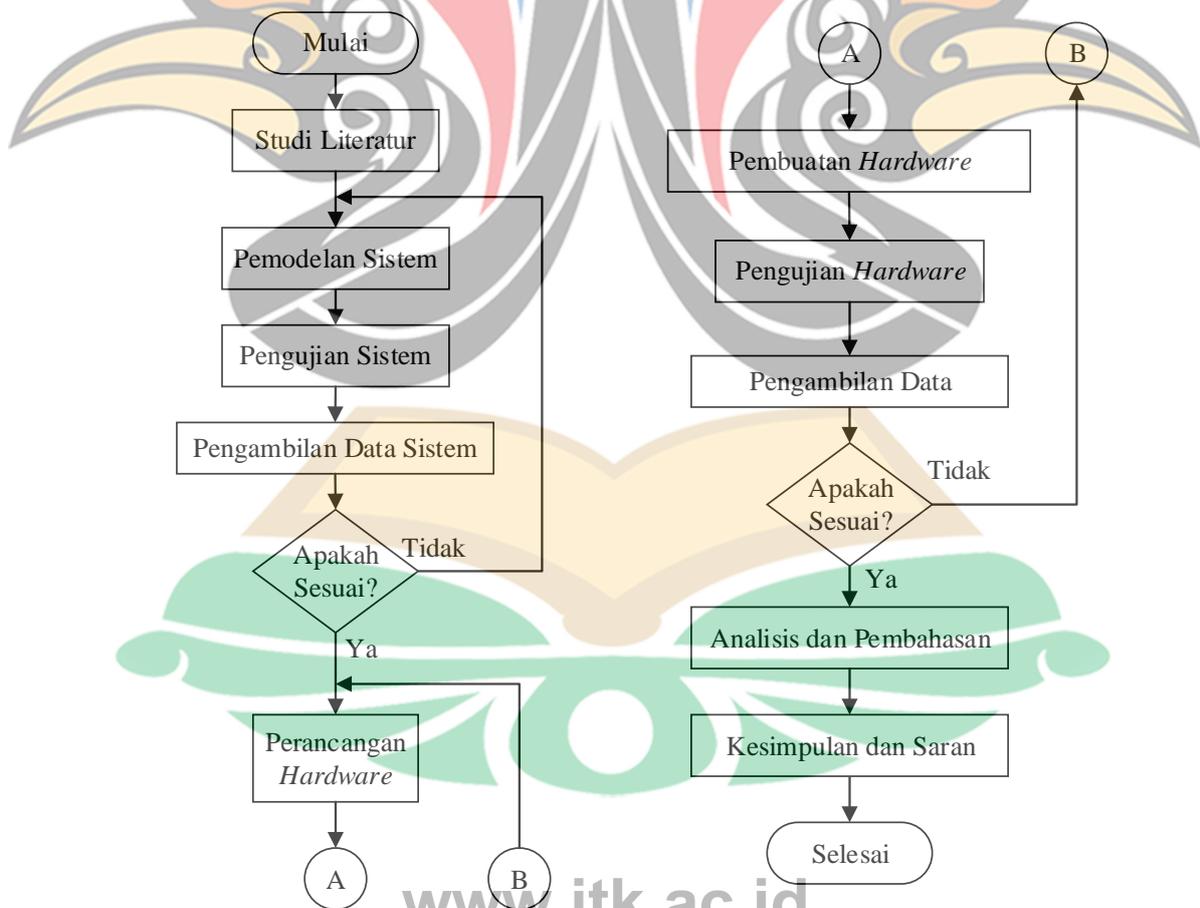


BAB III
METODOLOGI PENELITIAN

Bab ini berisi mengenai metode penelitian yang akan dilakukan, dimulai dari diagram alir yang merepresentasikan alur dari penelitian yang dilakukan agar terstruktur dan sistematis serta *time line* dari penelitian tersebut.

3.1 Diagram Alir Pengerjaan Tugas Akhir

Diagram alir pengerjaan merupakan tahapan-tahapan dalam penyelesaian tugas akhir. Tahapan-tahapan tersebut digunakan agar tidak terjadi kesalahan dalam proses pengerjaan sehingga mendapatkan hasil yang sesuai dengan tujuan dari tugas akhir. Adapun diagram alir pengerjaan tugas akhir pada gambar 3.1.



