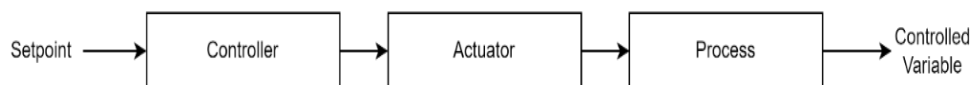


BAB II

LANDASAN TEORI

2.1 Sistem Kontrol

Sistem kendali atau sistem kontrol yaitu sistem yang memiliki fungsi mengatur, memerintah, dan mengendalikan proses pada industri. Sistem kendali yang baik adalah sistem yang mempunyai respon cepat, peka dengan perubahan sistem, mempunyai kesalahan pengendali yang kecil dan dapat mencapai *set point* walaupun ada gangguan (Lestari & Abdul Hadi, 2012). Sistem kontrol mempunyai kontroler dan *actuator*. *Input* yang biasa disebut *setpoint* dan *actuator* yaitu alat elektromekanik yang mengambil sinyal dari kontroler dan diubah dalam mekanik. Kebanyakan sistem kontrol menggunakan istilah *error* sebagai *feedback* bagi sistem. Dimana *error* itu adalah perbedaan dari nilai *setpoint* dan *present value* (Mauludin & Kurniawan, 2001). Peranan sistem kontrol memiliki peran yang penting dalam perkembangan ilmu dan teknologi. Sistem kontrol memerlukan komputer sebagai perangkat pengendali dan monitoring secara *real time* untuk mendapatkan data yang lebih presisi dan akurat (Agung, F, S; Putranto, 2015). Umumnya, gambar 2. 1 adalah diagram blok dari sistem kontrol.



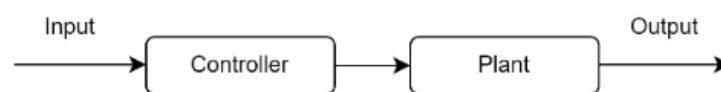
Gambar 2. 1 Diagram Blok Sistem Kontrol

Sumber: (Pribadi)

Sistem kontrol dibagi menjadi dua bagian, yaitu sistem kontrol *loop* terbuka dan sistem kontrol *loop* tertutup. Perbedaan dari keduanya yaitu ada atau tidak adanya *feedback* pada sistem. Suatu sistem yang memiliki *feedback outputnya* akan berpengaruh pada proses kontrol (Mauludin & Kurniawan, 2001).

2.1.1 Sistem *Loop* Terbuka

Sistem *loop* terbuka yaitu sistem yang tidak memiliki umpan balik (*feedback*), dimana suatu sinyal masukan diberikan kepada sistem, keluarannya bertindak sebagai penggerak dan dan pengendali proses untuk menghasilkan keluaran yang diinginkan (Saputri, 2016). Gambar 2. 2 adalah bentuk diagram blok untuk sistem kontrol *open loop*.

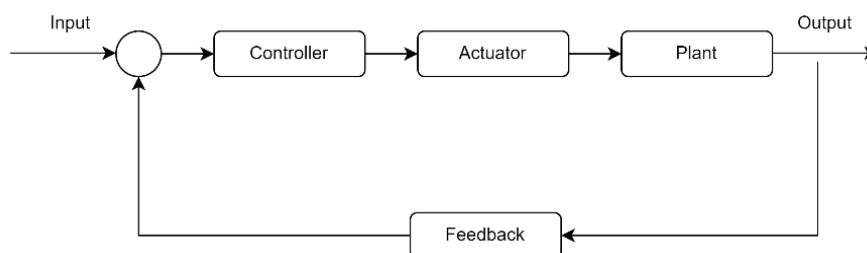


Gambar 2. 2 Diagram Blok *Open Loop*

Sumber: (Pribadi)

2.1.2 Sistem *Loop* Tertutup

Sistem *loop* tertutup yaitu sistem yang memiliki umpan balik (*feedback*) dimana keluaran yang dihasilkan dan sinyal *input* dimasukkan ke dalam sistem, sehingga hasil dari sinyal *output* dengan sinyal *input* disebut *feedback*. Sinyal *feedback* diumpankan pada *controller* untuk memperkecil kesalahan sehingga nilai keluaran sistem semakin mendekati harga yang diinginkan (Saputri, 2016). Gambar 2. 3 adalah bentuk diagram blok untuk sistem kontrol *close loop*.

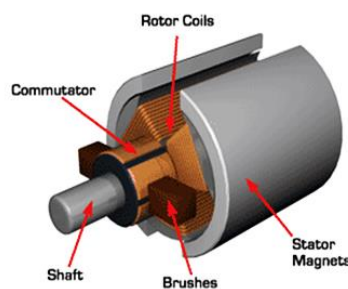


Gambar 2. 3 Diagram Blok Close Loop

Sumber: (Pribadi)

2.2 Motor DC

Motor DC (arus searah) adalah salah satu jenis motor listrik yang banyak digunakan di lingkungan industri dan kehidupan sehari-hari. Motor DC diaplikasikan pada peralatan portabel. Sumber masukan motor DC menggunakan daya listrik tegangan dan arus dimana keluarannya berbentuk daya mekanik yaitu berupa kecepatan rotor dan torsi. Pengendalian motor DC memiliki karakteristik variabel yang berperan sebagai fungsi waktu pada kondisi transien dan tunaknya adalah hal yang perlu diperhatikan. Sumber motor DC selain menggunakan daya listrik bisa juga menggunakan sumber jala-jala AC melalui sistem penyearah. Karakteristik penyearah memiliki karakter yang berbeda dan mempengaruhi tegangan masukan pada motor DC sehingga akan mempengaruhi bentuk gelombang arus yang dihasilkan, torsi dan kecepatan rotor (Hudati et al., 2021). Gambar 2. 4 adalah gambar dengan menunjukkan bagian-bagian dari motor DC yang pada umumnya memiliki 5 bagian konstruksi yang terdapat pada motor DC.

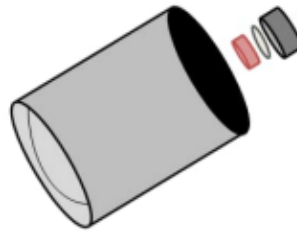


Gambar 2. 4 Motor DC
Sumber: (Hudati et al., 2021)

Umumnya, konstruksi pada motor DC meliputi:

a. Stator

Gambar 2. 5 adalah gambar dari bagian motor DC yang bernama stator yang berfungsi untuk menghasilkan medan magnet.



