

BAB IV

PEMBUATAN ALAT

Realisasi alat *Dual-Mode DC-DC Boost Converter* berbasis kontrol PID mencakup implementasi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk menunjang operasional sistem secara menyeluruh. Tahapan ini dilaksanakan guna menjamin setiap komponen berfungsi optimal sesuai spesifikasi teknis yang direncanakan.

Prosedur pengerjaan alat diawali dari penyiapan komponen dan material, pemodelan mekanis sasis koper aluminium beserta panel akrilik, penyusunan tata letak rangkaian pada PCB bolong, penyolderan modul daya dan kontrol, hingga pengodean program pada mikrokontroler Arduino Nano. Pelaksanaan secara terstruktur ini bertujuan untuk mengoptimalkan integrasi antarkomponen sehingga tercipta kinerja sistem yang stabil.

Selain proses fabrikasi, bab ini turut memaparkan pengujian tingkat dasar serta kalibrasi sistem yang telah dirakit. Evaluasi dilakukan untuk memverifikasi fungsionalitas tiap elemen, mencakup akurasi pembacaan sensor tegangan, penayangan informasi pada LCD 16x2 I2C, pengondisian *duty cycle* PWM 62,5 kHz, dinamika respon PID moda *closed loop*, hingga linieritas sirkuit pembagi tegangan. Melalui serangkaian tahapan pembuatan dan pengujian tersebut, alat *Dual-Mode DC-DC Boost Converter* diharapkan mampu beroperasi secara andal serta mempertahankan regulasi tegangan keluaran secara akurat dan presisi.

4.1 Alat dan Bahan

Pada tahap pembuatan alat *Dual-Mode DC-DC Boost Converter* ini, dilakukan proses realisasi dari hasil perancangan yang telah dijelaskan pada Bab III. Proses ini melibatkan penggunaan berbagai alat kerja serta bahan atau komponen yang dibutuhkan untuk membangun sistem secara fisik. Alat kerja digunakan untuk mendukung proses pengerjaan seperti penyolderan, pemotongan dan perakitan sasis, serta pengujian rangkaian, sedangkan bahan merupakan komponen utama yang dirakit menjadi satu kesatuan sistem sesuai dengan desain

yang telah dirancang sebelumnya. Pemilihan alat dan bahan disesuaikan dengan kebutuhan selama proses pembuatan, mulai dari tahap pemodelan dan perakitan mekanis sasis koper aluminium beserta panel akrilik, pemasangan rangkaian elektronik pada PCB bolong, hingga pengujian dan kalibrasi sistem. Seluruh alat dan bahan yang digunakan dalam proses ini disajikan dalam bentuk tabel untuk memudahkan identifikasi serta memberikan gambaran umum mengenai komponen yang digunakan dalam pembuatan *Dual-Mode DC-DC Boost Converter*.

Tabel 4. 1 Alat dan Bahan

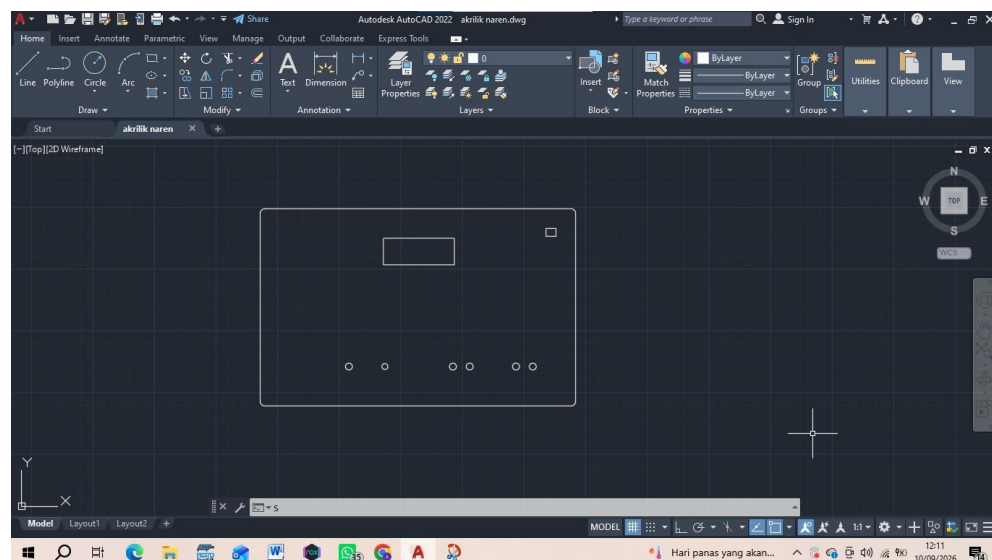
Nama Alat dan Bahan	Jumlah	Satuan
Arduino Nano	1	Unit
MOSFET IRLZ44N	1	Unit
Induktor 330 μ H	1	Unit
Dioda Schottky	1	Unit
Kapasitor Elko 47 μ f	1	Unit
Resistor 100 k Ω	1	Unit
Resistor 1 k Ω	1	Unit
Resistor 10 k Ω	1	Unit
LCD 16X2 I2C	1	Unit
Potensiometer 10 k Ω	1	Unit
Knop Potensiometer	1	Unit
Toggle Switch/Saklar	1	Unit
Kabel C to Female Type C	1	Unit
Heatsink MOSFET	1	Unit
PCB Bolong	1	Unit
Kabel Jumper Male to Male	3	Unit
Kabel Jumper Male to Female	4	Unit
Kabel serabut	1	m
Kabel Alligator to Jack Banana	2	Unit
Jack Banana Female	1	Unit
Akrilik	32x20	cm
Koper Aluminium 32x20x10	32x20x10	cm
Solder 60W	1	Unit
Timah	25	m
Multimeter Digital	1	Unit

