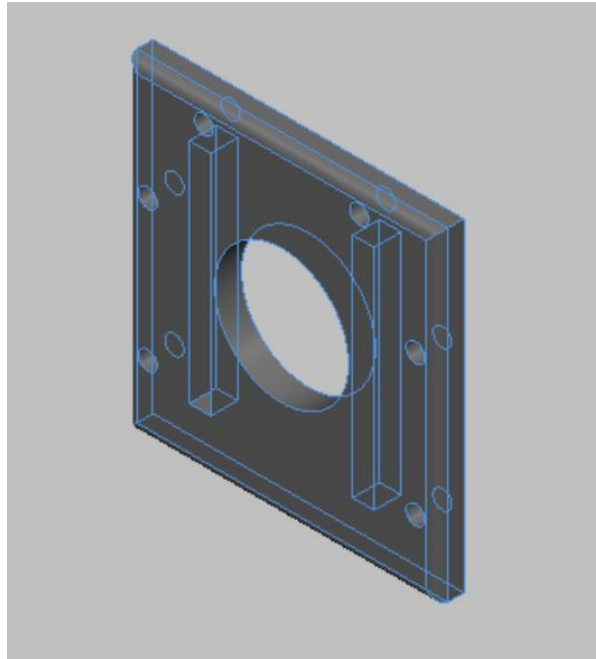
Desain 3D yang telah selesai kemudian dicetak menggunakan *printer 3D* dengan material PLA+. Proses pencetakan dilakukan dengan pengaturan resolusi cetak yang sesuai agar hasil cetakan memiliki presisi yang cukup tinggi, terutama pada bagian *snap-fit* dan dudukan komponen. Seluruh bagian dicetak terpisah sesuai dengan fungsi dan dirakit setelahnya Gambar 4. Menunjukkan proses pencetakan 3D Print. Secara keseluruhan, sistem ini terdiri dari sembilan komponen mekanik utama yang seluruhnya dicetak menggunakan teknologi 3D printing dengan material PLA+. Komponen-komponen tersebut meliputi:

1. Modul sensor yang ditunjukkan oleh Gambar 4.4, berfungsi untuk mendeteksi posisi objek yang sedang discan



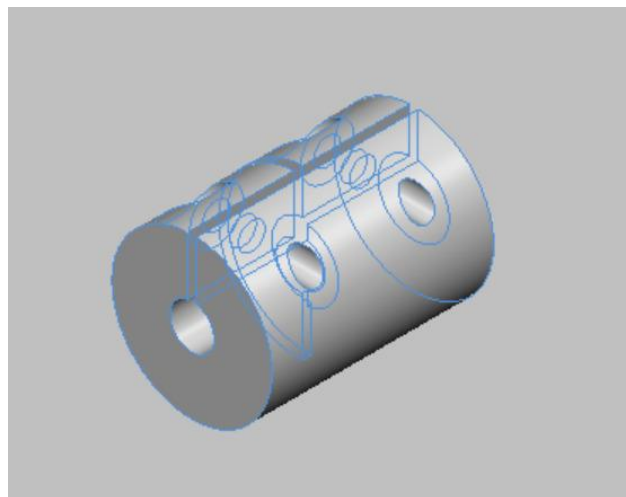
Gambar 4. 4 Modul Sensor

2. Mounting Bracket Stepper yang ditunjukkan oleh gambar 4.5, digunakan untuk memutar platform / *turntable*.



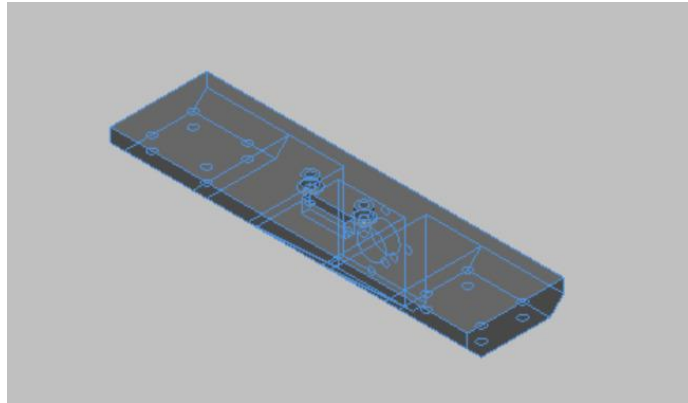
Gambar 4. 5 Mounting Bracket Stepper

3. Coupler yang ditunjukkan oleh gambar 4.6, berfungsi untuk menghubungkan poros motor stepper dengan platform atau dudukan objek.



