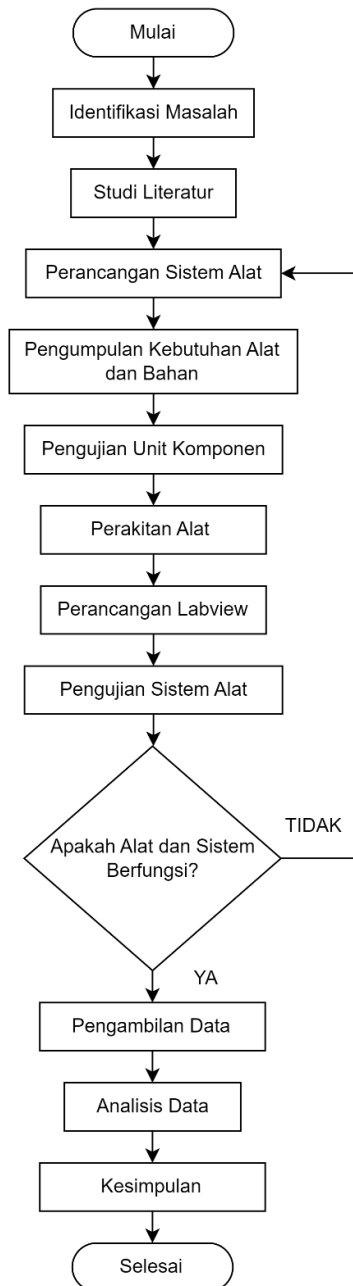


BAB III

METODE PENELITIAN

3.1 *Flowchart* Penelitian

Tahap penelitian dilakukan agar sisten yang dibuat memiliki hasil yang sesuai dengan yang diinginkan. Gambar 3. 1 adalah gambar dari *flowchart* penelitiannya.



Gambar 3. 1 Flowchart Penelitian

3.1.1 Mulai

Tahap pertama adalah memulai penelitian.

3.1.2 Identifikasi Masalah

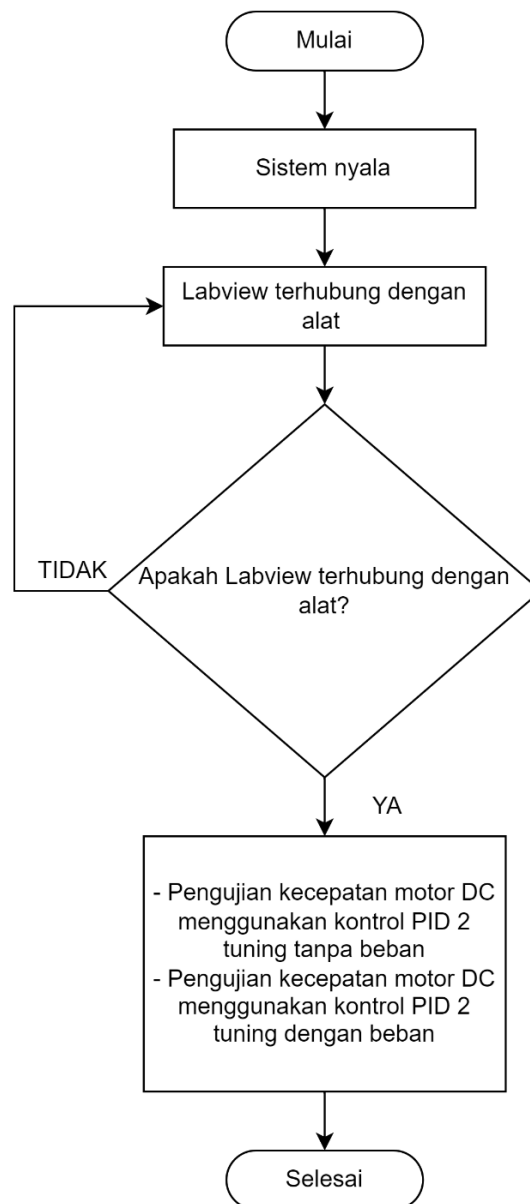
Tahap kedua adalah identifikasi masalah. Tahap ini yaitu menentukan tema yang akan diangkat sebagai topik dari penelitian tugas akhir. Identifikasi masalah dibuat untuk memudahkan mencari tema yang dibuat, yaitu hasil perbandingan pada kecepatan motor DC dengan metode PID menggunakan dua jenis tuning yang berbeda.

3.1.3 Studi Literatur

Tahap ketiga adalah studi literatur. Tahap ini yaitu mencari berbagai referensi pendukung untuk menunjang tugas akhir. Referensi yang digunakan berupa skripsi yang sudah melakukan penelitian tentang ini sebelumnya, dan juga berupa jurnal nasional dan internasional, maupun buku. Hal ini dilakukan agar memudahkan dalam perancangan dan pengujian yang dilakukan supaya tercapai hasil seperti yang diinginkan.

3.1.4 Perancangan Sistem Alat

Tahap keempat adalah perancangan sistem alat. Tahap ini yaitu dilakukan proses perancangan agar sistem alat dapat berfungsi dan terhubung satu sama lain supaya mendapatkan hasil perancangan yang berfungsi sehingga tujuan tercapai. Perancangan sistem alat yaitu meliputi terhubungnya Labview dengan alat yang dibuat sehingga berkesinambungan satu sama lain. Gambar 3. 2 adalah gambar *flowchart* dari sistem yang dibuat.

Gambar 3. 2 *Flowchart* Sistem Alat

Dimulai dengan menyalakan sistem, lalu menghubungkan sistem alat dengan Labview untuk selanjutnya data kecepatan motor terkirim dan terbaca juga oleh Arduino yang mana data juga dapat ditampilkan pada Labview berupa grafik. Selanjutnya pengujian kecepatan motor DC menggunakan kontrol PID dengan dua metode tuning Ziegler-Nichols dan Cohen-Coon juga dua kondisi yaitu tanpa beban dan dengan beban diinputkan parameter K_p , K_i , K_d nya pada Arduino dan ditampilkan grafiknya pada Labview.