Gambar 2. 5 Stator

Sumber: (Carin et al., 2018)

Bagian stator adalah bagian motor DC yang diam. Bagian stator itu didalamnya terdapat magnet permanen yang dapat menghasilkan medan magnet untuk memutar rotor. Stator disebut sebagai medan magnet utama, karena ketika stator diberi catu daya DC pada kumparan medan akan menghasilkan medan magnet konstan.

b. Rotor

Gambar 2. 6 adalah gambar bagian rotor dari sebuah motor DC untuk menghasilkan gerakan pada motor DC.



Gambar 2. 6 Rotor

Sumber: (Carin et al., 2018)

Bagian rotor adalah bagian motor DC yang bergerak. Bagian rotor merupakan bagian motor DC yang terbentuk dari inti besi yang dililitkan kumparan. Kumparan yang dimaksud adalah kumparan saat GGL induksi pada motor dihasilkan. Rotor digunakan sebagai penggerak.

Kecepatan putar rotor tergantung dari sumber tegangan dan medan magnet yang dihasilkan. Jika sumber tegangan lebih besar, maka putaran rotor akan semakin cepat begitupun sebaliknya, karena antara keduanya berbanding lurus.

c. *Brushes*

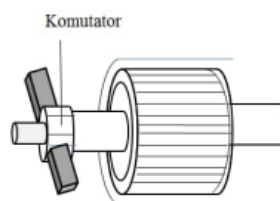
Bagian *brushes* terletak pada sisi komutator sebagai penyalur listrik ke motor. Bahan *brushes* terbuat dari karbon, logam graphite yang dilengkapi pegas penekan. Bahan *brushes* memiliki nilai konduktivitas yang tinggi untuk mengurangi rugi-rugi listrik sehingga menghasilkan nilai koefisien gesekan dan mengurangi keausan (Hudati et al., 2021).

d. *Shaft*

Shaft/batang biasanya berada di tengah motor untuk menahan beban aplikasi. Batang ini melekat pada pelat yang ditempelkan pada poros. Batang ini biasanya terbuat dari logam yang dikeraskan atau baja.

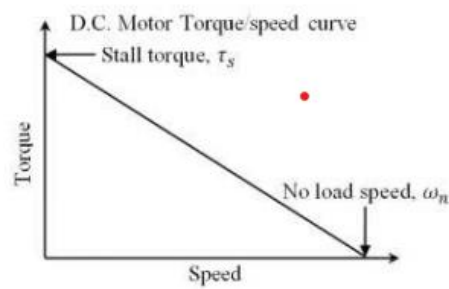
e. Komutator

Komutator adalah tembaga yang dikeraskan dan diisolasi dengan sejenis mika. Gambar 2. 7 adalah gambar bagian motor DC bernama komutator yang berfungsi untuk mengumpulkan arus induksi dari konduktor jangkar dan mengkonversinya menjadi arus searah melalui *brushes*.



Gambar 2. 7 Komutator
Sumber: (Carin et al., 2018)

Karakteristik yang dimiliki motor DC digambarkan melalui kurva daya dan kurva torsi/kecepatan seperti yang ditunjukkan pada gambar 2. 8.



Gambar 2. 8 Kurva Karakteristik Motor DC

Sumber: (Darmana et al., 2018)

Dari kurva itu dapat dianalisa batasan-batasan kerja dari motor serta kerja optimum dari motor tersebut. Hubungan antara torsi dan kecepatan motor DC terlihat bahwa torsi berbanding terbalik dengan kecepatan putaran. Dapat disimpulkan bahwa besar torsi yang dihasilkan motor dengan kecepatan putaran motor terdapat *trade off* (Darmana et al., 2018).

2.2.1 Motor DC JGA25-370



Gambar 2. 9 Motor DC JGA25-370

Sumber: (Seed Studio, n.d.)

Motor DC JGA25-370 pada gambar 2.9 ini merupakan motor roda gigi yang menggunakan satu set roda gigi untuk mengubah kecepatan tinggi asli dan torsi rendah dari motor menjadi kecepatan rendah dan torsi tinggi. Berikut spesifikasi dari motor DC JGA25-370. (Seed Studio, n.d.)

Tabel 2. 1 adalah tabel spesifikasi dari Motor DC JGA25-370.

Tabel 2. 1 Spesifikasi Motor DC JGA25-370

V	<i>No-Load</i>		<i>Maximum Efficiency Pointed</i>				<i>Blockage</i>	
	Rpm	A	Rpm	A	Kg.cm	W	Kg.cm	A
6	190	0.2	133	0.5	0.75	1.1	4.0	2.1
12	350	0.1	245	0.65	1.4	2.4	5.2	2.2

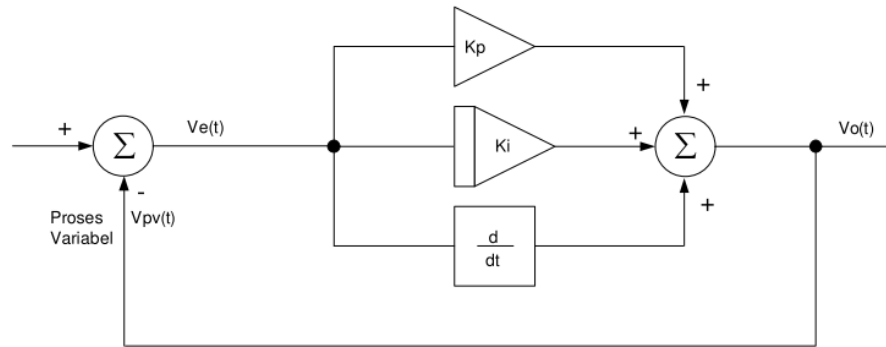
2.3 Kontroler PID

Kontroler PID (*Proportional-Integral-Derivatif*) adalah sebuah pengendali yang populer saat ini karena keefektifannya dan sederhana dalam pengimplementasiannya juga penggunaannya. Konfigurasi standar dari kontroler PID mempunyai parameter K_p , K_i , K_d yang ditentukan agar karakteristik *plant* sesuai dengan kriteria yang diharapkan. Spesifikasi umum dalam desain adalah *rise-time*, *settling-time*, *maximum overshoot* dan *error steady state* terhadap *input* yang diberikan (Palang et al., n.d.).