Gambar 3.1 Diagram Alir Pengerjaan

3.1.1 Studi Literatur www.itk.ac.id

Pada tahap penelitian ini dilakukan proses pencarian serta pengkajian literatur berupa buku, jurnal ilmiah nasional dan internasional, maupun sumber-sumber lain yang digunakan untuk membantu pengerjaan tugas akhir. Studi literatur pada penelitian ini dititik beratkan pada materi-materi yaitu :

1. Motor PMDC
2. Kontroler PI dan PID
3. *Software* Matlab
4. *Microcontroller* Arduino
5. Motor *driver*

3.1.2 Perancangan Sistem

Pada tahap ini dilakukan serangkaian pemodelan subsistem pada sistem pengendalian kecepatan motor DC dan juga perancangan parameter pengendalian. Perancangan sistem dilakukan dengan simulasi menggunakan *Simulink* pada *software* Matlab. Sebelum dilakukan simulasi terhadap sistem dilaksanakan terlebih dahulu perhitungan parameter dengan cara memodelkan parameter yang digunakan untuk simulasi.

Pemodelan subsistem yang dimaksudkan yaitu pemodelan dari parameter motor DC, pemodelan *driver* motor dan juga pemodelan sensor kedalam bentuk transfer function yang digunakan untuk pencarian nilai PID yang idela untuk menggerakkan sistem. Dilakukan pula perancangan nilai *setpoint* yang akan diberikan pada sistem nantinya pada tahap simulasi. Nilai *setpoint* yang dirancang nantinya akan terdiri dari nilai *setpoint* konstan kecepatan dari motor DC.

3.1.3 Pengujian Sistem

Setelah perancangan sistem, maka perlu dilakukan pengujian setiap subsistem penyusun penggerak motor. Pengujian sistem perlu dilakukan yang bertujuan memastikan bahwa sistem tersebut berjalan dengan baik dan sesuai dengan nilai *set-point*. Pengujian sistem dilakukan dengan beberapa metode pengujian yaitu pengujian *open loop*, pengujian *close loop* dan pengujian dengan kontroler PI dan

PID. Pada pengujian dengan kontroler PI dan PID dilakukan dengan perhitungan dengan melihat bentuk fungsi alih dari *plant* sehingga didapatkan nilai KP, KI dan KD dan memberikan variasi nilai *set point* kecepatan. Setelah semua berhasil diuji, selanjutnya adalah pengambilan data.

3.1.4 Pengambilan Data

Setelah semua sistem diuji pada simulasi Matlab maka langkah selanjutnya adalah pengambilan data. Pengambilan data dilakukan pada tiap-tiap pengujian. Data yang diperoleh adalah nilai tegangan, kecepatan motor, arus dan respons sistem. Semua data tersebut diambil dengan memvariasikan nilai setpoint dan tegangan *input*. Apabila data yang diambil semua maka dapat dilanjutkan ke tahap pembuatan alat/*hardware*. Apabila belum sesuai maka mengulang ke tahap pemodelan lagi.

3.1.5 Penentuan Keputusan

Tahap ini dilakukan setelah pengambilan data simulasi selesai. Dari hasil simulasi akan didapatkan hasil berupa hasil respons sistem dan akan dilihat nilai *error* dari keluaran respons sistem kemudian akan dilihat juga nilai *overshoot*, *settling time*, dan *rise time*. Tahap ini dilakukan untuk melakukan perancangan ulang parameter pengendali PID apabila respons sistem tidak sesuai dengan nilai sesuai.

3.1.6 Perancangan Hardware

Perancangan *hardware* mengacu pada perancangan sistem yang telah dilakukan. Motor PMDC sebagai *plant* dirangkai dengan *microcontroller* Arduino, motor *driver* dan diberi nilai tegangan masukan. Pada tahap ini juga dilakukan disain perangkat dan penyediaan komponen yang akan digunakan didalam penelitian.

3.1.7 Pembuatan Hardware

Pembuatan *hardware* dilakukan sesuai dengan perancangan sistem yang telah dilakukan. Pembuatan *hardware* dilakukan dengan merangkai komponen-

komponen yang digunakan dalam proses penelitian sesuai dengan disain yang telah dibuat.

www.itk.ac.id

3.1.8 Pengujian *Hardware*

Pengujian *hardware* dilakukan agar alat yang telah terpasang dipastikan dapat berjalan sesuai dengan yang diinginkan oleh sistem. Pengujian dilakukan dengan metode pengujian *open loop*, *close loop* dan pengujian dengan kontroler PI dan PID.