3.1.5 Pengumpulan Kebutuhan Alat dan Sistem

Tahap kelima adalah pengumpulan kebutuhan alat dan sistem. tahap ini dilakukan pengumpulan kebutuhan alat, komponen, dan bahan yang dapat membantu terhadap fungsional sistem yang dirancang sehingga mempermudah dalam proses perancangan alat juga sistem, serta meminimalisir pengeluaran biaya yang berlebih. Adapun daftar kebutuhan yang diperlukan dalam perancangan terdaftar pada tabel 3. 1 dan tabel 3. 2.

Tabel 3. 1 adalah tabel kebutuhan komponen yang akan digunakan dalam proses perancangan alat.

Tabel 3. 1 Kebutuhan Komponen

No	Nama Komponen	Spesifikasi	Jumlah
1	Arduino	Nano	1
2	Driver Motor	L298N	1
3	Motor DC	JGA25-370	1
4	Generator DC		1
5	Adaptor	12V	1

Komponen yang akan digunakan untuk proses perancangan alat diantaranya Arduino Nano, Driver Motor L298N, Motor DC JGA25-370, Generator DC, Adaptor 12V, dan LED.

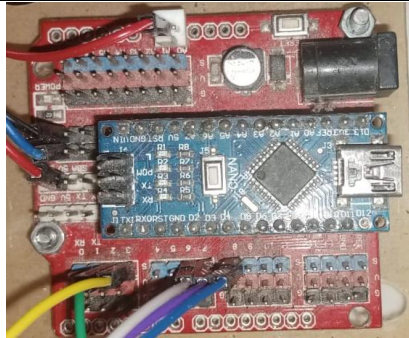
Tabel 3. 2 adalah daftar tabel kebutuhan alat dan bahan yang akan digunakan proses perancangan alat.

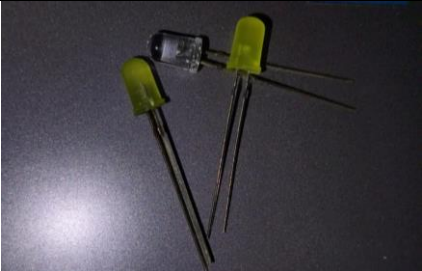
Tabel 3. 2 Kebutuhan Alat dan Bahan

No	Nama Alat dan Bahan	Jumlah
1	Akrilik	1
2	<i>Tachometer</i> Krisbow KW06-563	1
3	Obeng	1
4	Kabel <i>Jumper</i>	10
5	Dudukan Motor	2
6	Tang	1
7	<i>Couple</i> Motor	1
8	<i>Power Supply</i>	1

Alat dan bahan yang akan digunakan untuk proses perancangan alat diantaranya adalah akrilik, *tachometer*, obeng, kabel *jumper*,udukan motor, tang, dan *couple* motor. Tabel 3. 3 adalah tabel pemilihan komponen yang akan digunakan dalam pembuatan alat dalam penelitian.

Tabel 3. 3 Pemilihan Komponen

Nama Komponen	Foto Komponen
Arduino Nano	
Motor DC JGA25-370	
Driver Motor L298N	
Generator DC	

Nama Komponen	Foto Komponen
LED	

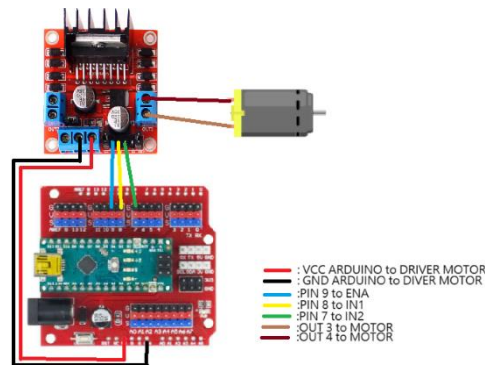
3.1.6 Pengujian Unit Komponen

Tahap keenam adalah pengujian unit komponen. tahap ini dilakukan pengujian setiap unit komponen yang akan digunakan untuk selanjutnya dirakit menjadi alat. Tujuannya adalah untuk mengetahui bahwa komponen yang akan digunakan dalam perancangan alat dapat berfungsi dengan baik, dan dapat digunakan untuk penelitian. Pengujian unit komponen terbagi menjadi pengujian Arduino Nano, Driver Motor L298N, dan Motor DC JGA25-370.

3.1.6.1 Pengujian Unit Arduino Nano

Pengujian unit Arduino nano supaya tahu kondisi dari Arduino Nano yang akan digunakan untuk proses penelitian. Pengujian Arduino akan digunakan untuk menjalankan perintah program yang sudah dibuat untuk melakukan komunikasi data yang menampilkan serial monitor dalam penerimaannya. Pengujian ini agar nantinya dapat terbaca hasil pengujian kecepatan pada unit motor dc.

Pengujian unit Arduino dilakukan dengan memberikan program untuk mengetahui kondisi komponen Arduino dapat berfungsi baik. Gambar 3. 3 adalah *wiring* untuk pengujian pada Arduino.

Gambar 3. 3 *Wiring* Pengujian Arduino Nano

Untuk pemasangan *wiring*nya dihubungkan berdasarkan gambar dengan tabel 3. 4 untuk pinnya. Pengujian dilakukan dengan cara memberikan juga program.

