

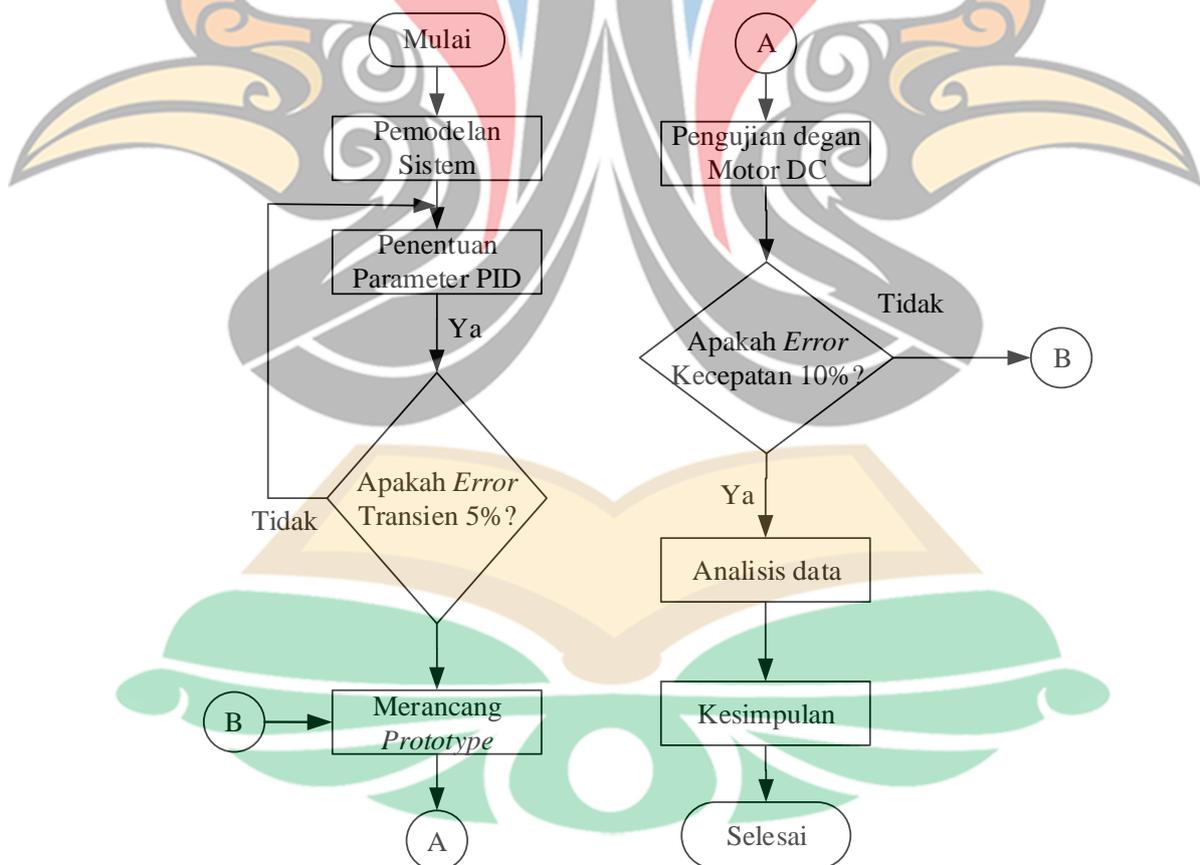
BAB III

METODE PENELITIAN

Metodologi penelitian merupakan kerangka dasar dari tahapan penyelesaian tugas akhir. Metodologi penelitian tugas akhir meliputi semua kegiatan yang dilaksanakan untuk memecahkan masalah dan melakukan proses analisa terhadap permasalahan yang telah ditetapkan pada tugas akhir.

3.1. Gambaran Umum

Diagram alir penelitian merupakan sebuah tahapan-tahapan yang dilakukan dalam menyelesaikan penelitian. Diagram alir pada penelitian tugas akhir ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram alir penelitian

3.1.1 Studi Literatur

Pada pelaksanaan penelitian ini dibutuhkan studi literatur. Karena studi literatur mempunyai peranan yang penting dalam suatu penelitian ini. Pencarian bahan-bahan keperluan penelitian ini diperoleh dari buku, jurnal ilmiah, ataupun laporan ilmiah yang berhubungan dengan materi penelitian pada tugas akhir ini. Materi yang diperlukan pada studi literatur ini yaitu motor DC, *rectifier fullwave*, PWM (*Pulse Width Modulation*) dan *controller* PID (*Proportional, Integral, Derivatif*).

3.1.2 Pemodelan Sistem

Pada pemodelan sistem dilakukan pemodelan dari *controlled rectifier* dan motor DC. Pada pemodelan sub sistem motor DC dilakukan dengan menggunakan *identification system* pada *software* Matlab dan untuk subsistem *controlled rectifier* dilakukan dengan menentukan nilai fungsi alih.

3.1.3 Penalaan *Controller*

Pada proses penentuan parameter nilai PID terlebih dahulu dilakukan dengan menentukan metode penalaan mana yang akan digunakan. Proses penentuan parameter dilakukan dengan terlebih dahulu melakukan simulasi respons sistem *open loop* dan respons sistem *close loop*. Hal tersebut dilakukan dengan tujuan melihat apakah sistem dapat ditala dengan metode Ziegler Nichols atau tidak, jika respons sistem tidak memenuhi syarat penalaan dengan metode Ziegler-Nichols maka proses penalaan dilakukan dengan menggunakan metode *trial and error*.