4.2 Perancangan Perangkat Keras (Hardware)

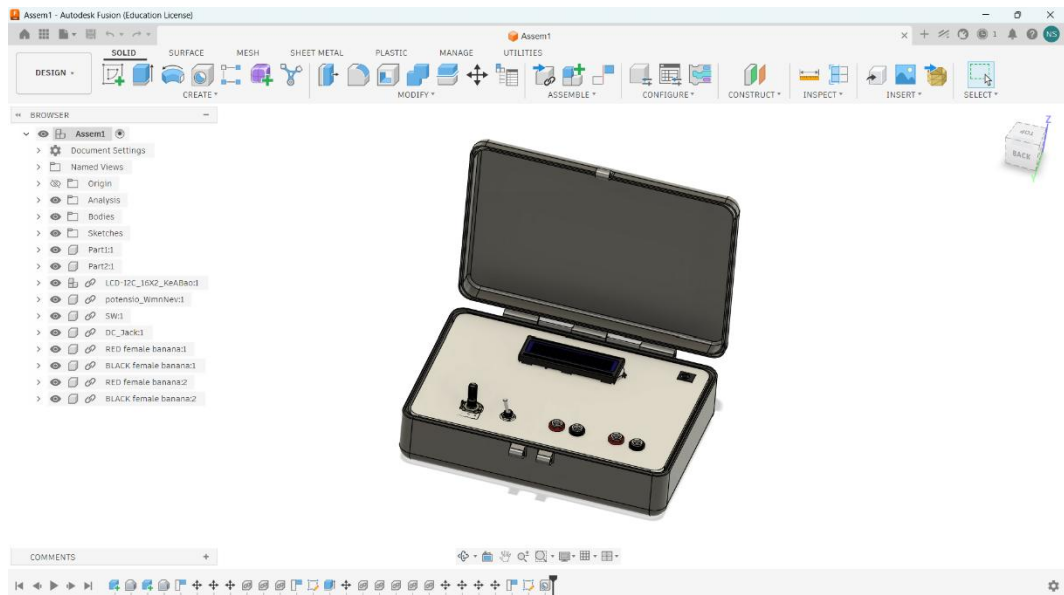
Perancangan perangkat keras (*hardware*) sistem *Dual-Mode DC-DC Boost Converter* ini dibagi menjadi empat tahapan utama:

1. Pemodelan Mekanis dan Desain Panel Akrilik (AutoCAD & 3D CAD)

Tahap awal diawali dengan pemodelan mekanis tiga dimensi (3D CAD) untuk mempresisikan tata letak komponen antarmuka dan sasis koper aluminium. Perancangan bidang antarmuka menggunakan lembaran akrilik bening berukuran $32\text{ cm} \times 20\text{ cm}$ dengan ketebalan 3 mm didesain menggunakan perangkat lunak AutoCAD. Tata letak komponen antarmuka seperti layar LCD 16x2 I2C, knob potensiometer, sakelar *toggle*, dan terminal *banana jack* diposisikan dan diukur secara presisi pada AutoCAD agar menghasilkan berkas pemotongan (*laser cutting*) yang akurat. Pemodelan mekanis ini juga merancang kedalaman penyangga PCB, posisi *heatsink* MOSFET IRLZ44N, dan bukaan ventilasi udara agar terintegrasi presisi di dalam kompartemen koper tanpa risiko saling berbenturan.