Tabel 3. 4 Pin *Wiring* Pengujian Arduino

Arduino Nano	Driver Motor L298N	Motor DC JGA25-370
VCC	VCC	
GND	GND	
PIN 6	ENA	
PIN 8	IN1	
PIN 7	IN2	
	OUT 3	ENCODER MOTOR
	OUT 2	ENCODER MOTOR

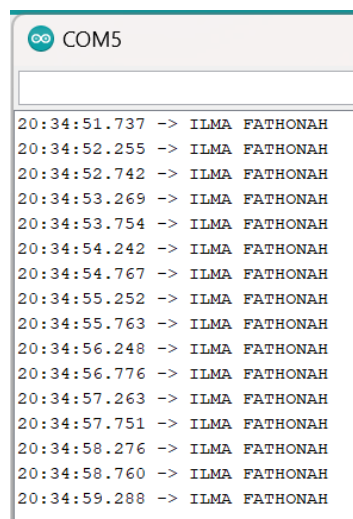
Tabel 3. 5 adalah program untuk pengujian unit Arduino menampilkan komunikasi serial monitor antara program dengan alat yang dirancang.

Tabel 3. 5 Program Pengujian Unit Arduino

Program	Keterangan
<code>void setup() {</code>	Untuk inisialisasi, pengaturan pin input/output, memulai komunikasi serial, dll
<code>delay(2000);</code>	Untuk mengatur jeda waktu antar perintah selama 2000 milidetik atau 2detik untuk ke baris kode berikutnya
<code>Serial.begin(9600);</code>	Untuk memulai komunikasi antara Arduino dan perangkat lain melalui port serial.

Program	Keterangan
	Kecepatan baud rate 9600, berarti data dikirim pada kecepatan 9600 bit per detik.
<code>void loop () {</code>	Fungsi yang akan menjalankan berulang kali setelah setup selesai
<code>Serial.println("ILMA FATHONAH");</code>	Untuk mengirimkan data ke monitor serial berupa tulisan "ILMA FATHONAH"
<code>delay(500);</code>	Untuk memberikan jeda waktu program selama 500 milidetik atau 0.5 detik

Gambar 3. 4 adalah hasil bahwa Arduino berfungsi untuk menampilkan serial monitor, yang mana Arduino dapat mengirim dan menerima data dengan baik.



Gambar 3. 4 Hasil Pengujian Serial Monitor Arduino Nano

Hasil dari program pengujian Arduino Nano diperintah untuk menampilkan serial monitor "ILMA FATHONAH". Tabel 3. 6 adalah tabel hasil pengujian komunikasi serial pada Arduino nano.

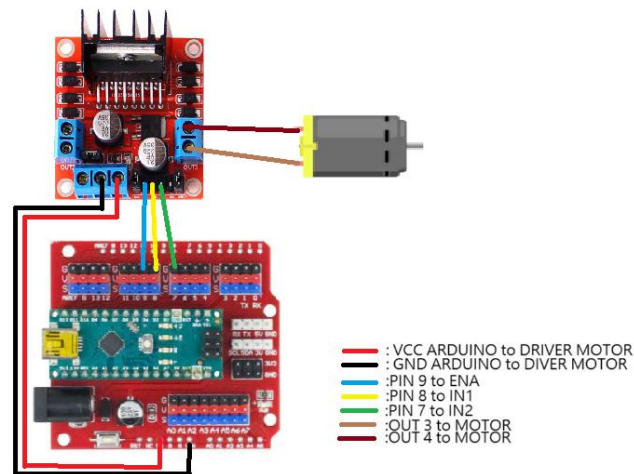
Tabel 3. 6 Hasil Pengujian Arduino Nano

Pengujian ke-	Komunikasi Serial	Keterangan
1	Ya	Baik
2	Ya	Baik
3	Ya	Baik
4	Ya	Baik
5	Ya	Baik
6	Ya	Baik
7	Ya	Baik
8	Ya	Baik

Pengujian ke-	Komunikasi Serial	Keterangan
9	Ya	Baik
10	Ya	Baik

3.1.6.2 Pengujian Unit Driver Motor L298N

Pengujian unit driver motor ini dilakukan supaya tahu kondisi dari driver motor L298N yang akan digunakan untuk penelitian ini berfungsi dengan baik. Pengujian akan digunakan untuk memberi perintah kecepatan pada motor dc dengan kondisi *high* dan *low*. Gambar 3.5 adalah *wiring* untuk pengujiannya.



Gambar 3. 5 *Wiring* Pengujian Driver Motor

Untuk pemasangan *wiring*nya dihubungkan berdasarkan gambar dengan tabel 3. 7 untuk pinnya. Pengujian dilakukan dengan cara memberikan juga program pada driver motor yang dapat mengontrol motor DC.

Tabel 3. 7 Pin *Wiring* Pengujian Driver Motor

Arduino Nano	Driver Motor L298N	Motor DC JGA25-370
VCC	VCC	
GND	GND	
PIN 6	ENA	
PIN 8	IN1	
PIN 7	IN2	
	OUT 3	ENCODER MOTOR

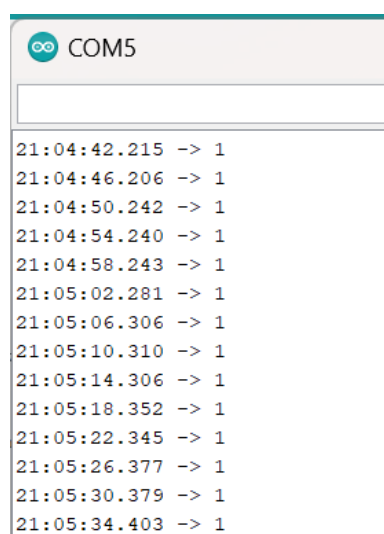
	OUT 2	ENCODER MOTOR
--	-------	---------------

Tabel 3. 8 adalah tabel program untuk pengujian driver motor dengan kondisi *high* dan *low*.