Pengendali PID terdiri dari tiga koefisien yaitu *Proporsional*, *Integral*, dan *Derivatif*. *Proporsional* adalah komponen yang membandingkan besar antara *set point* dan variabel proses, dimana perbedaan keduanya disebut *error*. Nilai P menentukan respon keluaran terhadap eror yang terjadi. Umumnya, dengan menaikkan nilai proporsional maka kecepatan dari respon akan meningkat namun menimbulkan osilasi yang besar pula. *Integral* berguna untuk mengurangi nilai *steady-state error* atau disebut nilai eror pada kondisi tetap. Pada komponen integral, eror pada setiap waktu akan terus diakumulasikan dan menyebabkan respon keluaran meningkat seiring berjalannya waktu. *Integral* bisa disebut sebagai penjumlah *error*. Respon integral meningkat sampai nilai *error* mengarah ke nilai nol. Dan *Derivatif* adalah komponen yang berguna untuk memprediksi besar eror di masa yang akan datang. Respon derivative bergantung pada perubahan dari eror pada setiap waktu yang dikalikan dengan nilai konstan yang telah dipilih.

Penggunaan respon keluaran akan lebih terkontrol karena eror dari keluaran lebih teredam (Ramadhan & Adinandra, n.d.).

Gambar 2. 10 adalah gambar blok diagram dari sisem kontrol PID.



Gambar 2. 10 Blok Diagram PID
Sumber: (Fauzy & Kasmungin, 2019)

Persamaan dari Kontrol PID adalah:

$$PID(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.1)$$

Keterangan:

K_p = konstanta proporsional

K_i = konstanta integral

K_d = konstanta detivatif

$e(t)$ = error (selisih setpoint dengan level actual)

Ketiga respon pengendali PID mempunyai kelebihan dan kekurangannya yang akan berpengaruh terhadap nilai K_p, T_i, T_d pada respon sistem. (Carin et al., 2018)

Berikut uraiannya:

a. Nilai besar dan kecil K_p akan berpengaruh terhadap pencapaian setpointnya.

Semakin besar akan semakin cepat, semakin kecil akan semakin lambat.

Namun, jika terlalu besar sistem output sistem tidak akan stabil atau sistem berisolasi. Pengaturan nilai K_p diatur untuk mengurangi *error steady state* tetapi tidak bisa menghilangkannya.

- b. K_i berfungsi untuk menghilangkan *error steady state* yang tidak bisa dihilangkan oleh K_p . Nilai K_i yang semakin besar akan mempercepat hilangnya *offset*. Namun jika semakin besar nilainya akan mengakibatkan osilasi pada output kontroler.
- c. K_d hanya berubah nilai keluarannya ketika ada perubahan error, sehingga ketika error tidak ada perubahan maka praktis kontrol ini tidak akan bereaksi. Jika nilai error berubah terhadap waktu, outputnya tergantung pada nilai T_d dan laju perubahan sinyal error.

Tabel 2. 2 adalah tabel untuk parameter PID dengan pembagian *rise time*, *overshoot*, *settling time*, dan *error* untuk penentuan parameter K_p , K_i , K_d nya.

Tabel 2. 2 Parameter PID

Parameter	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error</i>
K_p	Berkurang	Bertambah	Perubahan sedikit	Berkurang
K_i	Berkurang	Bertambah	Bertambah	Menghilangkan
K_d	Perubahan sedikit	Berkurang	Berkurang	Perubahan sedikit

2.3.1 Tuning Kontrol PID

Berikut adalah beberapa metode tuning dari sistem kontrol PID:

a. Metode Ziegler-Nichols

Metode Ziegler-Nichols terdiri dari metode pertama dan metode kedua. Metode pertama mempunyai dua parameter yaitu L dan T dan kemudian digunakan untuk mencari nilai K_p , K_i , dan K_d . Nilai parameternya diperoleh dari titik infleksi pada kurva S yang diperoleh dari garis singgung sumbu x dan sumbu y . Metode kedua juga mempunyai dua parameter yaitu K_u dan T_u . K_u

adalah nilai kritis K_p ketika sistem beresilasi dengan baik, dan T_u adalah jarak antara puncak gelombang dalam satuan detik (Nichols & Nichols, 2022).

Tabel 2. 3 adalah untuk mencari nilai K_p , T_i , T_d berdasarkan L dan T yang didasarkan terhadap kurva reaksi sistem *open loop*.

Tabel 2. 3 Nilai K_p , T_i , T_d berdasarkan L dan T

Tipe Pengendali	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Tabel 2. 4 adalah untuk mencari K_p , T_i , T_d berdasarkan K_u dan T_u berdasarkan metode osilasi pada sistem *close loop*.

Tabel 2. 4 Nilai K_p , T_i , T_d berdasarkan K_u dan T_u

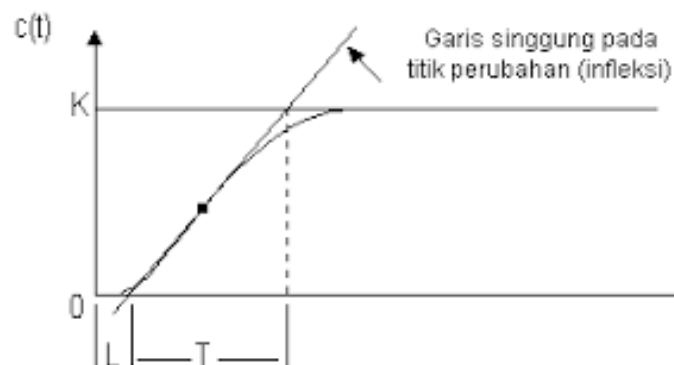
Tipe Pengendali	K_p	T_i	T_d
P	$\frac{K_u}{2}$	-	-
PI	$\frac{2K_u}{5}$	$\frac{4T_u}{5}$	-
PID	$\frac{3K_u}{5}$	$\frac{T_u}{2}$	$\frac{3T_u}{25}$

Kedua metode Ziegler-Nichols bertujuan untuk menghasilkan respon sistem dengan lonjakan maksimum 25%. Untuk kendali *tuning* lup terbuka disebut juga dengan metode kurva reaksi, sedangkan kendali *tuning* lup tertutup disebut dengan metode osilasi. Metode kurva reaksi merupakan *plant* tanpa ada pengendali umpan balik (*feedback controller*). Metode kurva mempunyai kelemahan yaitu ketidakmampuannya melakukan *tuning* pada sistem yang mempunyai *plant*

integrator dan *plant* yang mempunyai akar-akar persamaan yang kompleks. Sedangkan metode osilasi mempunyai beberapa tahapan dalam melakukan *tuning*, yaitu sebagai berikut:

- Plant* disusun serial dengan pengendali PID
- Parameter integrator diatur tak berhingga ($T_i = \infty$) dan parameter derivative diatur nol ($T_d = 0$)
- Parameter proporsional dinaikkan secara bertahap, mulai dari nol sampai respon sistem berosilasi dengan magnitude tetap.