3.1.9 Pengambilan Data

Pengambilan data dilakukan ketika semua *hardware* telah diselesaikan dan siap untuk dijalankan. Adapun data yang diambil adalah :

1. Tegangan motor.
2. Kecepatan motor.
3. Nilai K_P , K_I , dan K_D

3.1.10 Penentuan Keputusan

Tahap ini dilakukan setelah pengambilan data pada *hardware* / alat selesai. Dari hasil *running* alat akan didapatkan hasil berupa hasil respons sistem dan akan dilihat nilai *error* dari keluaran respons sistem. Kemudian akan dilihat nilai *overshoot*, *settling time*, dan *rise time*. Tahap ini dilakukan untuk melakukan perancangan ulang parameter pengendali PID apabila respons sistem tidak sesuai dengan nilai sesuai.

3.1.11 Analisis dan Pembahasan

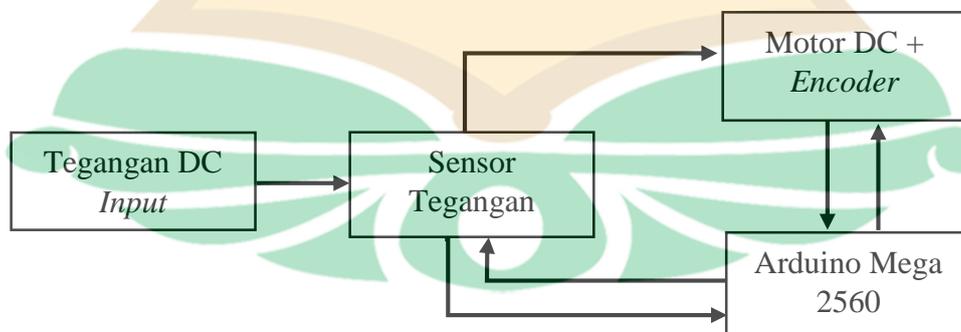
Analisis dan pembahasan dilakukan setelah didapatkan data dari simulasi dan implementasi. Analisis meliputi hasil data dari pengujian *open loop*, *close loop* dan pengujian dengan kontroler PI dan PID. Analisis diperlukan untuk membuktikan dan mencari nilai kecepatan yang sesuai dengan *set point*.

www.itk.ac.id

3.2 Pemodelan Fungsi Alih Motor

Pemodelan sistem merupakan tahap perancangan suatu model matematika dari sistem, baik sistem proses, sistem mekanik maupun sistem elektrik yang mampu merepresentasikan kinerja sistem mendekati kondisi *real* di lapangan. Pada tugas akhir ini, dilakukan serangkaian pemodelan subsistem dari sistem pengendalian pengendalian kecepatan motor DC sebagai penggerak *prototype conveyor* yaitu berupa pemodelan Motor PMDC tipe JGA25-370.

Pemodelan fungsi alih dilakukan dengan menggunakan cara sistem identifikasi dengan menggunakan *software* Matlab. Identifikasi sistem dilakukan untuk menyelesaikan merancang model matematika dari sistem dinamis berdasarkan data sistem yang diperoleh dari *plant*. Identifikasi sistem berguna untuk merepresentasikan sistem dinamis dan memastikan beberapa struktur model *black-box* linier dan nonlinier (Naung, dkk, 2018). Struktur *black-box* nonlinier dalam *toolbox* identifikasi sistem telah digunakan untuk mewakili waktu yang kontinu dari fungsi alih motor DC. Dalam arti yang lebih luas, identifikasi sistem adalah cara mudah untuk mendapatkan model matematika untuk sistem yang paling kompleks seperti motor DC (Naung, dkk, 2018). Pemodelan dilakukan dengan cara memberikan tegangan *input* pada motor dan *output* berupa kecepatan. Tegangan yang diberikan dimulai dari tegangan 1 volt dan dinaikan secara konstan hingga ke tegangan 12 volt kemudian tegangan diturunkan kembali dari 12 volt ke 1 volt. Tegangan dan kecepatan di rekam seiring dengan kenaikan tegangan. Gambar rangkaian skematik pengambilan data *input* tegangan dan data *output* kecepatan ditunjukkan pada gambar 3.2.