Tabel 3. 8 Program Pengujian Driver Motor

Program	Keterangan
<code>void loop() {</code>	Fungsi berulang yang akan di eksekusi terus-menerus
<code>digitalWrite(IN1, HIGH);</code>	Mengatur pin yang terhubung ke IN1 menjadi HIGH yang mengeluarkan tegangan tinggi untuk mengendalikan perangkat yang terhubung pin IN1
<code>digitalWrite(IN2, LOW);</code>	Mengatur pin yang terhubung ke IN2 menjadi LOW yang akan mengeluarkan tegangan rendah.
<code>delay(2000);</code>	Waktu tunggu 2 detik untuk memberikan waktu agar sinyal dapat dikirim ke perangkat
<code>Serial.println(OUTPUT);</code>	Mencetak nilai output ke serial monitor dan diikuti oleh garis baru
<code>delay(2000);</code>	Waktu tunggu 2 detik sebelum melanjutkan ke iterasi berikutnya dari loop untuk melihat hasil dan perintah selanjutnya

Hasil pengujian adalah hasil dari perintah untuk mengirimkan kondisi yang menunjukkan *high* dan *low* kondisi motor. Gambar 3. 6 adalah hasil serial monitor pengujian driver motor.



Gambar 3. 6 Hasil Pengujian Kondisi Driver Motor

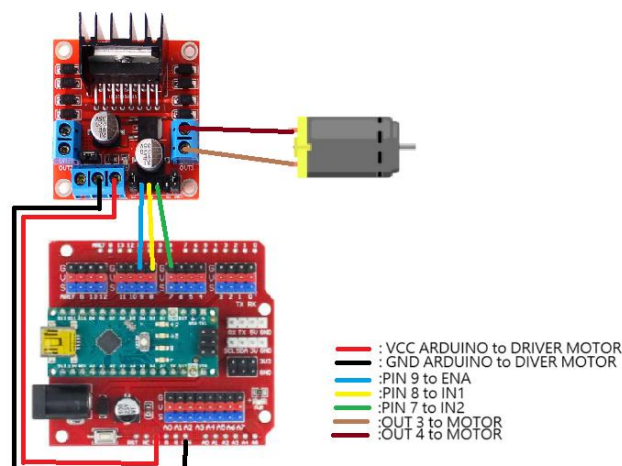
Hasil pengujian menunjukkan angka 1 yang artinya menunjukkan kondisi status motor sedang berputar atau aktif. Tabel 3. 9 adalah tabel hasil pengujian dari driver motor.

Tabel 3. 9 Hasil Pengujian Kondisi Driver Motor

Pengujian ke-	Status	Keterangan
1	1	Aktif
2	1	Aktif
3	1	Aktif
4	1	Aktif
5	1	Aktif
6	1	Aktif
7	1	Aktif
8	1	Aktif
9	1	Aktif
10	1	Aktif

3.1.6.3 Pengujian Unit Motor DC JGA25-370

Pengujian unit Motor DC JGA25-370 dilakukan supaya tahu kondisi motor dc yang akan digunakan untuk penelitian. Pengujian unit motor DC JGA-25-370 dilakukan dua pengujian, yaitu pengujian pembacaan encoder dimana PWM berubah menjadi RPM, dan pengujian pembaca *input output* RPM pada motor. Gambar 3. 7 adalah *wiring* untuk pengujian pada motor DC.



Gambar 3. 7 Wiring Pengujian Motor DC JGA-25-370

Untuk pemasangan *wiringnya* dihubungkan berdasarkan gambar dengan tabel 3. 10 untuk pinnya. Pengujian dilakukan dengan cara memberikan juga program.

Tabel 3. 10 Pin *Wiring* Pengujian Motor DC

Arduino Nano	Driver Motor L298N	Motor DC JGA25-370
VCC	VCC	
GND	GND	
PIN 6	ENA	
PIN 8	IN1	
PIN 7	IN2	
	OUT 3	ENCODER MOTOR
	OUT 2	ENCODER MOTOR

Tabel 3. 11 adalah tabel program dengan keterangannya untuk pengujian motor DC dengan pengujian PWM ke RPM.

Tabel 3. 11 Program Pengujian PWM ke RPM

Program	Keterangan
// Variabel global <code>volatile float velocity_i = 0;</code> <code>volatile long prevT_i = 0;</code>	Menyimpan dan menghitung nilai kecepatan motor berdasarkan pembacaan encoder berdasarkan selisih waktu
<code>void loop() {</code> <code>int pwr = 255;</code> <code>int dir = 1;</code>	Variabel yang digunakan untuk menentukan kekuatan atau kecepatan motor menggunakan nilai PWM dari 0-255 dengan variabel 1 untuk menentukan arah putaran motor
<code>float velocity = 0;</code> <code>noInterrupts();</code> <code>velocity = velocity_i;</code> <code>interrupts();</code>	Menyimpan nilai kecepatan motor berdasarkan nilai dari interrupt sementara. Nilai velocity digunakan dalam perhitungan berdasarkan encoder
// Hitung RPM <code>float rpm = velocity / 88.5 * 60.0;</code>	Rumus RPM yang mengindikasikan kecepatan motor berputar dalam 1 menit
// Tampilkan data ke Serial Monitor <code>Serial.print(pwr);</code> <code>Serial.print(",");</code> <code>Serial.println(rpm);</code>	Untuk menampilkan nilai PWR yang diinput dan nilai hasil RPM yang dihitung
<code>void readEncoder() {</code> <code>int b =</code> <code>digitalRead(ENCB);</code> <code>int increment = (b > 0)</code> <code>? 1 : -1;</code>	Membaca sinyal encoder untuk menghitung arah putaran motor yang digunakan untuk mendeteksi rotasi dan arah putaran motor
// Hitung deltaT untuk kecepatan	Menghitung kecepatan motor berdasarkan waktu yang dibutuhkan untuk setiap

Program	Keterangan
<pre> long currT = micros(); float deltaT = ((float)(currT - prevT_i)) / 1.0e6; if (deltaT > 0) { velocity_i = increment / deltaT; } prevT_i = currT; } </pre>	perubahan sinyal encoder sehingga pengukuran revolusi perdetik kecepatan motor lebih akurat.