Gambar 4. 1 Desain Akrilik



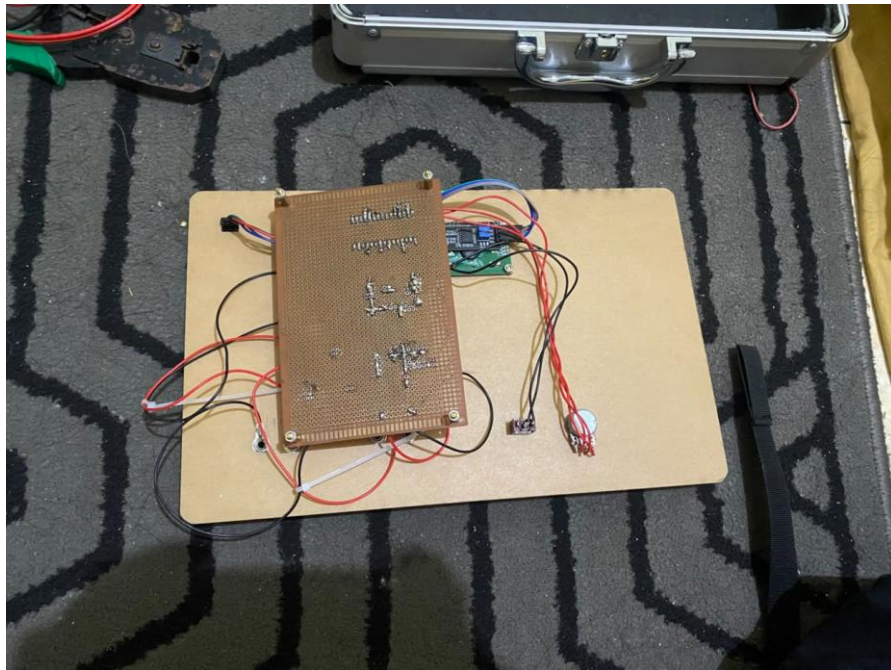
Gambar 4. 2 Pembuatan Desain 3D

2. Perancangan Rangkaian dan Penataan Komponen

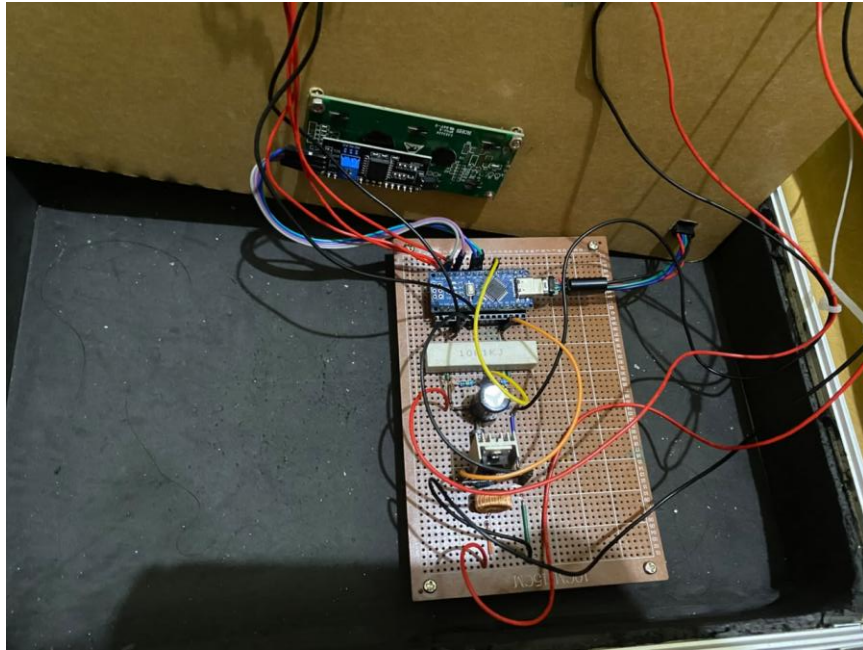
Proses dilanjutkan dengan perancangan rangkaian elektronik untuk memetakan hubungan antar komponen daya dan kontrol. Rangkaian mengintegrasikan komponen daya utama seperti MOSFET IRLZ44N, induktor toroid, dioda *fast recovery*, kapasitor filter keluaran, sirkuit pembagi tegangan R_1 dan R_2 , serta mikrokontroler Arduino Nano. Tata letak penempatan komponen pada PCB bolong (*perfboard/matrix board*) direncanakan secara sistematis guna meminimalkan panjang jalur koneksi antar komponen. Konsep *Star Grounding* diterapkan secara manual dengan memisahkan jalur *Power Ground* (arus besar) dan *Signal Ground* (sensor analog pin A1) pada papan PCB bolong, yang kemudian disatukan pada satu titik node pusat guna mencegah pengaruh *noise* akibat pensaklaran frekuensi 62,5 kHz.

3. Penyolderan dan Perakitan Fisik

Tahap berikutnya adalah penyolderan komponen secara manual pada PCB bolong sesuai dengan peta skematik dan rangkaian yang telah dirancang. Jalur tembaga/kawat penghantar berdiameter tebal ditempelkan dan dilapisi timah solder (*soldering trace reinforcement*) pada titik-titik jalur arus besar guna memperkuat konduktivitas serta menekan nilai resistansi jalur. MOSFET IRLZ44N dipasangkan pada *heatsink* aluminium TO-220 lengkap dengan mika isolator dan *thermal grease* untuk mengoptimalkan pelepasan panas. Hasil pemotongan presisi lembaran akrilik 32 cm × 20 cm setebal 3 mm dari proses *laser cutting* kemudian dipasang komponen antarmuka, dilanjutkan dengan perakitan papan PCB bolong secara kokoh di dalam koper aluminium.



Gambar 4. 3 Hasil Penyolderan



Gambar 4. 4 Perakitan Komponen

4. Pengujian dan Verifikasi Mandiri Hardware

Tahapan perancangan perangkat keras diakhiri dengan pengujian fisik secara mandiri sebelum mikrokontroler dihubungkan ke sumber daya tinggi. Pengujian ini meliputi uji kontinuitas menggunakan multimeter digital untuk memastikan tidak ada hubungan singkat (*short-circuit*) antara jalur daya masukan, keluaran, logika, dan *ground*. Selain itu, dilakukan verifikasi linieritas pembacaan sinyal pada titik *node* pembagi tegangan pada pin A1 Arduino Nano saat disuntikkan tegangan variabel eksternal. Hal ini bertujuan untuk menjamin bahwa sinyal umpan balik yang masuk ke mikrokontroler selalu berada dalam batas aman di bawah 5 V ketika sistem memacu tegangan masukan 20 V DC hingga mencapai *setpoint* 40 V DC.

4.3 Perancangan Perangkat Lunak (Software)

Pada tahap ini dilakukan proses pembuatan perangkat lunak (*software*) yang digunakan untuk mengendalikan seluruh sistem kerja pada alat *Dual-Mode DC-DC Boost Converter*. Perangkat lunak berfungsi sebagai penghubung antara komponen perangkat keras daya dengan sistem kontrol digital sehingga setiap elemen dapat bekerja sesuai dengan fungsi yang telah dirancang. Program dibuat menggunakan bahasa pemrograman Arduino (C/C++) dan diunggah ke mikrokontroler Arduino Nano sebagai pusat pengendali sistem.

Pembuatan perangkat lunak dilakukan dengan menyusun logika program agar mampu mengatur proses konversi daya secara presisi, mulai dari pembacaan input moda operasi, pengolahan sinyal umpan balik tegangan, komputasi kendali PID, hingga pemicuan sakelar daya. Program dirancang untuk mengatur komunikasi dan pengondisian sinyal antara Arduino Nano dengan berbagai komponen, meliputi layar LCD 16x2 I2C, sakelar *toggle* pemilih moda, potensiometer, sirkuit pembagi tegangan, serta MOSFET IRLZ44N.