3.1.4 Simulasi Keseluruhan Sistem

Pada tahap ini dilakukan simulasi sistem dengan bantuan *software* PSIM untuk melihat hasil respons sinyal keluaran. Hasil respons tersebut berupa respons tegangan dan respons kecepatan sistem setelah dilakukan penambahan *controller*. Selanjutnya dilakukan analisis mengenai respons tegangan dan respons kecepatan.

3.1.5 Perancangan *Prototype*

Perancangan *controlled rectifier* ini dilakukan untuk membuat desain dari *controlled rectifier*. Setelah membuat desain dari *controlled rectifier* kemudian

menentukan komponen yang diperlukan. Adapun komponen yang dibutuhkan untuk membuat *fullwave controlled rectifier* ini adalah *Silicon Controlled Rectifier* (SCR), motor DC, sensor kecepatan *rotary encoder* dan mikrokontroler.

3.1.6 Pengambilan Data

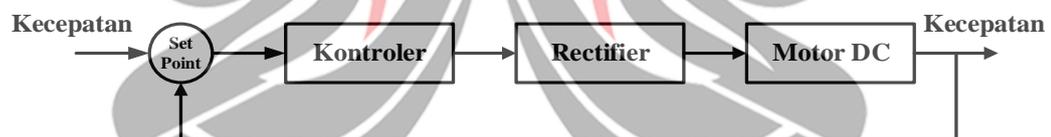
Pengambilan data dilakukan dengan melihat hasil respons tegangan, respons kecepatan dan parameter nilai PID. Selanjutnya dilakukan analisis mengenai respons tegangan dan respons kecepatan

3.1.7 Analisis Data

Analisis data dilakukan dengan melihat respons tegangan, respons kecepatan motor DC, parameter PID serta sudut penyalan untuk SCR sebagai *input* motor DC. Hasil dari simulasi dan implementasi akan dibandingkan.

3.2. Diagram Blok Sistem

Pada gambar 3.2 merupakan blok sistem dari sistem *controlled rectifier*. Sistem diberikan *setpoint* atau kecepatan referensi berupa kecepatan. Kecepatan referensi diberikan ke sistem kemudian dibandingkan dengan kecepatan *feedback*.



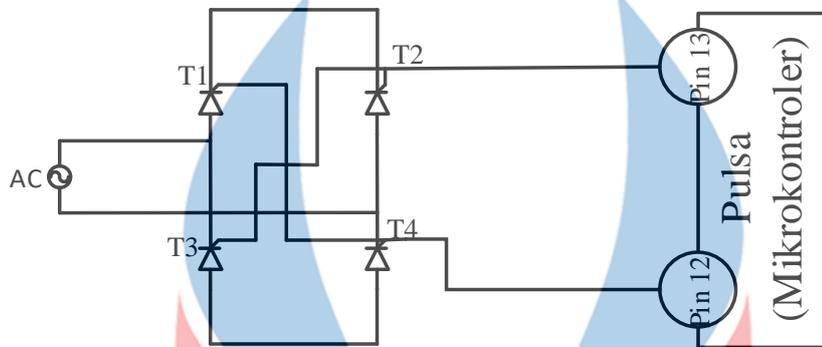
Gambar 3.2 Diagram blok sistem *controlled rectifier*

3.3. Rangkaian *Fullwave Controlled Rectifier*

Fullwave controlled rectifier terdiri dari empat komponen SCR yang fungsinya sebagai saklar terkontrol. Saklar dari SCR ini diatur dengan menggunakan pulsa dari mikrokontroler. Rangkaian *Fullwave controlled rectifier* digambarkan pada Gambar 3.3.

Controlled rectifier satu fasa menggunakan empat komponen SCR. Pada Gambar 3.3 SCR dilambangkan (T). Pada prinsip kerja dari *controlled rectifier* T1 dan T4 menyala bersama dan T2 dan T3 akan on 180° setelah T1 dan T4. T1 dan T4 dihubungkan pada pin 12 pada pin arduino, T2 dan T3 dihubungkan pada pin

13. Pin 12 dan pin 13 merupakan pulsa pembangkitan untuk membuka *gate* pada SCR.



Gambar 3.3 Wiring fullwave controlled rectifier

3.4. Pemodelan Sistem

Pemodelan sub sistem dilakukan menjadi dua bagian. Untuk menentukan fungsi alih dari motor DC dilakukan dengan menggunakan *identification system* dan untuk pemodelan fungsi alih *rectifier controlled* dilakukan dengan memodelkan persamaan 2.12.

3.4.1 Pemodelan Fungsi Alih Motor DC

Parameter pemodelan sistem diperlukan untuk memenuhi variabel-variabel sistem yang digunakan pada pemodelan sistem. Pemodelan sistem dilakukan dengan mengacu pada persamaan sistem yang akan dimodelkan yaitu Motor DC. Penentuan parameter ini melihat spesifikasi pada motor DC diantaranya tegangan motor DC dan kecepatan motor DC.

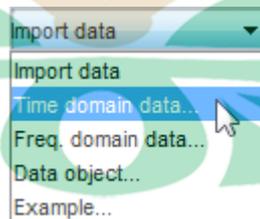
Identifikasi sistem (*System Identification*) merupakan salah satu fitur software matlab. Dimana *system identification* dapat memodelkan suatu sistem atau *plant* secara matematis dengan data berupa *input* dan *output*. Proses identifikasi diperlukan untuk mengetahui karakteristik dari suatu sistem berdasarkan model matematis yang telah didapatkan. Pada penelitian ini dilakukan dengan pengukuran yaitu memberikan nilai input tegangan motor DC kemudian mengamati output sebagai kecepatan.