Dari gambar 2. 11 kurva S konstanta L (*dead time*) dan T (waktu tunda) terlihat bahwa kurva ini berubah naik setelah selang waktu waktu mati. Sedangkan waktu tunda menggambarkan perubahan kurva setelah 66% dari keadaan mantapnya (*set point*).



Gambar 2. 11 Kurva Reaksi berbentuk S
Sumber: (Lestari & Abdul Hadi, 2012)

b. Cohen-Coon

Metode Cohen-Coon adalah pengembangan dari metode Ziegler-Nichols dimana versi ini adalah versi yang lebih kompleks dari metode Ziegler-Nichols. Perbedaannya adalah Cohen-Coon memberikann waktu naik yang lebih cepat. Dengan metode Cohen-Coon sebuah respon sistem dimodelkan ke perubahan langkah sebagai respon orde pertama ditambah waktu mati, sehingga didapat

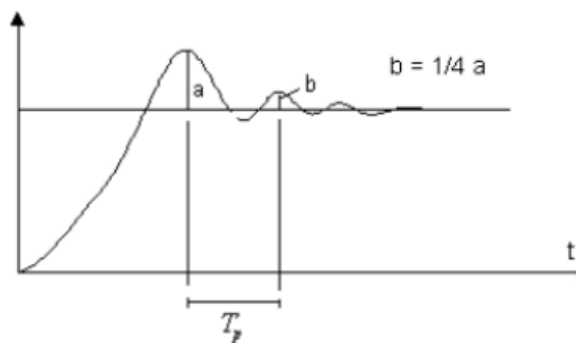
tiga parameter yaitu K_c , T_i , dan T_d . K_c adalah kondisi mapan keluaran dibagi dengan perubahan langkah input, T_i adalah konstanta waktu efektif dari respon orde pertama, dan T_d adalah waktu mati (Makasudede, 1953).

Tabel 2. 5 adalah tabel dari perumusan penelaan parameter PID berdasarkan kurva *open loop*.

Tabel 2. 5 Perumusan PID Berdasarkan Kurva Cohen-Coon

Tipe Pengendali	K_c	T_i	T_d
P	$\frac{1}{K} \left(\frac{T}{L} \right) \left[1 + \frac{1}{3} \left(\frac{L}{T} \right) \right]$	-	-
PI	$\frac{1}{K} \left(\frac{T}{L} \right) \left[0.9 + \frac{1}{12} \left(\frac{T}{L} \right) \right]$	$L \left[\frac{30 + 3 \left(\frac{L}{T} \right)}{9 + 20 \left(\frac{L}{T} \right)} \right]$	-
PD	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{5}{4} + \frac{1}{6} \left(\frac{L}{T} \right) \right]$	-	$L \left[\frac{6 - 2 \left(\frac{L}{T} \right)}{22 + 3 \left(\frac{L}{T} \right)} \right]$
PID	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{4}{3} + \frac{1}{4} \left(\frac{L}{T} \right) \right]$	$L \left[\frac{32 + 6 \left(\frac{L}{T} \right)}{13 + 8 \left(\frac{L}{T} \right)} \right]$	$L \left[\frac{4}{11 + 2 \left(\frac{L}{T} \right)} \right]$

Gambar 2. 12 adalah gambar dari respon *quarter amplitude decay* yang memiliki respon untuk mengatasi respon yang teredam.



Gambar 2. 12 Respon *Quarter Amplitude Decay*

Sumber: (Mesin et al., 2021)

Penyetelan menggunakan metode Cohen-Coon dapat dicari dengan langkah sebagai berikut:

- a. Kontroller Kp diatur hingga model memiliki respon teredam
- b. Dari respon tersebut dapat diidentifikasi nilai T_p (waktu periode 1 amplitudo) dan L (waktu awal respon mengalami peningkatan)
- c. Data yang sudah didapat dapat dihitung berdasarkan konstanta diatas.

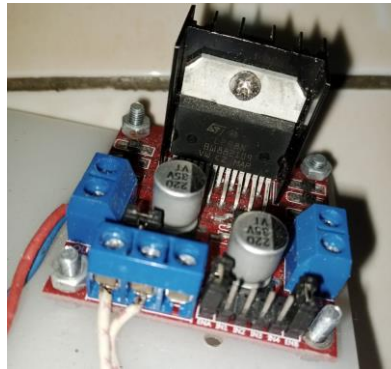
Maka akan didapat nilai masing-masing konstanta yang ingin digunakan. Dan metode ini juga digunakan sebagai dasar tuning dari model objek yang dikendalikan (Mesin et al., 2021).

2.4 Driver Motor L298N

Driver motor L298N adalah sebuah driver motor *H-bridge* yang dapat mengendalikan beban-beban induktif. Driver motor L298N mempunyai dua buah *H-bridge* yang dapat mengendalikan kecepatan dan arah 2 buah motor DC. Kedua

H-bridge dalam modul ini bisa diparalel untuk meningkatkan kemampuan menopang arus mencapai 4 ampere (Antoni & Sulistiyowati, n.d.).

Gambar 2. 13 adalah gambar dari driver motor L298N yang dapat berfungsi untuk mengatur kecepatan dan arah putaran pada motor DC.



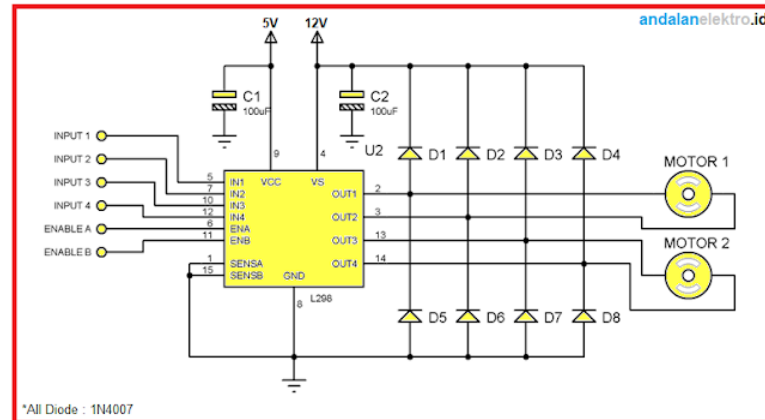
Gambar 2. 13 Driver Motor L298N

Sumber: (Pribadi)

Adapun spesifikasi dari modul IC L298N ini adalah:

- a. Tegangan operasi mencapai 46V.
- b. Mampu mengendalikan stepper bipolar 2A.
- c. Mampu mengontrol arah motor DC dengan arus maksimal 2A di setiap *H-bridge*.
- d. Mendukung kontrol PWM (*Pulse Width Modulation*) dengan frekuensi mencapai 20KHz.
- e. Mempunyai sensor arus keluaran.