Pada alat ini, perangkat lunak memiliki beberapa fungsi utama, yaitu mengonfigurasi register *Timer1* untuk menghasilkan frekuensi *switching* PWM sebesar 62,5 kHz, membaca status sakelar *toggle* untuk penentuan moda *open-loop* atau *closed-loop*, mengolah pembacaan sinyal analog tegangan keluaran (V_{out}) via pin A1, mengeksekusi perhitungan galat serta komputasi PID secara diskrit, dan membatasi persentase *duty cycle* pada rentang aman 10% hingga 85%. Selain itu, program dirancang agar mampu menyajikan informasi operasional secara *real-time* melalui LCD 16x2 I2C sehingga pengguna dapat memantau tegangan *setpoint*, tegangan keluaran aktual, serta besaran *duty cycle* dengan mudah.

Subbab ini memaparkan tahapan persiapan pemrograman, proses pembuatan program Arduino, cara kerja program pada setiap komponen yang digunakan, serta implementasi perangkat lunak terhadap sistem *Dual-Mode DC-DC Boost Converter* secara keseluruhan.

4.3.1 Pembuatan Program Arduino

Pada tahap ini dilakukan proses pembuatan program yang digunakan untuk mengendalikan seluruh sistem pada alat *Dual-Mode DC-DC Boost Converter*. Program dibuat menggunakan Arduino IDE dengan bahasa pemrograman Arduino (C/C++) dan diunggah ke Arduino Nano sebagai mikrokontroler utama sistem. Pembuatan program dilakukan berdasarkan alur kerja yang telah dirancang sebelumnya agar seluruh komponen daya dan kontrol dapat bekerja secara terintegrasi dan sesuai dengan fungsi yang diinginkan.

Arduino Nano berfungsi sebagai pusat pengendali sistem yang bertugas membaca masukan dari pengguna dan sensor, mengolah data yang diterima melalui algoritma kendali, kemudian memberikan perintah kepada komponen luaran. Masukan pada sistem berasal dari sirkuit pembagi tegangan, potensiometer, dan sakelar *toggle* pemilih moda, sedangkan luaran sistem terdiri dari modul layar LCD 16x2 I2C dan sakelar daya MOSFET IRLZ44N yang dikendalikan melalui sinyal PWM frekuensi tinggi 62,5 kHz.

Tahap awal pembuatan program dilakukan dengan menambahkan pustaka (*library*) yang dibutuhkan untuk mendukung komunikasi antar-komponen dan algoritma kendali. Pada sistem ini digunakan pustaka `PID_v1.h` untuk kalkulasi kendali PID diskrit, `Wire.h` untuk komunikasi serial I2C, serta `LiquidCrystal_I2C.h` untuk mengendalikan tampilan LCD 16x2. Pustaka tersebut digunakan agar proses komputasi kendali dan komunikasi data dengan layar LCD menjadi lebih sederhana dan efisien.

Setelah proses pemanggilan pustaka dilakukan, tahap berikutnya yaitu menentukan konfigurasi pin masukan dan luaran pada Arduino Nano. Konfigurasi pin dilakukan untuk mendefinisikan fungsi setiap pin yang digunakan pada sistem. Pin luaran Digital D9 dikonfigurasi untuk mengendalikan sinyal pemicu *Gate* MOSFET melalui register *Timer1*, sedangkan pin masukan Analog A1 digunakan untuk membaca sinyal umpan balik pembagi tegangan, Analog A7 untuk membaca masukan potensiometer, dan Digital D4 untuk membaca kondisi sakelar *toggle*.

Tahap selanjutnya yaitu melakukan deklarasi variabel yang digunakan dalam program. Variabel digunakan untuk menyimpan parameter sistem seperti target tegangan acuan (*Setpoint*), tegangan keluaran terukur (*Input*), nilai manipulasi keluaran PWM (*Output*), konstanta bobot gain PID (K_p , K_i , K_d), faktor rasio kalibrasi sensor, status pemilih moda, serta parameter pewaktu *non-blocking* untuk pengiriman data serial dan pembaruan layar LCD.

Setelah proses deklarasi selesai dilakukan, program kemudian dibagi menjadi beberapa fungsi dan struktur logika untuk mempermudah pengelolaan program dan meningkatkan keterbacaan kode. Fungsi-fungsi tersebut meliputi proses pengondisian sinyal sensor (*oversampling filter noise*), konfigurasi register internal *Timer1* PWM 62,5 kHz, pemrosesan logika kendali *dual-mode*, komputasi algoritma PID, pencatatan data telemetri serial, serta pembaruan tampilan LCD.

Pada program juga dibuat sistem pemilihan moda operasional menggunakan sakelar *toggle* dan potensiometer agar pengguna dapat melakukan pengaturan parameter kerja alat secara langsung. Sistem navigasi tersebut digunakan untuk memilih antara moda *Open-Loop* (pengaturan *duty cycle* manual) atau moda *Closed-Loop* (pengaturan *setpoint* tegangan otomatis berbasis PID) sebelum dan selama proses konversi daya berlangsung.

Setelah seluruh fungsi program selesai dibuat, dilakukan proses integrasi antar-fungsi agar seluruh bagian sistem dapat bekerja secara berurutan. Integrasi program dilakukan mulai dari proses inisialisasi sistem pada fungsi *setup()*, pembacaan sinyal sensor terfilter, pemrosesan logika kendali *dual-mode*, penyesuaian pemicuan MOSFET, transmisi data serial ke komputer, hingga pembaruan informasi parameter pada LCD secara *non-blocking*.