Dalam proses identifikasi ini, data *input-output* yang ada dimasukkan ke dalam aplikasi dengan memilih “*time domain data*” saat menekan akan timbul tulisan “*import data*”. Setelah itu dilakukan preprocessing. Tampilan dari *system identification toolbox* pada matlab dapat dilihat pada Gambar 3.3.



Gambar 3.4 Tampilan sistem *system identification*

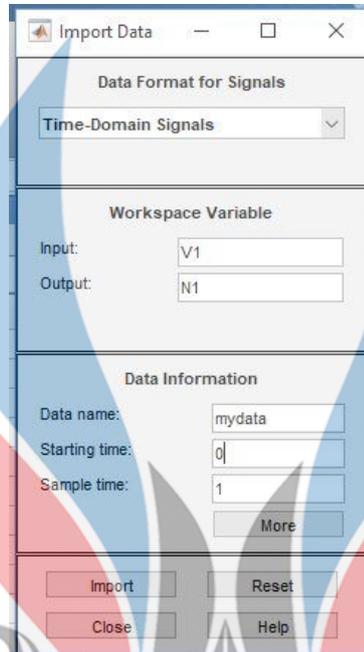
Untuk memasukkan nilai pada *identification system* data yang akan diolah dimasukkan di *command window* Matlab. Pada *command window* memuat data ke dalam ruang kerja Matlab sebagai dua vektor kolom, masing-masing V1 dan N1. Variabel V1 adalah data input dan N1 adalah data *output*. V1 adalah data tegangan dan N1 adalah data kecepatan. Mengimpor data ke dalam *identification system* dapat mengimpor data *single-input / single-output* (SISO) data V1 dan N1 ke dalam aplikasi dari ruang kerja Matlab. Untuk import data dipilih import data kemudian *time domain data*.



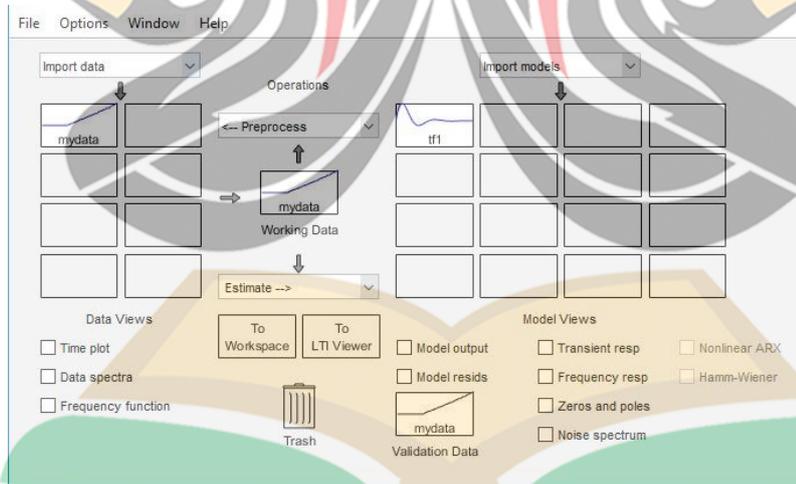
Gambar 3.5 Tampilan *box import data*

www.itk.ac.id

Box import data nilai variabel *input* adalah V1 (tegangan) dan variabel *output* (kecepatan). 'Starting time' 0 dan 'Sample time' 1. Nilai ini adalah waktu sampel aktual dalam pengambilan data.



Gambar 3.6 Time domain data



Gambar 3.7 Tampilan *identification system* penelitian

Proses *identification system* dilakukan untuk 3 data. Setelah dilakukan proses *identification system* maka dapat menampilkan hasil nilai fungsi alih dari motor DC. Dari hasil proses identifikasi sistem terhadap ketiga data tersebut menghasilkan nilai fungsi alih yang berbeda-beda. Dalam proses penelitian ini hanya diambil satu

nilai fungsi alih. Pengambilan nilai fungsi alih terbaik berdasarkan nilai *fit to estimation data* terbesar. Fit adalah nilai terbaik antara data referensi dan data pengujian. Nilai fit dari data pengujian dan referensi dapat dihitung menggunakan rumus pada Matlab.

$$fit = \frac{|x - x_{ref}|^2}{N_s}$$

X adalah data pengujian, x ref adalah data referensi, N_s adalah banyaknya data sampel. Berikut adalah hasil nilai fungsi alih dari tiga data yang telah di ambil

Tabel 3.1 Nilai Fungsi Alih Tiap Data

Data Ke	Nilai fungsi alih	Nilai <i>fit to estimation</i>
Data 1	$\frac{\omega(s)}{V_{in}(s)} = \frac{25.52}{s^2 + 3.411s + 3.192}$	80,75 %
Data 2	$\frac{\omega(s)}{V_{in}(s)} = \frac{2.04 \times 10^4}{s^2 + 2130s + 2516}$	89,85 %
Data 3	$\frac{\omega(s)}{V_{in}(s)} = \frac{2.539}{s^2 + 0.5406s + 0.3193}$	90,01 %

Nilai fit terbaik didapatkan pada data ke-3. Sehingga nilai fungsi alih dari sistem dipilih data 3.