Gambar 3.2 Rangkaian Skematik Pengambilan Data Tegangan dan Kecepatan

Pengambilan data dilakukan dengan 3 kali percobaan. Data dari 3 kali proses pengambilan data menghasilkan data yang berbeda. Data tegangan terhadap kecepatan terdapat pada lampiran A.

Proses identifikasi sistem dilakukan dengan menggunakan *software* Matlab dengan menggunakan nilai tegangan terhadap kecepatan. Nilai tegangan masukan kepada motor sebagai parameter *input* dan nilai kecepatan sebagai parameter *output*. Nilai tegangan dan kecepatan yang di plot pada bagian *new variable*. Kemudian hasil plot tegangan dan kecepatan masuk kedalam jendela *workspaces*. Plot tegangan dan kecepatan nantinya dipanggil sebagai variabel *input* dan *output* pada identifikasi sistem. Kemudian masuk ke dalam menu *Identification System*. Di dalam menu dimasukan data *input* berupa nilai tegangan dan nilai kecepatan yang telah dikondisikan. Kemudian dipilih proses *transfer fuction* maka setelah itu akan didapatkan hasil nilai fungsi alih.

Proses identifikasi sistem dilakukan untuk 3 data pada lampiran A. Setelah dilakukan proses identifikasi sistem 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.

Dari proses identifikasi sistem 3 data yang digunakan maka didapatkan hasil nilai fungsi alih yang berbeda. Hasil nilai fungsi alih dari tiga data yang telah di ambil ditunjukkan pada tabel 3.1.

Tabel 3.1 Nilai Fungsi Alih Setiap Data

Data Ke	Nilai fungsi alih	Nilai fit to estimation
Data 1	$\frac{\omega(s)}{V_{in}(s)} = \frac{0,09316}{s^2 + 0,02739s + 0.01186}$	80,75 %
Data 2	$\frac{\omega(s)}{V_{in}(s)} = \frac{2.22e04}{s^2 + 2214s + 2725}$	89,85 %
Data 3	$\frac{\omega(s)}{V_{in}(s)} = \frac{91,31}{s^2 + 14,06s + 11,47}$	90,94 %

Nilai *fit to estimation* terbesar dari 3 data adalah data yang ketiga dengan nilai sebesar 90,94 %. Sehingga nilai fungsi alih data 3 digunaka sebagai nilai *plant* penelitian.

3.3 Sistem dengan Penambahan Kontroler PI

Penggunaan kontroler PI dilakukan dengan penambahan kontroler PI pada sistem pengaturan kecepatan motor. Kontroler PID mengatasi kecepatan motor yang tidak dapat mencapai nilai *set point*, *overshoot*, *settling time* dan *risetime*. Sinyal kontrol dari PI yang berupa tegangan *input* ke sistem motor PMDC (Deshpande, 1990).

Percobaan dilakukan pada sistem dengan melakukan penambahan kontroler PI. Gangguan yang terjadi pada sistem motor PMDC adalah kecepatan tidak sampai *set point*. Jika terjadi perubahan beban pada sistem maka sistem akan mengalami gangguan karena disebabkan sistem pada motor PMDC tersebut saling terinterkoneksi menjadi satu. Gangguan-gangguan yang terjadi karena perubahan beban sehingga kecepatan tidak sampai *set point* yang diinginkan.

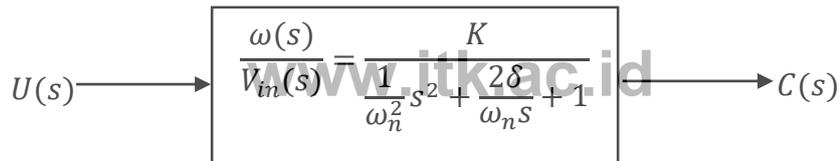
Perencanaan parameter kontroler PI dapat dijabarkan sebagai berikut : Dengan menggunakan sistem identifikasi sistem maka diperoleh model matematika (fungsi alih) dari motor DC dengan penguatan medan konstan yaitu :

$$\frac{\omega(s)}{V_{in}(s)} = \frac{91,31}{s^2 + 14,06s + 11,47} \quad (3.1)$$

Setelah *plant* didapatkan fungsi alih, maka dapat dicari nilai kontroler PI. *Plant* yang dihasilkan adalah *plant* orde kedua sehingga dapat dipresentasikan dalam bentuk berikut ini.