Driver motor L298N mempunyai proteksi *over temperature supply voltage* sebesar +5VDC yang dapat memudahkan perancangan motor driver menggunakan IC L298N untuk mengetahui alokasi pin terlebih dahulu. Gambar 2. 14 adalah skema rangkaian dari driver motor L298N.



Gambar 2. 14 Skema Rangkaian IC L298N
Sumber: (Elektro, 2020)

Berikut cara kerja dari rangkaian IC L298N:

- Input 1 dan input 2 digunakan untuk mengontrol Motor 1. Motor hanya akan berputar apabila Enable A diberikan logika *high*. Apabila pin enable diberikan logika *low* motor tidak akan berputar.
- Apabila input 1 *high* dan input 2 *low* maka motor akan berputar dengan arah tertentu.
- Apabila input 1 *low* dan input 2 *high* maka motor akan berputar kearah sebaliknya.
- Apabila input 1 dan 2 logikanya sama *high* atau *low* maka motor tidak akan berputar.
- Input 3 dan input 4 digunakan untuk mengontrol motor 2. Motor hanya akan berputar apabila Enable B diberikan logika *high*. Apabila pin enable diberikan logika *low* motor tidak akan berputar.
- Apabila input 3 *high* dan input 4 *low* maka motor akan berputar dengan arah tertentu.
- Apabila input 3 *low* dan input 4 *high* maka motor akan berputar kearah sebaliknya.

- h. Apabila input 3 dan 4 logikanya sama *high* atau *low* maka motor tidak akan berputar.
- i. Putaran motor searah jarum jam disebut CW (*Clock Wise*) sedangkan putaran motor yang berlawanan arah jarum jam disebut CCW (*Counter Clock Wise*)

Kapasitor 1 dan 2 berfungsi sebagai *Decoupling* untuk menghilangkan tegangan liar yang berasal dari *power supply*. Sedangkan 8 buah II-17ekni 1N 4007 berfungsi sebagai proteksi terhadap induksi yang diakibatkan oleh perubahan putaran motor secara tiba-tiba.

2.5 Rotary Encoder

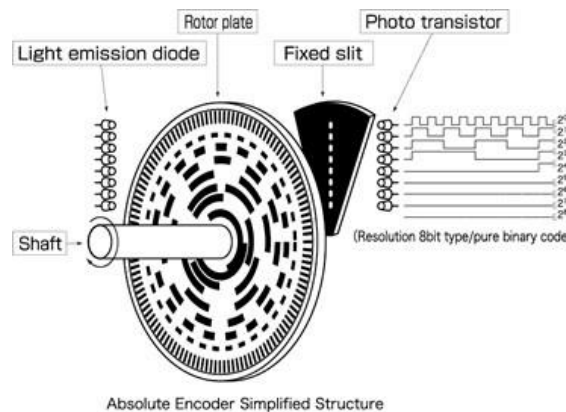
Rotary encoder adalah perangkat elektromekanik untuk memonitor gerakan dan posisi yang umumnya menggunakan sensor II-17ekni untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Suatu poros yang berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* dan diteruskan oleh rangkaian kendali. *Rotary encoder* biasanya digunakan pada pengendalian robot, *motor drive*, dan sebagainya. *Rotary encoder* tersusun dari piringan tipis berlubang pada bagian lingkaran piringan sehingga II-17eknik LED akan menuju piringan dan *photo-transistor* dapat mendeteksi ketika motor berputar dan menghasilkan suatu pulsa gelombang. Semakin banyak deretan pulsa yang dihasilkan pada suatu putaran itu menentukan akurasi *rotary encoder* dan mengakibatkan semakin banyaknya jumlah lubang yang dapat dibuat (Dwi Ermansyah, 2016).

Rotary Encoder memiliki dua jenis utama yaitu:

a. *Absolute Rotary Encoder*

Prinsip kerja dari *absolute rotary encoder* itu menggunakan piringan yang memiliki pola unik serta sinyal yang tersusun dalam jalur *concentric* yang menghasilkan kode digital berbeda satu sama lain untuk menyatakan sejumlah posisi tertentu dari poros yang dihubungkan.

Gambar 2. 15 adalah gambar dari *rotary encoder absolute* yang merupakan bentuk piringan yang tersusun dari segmen-segmen cincin konsentris yang dimulai dari bagian tengah piringan ke arah tepi luar yang jumlah segmennya selalu dikali dua jumlah segmen sebelumnya.



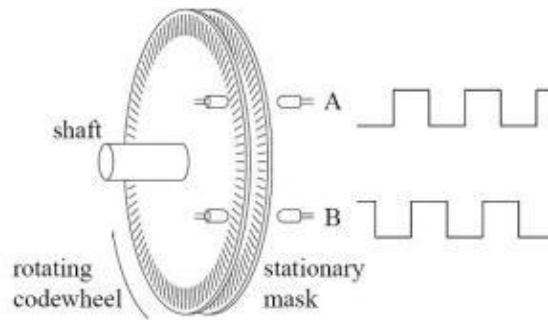
Gambar 2. 15 *Rotary Encoder Absolute*

Sumber: (Dwi Ermansyah, 2016)

Kelipatan tersebut akan membuat sistem biner yang dibaca dan menghasilkan posisi poros yang dapat diketahui dan diukur lalu diteruskan ke rangkaian pengendali. Semakin banyak bit yang dipakai maka posisi yang diperoleh semakin banyak.

b. *Incremental Rotary Encoder*

Encoder incremental terdiri dari dua *track* atau *single* dan dua sensor yang disebut *channel A* dan *B* seperti pada gambar 2. 16.



Gambar 2. 16 *Rotary Encoder Incremental*

Sumber: (Dwi Ermansyah, 2016)

Ketika poros berputar, deretan pulsa akan muncul di masing-masing *channel* pada frekuensi yang proporsional dengan kecepatan putar sedangkan hubungan fasa antara *channel A* dan *B* menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat ditentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda fasa seperempat putaran (*quadrature signal*). Seringkali terdapat *output channel* ketiga disebut *INDEX*, yang menghasilkan satu pulsa perputaran berguna untuk menghitung jumlah putaran yang terjadi (Dwi Ermansyah, 2016).

Tabel 2. 6 adalah tabel *output rotary encoder incremental* untuk rotasi yang searah dengan jarum jam dan berlawanan arah jarum jam.