A. Pemodelan Fungsi Alih *Controlled Rectifier*

Pemodelan fungsi alih *fullwave controlled rectifier* mengacu pada perancangan *prototype*. Sumber tegangan yang diberikan sebesar 12 V dengan frekuensi 50 Hz. Dengan diberikan besar sudut penyalaan pada *fullwave controlled rectifier* yaitu 45° dan perancangan *fullwave controlled rectifier* pada tugas akhir ini hanya dibebani resistansi beban ($RL = 330\Omega$). Persamaan 2.13 digunakan untuk menghitung nilai v_{rec} yang nilai rata-ratanya dapat dicari sebagai berikut :

$$v_{rec} = \frac{2\sqrt{2}}{\pi} v_{ac} \cos \alpha_r$$

$$V_{rec} = \frac{2\sqrt{2}}{\pi} V_{ac} \cos\alpha$$

$$V_{rec} = \frac{2\sqrt{2}}{3.14} \times 12 \times \cos(45^\circ)$$

$$= 7.64 \text{ Volt}$$

Untuk mendapatkan fungsi alih dari *fullwave controlled rectifier*, hasil V_{rec} disubstitusikan ke persamaan 2.12 sebagai berikut

$$\frac{di_o}{dt} = -RLi_o + V_{rec}$$

$$sI_o = -RLI_o + V_{rec}$$

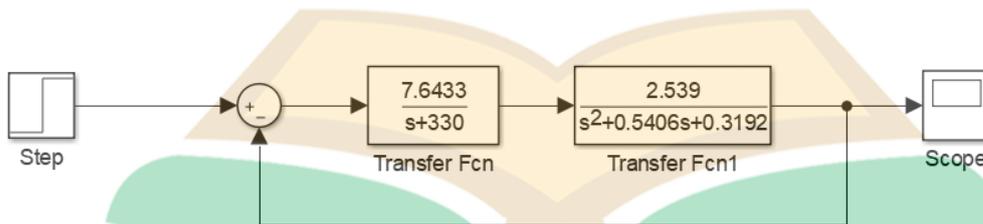
$$(S + RL)I_o = V_{rec}$$

$$I_o = \frac{V_{rec}}{s + RL}$$

$$= \frac{7.64}{s + 330}$$

3.5. Penalaan *Controlled PID*

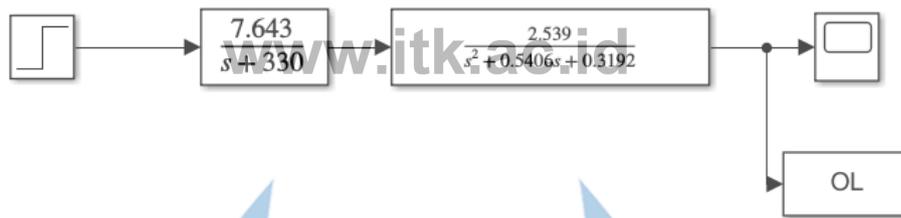
Setelah dilakukan pemodelan sistem dalam bentuk fungsi alih, untuk masing-masing komponen telah didapatkan, maka tahap selanjutnya adalah memodelkan sistem ke dalam diagram blok sistem keseluruhan. Dengan mengacu pada gambar 3.2 maka pemodelan diagram blok fungsi alih dari sistem secara keseluruhan dapat dilihat pada gambar 3.8.



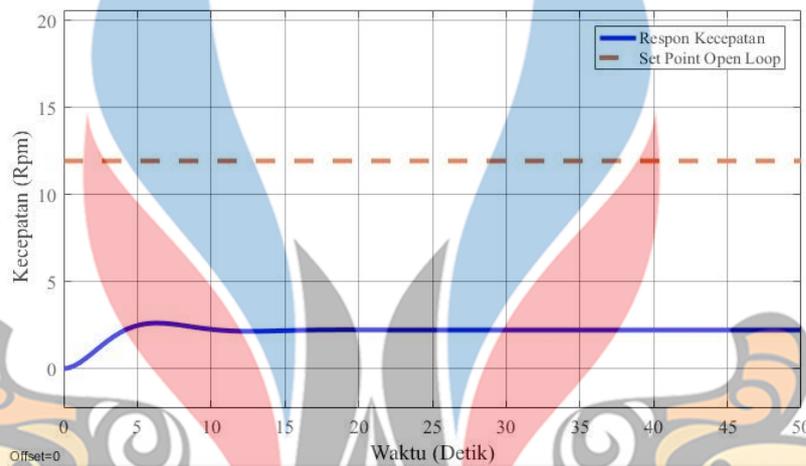
Gambar 3.8 Sistem dalam fungsi alih

3.5.1 Simulasi *Open loop*

Diagram blok sistem pengendalian dengan *controller PID open loop* ditunjukkan oleh gambar 3.9. Simulasi pengujian *open loop* dilakukan dengan memberikan nilai referensi tegangan.