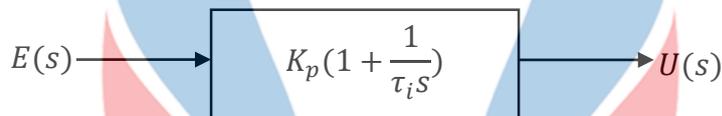
$$\frac{\omega(s)}{V_{in}(s)} = \frac{91,31}{s^2 + 14,06s + 11,47}$$
$$n(s) = \frac{7,96}{0,087s^2 + 1,2258s + 1} \quad (3.2)$$

Suatu *plant* orde kedua tanpa *delay* dapat direpresentasikan dalam bentuk diagram blok ditunjukkan pada gambar 3.3.



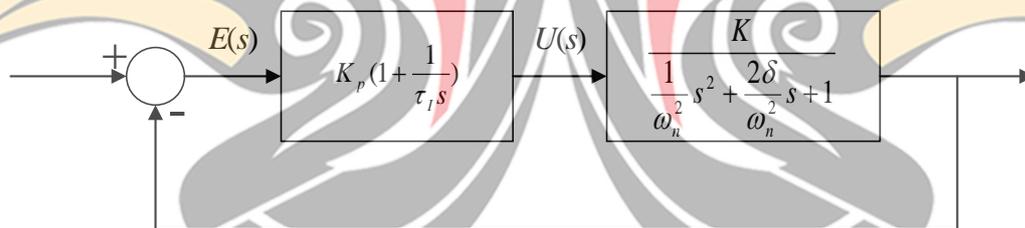
Gambar 3.3 Diagram Blok *Plant* Orde Kedua *Open Loop*

K adalah *Gain overall*, ω_n adalah frekuensi alami tak teredam, dan δ adalah rasio peredaman. Sedangkan diagram blok dari kontroler PI ditunjukkan pada gambar 3.4.



Gambar 3.4 Diagram Blok Kontroler PI

K_p adalah penguatan proporsional dan τ_i adalah waktu integral. Jika kontroler proporsional ditambah integral diterapkan pada *plant* orde kedua tanpa *delay* dalam suatu sistem pengaturan maka diagram bloknya adalah pada gambar 3.5.



Gambar 3.5 Diagram Blok Kontroler PI Diterapkan pada *Plant* Orde Kedua.

Oleh karena itu berdasarkan persamaan (3.2), perencanaan parameter kontroler PI dapat dijabarkan sebagai berikut :

Dari persamaan tersebut maka diperoleh nilai $K = 7,96$ sedangkan nilai ω_n diperoleh dari persamaan :

$$\frac{1}{\omega_n^2} = 0,087$$

$$\omega_n = \sqrt{\frac{1}{0,087}}$$

$$\omega_n = 3,39 \quad (3.3)$$

Nilai δ dapat diperoleh dari persamaan berikut :

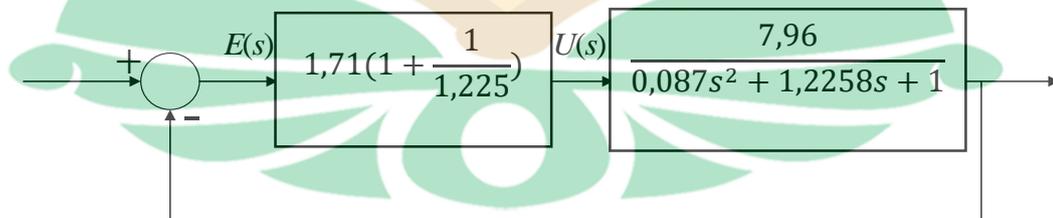
$$\begin{aligned} \frac{2\delta}{\omega_n} &= 1,2258 \\ \delta &= \frac{(1,2258)(3,39)}{2} \\ \delta &= 2,077 \end{aligned} \quad (3.4)$$

Besarnya waktu integral (τ_i) dapat diperoleh dari persamaan berikut :

$$\begin{aligned} \tau_i &= \frac{2\delta}{\omega_n} \\ \tau_i &= \frac{(2)(2,077)}{3,39} \\ \tau_i &= 1,225 \end{aligned} \quad (3.5)$$