4.3.2 Cara Kerja Program

Pada tahap ini dijelaskan cara kerja program yang digunakan pada modul praktikum *Dual-Mode DC-DC Boost Converter* berbasis Arduino Nano berdasarkan fungsi-fungsi dan blok logika yang telah dibuat pada skrip program. Setiap bagian program memiliki tugas dan peran yang spesifik dalam mengendalikan sirkuit konverter daya sehingga seluruh komponen—mulai dari

mikrokontroler, sirkuit pensaklaran MOSFET, sensor umpan balik, hingga modul antarmuka—dapat bekerja secara terintegrasi sesuai rancangan. Pembagian struktur program ke dalam fungsi dan blok logika yang terorganisasi dilakukan untuk mempermudah pengelolaan alur komputasi, meningkatkan keterbacaan kode, serta memfasilitasi proses penelusuran kesalahan (*debugging*) pada sistem kendali.

Program pada alat ini dirancang untuk menjalankan proses komputasi secara kontinu dan responsif, dimulai dari tahapan inisialisasi register PWM 62,5 kHz dan modul tampilan, pembacaan masukan parameter dari potensiometer dan sakelar mode, eksekusi penyaringan sinyal sensor tegangan (*oversampling filter*), perhitungan komputasi algoritma kendali PID, hingga pembaruan register pemicu *Gate* MOSFET IRLZ44N. Selain itu, program dirancang secara efisien agar mampu mengelola pengondisian sinyal terfilter untuk meredam gangguan *switching noise*, membatasi persentase *duty cycle* demi keamanan komponen (*clamping protection*), serta memperbarui data parameter kelistrikan pada layar LCD 16x2 I2C secara *real-time* tanpa menginterupsi siklus pensaklaran utama.

Pada subbab ini akan dijelaskan cara kerja setiap fungsi dan susunan blok kode yang digunakan pada program Arduino Nano beserta potongan skrip yang berkaitan. Penjelasan ini disusun untuk memberikan gambaran yang jelas mengenai alur kerja perangkat lunak, logika pengondisian *Dual-Mode*, serta keterkaitan antar-fungsi dalam mengoordinasikan seluruh sistem pada modul praktikum *Dual-Mode DC-DC Boost Converter*.

a. Deklarasi Pustaka (*Library Declaration*)

```
1  #include <PID_v1.h>
2  #include <Wire.h>
3  #include <LiquidCrystal_I2C.h>
```

Gambar 4. 5 Deklarasi Pustaka

Pemrograman mikrokontroler Arduino Nano diawali dengan mendeklarasikan pustaka-pustaka (*libraries*) yang dibutuhkan untuk mendukung operasi sistem. Pustaka `#include <PID_v1.h>` diimpor untuk mengeksekusi komputasi matematis algoritma kendali PID (*Proportional-Integral-Derivative*)

secara *real-time*. Pustaka `#include <Wire.h>` digunakan untuk mengaktifkan protokol komunikasi serial I2C (*Inter-Integrated Circuit*) bawaan mikrokontroler. Pustaka ini bekerja secara bersamaan dengan `#include <LiquidCrystal_I2C.h>` yang berfungsi mengendalikan modul layar LCD 16x2 melalui penggerak bus I2C PCF8574.

b. Pemetaan Pin dan Deklarasi Variabel Global

```

5 // 1. DEFINISI PIN & ALAMAT LCD
6 const int pinPWM      = 9; // Output PWM MOSFET (Pin D9 - Timer1 OC1A 62.5 kHz)
7 const int pinSensor    = A1; // Pin Sensor Pembagi Tegangan Output
8 const int pinPotensio  = A7; // Pin Potensiometer Pengatur
9 const int pinSaklar    = 4; // Pin Saklar Toggle Mode (Pin D4)
10
11 // Inisialisasi LCD I2C alamat 0x27, 16 Kolom, 2 Baris
12 LiquidCrystal_I2C lcd(0x27, 16, 2);
13
14 // 2. KALIBRASI SENSOR & VARIABEL PID
15 float rasioSensor = 101.0; // Nilai rasio hasil kalibrasi presisi terakhir
16 double Setpoint, Input, Output;
17
18 // Parameter PID Stabil
19 double Kp = 1, Ki = 3, Kd = 0.01;
20 PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
21
22 // Timing non-blocking untuk LCD & Serial Logger
23 unsigned long previousMillis = 0;
24 const long intervalLCD = 200; // Refresh tampilan LCD setiap 200 ms
25

```

Gambar 4. 6 Pemetaan Pin dan Deklarasi Variabel Global

Kode program pada Subbab 4.3.2 b berfungsi untuk mengalokasikan pin *input/output* (I/O) mikrokontroler Arduino Nano yang terhubung langsung dengan perangkat keras *DC-DC Boost Converter*. Alokasi tersebut meliputi penggunaan pin D9 (PWM_PIN) sebagai keluaran sinyal *Pulse Width Modulation* (PWM) untuk mengendalikan *gate driver* MOSFET, serta pin A1 (VOLTAGE_SENS_PIN) sebagai masukan analog untuk membaca sinyal balik (*feedback*) tegangan keluaran (V_{out}) melalui rangkaian sensor pembagi tegangan (*voltage divider*). Selain itu, pin A7 dikonfigurasi sebagai masukan analog pendukung untuk pembacaan parameter kontrol seperti potensiometer penyesuaian *setpoint*, dan pin D4 disiapkan sebagai jalur komunikasi digital pendukung untuk antarmuka modul tampilan LCD maupun indikator status sistem.