Gambar 3.9 Simulasi *open loop*

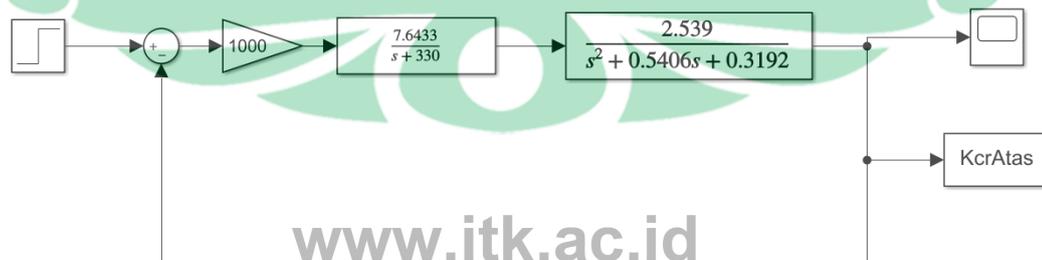


Gambar 3.10 Respons sistem *open loop*

Respons sistem *open loop* sesuai dengan simulasi gambar 3.9 dapat dilihat pada gambar 3.10. Respons yang diperoleh tidak berbentuk tangensial, sehingga metode pertama Ziegler-Nichols tidak dapat digunakan. Respons pada gambar tidak berbentuk tangensial karena tidak memiliki titik balik

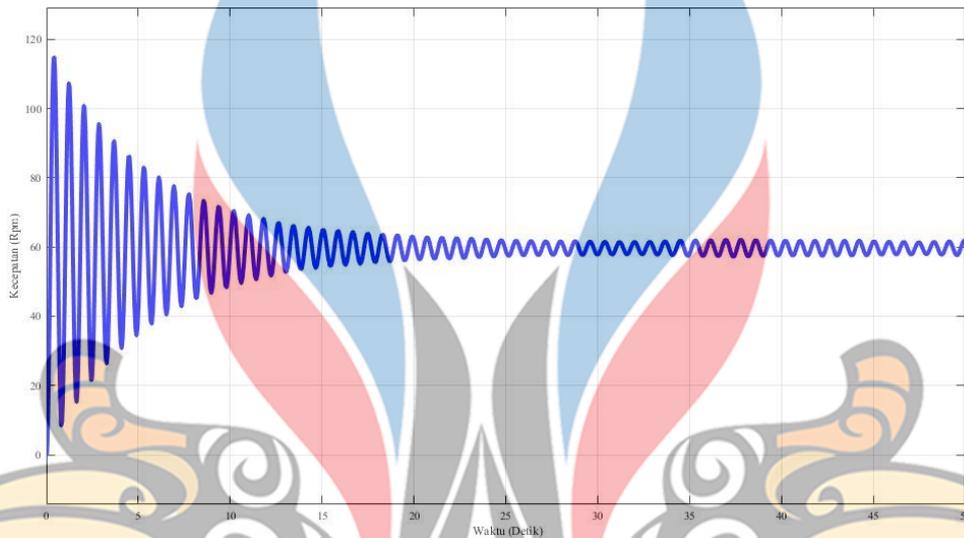
A. Simulasi *Close Loop*

Metode Diagram Ziegler-Nichols-2 dilakukan dengan menggunakan blok pengujian *closed loop* untuk dapat ditunjukkan oleh gambar 3.11.



Gambar 3.11 Simulasi *closed loop* dengan $K_{cr}=1000$

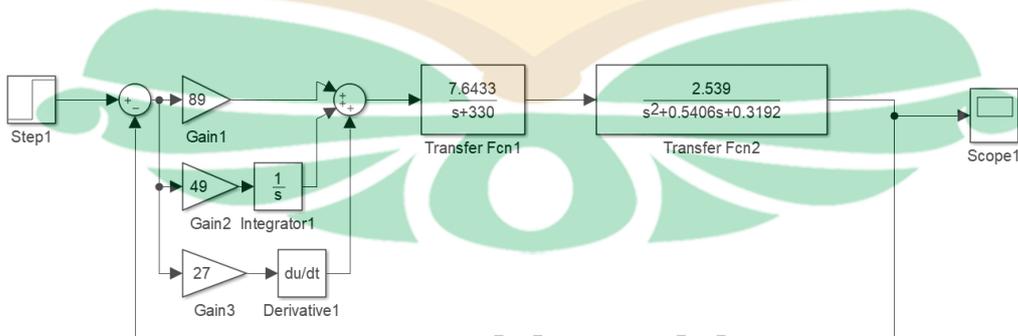
Pengujian *closed loop* dilakukan dengan memvariasikan nilai K_{cr} . Ketika memvariasikan nilai K_{cr} sebesar 1000 didapatkan juga respons yang tidak beresilasi. Oleh karena itu digunakan metode *trial and error* untuk penalaan parameter PID dengan melihat karakteristik respons sistem yang dimiliki.