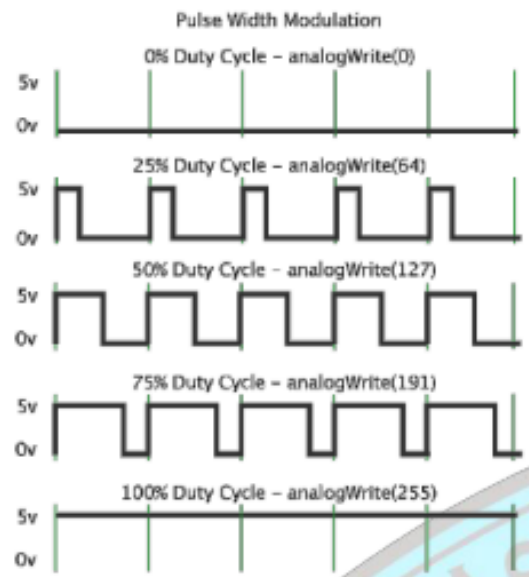
Tabel 2. 6 *Ouput Rotary Encoder Incremental*

Coding untuk rotasi berlawanan arah jarum jam			Coding untuk rotasi searah jarum jam		
Tahap	A	B	Tahap	A	B
1	1	0	1	0	0
2	1	1	2	0	1
3	0	1	3	1	1
4	0	0	4	1	0

2.6 PWM (*Pulse Width Modulation*)

Pulse Width Modulation (PWM) adalah teknik yang mengubah sinyal digital gelombang untuk mendapat tegangan rata-rata berbeda (Putra et al., 2021). Metode PWM digunakan untuk mengatur kecepatan sebuah motor DC dengan membuat gelombang persegi yang memiliki perbandingan pulsa *high* terhadap pulsa *low* tertentu yang diskalakan dari 0-100% (Arduino, n.d.). Sinyal PWM memiliki amplitude dan frekuensi dasar yang tetap, tetapi lebar pulsa yang bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitude sinyal asli yang belum termodulasi. Yang artinya sinyal PWM memiliki gelombang yang tetap namun *duty cycle* bervariasi (antara 0%-100%).

Gambar 2. 17 adalah gambar dari pulsa PWM yang membentuk beberapa persen *duty cycle*.



Gambar 2. 17 *Pulse Width Modulation*
Sumber: (Gafur et al., 2020)

Aplikasi PWM berbasis mikrokontroler yaitu pengendalian motor DC, motor servo, dan pengaturan nyala terang LED (Setiawan, 2017). Modulasi PWM diperoleh dengan bantuan sebuah gelombang kotak yang siklus kerja (*duty cycle*) gelombang dapat diubah untuk mendapatkan sebuah tegangan keluaran yang bervariasi dan merupakan nilai rata-rata dari gelombang tersebut (Arduino, n.d.).

PWM berkerja sebagai *switching power supply* untuk mengontrol *on* dan *off*. Tegangan DC dikonversikan menjadi sinyal menjadi sinyal kotak bolak-balik, saat *on* mendekati tegangan puncak dan saat *off* menjadi nol (0) volt. Jika frekuensi *switching* cukup tinggi maka temperature desktop akan dikendalikan sesuai yang diharapkan yaitu dengan mengatur *duty cycle* dari sinyal yang disebabkan oleh PWM. *Duty cycle* adalah perbandingan antara pulsa *high* dengan pulsa *low* pada suatu gelombang. Dimana rumus *duty cycle* adalah: (Arduino, n.d.)

$$D = \frac{T_{on}}{T_{on} + T_{off}} \times 100\% \quad (2.2)$$

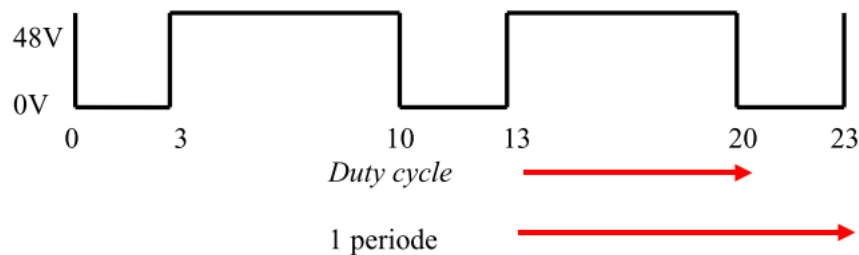
Keterangan:

T_{on} = Waktu Pulsa *High*

T_{off} = Waktu Pulsa *Low*

D = *Duty cycle* (lamanya pulsa *high* dalam satu periode)

Gambar 2.18 adalah gambar bentuk dari *duty cycle* sehingga membentuk gelombang dalam satu periode.



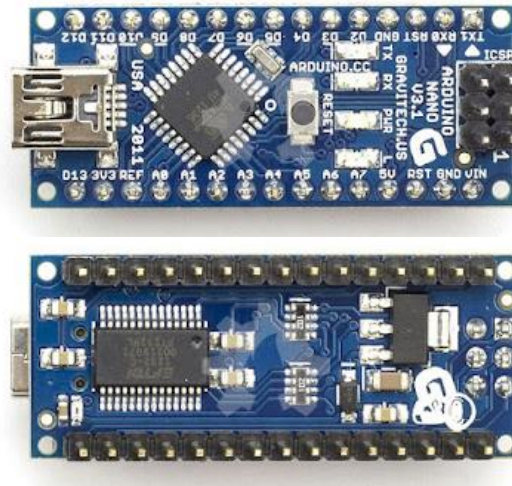
Gambar 2. 18 *Duty Cycle*
Sumber: (Teknik et al., n.d.)

Duty cycle adalah representasi sinyal dalam satu periode dan dinyatakan dalam bentuk (%) dengan nilai 0%-100%. Semakin tinggi frekuensi kerja PWM, maka semakin baik kerja motor. Jika frekuensi semakin kecil maka akan terdengar dengungan saat motor dijalankan dengan *duty cycle* kecil, dan semakin tinggi frekuensi suara dengungan semakin kecil. Dengan mengatur lamanya waktu motor untuk berputar dalam satu periode pulsa, maka besarnya *duty cycle* pulsa kotak yang dikirimkan pun berubah sesuai dengan *duty cycle* atau waktu motor berputar dan kecepatan motor pun berubah (Teknik et al., n.d.).

2.7 Arduino Nano

Arduino nano adalah suatu papan rangkaian elektronik *open source* menggunakan mikrokontroler Atmega 328 yang memiliki 12 pin *input*, 8 pin *input* analog, 16 MHz *crystal oscillator*, sambungan USB, ICSP *header*, dan tombol reset.

Gambar 2. 19 adalah bentuk fisik dari komponen Arduino nano yang mempunyai beberapa pin dengan masing-masing fungsi.