Tabel 3. 12 adalah tabel hasil pengujian dari percobaan PWM menjadi RPM pada motor dengan dilakukan beberapa kali pengujian dengan kombinasi nilai PWM yang berbeda sehingga menghasilkan nilai RPM yang berbeda juga. Sebagai pengukur pembandingnya digunakan tachometer Krisbow KW06-563 untuk pengukuran lainnya.

Tabel 3. 12 Hasil Pengujian PWM ke RPM

PWM	Nilai RPM Sensor	Nilai RPM Krisbow KW06-563	Error
0	0	0	0%
50	158.11	154	2.66%
100	607.5	596.5	1.84%
150	763.48	746.2	2.31%
200	834.93	813.1	2.68%
255	911.24	876.6	3.95%
Rata-rata Error			2.24%

Pengujian PWM ke RPM dilakukan pengujian beberapa kali dengan memasukkan nilai PWM dari 0-255 sehingga menghasilkan nilai RPM dan dibandingkan nilai keluaran kecepatannya menggunakan alat tachometer Krisbow KW06-563. Sehingga dihasilkan nilai rata-rata error antara pembacaan sensor dengan pengukuran tachometer sebesar 2.24409%.

Tabel 3. 13 adalah tabel program dengan keterangannya untuk pengujian motor DC dengan pengujian pembacaan kecepatan RPM pada motor dengan memasukkan nilai *setpoint*.

Tabel 3. 13 Program Pengujian Pembacaan RPM

Program	Keterangan
<pre>// Variabel global volatile float velocity_i = 0; volatile long prevT_i = 0;</pre>	Menyimpan dan menghitung nilai kecepatan motor berdasarkan pembacaan encoder berdasarkan selisih waktu
<pre>void loop() { float sp = 1000; //setpoint RPM</pre>	Fungsi yang dijalankan berulang dengan menyimpan nilai <i>setpoint</i> RPM sebagai target
<pre>float velocity = 0; noInterrupts(); velocity = velocity_i; interrupts();</pre>	Menyimpan nilai kecepatan motor berdasarkan nilai dari interrupt sementara. Nilai velocity digunakan dalam perhitungan berdasarkan encoder
<pre>// Hitung RPM float rpm = velocity / 88.5 * 60.0;</pre>	Rumus RPM yang mengindikasikan kecepatan motor berputar dalam 1 menit
<pre>int dir = 1; int pwr = 255;</pre>	Pengatur arah putaran motor dan mengatur kecepatan motor menggunakan PWM
<pre>if (rpm < sp){ dir = 1; pwr = 255; } else if (rpm > sp){ dir = 0; pwr = 0; }</pre>	Jika RPM kurang dari nilai <i>setpoint</i> motor berputar lebih lambat dari yang diinginkan. Jika RPM lebih dari nilai <i>setpoint</i> motor berputar lebih cepat dari yang diinginkan. Dengan pwr 255 yang mengatur kecepatan motor ke nilai maksimum mendekati nilai <i>setpoint</i> , dan pwr 0 yang mengatur motor ke 0 yang mana motor dimatikan atau tidak berputar menurunkan RPM
<pre>// Tampilkan data ke Serial Monitor Serial.print(sp); Serial.print(","); Serial.print(rpm); Serial.print(","); Serial.println(pwr);</pre>	Untuk menampilkan nilai <i>setpoint</i> , RPM, dan pwr pada serial monitor
<pre>void readEncoder() { int b = digitalRead(ENCB); int increment = (b > 0) ? 1 : -1;</pre>	Membaca sinyal encoder untuk menghitung arah putaran motor yang digunakan untuk mendeteksi rotasi dan arah putaran motor
<pre>// Hitung deltaT untuk kecepatan long currT = micros(); float deltaT = ((float) (currT - prevT_i)) / 1.0e6; if (deltaT > 0) { velocity_i = increment / deltaT; }</pre>	Menghitung kecepatan motor berdasarkan waktu yang dibutuhkan untuk setiap perubahan sinyal encoder sehingga pengukuran revolusi perdetik kecepatan motor lebih akurat.

Program	Keterangan
<pre>prevT_i = currT; }</pre>	

Tabel 3. 14 adalah tabel hasil pengujian dari pengujian RPM pada motor DC yang diinputkan menjadi nilai *setpoint* dengan menghasilkan *output* pada motor yang juga dibandingkan nilai kecepatannya dengan menggunakan tachometer Krisbow KW06-563.

Tabel 3. 14 Hasil Pengujian RPM Motor DC

Percobaan ke-	Setpoint RPM	Nilai RPM Serial Monitor	Tachometer Krisbow KW06-563	Error	Rata-rata Error
1	1000	856.02	833.5	2.701%	2.284%
	1000	860.36	840.7	2.338%	
	1000	843.24	828.2	1.815%	
2	800	792.02	787.7	0.548%	0.413%
	800	788.33	784	0.552%	
	800	766.93	768	0.139%	
3	600	566.86	566.2	0.116%	0.370%
	600	590.56	590.2	0.060%	
	600	572.61	567.3	0.936%	
4	400	380.03	376.7	0.883%	2.001%
	400	372.51	356.4	4.520%	
	400	373.33	375.7	0.630%	
5	200	187.49	179.5	4.451%	4.474%
	200	183.83	175.5	4.746%	
	200	199.17	190.6	4.496%	
6	0	0	0	0%	0%
	0	0	0	0%	
	0	0	0	0%	