Besarnya penguatan proporsional (K_p) dapat diperoleh dari persamaan berikut:

$$\begin{aligned} K_p &= \frac{2\delta}{\tau^* \omega_n k} \\ K_p &= \frac{(2)(2,077)}{(0,09)(3,39)(7,96)} \\ K_p &= 1,71 \end{aligned} \quad (3.6)$$



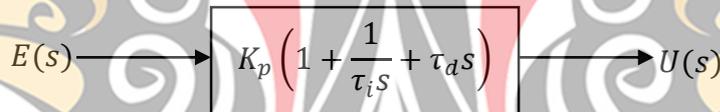
Gambar 3.6 Diagram Blok Sistem Pengaturan PI Motor PMDC

Setelah semua nilai parameter didapatkan maka diagram blok dalam suatu sistem pengaturan ditunjukkan pada gambar 3.6.

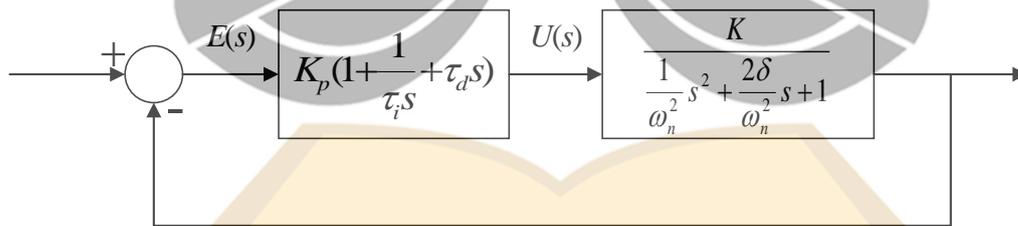
3.4 Sistem dengan Penambahan Kontroler PID

Pengaplikasian kontroler PID dilakukan dengan penambahan kontroler PID pada sistem. Kontroler PID (proporsional, integral dan diferensial) merupakan kontroler yang aksi kontrolernya mempunyai sifat proporsional, integral dan diferensial terhadap sinyal kesalahan. Perancangan kontroler PID pada tugas akhir ini diterapkan pada *plant* orde kedua sama seperti *plant* pada bagian kontroler PI. Dengan menentukan penguatan proporsional K_p , waktu integral τ_i , dan waktu differensial τ_d yang tepat diharapkan respons *plant* sesuai dengan performasi yang diinginkan.

Plant orde kedua telah di presentasikan pada gambar 3.6. Sedangkan diagram blok dari kontroler PID adalah pada gambar 3.7 dan 3.8.



Gambar 3.7 Diagram Blok Kontroler PID



Gambar 3.8 Diagram Blok Kontroler PID Diterapkan pada *Plant* Orde Kedua

K_p adalah penguatan proporsional, τ_i adalah waktu integral dan waktu τ_d adalah waktu diferensial. Jika kontroler proporsional ditambah integral dan ditambah differensia diterapkan pada *plant* orde kedua tanpa *delay* dalam suatu sistem pengaturan maka diagram bloknnya adalah pada gambar 2.8.

Perancangan parameter kontroler PID memiliki perhitungan yang sama seperti perancangan kontroler PI namun harus mencari nilai parameter τ_d . Tabel 3.2 adalah hasil nilai-nilai perhitungan yang telah dihasilkan.

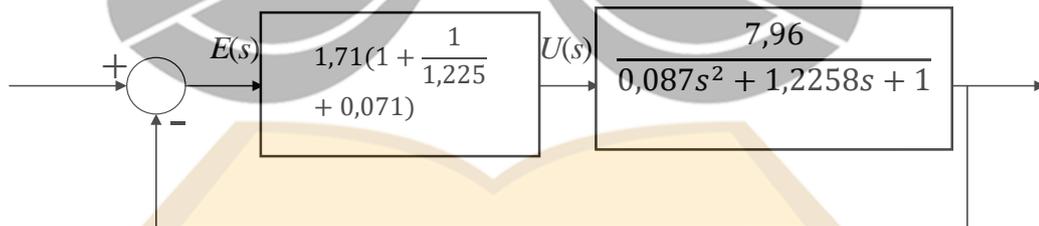
Tabel 3.2 Nilai Perhitungan PI

No	Parameter	Nilai
1	K	7,96
2	ω_n	3,39
3	δ	2,077
4	τ_i	1,225
5	K_p	1,5394

Besarnya nilai penguatan differensial (τ_d) dapat diperoleh dari persamaan berikut:

$$\begin{aligned} \tau_d &= \frac{1}{2\delta\omega_n} \\ \tau_d &= \frac{1}{2 \times 2,077 \times 3,39} \\ \tau_d &= 0,071 \end{aligned} \quad (3.7)$$

Setelah semua nilai parameter didapatkan maka diagram blok dalam suatu sistem pengaturan adalah pada gambar 3.9.