Gambar 2. 19 Arduino Nano
Sumber: (Razor, n.d.)

Arduino nano mempunyai beberapa kategori pin yang mempunyai fungsi masing-masing:

a. Pin input/output digital

Fungsinya adalah untuk membaca sinyal digital yaitu nilai 0 dan 1 atau disebut dengan logika TRUE dan FALSE. Jumlah pin digital Arduino nano adalah sebanyak 14pin.

b. Pin input analog

Fungsinya adalah untuk membaca sinyal analog untuk diubah ke bentuk sinyal digital. Jumlah pin input analog adalah sebanyak 8pin.

c. Pin tegangan

Fungsinya yaitu untuk mengatur tegangan yang ada pada Arduino.

d. Pin reset

Fungsinya yaitu untuk merestart ulang program yang sedang berjalan pada Arduino dengan cara menghubungkan pin reset ke salah satu pin digital lalu masukkan script khusus. Atau bisa juga dengan menekan tombol reset yang tersedia pada board.

Tabel 2. 7 adalah tabel spesifikasi dari komponen Arduino Nano.

Tabel 2. 7 Spesifikasi Arduino Nano

Jenis Mikrokontroler	Atmega328
Tegangan Operasi	5 Volt
Tegangan Disarankan	7 – 12 Volt
Batas Tegangan	6 – 20 volt
Pin Input/Output Digital	14
Pin PWM	6
Pin Input Analog	8
Arus Per Pin	40 Ma
Memori Flash	32 KB (2 KB untuk <i>bootloader</i>)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Panjang x Lebar	4,3 cm x 1,8 cm
Berat	5 gram

2.8 LabVIEW

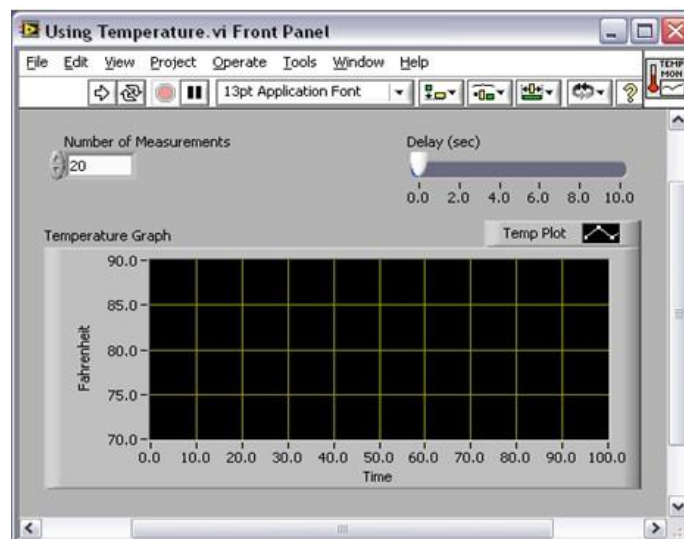
LabVIEW yaitu sebuah *software* pemrograman yang diproduksi oleh sebuah perusahaan bernama *National Instrument* yang mempunyai fungsi dan peranan yang sama seperti bahasa pemrograman lainnya. Perbedaannya LabVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram. Program LabVIEW dikenal dengan sebutan *Virtual Instrument* (VI) karena penampilan dan operasinya dapat meniru sebuah instrument. Dalam pemrograman LabView, pertama-tama itu membuat front panel atau user interface dengan menggunakan

kontrol dan indikator, kontrol yang dimaksud adalah peralatan input, sedangkan indikator yang dimaksud adalah peralatan display. Setelah menyusun user interface, lanjutkan dengan menyusun diagram blok yang berisi kode-kode Vis untuk mengontrol front panel pemrograman (Pongoh et al., 2021).

Berikut penjelasan dari beberapa komponen LabView:(Azzahra, 2021)

a. *Front Panel*

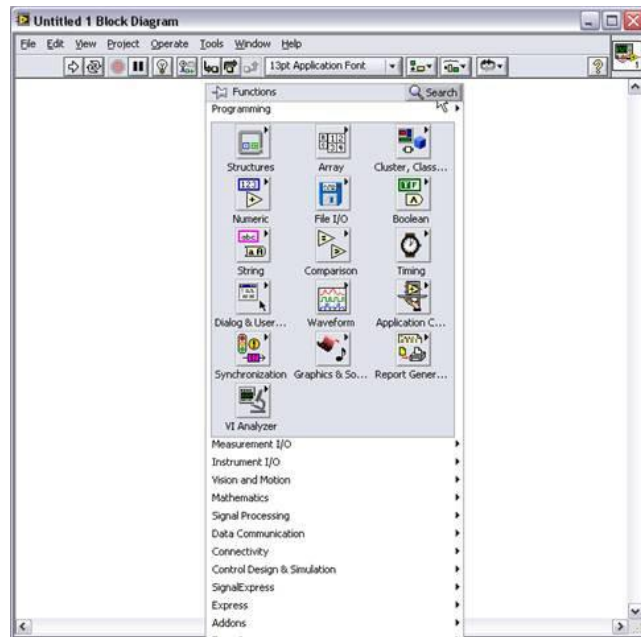
Front panel adalah bagian yang didalamnya terdapat kontrol dan indikator yang digunakan untuk membangun VI, menjalankan program, dan mendebug program. Gambar 2. 20 adalah gambar dari *front panel* pada aplikasi Labview sehingga program dapat dijalankan.



Gambar 2. 20 *Front Panel* LabView
(Instruments, 2020)

b. *Block Diagram*

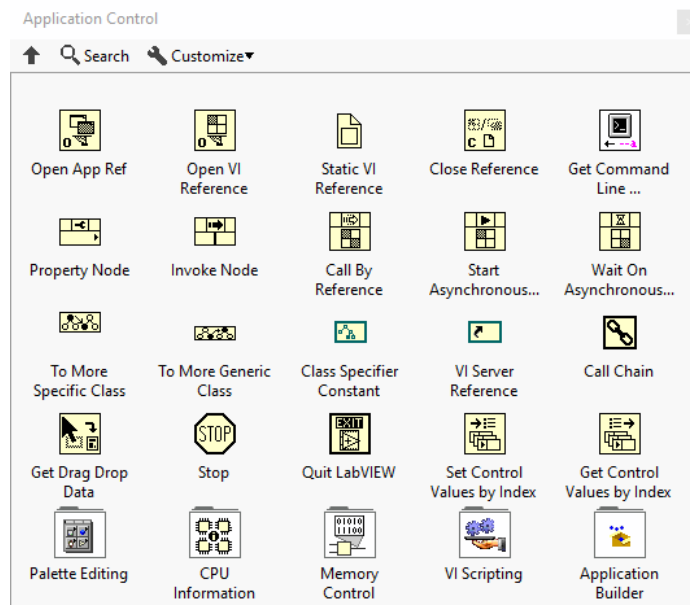
Block diagram adalah bagian yang berisi *source* kode yang dibuat dan berfungsi sebagai intruksi untuk *front panel*. Gambar 2. 21 adalah gambar blok diagram pada aplikasi Labview.