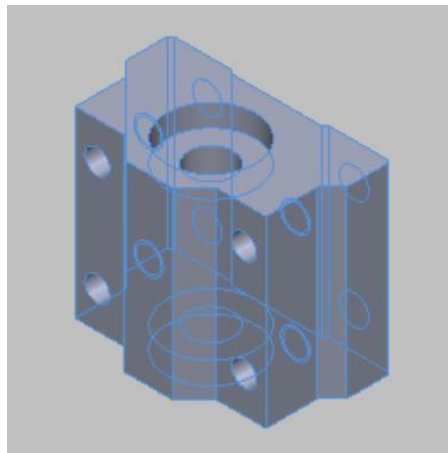
Gambar 4. 6 Coupler

4. Base Frame yang ditunjukkan oleh gambar 4.7, berfungsi sebagai tempat pemasangan komponen-komponen lain.



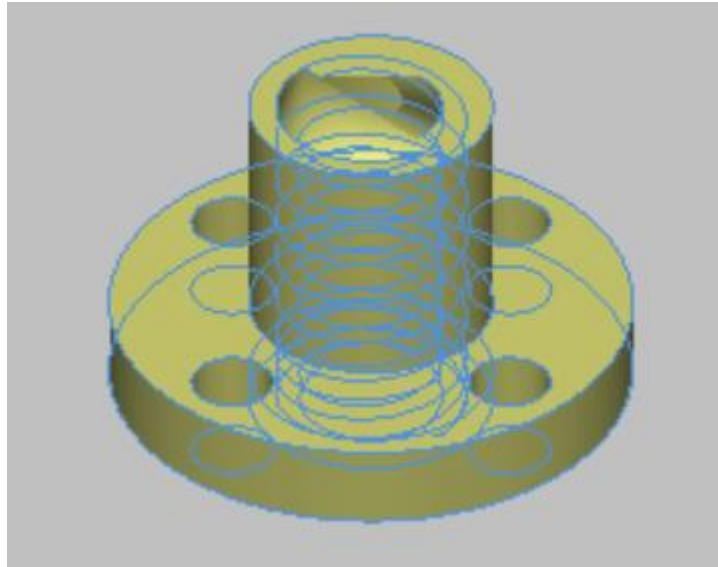
Gambar 4. 7 Base Frame

5. Sensor Holder yang ditunjukkan oleh gambar 4.8, berfungsi untuk tempat memasang sensor jarak ToF.



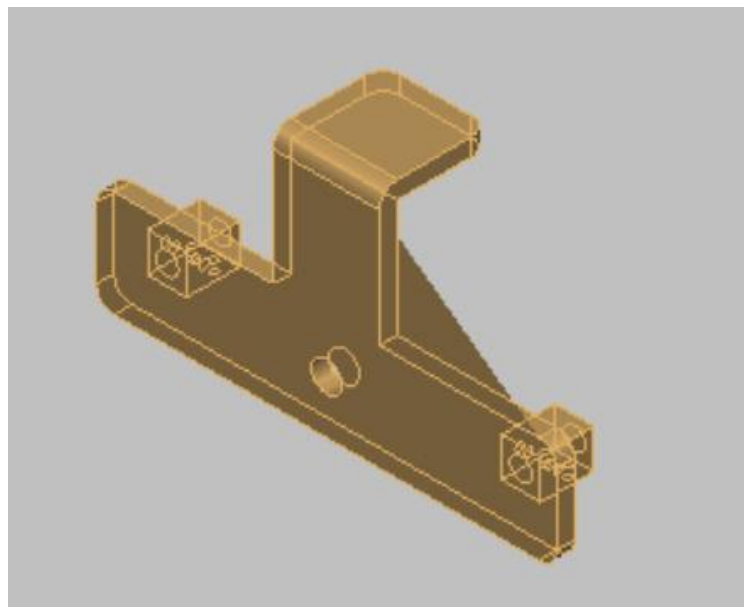
Gambar 4. 8 Sensor Holder

6. Flange Mount yang ditunjukkan oleh gambar 4.9, sebagai tempat duduk poros pemutar yang bisa dipasangkan ke motor stepper.