Selain pemetaan pin, bagian ini juga mendeklarasikan variabel-variabel global yang digunakan dalam proses kalkulasi algoritma kendali lingkaran tertutup (*closed-loop PID*). Deklarasi tersebut mencakup penentuan konstanta *gain* proporsional, integral, dan diferensial ($K_p = 1$, $K_i = 3$, $K_d = 0,01$), serta variabel penampung *setpoint* tegangan acuan, nilai *error* saat ini, akumulasi *error* (*integral*), *error* sebelumnya (*previous_error*), dan nilai *duty cycle* hasil kalkulasi (*output_pwm*). Untuk menjamin eksekusi algoritma PID berjalan secara presisi pada interval waktu konstan (*sampling rate*), didaftarkan pula variabel pewaktu berbasis unsigned long seperti *lastTime* dan *sampleTime* yang memungkinkan kalkulasi dilakukan secara *non-blocking* tanpa mengganggu alur pemrosesan utama pada mikrokontroler.

c. Inisialisasi Objek dan Konfigurasi PID

```
10 LiquidCrystal_I2C lcd(0x27, 16, 2);
17 PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

Gambar 4. 7 Inisialisasi Objek dan Konfigurasi PID

Tahap inisialisasi objek dilakukan untuk mendaftarkan konfigurasi perangkat keras ke dalam modul perangkat lunak. Pembentukan objek diawali dengan penginstansiasian modul tampilan melalui instruksi `LiquidCrystal_I2C lcd(0x27, 16, 2);`. Deklarasi ini membentuk objek antarmuka bernama `lcd` dengan mendaftarkan alamat bus komunikasi I2C 0x27 pada modul ekspander PCF8574 serta menetapkan dimensi fisik layar LCD sebesar 16 kolom dan 2 baris.

Pendaftaran objek kemudian dilanjutkan dengan menginisialisasi modul kendali otomatis melalui perintah `PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);`. Baris kode ini menginstansiasi objek kontrol bernama `myPID` dari pustaka `PID_v1` dengan mendaftarkan alamat memori variabel utama (`&Input`, `&Output`, `&Setpoint`) menggunakan *pointer*, memasukkan nilai konstanta bobot *gain* (K_p , K_i , K_d), serta menetapkan mode operasi `DIRECT`. Konfigurasi `DIRECT` tersebut menentukan bahwa arah perubahan nilai *Output* PWM berbanding lurus dengan peningkatan tegangan keluaran konverter, sehingga setiap kenaikan sinyal pemicu kontrol akan secara langsung meningkatkan level tegangan keluaran fisik sistem.

d. Inisialisasi Sistem (*Setup Function*)

```

42 void setup() {
43     Serial.begin(9600);
44
45     pinMode(pinPWM, OUTPUT);
46     pinMode(pinPotensio, INPUT);
47     pinMode(pinSensor, INPUT);
48     pinMode(pinSaklar, INPUT_PULLUP);
49
50     // SETTING TIMER1 (PIN D9): FAST PWM 8-BIT FREKUENSI 62.5 kHz
51     TCCR1A = (1 << COM1A1) | (1 << WGM10); // Clear OC1A on Compare Match, 8-bit Fast PWM
52     TCCR1B = (1 << WGM12) | (1 << CS10); // Prescaler 1 (62.5 kHz)
53
54     // Inisialisasi LCD
55     lcd.init();
56     lcd.backlight();
57     lcd.clear();
58     lcd.setCursor(0, 0);
59     lcd.print("BOOST CONVERTER");
60     lcd.setCursor(0, 1);
61     lcd.print("DC-DC DUAL MODE");
62     delay(1500);
63     lcd.clear();
64
65     // Inisialisasi Pembacaan Awal Sensor
66     Input = bacaTeganganFilter();
67
68     myPID.SetMode(AUTOMATIC);
69
70     // Batas atas terkunci rapat di 200 (~78% Duty Cycle maks)
71     myPID.SetOutputLimits(0, 200);
72     myPID.SetSampleTime(10); // Kecepatan hitung PID 10ms
73 }

```

Gambar 4. 8 Inisialisasi Sistem (Setup Function)

Fungsi `setup()` dieksekusi satu kali saat mikrokontroler pertama kali menerima daya. Jalur komunikasi serial diaktifkan via `Serial.begin(9600)`; untuk keperluan pemantauan data. Arah arus data diset melalui `pinMode()`, di mana `pinPWM` dikonfigurasi sebagai *OUTPUT*, sedangkan `pinPotensio` dan `pinSensor` dikonfigurasi sebagai *INPUT*. Pin `pinSaklar` dikonfigurasi sebagai `INPUT_PULLUP` untuk mengaktifkan resistor *internal pull-up* mikrokontroler.

Layar LCD diinisialisasi melalui `lcd.init()`, lampu latar dinyalakan dengan `lcd.backlight()`, dan pesan pembuka (*splash screen*) "BOOST CONVERTER DC-DC DUAL MODE" ditampilkan selama 1,5 detik. Sebelum memasuki siklus komputasi, program melakukan pembacaan awal tegangan melalui instruksi `Input = (analogRead(pinSensor) * 5.0 / 1023.0) * rasioSensor;`. Selanjutnya, algoritma PID diaktifkan secara otomatis melalui `myPID.SetMode(AUTOMATIC)`; dan batas maksimum register keluaran PWM dibatasi pada nilai 0 – 200 menggunakan

instruksi `myPID.SetOutputLimits(0, 200);`. Nilai batas 200 ini setara dengan *Duty Cycle* maksimum $\approx 78,4\%$, yang berfungsi membatasi durasi konduksi MOSFET demi mencegah pembakaran komponen akibat *over-duty*.

e. Pengondisian Sinyal Sensor dan Pembacaan Input