Gambar 3.9 Diagram Blok Sistem Pengaturan PID Motor PMDC

3.5 Perancangan Simulasi Sistem dengan Simulink

Perancangan simulasi sistem dilakukan untuk menguji sistem berdasarkan nilai fungsi alih motor dan nilai kotroler PI atau PID yang telah di dapatkan. Simulasi sistem dilakukan dengan kondisi *open loop*, *close loop* tanpa kontroler dan *close loop* dengan kontroler PI dan PID. Kemudian untuk masing-masing kondisi

dilakukan variasi nilai *set point* kecepatan yaitu 80 rpm, 100 rpm dan 120 rpm. Simulasi dilakukan untuk mengetahui respons sistem dari masing-masing kondisi.

3.5.1 Simulasi Sistem *Open Loop*

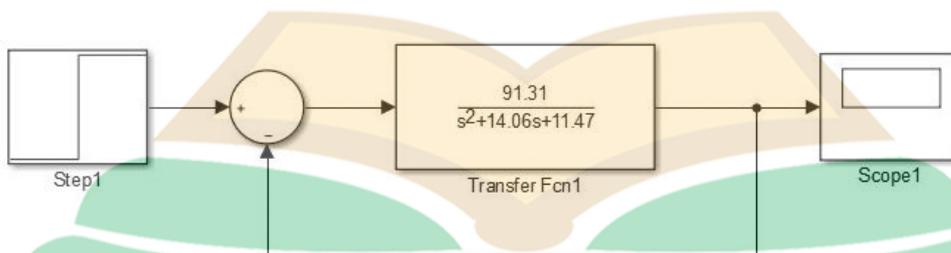
Simulasi *open loop* dilakukan untuk mengetahui respons motor saat digerakan. Pada simulasi *open loop* tidak terdapat nilai *feed back* sehingga sistem tidak dapat diatur kecepatannya dan tidak dapat menuju *set point* yang diinginkan. Dari hasil respons sistem saat *open loop* dapat diketahui apakah respons sistem baik atau buruk. Gambar 3.10 adalah diagram blok sistem *open loop* pada *Simulink*.



Gambar 3.10 Simulasi Sistem dengan *Open loop*

3.5.2 Simulasi Sistem *Close Loop*

Simulasi *close loop* dilakukan untuk mengetahui respons motor saat digerakan. Pada simulasi *close loop* terdapat *feed back* namun sistem tidak dapat diatur kecepatannya dan tidak dapat menuju *set point* yang diinginkan. Dari hasil respons sistem saat *open loop* dapat diketahui apakah respons sistem baik atau buruk. Gambar 3.11 adalah diagram blok sistem *open loop* pada *Simulink*.

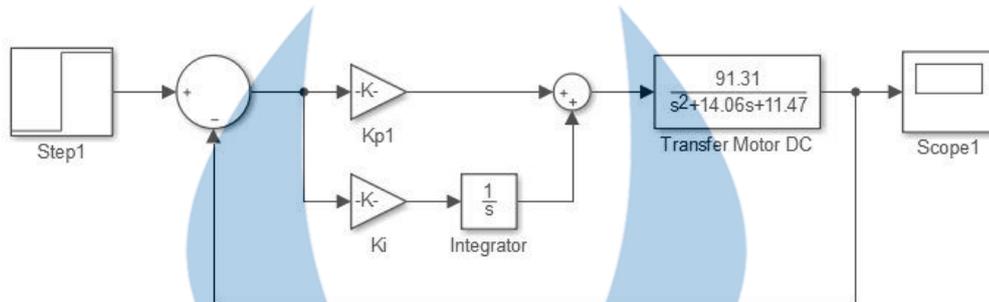


Gambar 3.11 Diagram Blok Simulasi Sistem dengan *Close loop*

3.5.3 Simulasi Sistem *Close Loop* dengan Kontroler PI

Simulasi sistem *close loop* dilakukan dengan menambahkan kontroler PI (proporsional dan integrator). Hasil nilai *feed back* digunakan referensi untuk