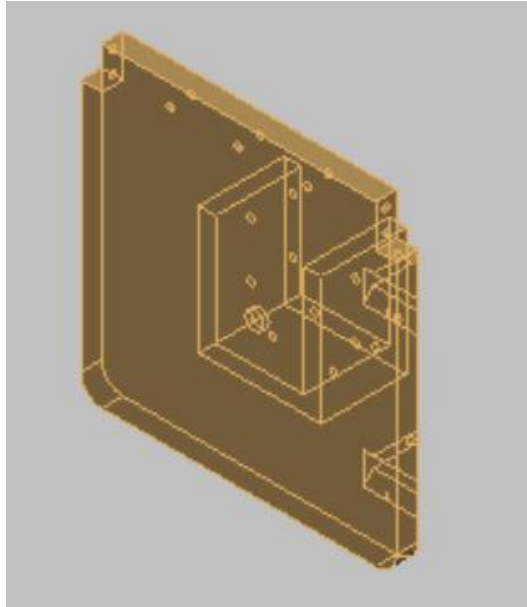
Gambar 4. 9 Flange Mount

7. Clamp yang ditunjukkan oleh gambar 4.10, berfungsi untuk menjepit atau memegang komponen elektronik.



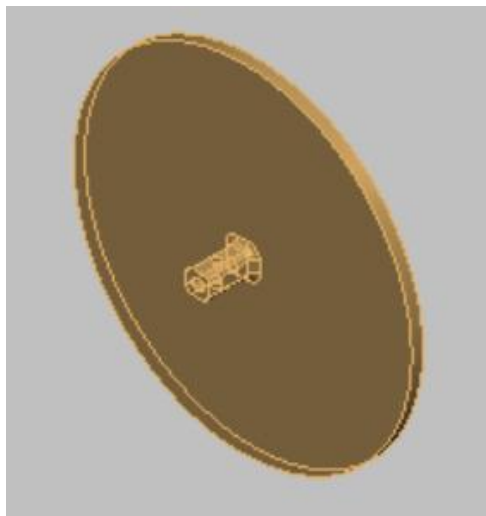
Gambar 4. 10 Clamp

8. Frame Utama yang ditunjukkan oleh gambar 4.11, sebagaiudukan utama untuk menopang dan menyatukan semua komponen mekanik dan elektronik pada 3D scanner.



Gambar 4. 11 Frame Utama

9. *Turntable* yang ditunjukkan oleh gambar 4.12, berfungsi sebagai tempat meletakkan objek yang akan dipindai dan memungkinkan objek berputar selama proses pemindaian.



Gambar 4. 12 Turntable

Setiap komponen yang telah didesain agar dapat dirakit secara presisi menggunakan sistem *snap-fit* maupun baut, tergantung pada kebutuhan kekuatan dan fleksibilitasnya. Desain modular ini tidak hanya memudahkan proses perakitan dan perawatan, tetapi juga memungkinkan alat untuk diperluas atau dimodifikasi di masa depan.

3. Perakitan Komponen Mekanik

Setelah proses pencetakan PCB selesai, bagian-bagian mekanik dari sistem 3D scanner dirakit secara modular. Komponen seperti motor stepper, sensor jarak (misalnya VL53L0X atau HC-SR04), dan Arduino Nano dipasang menggunakan baut danudukan khusus agar stabil selama proses pemindaian berlangsung. Penempatan yang presisi sangat penting untuk menjaga akurasi hasil pemindaian.

Sementara itu, struktur mekanik lainnya sepertiudukan objek dan rangka penyangga sensor/motor dirancang menggunakan sistem kunci (snap-fit) atau sekrup standar, sehingga seluruh unit dapat dibuka dan ditutup kembali dengan mudah. Pendekatan ini sangat bermanfaat untuk kebutuhan kalibrasi, perawatan rutin, penggantian sensor, atau pengembangan sistem lebih lanjut.