```

46 void loop() {
47     int nilaiADC = analogRead(pinSensor);
48     double teganganPin = (nilaiADC * 5.0) / 1023.0;
49     Input = teganganPin * rasioSensor;
50
51     int nilaiPot = analogRead(pinPotensio);
52     int modeState = digitalRead(pinSaklar);

```

Gambar 4. 9 Pengondisian Sinyal Sensor dan Pembacaan Input

Siklus utama program dieksekusi secara terus-menerus di dalam fungsi `loop()`. Setiap siklus diawali dengan tahap pengondisian sinyal analog yang diperoleh dari sirkuit pembagi tegangan (*voltage divider*). Pembacaan diawali melalui instruksi `int nilaiADC = analogRead(pinSensor);` untuk mengambil data mentah digital ADC 10-bit ($0 - 1023$) pada Pin Analog A1. Selanjutnya, baris kode `double teganganPin = (nilaiADC * 5.0) / 1023.0;` mengonversi nilai diskrit digital tersebut menjadi skala tegangan analog aktual yang diterima pin mikrokontroler ($0 - 5,0$ V). Nilai tegangan pin ini kemudian dikalikan dengan faktor pengali rasio 101,0 ($K = 101$) melalui persamaan `Input = teganganPin * rasioSensor;` untuk merekonstruksi besarnya nilai tegangan fisik keluaran konverter (V_{out}) pada terminal beban.

Setelah proses pengondisian sinyal umpan balik sensor selesai, program melanjutkan ke pembacaan antarmuka masukan dari pengguna. Posisi sudut potensiometer dibaca menggunakan perintah `int nilaiPot = analogRead(pinPotensio);` pada Pin Analog A7 untuk memperoleh nilai acuan skala $0 - 1023$. Bersamaan dengan itu, status logika sakelar pemilih mode dibaca melalui perintah `int modeState = digitalRead(pinSaklar);` pada Pin Digital D4 guna menentukan apakah sistem akan dieksekusi dalam mode jerat terbuka (*Open Loop*) atau mode jerat tertutup (*Closed Loop*) berbasis PID.

f. Pemrosesan Logika Kendali *Dual-Mode*

```

if (modeState == LOW) {
    // MODE 1: OPEN LOOP
    if (lastModeState != LOW) {
        myPID.SetMode(MANUAL);
        lastModeState = LOW;
    }
    Output = map(nilaiPot, 0, 1023, 0, 200);
    OCR1A = (int)Output;
} else {
    // MODE 2: CLOSED LOOP (PID)
    if (lastModeState != HIGH) {
        myPID.SetMode(AUTOMATIC);
        lastModeState = HIGH;
    }
}

```

Gambar 4. 10 Pemrosesan Logika Kendali Dual-Mode

Pemrosesan sinyal pemicu *Pulse Width Modulation* (PWM) dieksekusi menggunakan struktur pengkondisian percabangan if-else berdasarkan nilai status logika dari sakelar pemilih mode. Percabangan logika ini bertugas menentukan mekanisme kendali pensaklaran MOSFET IRLZ44N, baik secara manual jerat terbuka (*Open Loop*) maupun secara otomatis jerat tertutup (*Closed Loop*) berbasis algoritma PID.

Pada mode *Open Loop* yang diaktifkan ketika sakelar bernilai *LOW* (if (modeState == LOW)), sistem beroperasi secara manual tanpa adanya koreksi umpan balik. Nilai mentah pembacaan potensiometer pada skala 0 hingga 1023 dipetakan secara linier menjadi nilai register keluaran PWM rentang 0 hingga 200 melalui instruksi `Output = map(nilaiPot, 0, 1023, 0, 200);`. Nilai variabel `Output` tersebut langsung diumpankan ke Pin Digital D9 melalui perintah `analogWrite(pinPWM, Output);` tanpa melewati siklus komputasi PID. Mekanisme

ini memungkinkan pengguna menyetir persentase *Duty Cycle* pemicuan secara langsung untuk menguji karakteristik dasar konverter.

Sebaliknya, pada mode *Closed Loop* PID yang diaktifkan ketika sakelar bernilai *HIGH* (else), sistem beralih penuh ke mode kendali otomatis. Pembacaan potensiometer dipetakan menjadi target tegangan acuan (*Setpoint*) dalam rentang 12,0 V hingga 50,0 V DC melalui perintah `Setpoint = map(nilaiPot, 0, 1023, 120, 500) / 10.0;`. Selanjutnya, instruksi `myPID.Compute();` dipanggil untuk mengalkulasi besarnya nilai variabel manipulasi Output PWM yang diperlukan secara otomatis berdasarkan selisih galat $e(t) = \text{Setpoint} - \text{Input}$. Sinyal PWM hasil kalkulasi PID ini kemudian dikirimkan ke Pin Digital D9 via `analogWrite(pinPWM, Output);` untuk menyetir waktu konduksi MOSFET secara dinamis, sehingga tegangan keluaran konverter terkunci secara presisi pada nilai target meskipun terjadi fluktuasi beban.

g. Pembaruan Tampilan LCD Secara *Non-Blocking*