Gambar 3.12 Respons *closed loop*

B. *Trial and Error*

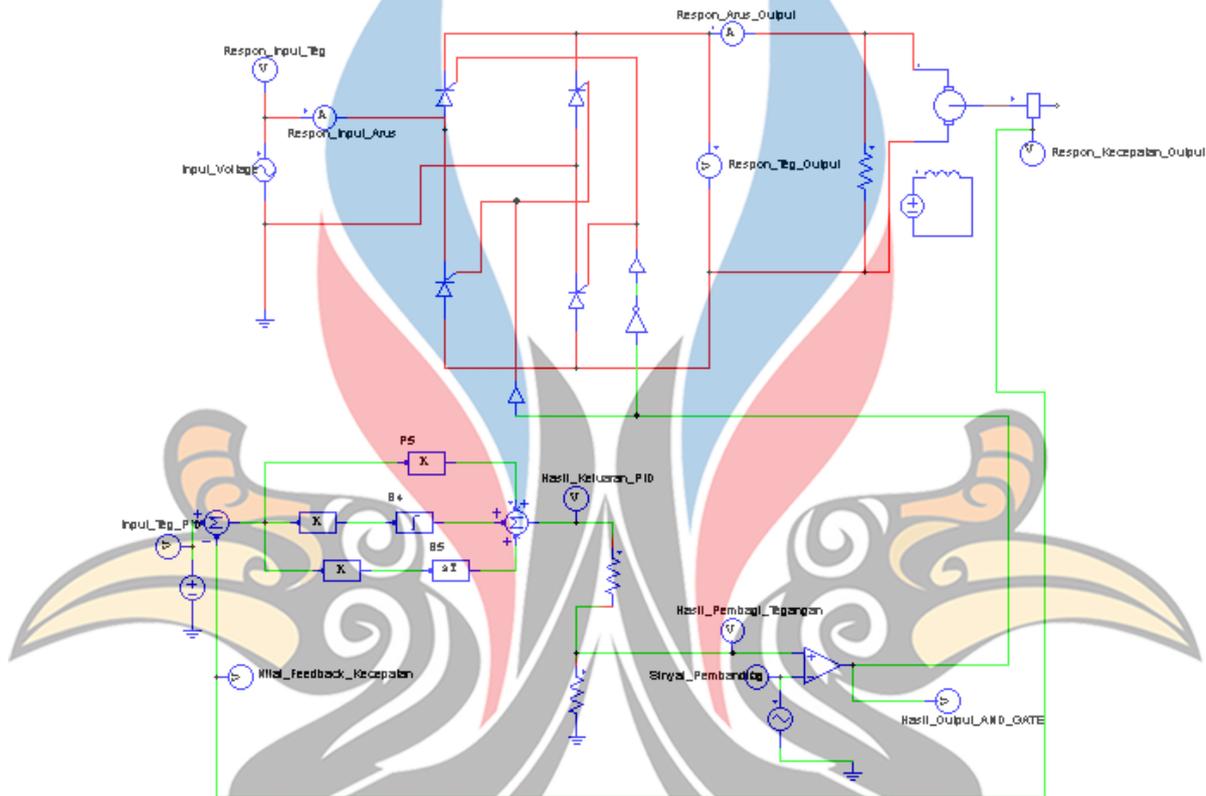
Metode *trial and error* dilakukan untuk mencari parameter PID terbaik dari sistem. Parameter PID dicari dengan menggunakan karakteristik masing-masing kendali P, I dan D yang ditampilkan pada tabel 2.1. Sistem dengan pengendali PID metode *trial and error* ditunjukkan oleh gambar 3.13



Gambar 3.13 Sistem dengan kendali PID

Penentuan parameter PID dengan metode *trial and error* menggunakan karakteristik PID. Karakteristik yang digunakan agar respons motor dengan nilai yang diinginkan menunjukkan kondisi *time rise* cepat dan *error steady-state* mendekati 0.

3.6. Perancangan Simulasi dengan PSIM



Gambar 3.14 Simulasi PSIM Sistem

Gambar 3.14 menunjukkan simulasi sistem dengan menggunakan *software* PSIM. Simulasi dimaksudkan untuk melihat respons tegangan dan kecepatan keluaran motor. Sistem pada PSIM ini terdiri dari *PID controller*, *rectifier*, dan beban motor. Parameter PID yang digunakan berdasarkan parameter yang telah didapatkan dengan metode *trial and error*.

3.7. Pengujian Simulasi

Pada tahap pengujian simulasi dilakukan menggunakan *software* Matlab. Pengujian dilakukan tanpa *controller* dan dengan penambahan *controller* PID. Variasi pengujian dilakukan dengan menggunakan *setpoint* 60 Rpm dan 120 Rpm.

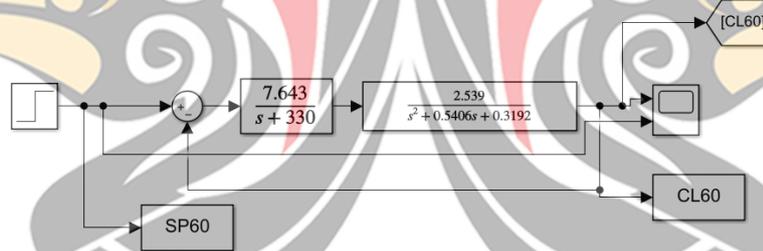
Pemilihan kedua *setpoint* ini untuk melihat respons sistem pada saat kecepatan *rating* dan kecepatan dibawah *rating*.

A. Tanpa Controller

Pada kondisi pengujian tanpa *controller* ini melihat respons sistem apabila tidak ditambahkan *controller* PID. Pengujian dilakukan untuk melihat respons sistem, apabila respons kecepatan dapat mencapai *setpoint* yang diinginkan dan cepat untuk *steady state*, maka tidak perlu dilakukan penambahan *controller*.