Desain modular ini memberikan fleksibilitas tinggi dalam integrasi dan penyesuaian ulang posisi komponen, tanpa mengorbankan kekuatan atau kestabilan struktur. Selain itu, penggunaan mekanisme rotasi berbasis motor stepper memberikan kontrol yang lebih presisi terhadap pergerakan sudut, yang sangat krusial dalam proses pemindaian 3D berbasis rotasi.



Gambar 4. 13 Komponen Mekanik yang Telah Terpasang

4.2 Pembuatan Program Pada Arduino IDE

Pemrograman mikrokontroler Arduino Nano dilakukan menggunakan perangkat lunak Arduino IDE dengan bahasa pemrograman C++. Program ini merupakan bagian sentral dari sistem karena mengatur kerja seluruh komponen perangkat keras seperti sensor ToF, motor stepper, driver motor stepper, modul SD card, LCD 16x2 I2C. Struktur kode dibagi ke dalam beberapa bagian utama untuk memastikan pengembangan yang modular dan efisien. Setiap bagian difokuskan pada satu fungsi spesifik agar lebih mudah dalam proses *debugging* dan pengujian. Pembagian program yang dikembangkan meliputi beberapa bagian, yaitu:

1. Program untuk menginisialisasi dan setup motor stepper.
2. Program untuk mengatur *microstepping* dan *homing* awal.
3. Program untuk membaca jarak dari sensor ToF.
4. Program untuk menghitung koordinat XYZ dalam penerapan rumus pythagoras.
5. Program untuk menyimpan data dalam SD card.
6. Program untuk mengontrol antarmuka LCD dan push button.

4.2.1 Inisialisasi dan Setup Motor Stepper

Motor stepper merupakan komponen penggerak utama pada sistem *3D Scanner* ini, dimana satu motor digunakan untuk memutar objek secara horizontal (*turntable*) dan satu lagi digunakan untuk menggerakkan secara vertikal sensor ToF.

Inisialisasi dan pengaturan awal kedua motor ini menjadi pondasi dari akurasi dan presisi sistem pemindaian.

Pada tahap awal inisialisasi, program mendefinisikan pin-pin digital yang digunakan sebagai sinyal kendali untuk setiap motor. Pin *dir_pin*, *step_pin*, dan *ena_pin* digunakan untuk mengatur arah, jumlah langkah, serta aktivasi motor. Pengaturan ini diterapkan untuk kedua motor secara terpisah. Program inisialisasi pin motor stepper yang digunakan akan dijelaskan pada Gambar 4.14

```
// === Stepper Setup
#define DIR1      5
#define PUL1      6
#define ENA1      7
#define DIR2      2
#define PUL2      3
#define ENA2      4
```

Gambar 4. 14 Pin Motor Stepper

Fungsi *setup()* pada Arduino juga digunakan untuk menyetel mode pin sebagai output, serta mengatur status awal HIGH atau LOW seperti yang dijelaskan pada Gambar 4.15

```
// TMC2208 Init
pinMode(DIR1, OUTPUT);
pinMode(PUL1, OUTPUT);
pinMode(ENA1, OUTPUT);
pinMode(DIR2, OUTPUT);
pinMode(PUL2, OUTPUT);
pinMode(ENA2, OUTPUT);

digitalWrite(ENA1, LOW); // Enable drivers
digitalWrite(ENA2, LOW); // Enable drivers
```

Gambar 4. 15 Pin Arduino

Aktivasi motor dilakukan melalui logika HIGH/LOW untuk *enable pin* (*enPinX*), serta sinyal pulsa digital ke *stepPinX* dalam fungsi *run_stepper()* dan *move_scanner()*. Fungsi ini bertugas mengatur arah rotasi dan jumlah langkah yang akan dijalankan. Untuk kebutuhan resolusi tinggi, motor stepper dikendalikan menggunakan pengaturan *microstepping* $\frac{1}{4}$, yang menghasilkan 1600 langkah per

rotasi penuh, hal tersebut yang memungkinkan peningkatan akurasi dalam pemindaian sudut pada rotasi horizontal objek.