```

if (currentMillis - previousMillis >= intervalLCD) {
    previousMillis = currentMillis;

    char strSP[6], strOUT[6];
    dtostrf(Setpoint, 4, 1, strSP);
    dtostrf(Input, 4, 1, strOUT);

    int dutyPercent = (Output / 255.0) * 100.0;
    int pwmVal = (int)Output;

    if (modeState == LOW) {
        lcd.setCursor(0, 0); lcd.print("OPENLOOP V:"); lcd.print(strOUT); lcd.print("V ");
        lcd.setCursor(0, 1); lcd.print("PWM:");
        if (pwmVal < 100) lcd.print(" ");
        if (pwmVal < 10) lcd.print(" ");
        lcd.print(pwmVal);
        lcd.print(" D:");
        if (dutyPercent < 100) lcd.print(" ");
        if (dutyPercent < 10) lcd.print(" ");
        lcd.print(dutyPercent); lcd.print("% ");
    } else {
        lcd.setCursor(0, 0); lcd.print("CLOSED V:"); lcd.print(strOUT); lcd.print("V ");
        lcd.setCursor(0, 1); lcd.print("SP:"); lcd.print(strSP); lcd.print("V D:");
        lcd.setCursor(12, 1);
        if (dutyPercent < 100) lcd.print(" ");
        if (dutyPercent < 10) lcd.print(" ");
        lcd.print(dutyPercent); lcd.print("%");
    }
}
}

```

Gambar 4. 11 Pembaruan Tampilan LCD Secara Non-Blocking

Bagian akhir fungsi `loop()` menangani pembaruan data pada layar LCD secara *non-blocking* menggunakan fungsi pewaktu `millis()`. Melalui kondisi `if (currentMillis - previousMillis >= intervalLCD)`, program hanya akan memperbarui tampilan layar setiap 200 ms sekali tanpa menghentikan eksekusi perhitungan PID.

Di dalam blok pewaktu ini, nilai register PWM dikonversi menjadi persentase *Duty Cycle* melalui persamaan float $\text{dutyPercent} = (\text{Output} / 255.0) * 100.0$. Jika sistem berada pada mode *Open Loop*, baris pertama LCD menampilkan status "OPEN Duty: XX%". Jika berada pada mode *Closed Loop*, baris pertama menyajikan target tegangan "PID SP: XX.XV". Sementara itu, baris kedua LCD secara kontinu menampilkan nilai tegangan keluaran aktual "Vout: XX.XV". Penggunaan teknik pewaktu *non-blocking* ini menjamin tampilan LCD tetap diperbarui secara responsif tanpa mengganggu frekuensi pensaklaran PWM dan kecepatan komputasi kendali.

4.4 Hasil Akhir Alat

Setelah proses perancangan, perakitan mekanik sasis koper, penyolderan komponen elektronika pada papan PCB bolong, serta pengembangan perangkat lunak selesai dilakukan, diperoleh sebuah modul praktikum *Dual-Mode DC-DC Boost Converter* berbasis Arduino Nano yang dapat digunakan sebagai media pembelajaran sistem konversi daya dan kendali PID secara interaktif.

Alat yang telah direalisasikan terdiri dari beberapa bagian utama, yaitu sasis koper aluminium portabel ($32\text{ cm} \times 20\text{ cm} \times 10\text{ cm}$) beralaskan panel akrilik 3 mm, sirkuit daya utama (*power stage*), sirkuit kendali dan sensor, serta antarmuka pengguna. Sirkuit daya utama menggunakan komponen induktor toroid $330\text{ }\mu\text{H}$, sakelar daya MOSFET IRLZ44N yang dilengkapi *heatsink* aluminium, Dioda Schottky SR3150, kapasitor filter $47\text{ }\mu\text{F}/160\text{ V}$. Seluruh komponen daya dan kontrol dirakit secara terintegrasi di atas papan PCB bolong (*perfboard*) dengan penebalan timah solder pada jalur arus besar.

Sistem kendali alat menggunakan mikrokontroler Arduino Nano sebagai pusat pemroses data yang bertugas mengatur seluruh proses pensaklaran PWM berfrekuensi tinggi 62,5 kHz dan komputasi algoritma PID. Mode operasi sirkuit dapat dialihkan secara instan menggunakan sakelar toggle SPDT pada pin D4 antara mode *Open Loop* (manual) dan *Closed Loop* (otomatis). Parameter operasional—seperti persentase *Duty Cycle* manual pada mode *Open Loop* maupun nilai target tegangan (*Setpoint* 12 V – 50 VDC) pada mode *Closed Loop*—dapat diatur secara mudah melalui potensiometer linier $10\text{ k}\Omega$ pada pin A7, kemudian seluruh informasi status mode, *Setpoint*, *Duty Cycle*, dan tegangan keluaran aktual (V_{out}) ditampilkan pada layar LCD 16x2 I2C.

Untuk menjaga keamanan operasional alat dan presisi pembacaan, sirkuit dilengkapi dengan sensor umpan balik pembagi tegangan ($100\text{ k}\Omega/1\text{ k}\Omega$, $K = 101$) pada pin A1 serta fitur proteksi batas *Duty Cycle* maksimal ($\approx 78,4\%$) berbasis *software*. Selain itu, unit kontrol disuplai secara independen menggunakan *charger* HP 5 VDC via kabel USB guna mengisolasi mikrokontroler dari gangguan *switching noise* dan *ground loop*.

Secara keseluruhan, modul *trainer kit Dual-Mode DC-DC Boost Converter* yang telah direalisasikan mampu melakukan proses konversi dan regulasi tegangan searah secara stabil sesuai dengan parameter yang ditentukan. Hasil realisasi alat ini selanjutnya diuji untuk mengetahui kinerja regulasi beban, akurasi pembacaan sensor, respons transien, tingkat kesalahan (*error rate*), serta kestabilan keluaran sebagaimana dibahas pada BAB V.