Gambar 2. 21 Blok Diagram LabView
(Instruments, 2020)

c. *Control Palette*

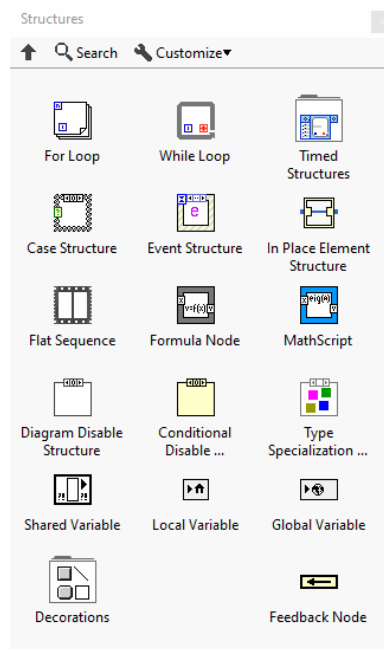
Control palette adalah indikator yang digunakan untuk membuat *user interface* atau *front panel*. Gambar 2. 22 adalah gambar dari beberapa pilihan *control pallette* pada aplikasi Labview yang dapat digunakan untuk membuat UI (*User Interface*).



Gambar 2. 22 *Control Palette* LabView
(WIKI, 2020a)

d. *Function Palette*

Function palette beberapa fungsi dari yang dapat digunakan untuk membangun diagram blok yang dapat digunakan oleh *user*. Gambar 2. 23 adalah gambar dari *function palette* sehingga dapat dibangunnya diagram blok pada labview.



Gambar 2. 23 *Function Palette* LabView
(WIKI, 2020b)

2.9 Review Hasil Penelitian Terkait

Penelitian ini didasarkan pada penelitian-penelitian sebelumnya yang sejenis mengenai pengendalian kecepatan motor DC menggunakan pengendali PID untuk pembanding kedepannya dan bisa menjadi metodologi pengembangan selanjutnya.

Maka dari itu, terdapat beberapa penelitian sejenis yang sudah dibuat terlebih dahulu yang dirangkum pada tabel 2. 8, diantaranya sebagai berikut:

Tabel 2. 8 *Review* Penelitian Terkait

No	Nama Peneliti	Judul Penelitian	Pembahasan
1	Muhammad Reza Aditya NP dan Rahmat Hidayat	Kendali Kecepatan Motor DC Menggunakan	Membahas tentang penerapan kontrol PID untuk mengendalikan kecepatan sudut motor DC menggunakan encoder sebagai

No	Nama Peneliti	Judul Penelitian	Pembahasan
		Pengendali PID dengan Encoder Sebagai <i>Feedback</i>	umpan balik. Pengimplementasiannya menggunakan Arduino Uno, motor DC dan encoder. Parameter PID diubah satu persatu untuk melihat pengaruh terhadap kurva respon kecepatan pada motor saat diubah.
2	Alifa Restu JW dan Andry Irawan	Pengaturan Kecepatan Motor DC dengan Kontrol Proporsional Integral Derivatif (PID) Berbasis Labview	Membahas tentang pengaturan dan pemantauan kecepatan motor DC secara <i>real-time</i> menggunakan Labview 2010. Parameter PIDnya dapat di tuning pada Labview untuk menghasilkan respon kecepatan dan Labview juga dapat menampilkan panel kontrol, grafik berbasis PID ke motor.
3	Fauzy dan Sugiatmo Kasmungin	Perancangan Sistem Kontrol Kecepatan Motor DC dengan PID Labview 2010	Membahas tentang perancangan sistem kontrol kecepatan motor DC menggunakan PID berbasis Labview dan perangkat NI-DAQ 6008. Sistemnya dirancang untuk mengatasi gangguan beban yang menyebabkan kecepatan motor lambat, dengan sensor proximity sebagai sistem kontrol <i>close-loop</i> .
4	Mila Diah IP, Alfian Ma'arif, dan Riky Dwi Puriyanto	Pengendali Kecepatan Sudut Motor DC Menggunakan Kontrol PID dan Tuning Ziegler-Nichols	Membahas tentang penerapan kontrol PI menggunakan tuning Ziegler-Nichols sebagai metode tuning untuk pengendalian kecepatan sudut pada motor DC. Menggunakan simulasi Matlab untuk pendekatan <i>trial and error</i> yang kurang stabil. Tuning Ziegler-Nichols memberikan performa lebih baik dan stabil dibanding metode <i>trial and error</i> .
5	Rahmad Sulistiyadi, Sugiarto, dan Oni Yuliani	Metode Penelaan Kontroler PID	Membahas 3 metode tuning PID yaitu Ziegler-Nichols, Chien-Hrones-Reswick, dan Cohen-Coon. Semua metode diuji pada berbagai tipe sistem orde dari orde-1 sampai orde tertinggi menggunakan simulasi Matlab dan HPS Technic Simulator PID. Metode Ziegler-Nichols lebih efektif diaplikasikan untuk sistem orde-1 sampai orde tertinggi, sedangkan yang 2 lainnya cocok untuk sistem orde-1 saja. Semakin tinggi orde sistemnya, konstanta waktu meningkat dan performa meredam sistem melemah.
6	Hadi Supriyanto, Fitria Suryatini, Abdur Rohman Harits Martawireja, dan Hendy Rudiansyah	Impelementasi Kontroler PID Dengan zmetode Tuning Ziegler-Nichols dan Cohen-Coon Pada Sistem Scada Kendali Level Air	Membahas membandingkan antara 2 metode Ziegler-Nichols dan Cohen-Coon untuk pengaturan level air terbaik. Metode Cohen-Coon lebih unggul daripada metode Ziegler-Nichols untuk sistem kendali level air berbasis PLC-SCADA. Monitoring dan analisis performa dilakukan melalui SCADA selama eksperimen.

Penelitian ini didasarkan pada penelitian-penelitian sebelumnya terkait yang sejenis mengenai peranan pengendali PID untuk kecepatan motor DC menggunakan metode Ziegler-Nichols dan Cohen-coon, untuk pembandingan ke depannya dan bisa menjadi metodologi pengembangan selanjutnya.