4.2.2 Pengaturan Microstepping dan Homing Awal

Setelah proses inisialisasi motor stepper dilakukan, tahap selanjutnya adalah pengaturan *microstepping* dan implementasi proses homing awal. Kedua hal tersebut krusial untuk menjamin bahwa sistem pemindaian memiliki resolusi tinggi dan posisi awal sensor yang terstandarisasi sebelum proses pemindaian dimulai.

- **Microstepping**

Motor stepper NEMA 17 dalam sistem ini dikendalikan menggunakan *driver motor* TMC2208, yang mendukung pengaturan *microstepping* hingga 1/16. Meskipun mampu hingga 1/16, untuk menjaga keseimbangan antara presisi dan kecepatan pemindaian, sistem menggunakan pengaturan $\frac{1}{4}$ *microstepping* sehingga satu rotasi penuh (360 derajat) terdiri dari 1600 langkah.

Microstepping ini tidak diatur melalui kode, melainkan melalui jumper konfigurasi pada PCB *driver* TMC2208 yang sudah ditentukan secara *hardware*. Dalam kode program, nilai ini diperhitungkan saat menentukan jumlah langkah per derajat putaran. Fungsi *run_stepper()* akan menerima parameter langkah berdasarkan nilai *microstepping* tersebut.

```
run_stepper(2, CW, r_step, 800);
```

Gambar 4. 16 Fungsi pemanggilan stepper dengan microstep

- 2 – motor ke 2 (horizontal)
- CW – arah putar searah jarum jam
- *r_step* – jumlah langkah per iterasi
- 800 – delay antar step

Proses homing awal diperlukan untuk memastikan sensor ToF memulai pemindaian dari posisi $Z=0$ (paling bawah). Dikarenakan tidak menggunakan limit switch fisik, sistem ini menerapkan pendekatan yang lebih sederhana dengan sensorik pasif yang mana nantinya sensor ToF akan mengukur jarak secara vertikal saat dinaikkan, dan berhenti jika mendeteksi penutup sensor pada jarak $<35\text{mm}$.

```
while(1){
    dist = read_sensor();
    if(dist < 35) break;
    run_stepper(1, UP, 50, 500);
```

Gambar 4. 17 potongan proses homing sensor ToF

Setelah sensor menyentuh penutup atas, maka sistem menurunkan sensor ke bawah kembali sejauh 140mm yang akan menjadikannya posisi dasar ($Z=0$) untuk memulai pemindaian.

```
if(step_count > 10) move_scanner(DOWN, 140, 500);
```

Gambar 4. 18 reset posisi ke titik awal

Dengan pendekatan ini, proses kalibrasi awal menjadi otomatis dan tidak memerlukan intervensi pengguna serta tidak memerlukan limit switch eksternal.

4.2.3 Pembacaan Jarak dari Sensor ToF

Sensor Time-of-Flight (ToF) yang digunakan pada sistem ini adalah V53L0X, sebuah sensor berbasis laser yang mampu mengukur jarak secara presisi dalam satuan milimeter. Dalam proyek ini, sensor ToF berfungsi sebagai komponen utama untuk menangkap bentuk geometri permukaan objek secara bertahap dan akurat. Setiap titik hasil pembacaan sensor dikonversikan menjadi koordinat 3D, sehingga akurasi pembacaan jarak sangat mempengaruhi hasil akhir *point cloud*.

Sensor VL53L0X dihubungkan ke Arduino Nano melalui protokol komunikasi I2C. Inisialisasi sensor dilakukan dalam fungsi *setup()* menggunakan *library* resmi *Adafruit_VL53L0X*. Selain inisialisasi dasar, sistem juga menetapkan *timeout* dan *timing budget* agar sesuai dengan kebutuhan kecepatan pembacaan saat proses pemindaian berlangsung.

```
// VL53L0X Init
Wire.begin();
Wire.setClock(100000);

sensor.setTimeout(50);
if(!sensor.init()){
  lcd.setCursor(0, 0);
  lcd.print("Sensor Error");
  while(1);
}
sensor.setMeasurementTimingBudget(20000);
delay(500);
```