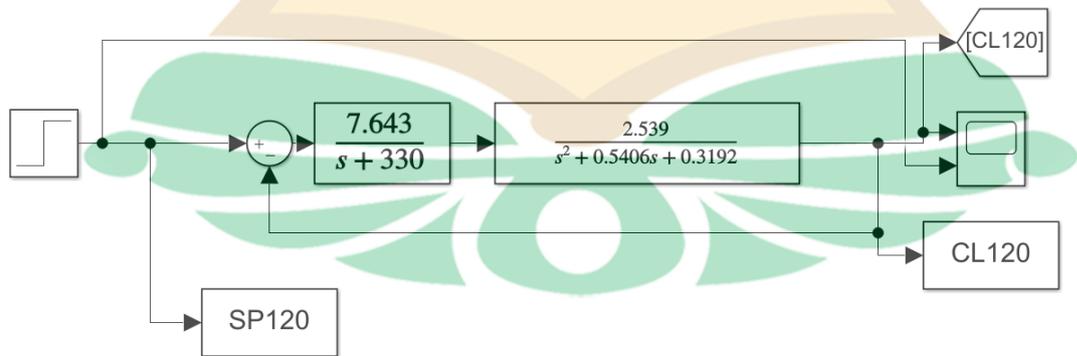
A.1. Setpoint 60 Rpm

Pengujian sistem ini dilakukan dalam bentuk *closed loop* dan tanpa menggunakan *controller* dengan *setpoint* berupa *step* sebesar 60. Nilai tersebut sesuai dengan salah-satu kecepatan referensi yang diberikan yaitu 60 Rpm. Gambar 3.15 menunjukkan sistem *closed loop* tanpa menggunakan *controller* dengan *setpoint* 60 Rpm.



Gambar 3.15 Sistem tanpa *controller* dengan *setpoint* 60 Rpm

A.2 Setpoint 120 Rpm



Gambar 3.16 Sistem tanpa *controller* dengan *setpoint* 120 Rpm

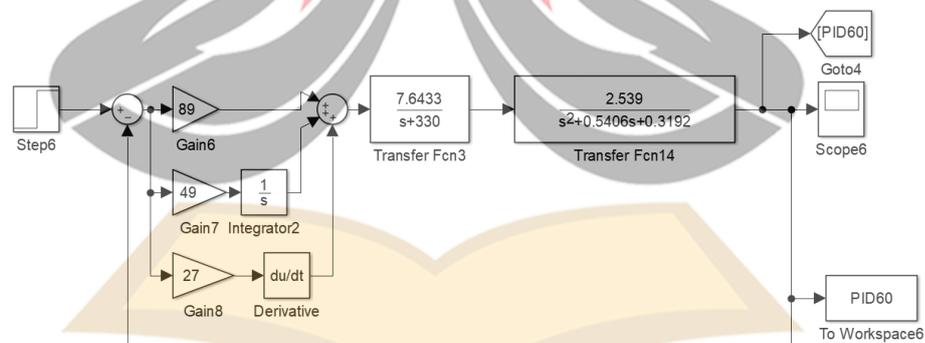
Pengujian sistem ini juga dilakukan dalam bentuk *closed loop* dan tanpa menggunakan *controller*, namun besar *setpoint* yang digunakan adalah 120. Nilai tersebut sesuai dengan salah-satu kecepatan referensi yang diberikan yaitu 120 Rpm. Pengujian dilakukan untuk melihat respons kecepatan yang dihasilkan tanpa menggunakan *controller*. Gambar 3.16 menunjukkan sistem *closed loop* tanpa menggunakan *controller* dengan *setpoint* 120.

B. Penambahan *Controller*

Pengujian dengan *controller* dilakukan untuk melihat respons sistem apabila telah ditambahkan *controller* PID. Pengujian dilakukan untuk melihat respons kecepatan yang dihasilkan setelah dilakukan penambahan *controller*.

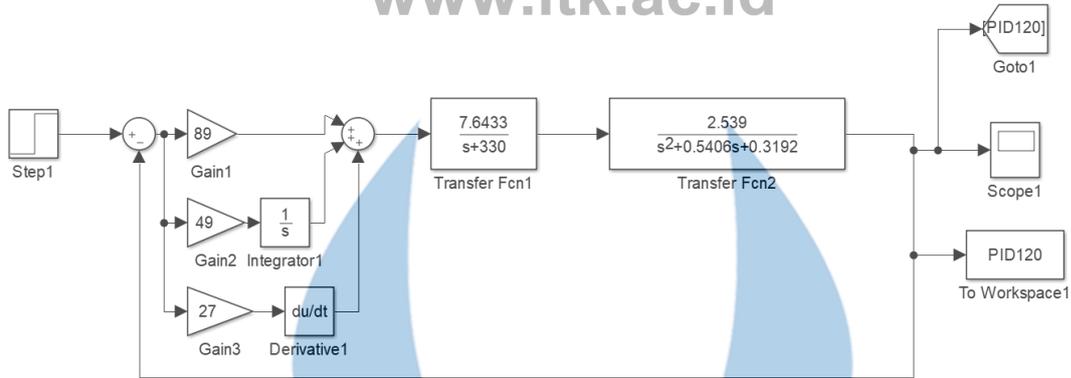
B.1. *Setpoint* 60 Rpm

Pengujian sistem ini dilakukan dalam bentuk *closed loop* dan menggunakan PID *controller* dengan *setpoint* berupa *step* sebesar 60 Rpm. Gambar 3.17 menunjukkan sistem *closed loop* dengan menggunakan *controller* PID dengan *setpoint* 60 Rpm.