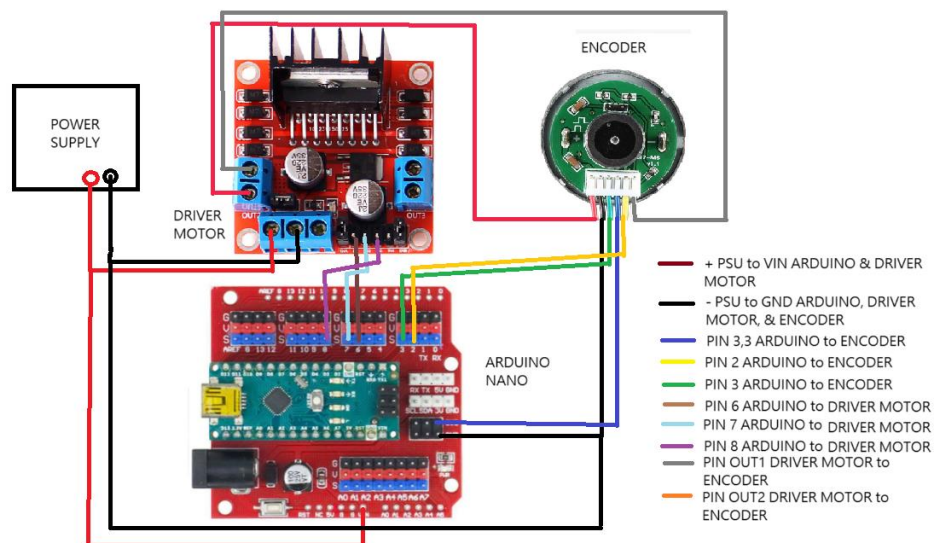
Hasil pengujian nilai RPM dengan memasukkan nilai *setpoint* dari 0-1000rpm dengan selisih nilai 200 dan membandingkan hasil kecepatan RPMnya dengan nilai pengukuran tachometer Krisbow KW06-563 maka didapatkan nilai rata-rata error pada setiap percobaannya adalah untuk *setpoint* 0RPM nilai rata-rata error 0%, *setpoint* 200RPM nilai rata-rata error 4.474%, *setpoint* 400RPM nilai rata-rata error 2.001%, *setpoint* 600RPM nilai rata-rata error

0.370%, *setpoint* 800RPM nilai rata-rata error 0.413%, *setpoint* 1000RPM nilai rata-rata error 2.284%.

3.1.7 Perakitan Alat

Tahap ketujuh adalah perakitan alat. Tahap ini dilakukan perakitan komponen menjadi alat sesuai dengan *wiring*, sehingga alat tersebut berfungsi dan sesuai dengan keinginan untuk selanjutnya dapat dilakukan proses pengujian penelitian tugas akhir.

Perakitan alat dibuat berdasarkan perancangan yang sudah dibuat sesuai dengan *wiring* dari gambar 3. 8 sehingga menjadi alat yang dapat digunakan untuk pengujian perbandingan nilai kecepatan pada motor DC dengan pengendali PID menggunakan metode Ziegler-Nichols dan Cohen-Coon.



Gambar 3. 8 *Wiring* Perakitan Alat

Tabel 3. 15 adalah tabel pin *wiring* untuk proses perakitan sehingga alat mencapai tujuan seperti yang diinginkan.

Tabel 3. 15 Pin *Wiring* Perakitan Alat

Arduino Nano	Driver Motor L298N	Motor DC JGA25-370
VCC	VCC	
GND	GND	GND MOTOR
PIN 6	ENA	

PIN 8	IN1	
PIN 7	IN2	
	OUT 3	ENCODER MOTOR
	OUT 2	ENCODER MOTOR
3.3V		VCC MOTOR

Setiap komponen yang terhubung antara Arduino nano, driver motor, motor DC JGA25-370 melakukan komunikasi satu sama lain. Arduino nano adalah pengendali utama dari sistem. Arduino bertugas untuk memproses data, memberikan sinyal kontrol, dan mengendalikan sistem sesuai intruksi yang diprogram. Arduino yang terhubung dengan motor DC JGA25-370 menghasilkan gerakan berupa kecepatan putaran motor yang dapat diatur melalui sinyal PWM Arduino. Motor DC dikontrol juga oleh driver motor L298N sebagai pengendali kecepatannya. Driver motor L298N adalah sebagai penghubung antara Arduino dan motor juga menerima sinyal PWM Arduino yang dapat mengatur kecepatan motor.

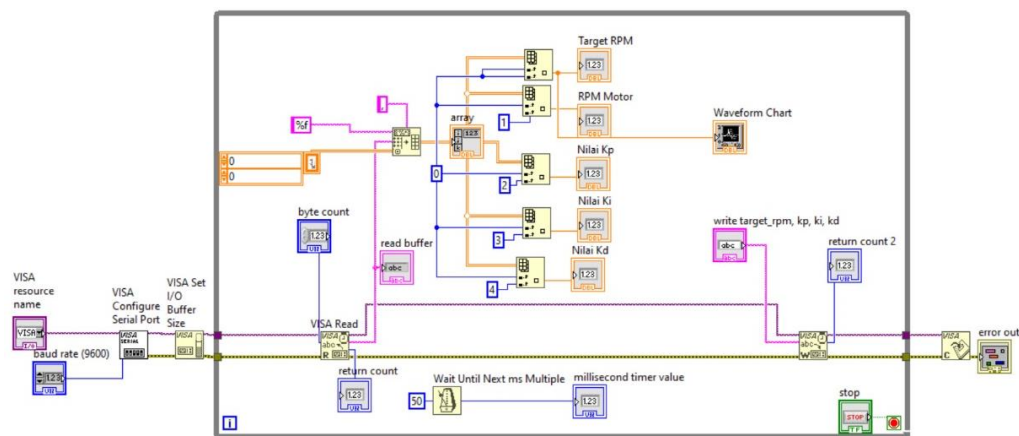
3.1.8 Perancangan Labview

Tahap kedelapan adalah perancangan Labview. Tahap ini dilakukan perancangan supaya pengujian pada hasil alat rancangan, proses tuning PID pada kecepatan motor dc nya dapat dilakukan melalui labview.