Gambar 4. 19 inisialisasi sensor ToF

Jika sensor gagal terdeteksi, sistem akan menampilkan pesan kesalahan pada LCD dan berhenti bekerja agar mencegah kesalahan pengukuran lebih lanjut. Proses pembacaan jarak dilakukan melalui fungsi *read_sensor()*, yang memanggil metode *readRangeSingleMillimeters()* dari library *sensor*. Hasil yang diperoleh merupakan nilai jarak dari sensor ke permukaan objek dalam satuan milimeter.

```
// Read Sensor Function
uint16_t read_sensor(void){
  uint16_t filtered_dist;

  if(!sensor.timeoutOccurred()){
    // Calculate average
    uint16_t sum = 0;
    for (int i = 0; i < numSamples; i++) {
      raw_samples[i] = sensor.readRangeSingleMillimeters();
      sum += raw_samples[i];
    }
    uint16_t avgDistance = (sum / numSamples) - tolerance;
    filtered_dist = pow(avgDistance/K1, 1/K2);
  }
}
```

Gambar 4. 20 Fungsi pembacaan sensor

Fungsi ini menggunakan *readRangeSingleMillimeters* untuk menyimpan hasil pengukuran.

Sensor ToF membaca jarak dari permukaan objek pada setiap sudut *turntable* (stepper horizontal) dan setiap layer ketinggian (stepper vertikal). Pembacaan tersebut akan dilakukan dalam loop utama pemindaian pada Gambar 4.21

```

save_dist = read_sensor();
if(save_dist > 200){
    Serial.println("No Object");
    no_object = true;
    break;
}
angle += DEGTORAD * 0.9;

xp = (cos(angle) * save_dist);
yp = (sin(angle) * save_dist);

```

Gambar 4. 21 Sistem pembacaan sensor

Nilai jarak yang dibaca akan disimpan dalam variabel *save_dist*, lalu digunakan untuk menghitung koordinat X, Y, dan Z berdasarkan sudut rotasi dan tinggi layer. Dalam beberapa kondisi, sensor bisa mengembalikan data tidak valid akibat permukaan objek terlalu gelap, terlalu jauh, atau berada di luar jangkauan. Untuk itu, nilai 999 akan ditandai dan bisa difilter saat pasca proses *point cloud* (meshlab).

4.2.4 Perhitungan Koordinat XYZ (Penerapan Rumus Pythagoras)

Setiap data jarak yang dibaca dari sensor Tof tidak memiliki arti geometris yang langsung dalam sistem tiga dimensi, kecuali jika dikonversi menjadi koordinat (X,Y, Z). Perhitungan koordinat XYZ menjadi tahap krusial dalam proses pemindaian 3D. Konversi ini memungkinkan sistem membangun representasi spasial dari objek secara utuh dalam bentuk *point cloud*.

Pemindaian dilakukan dengan memutar objek secara horizontal sembari sensor tetap menghadap objek dari titik penempatan sensor. Dengan mengasumsikan bahwa posisi sensor tetap dan objek berputar, maka konversi dari sistem polarkoordinat menjadi koordinat kartesian dilakukan menggunakan rumus trigonometri yang akan dijelaskan pada Gambar 4.22

```
angle += DEGTORAD * 0.9;

xp = (cos(angle) * save_dist);
yp = (sin(angle) * save_dist);
```

Gambar 4. 22 Program tranformasi koordinat

Sistem menggunakan pengkondisian tambahan untuk menghindari error akibat data tidak valid. Jika *save_dist* > 200, maka data di lewati dan tidak disimpan.

```
if(save_dist > 200){
    Serial.println("No Object");
    no_object = true;
    break;
}
```

Gambar 4. 23 Pengkondisian tambahan untuk menghindari error

Hal ini ditambahkan guna menjaga integritas hasil *pointcloud* agar tidak tercemar oleh data ekstrem yang dapat merusak visualisasi 3D. Dengan metode ini, seluruh titik yang terukur pada permukaan objek dikonversi menjadi sistem koordinat kartesian yang kemudian akan disimpan dan divisualisasikan menggunakan perangkat lunak seperti *meshlab*.