Gambar 3.17 Sistem *close loop* dengan pid *controller* (*Setpoint* 60 Rpm)

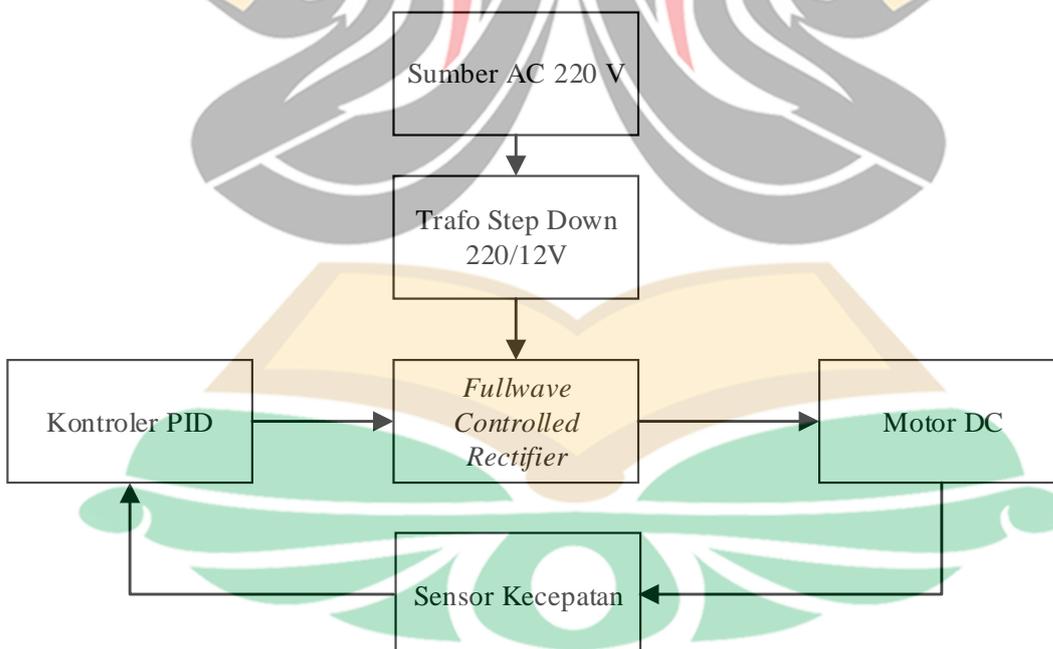
B.2 Setpoint 120 Rpm



Gambar 3.18 Sistem *close loop* dengan PID controller (Setpoint 120 Rpm)

Pengujian sistem ini dilakukan dalam bentuk *closed loop* dan menggunakan PID controller dengan *setpoint* berupa *step* sebesar 120. Gambar 3.18 menunjukkan sistem *closed loop* dengan menggunakan controller PID dengan *setpoint* 120.

3.8. Perancangan Prototype



Gambar 3.19 Blok diagram *prototype*

Pada Blok diagram ini menjelaskan komponen utama pengendali kecepatan dan putaran motor DC satu fasa yang terdiri dari beberapa bagian blok masing-masing yang memiliki fungsi kerja yang penting dan saling mendukung satu sama lainnya. Blok diagram pada Gambar 3.19 merupakan alat sebuah pengendali kecepatan motor DC.

Perencanaan perangkat keras ini meliputi *controlled rectifier fullwave* dengan SCR untuk mengubah tegangan AC ke tegangan DC sekaligus sebagai tempat pengaturan dengan modulasi lebar pulsa. Dengan menggunakan *controller* PID sebagai pengendali mikrokontroler Arduino.

3.8.1 SCR C106M

SCR C106M memiliki kapasitas yang besar. SCR tersebut memiliki tegangan maksimum sebesar 400 V dan arus 4 A. Sehingga SCR mampu digunakan untuk membuat rangkaian *controlled rectifier* sebagai sumber motor DC dengan tegangan 12 V dan arus 0,45 A.

3.8.2 Sensor Encoder

Sensor yang digunakan dalam tugas akhir ini adalah sensor *encoder*. Sensor *encoder* dipasang pada motor DC untuk membaca besar nilai kecepatan motor DC. Hasil pembacaan sensor digunakan sebagai *feedback* dan nilai referensi dalam rangkaian *controlled rectifier*.

3.8.3 Arduino Mega

Mikrokontroler yang digunakan adalah arduino mega. arduino mega sebagai *controller* pada rangkaian *controlled rectifier*. Penggunaan *controller* dengan menambahkan PID dan pembangkitan sinyal PWM untuk SCR. Selain sebagai *controller* arduino juga digunakan sebagai pengolah data dari kecepatan *setpoint* yang dibandingkan dengan kecepatan *feedback*.



Gambar 3.20 *Prototype controlled rectifier*

3.9. Pengujian *Prototype*

Pengujian *prototype* dilakukan dengan menggunakan *controller* dan tanpa *controller*. Pengujian dilakukan dengan variasi *setpoint* 60 Rpm dan 120 Rpm. Dalam tugas akhir ini waktu yang dibutuhkan untuk mengambil data selama 30 detik.

3.9.1 Tanpa *Controller*

Pengujian tanpa *controller* sistem diuji dalam keadaan *close loop*. Pengujian tersebut dilakukan dengan melakukan variasi kecepatan 60 Rpm dan 120 Rpm. Pengujian tersebut dilihat hasil respons kecepatan sistem jika tanpa *controller*.

3.9.2 Penambahan *Controller*

Pengujian dengan *controller* sistem diuji dalam keadaan *close loop*. Pengujian tersebut dilakukan dengan melakukan variasi kecepatan 60 Rpm dan 120 Rpm. Pengujian tersebut dilihat hasil respons kecepatan sistem jika ditambahkan dengan *controller* PID.