Perancangan Labview sebagai *user interface* digunakan untuk membaca kecepatan pada motor DC dan mengirimkan data dari Arduino ke Labview. Penggunaan Labview sebagai *user interface* juga untuk mengimplementasikan penampil kecepatan pada motor DC. Kecepatan motor akan ditampilkan dalam bentuk grafik atau indikator numerik dengan menunjukkan berapa banyak sinyal PWM yang dikirim ke motor untuk mengatur kecepatannya.

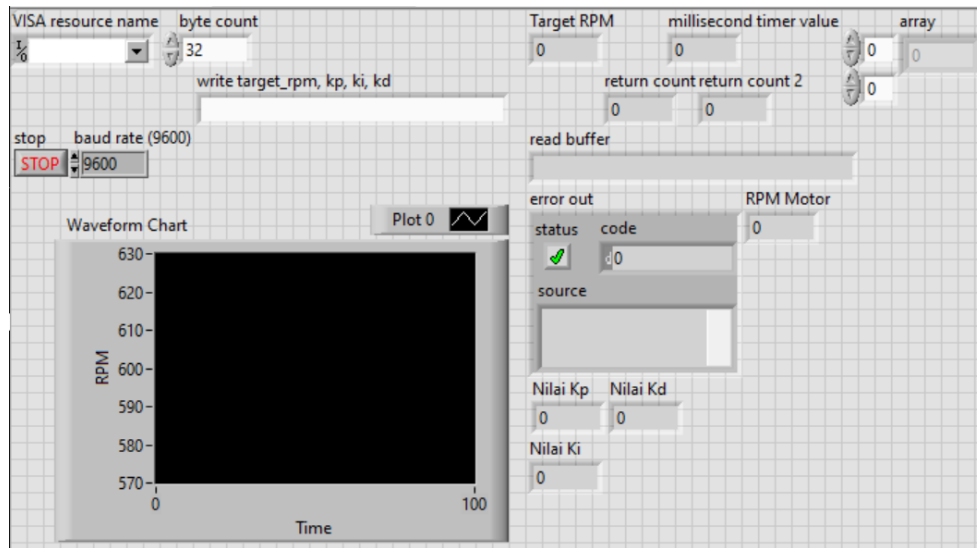
Untuk menghubungkan antara program dari Arduino dan Labview perlu mengunduh *library* encoder pada Arduino agar dapat terhubung ke Labview. Berikutnya adalah membuat blok diagram pada jendela block diagram. Pada front panel akan memunculkan palet control untuk menampilkan hasil dari rangkaian block diagram. Jendela diagram blok berfungsi untuk membangun pemrograman pada Labview menggunakan representasi grafis.

Gambar 3. 9 adalah diagram block dari penelitian ini yang berfungsi sebagai pemroses data.



Gambar 3. 9 Diagram Block Labview

Jendela front panel mirip panel kontrol dimana pengguna dapat memasukkan data melalui elemen kontrol dan elemen indikator akan menampilkan hasil atau status dari data. Gambar 3. 10 adalah front panel dari penelitian ini yang berfungsi sebagai penampil data.



Gambar 3. 10 Front Panel Labview

Hasil rancangan diimplementasikan pada penelitian dimana alat, Arduino, dan Labview dapat terhubung. Pengujian berhasilnya Labview terhubung adalah grafik pada Labview muncul atau menampilkan hasil pembacaan grafiknya.

3.1.9 Pengujian Sistem Alat

Tahap kesembilan adalah pengujian sistem alat. Tahap ini dilakukan pengujian pada sistem alat yang sudah dirancang dan diberi program untuk mengetahui perakitan alat dan program yang dibuat sudah sesuai atau tidak. Hasil pengujian sistem alat yaitu untuk memudahkan proses pengambilan data.

Pengujian sistem alat merupakan hasil pengujian dari sistem alat yang terhubung dengan Labview, dan juga pembacaan hasil kecepatan pada motor DC. Yang mana Labview dapat menampilkan grafik pembacaan kecepatan motor DC yang diuji dengan beberapa pengujian, dan juga pengujian pada penerapan beberapa metode PID yang akan dilakukan atau diterapkan pada pengujian motor.

3.1.10 Pengambilan Data

Tahap kesepuluh adalah pengambilan data. Tahap ini dilakukan proses pengambilan data pada saat pengujian berupa pengujian kecepatan motor dc menggunakan sistem PID dengan tuning dua metode dan dua kondisi ketika tanpa beban atau dengan beban.

3.1.11 Analisis Data

Tahap kesebelas adalah analisis data. Tahap ini dilakukan analisa data hasil dari pengambilan data saat pengujian yang sudah dilakukan.

3.1.12 Kesimpulan

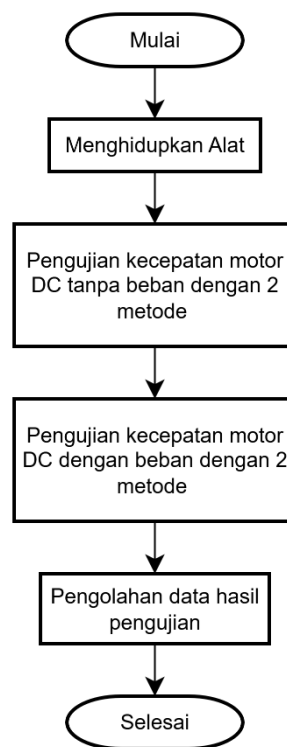
Tahap keduabelas adalah Kesimpulan. Tahap ini dilakukan memberi kesimpulan pada data hasil analisis terhadap alat yang sudah dirancang dan diuji.