4.2.5 Penyimpanan Data ke SD Card

setelah proses pembacaan jarak dan perhitungan koordinat 3d (X, Y, Z) selesai dilakukan, data hasil pemindaian perlu disimpan agar dapat diproses lebih lanjut pada tahap visualisasi *pointcloud* menggunakan perangkat lunak eksternal (*meshlab*). Untuk nantinya akan diproses melalui *meshlab*, sistem akan menggunakan modul pembaca SD card yang terhubung langsung dengan Arduino Nano.

Modul SD card terhubung melalui protokol komunikasi SPI, dan memanfaatkan *library* standar SD.h. Inisialisasi dilakukan pada fungsi *setup()* untuk memastikan SD card dapat diakses dan file dapat dibuat.

```
// SDCard Init
if(!SD.begin(10)){
  lcd.clear();
  lcd.print("SD Card Fail!");
  while(1);

  SD.remove(filename);

  dataFile = SD.open(filename, FILE_WRITE);
  if(dataFile){
    dataFile.println("Distance Log Start");
    dataFile.close();
  }
}
```

Gambar 4. 24 Inisialisasi SD card

Jika SD card tidak terdeteksi atau gagal diinisialisasi, sistem akan menampilkan pesan kesalahan dan berhenti agar tidak terjadi proses *scanning* tanpa penyimpanan data. File bernama *measure.txt* akan dibuka atau dibuat ulang setiap kali alat dinyalakan.

Sistem menyimpan setiap titik hasil pemindaian dalam format CSV-like, yaitu setiap baris terdiri dari tiga nilai X, Y, dan Z yang dipisahkan oleh koma, lalu diikuti dengan karakter newline (`\n`). Penulisan dilakukan menggunakan fungsi *writeToSD()* yang akan ditunjukkan pada Gambar 4.25

```
// Write To SDCard
void writeToSD(float x_data, float y_data, float z_data){
  dataFile = SD.open(filename, FILE_WRITE);

  if(dataFile){
    dataFile.print(x_data); dataFile.print(",");
    dataFile.print(y_data); dataFile.print(",");
    dataFile.println(z_data);
    dataFile.close();
  }
}
```

Gambar 4. 25 Fungsi penyimpanan data

Yang nantinya akan menghasilkan data dalam format txt yang akan diproses melalui meshlab. Setiap hasil pembacaan dan perhitungan XYZ yang valid akan disimpan kedalam SD card secara *real time* selama proses *scanning*.

4.2.6 Kontrol Antarmuka LCD dan Push Button

Sistem 3D *Scanner* ini dilengkapi dengan antarmuka pengguna sederhana berupa LCD 16x2 I2C dan dua buah *push button*. Keberadaan antarmuka ini memberikan pengalaman interaktif bagi pengguna dalam menginisialisasi, memantau serta mengontrol proses *scanning* tanpa perlu terhubung pada komputer.

LCD 16x2 dihubungkan ke Arduino melalui antarmuka I2C, yang menggunakan pin SDA dan pin SCL. Untuk mengontrol LCD, digunakan *library LiquidCrystal_I2C.h*.

```
// === LCD Setup ===
#define LCD_ADDRESS 0x27
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

// LCD-Init
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("VL53L0X Init...");
delay(250);
```

Gambar 4. 26 Inisialisasi LCD

Pada sistem 3D *Scanner* ini menggunakan 2 buah *push button* yang dihubungkan ke pin digital Arduino. Tombol digunakan untuk memulai proses pemindaian serta mengatur maupun menghentikan sistem secara manual. *Push button* dibaca menggunakan *digitalRead()* dan dikondisikan untuk mendeteksi tekanan tombol (logika LOW atau HIGH tergantung konfigurasi *wiring*).

```
if(btn1_state == LOW){
    lcd.clear();
    running_state = 1;
    delay(50);
}

else if(btn2_state == LOW){
    lcd.clear();
    delay(50);
}
```

Gambar 4. 27 Pembacaan push button

Button 1 digunakan untuk memulai proses *scanning*, sedangkan button 2 dapat digunakan untuk *homing*.