3.1.13 Selesai

Tahap ketigabelas adalah selesai. penelitian selesai setelah semua tahap sudah memenuhi spesifikasi.

3.2 *Flowchart* Pengujian Sistem Alat

Tahapan ini adalah tahapan pengujian pada sistem alat yang dirancang untuk pengujian sistem pada alat yang sudah dibuat. Gambar 3. 11 adalah gambar dari *flowchart* pengujian pada sistem alat.



Gambar 3. 11 *Flowchart* Pengujian Sistem Alat

Flowchart pengujian sistem alat bertujuan supaya sistem dan alat yang sudah dirancang berhasil sesuai dengan yang diinginkan dan alat bekerja secara optimal agar dapat melakukan pengujian kecepatan motor DC dengan menggunakan *user interface* Labview untuk tuning melakukan 2 metode tuning PID Ziegler-Nichols dan Cohen-Coon. Pengujian kecepatan juga dilakukan dua keadaan pengujian, tanpa beban dan dengan menggunakan beban berupa generator DC.

Hasil pengujian akan ditampilkan pada Labview berupa grafik hasil tuning PID. Untuk tabel pengujian yang akan dilakukan terbagi menjadi tiga tahapan proses pengujian. Diantaranya:

1. Pengujian untuk perancangan rangkaian sistem kendali pada kecepatan motor DC dengan *user interface* Labview.
2. Pengujian tuning PID untuk kecepatan motor DC menggunakan metode Ziegler-Nichols dan Cohen-Coon tanpa beban.

3. Pengujian tuning PID ketika motor DC diberikan beban.

Adapun tabel-tabel dibawah adalah tabel yang selanjutnya akan digunakan pada proses pengujian sistem alat.

1. Pengujian perancangan kecepatan motor DC dengan *user interface* Labview

Pengujian perancangan kecepatan motor DC dengan *user interface* Labview untuk mengetahui jika *user interface* Labview berfungsi dan terhubung ke alat.

2. Pengujian tuning PID untuk kecepatan motor DC dengan 2 metode tanpa beban.

Pengujian kecepatan motor menggunakan sistem kontrol PID dengan 2 metode tuning yang digunakan yaitu metode Ziegler-Nichols dan Cohen-Coon tanpa beban.

3. Pengujian tuning PID ketika motor DC diberikan beban.

Pengujian kecepatan menggunakan beban untuk mengetahui respon dari 2 metode tuning pada motor DC.

Adapun tabel-tabel dibawah ini adalah tabel untuk selanjutnya dilakukan proses pengujian pada sistem alat untuk mencatat nilai kecepatan motor DC menggunakan metode PID dengan tuning Ziegler-Nichols dan Cohen-Coon.

1. Pengujian Perancangan Rangkaian Sistem Yang Terhubung Dengan UI LabView

Pengujian perancangan rangkaian sistem yang terhubung dengan LabView dilakukan pengujian sistem komunikasi antara sistem alat dan Labview sudah terhubung sehingga LabView dapat menampilkan hasil grafik dari kecepatan motor DC pada alat.

2. Pengujian Kecepatan Motor DC Menggunakan 2 Metode Tuning Tanpa Beban

Pengujian kecepatan Motor DC menggunakan 2 metode yaitu metode tuning Ziegler-Nichols dan Cohen-Coon untuk menemukan nilai parameter PID yang sesuai dengan tabel penelaan pada 2 metode tersebut.. Pengujiannya adalah per parameter P, PI, PID pada metode Ziegler-Nichols dan P, PI, PD, PID pada metode Cohen-Coon. Adapun tabel 3. 16 sampai tabel 3. 22 adalah tabel untuk pengujian kecepatan motor menggunakan 2 metode tuning tanpa menggunakan beban.

Tabel 3. 16 Pengujian Metode Ziegler-Nichols Parameter P Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 17 Pengujian Metode Ziegler-Nichols Parameter PI Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 18 Pengujian Metode Ziegler-Nichols Parameter PID Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 19 Pengujian Metode Cohen-Coon Parameter P Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 20 Pengujian Metode Cohen-Coon Parameter PI Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 21 Pengujian Metode Cohen-Coon Parameter PD Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 22 Pengujian Metode Cohen-Coon Parameter PID Tanpa Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

3. Pengujian Kecepatan Motor DC Menggunakan 2 Metode Tuning Dengan Beban

Pengujian kecepatan menggunakan beban bertujuan untuk mengetahui sistem kontrol PID dapat mempertahankan kecepatan motor walaupun ada variasi dalam beban yang diterimanya karena kehadiran beban dapat mempengaruhi performa motor dalam hal kestabilan kecepatan dan responnya. Metode PID pengujiannya menggunakan dua metode tuning Ziegler-Nichols dan Cohen-Coon pada parameter PID sebagai pembanding metode yang lebih optimal. Adapun tabel 3. 23 dan tabel 3. 24 adalah tabel untuk pengujian kecepatan motor menggunakan 2 metode tuning dengan menggunakan beban. Adapun beban yang diberikan adalah *input* tegangan

sebesar 1 V 5A yang diberikan oleh *power supply* kepada generator DC yang sudah di-*couple* dengan motor DC.

Tabel 3. 23 Pengujian Metode Ziegler-Nichols Parameter PID Menggunakan Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>

Tabel 3. 24 Pengujian Metode Cohen-Coon Parameter PID Menggunakan Beban

RPM	Karakteristik			
	<i>Rise-Time</i>	<i>Overshoot</i>	<i>Settling-Time</i>	<i>Steady-State Error</i>