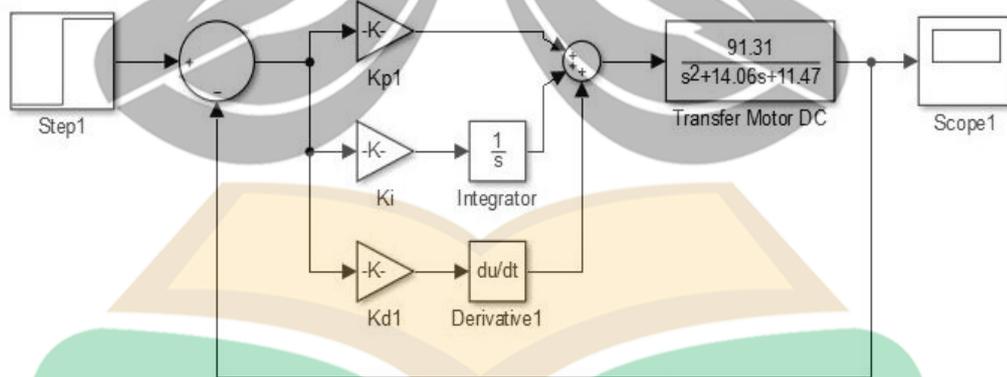
memperbaiki respons dari hasil respons sistem saat digerakan dan mempercepat kecepatan dapat menuju *set point* yang telah ditentukan. Gambar 3.12 adalah diagram blok sistem dengan kontroler PI.



Gambar 3.12 Diagram Blok Simulasi Sistem dengan Kontroler PI

3.5.4 Simulasi Sistem *Close Loop* dengan Kontroler PID

Simulasi sistem *close loop* dilakukan dengan menambahkan kontroler PID (proporsional, integrator dan derivatif). Hasil nilai *feed back* digunakan referensi untuk memperbaiki respons dari hasil respons sistem saat digerakan dan mempercepat kecepatan dapat menuju *set point* yang telah ditentukan. Gambar 3.13 merupakan diagram blok sistem dengan kontroler PID.

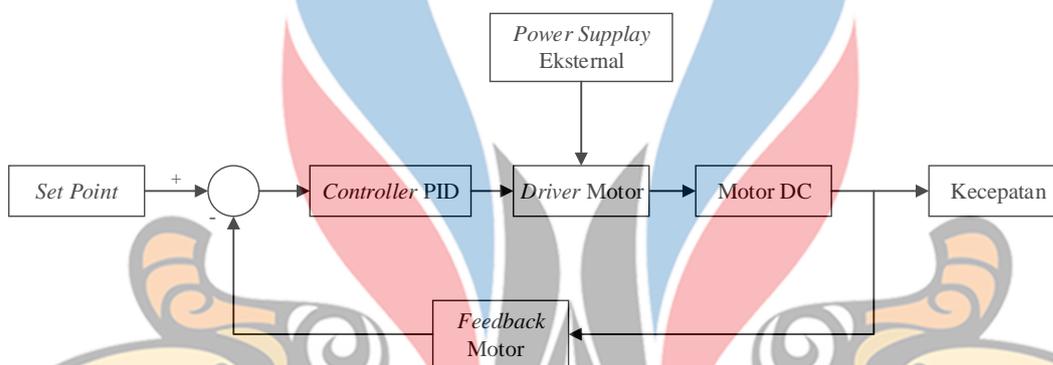


Gambar 3.13 Diagram Blok Simulasi Sistem dengan Kontroler PID

3.6 Perancangan Sistem Pengaturan Kecepatan Motor DC

Dilakukan perancangan sistem pengereman otomatis bertujuan untuk mempermudah merealisasikan *prototype* sistem pengendalian kecepatan motor yang akan dibuat. Blok diagram sistem pengaturan kecepatan ditunjukkan pada gambar 3.7.

Cara kerja dari sistem pengendalian kecepatan motor DC yang akan dirancang adalah kontroler PI dan PID mengeluarkan sinyal *Pulse Width Modulation* (PWM) untuk motor *driver*. Sinyal PWM berfungsi untuk mengatur tegangan eksternal yang masuk agar nilai tegangan yang keluar dari motor *driver* sesuai dengan *set point*. Besar nilai tegangan berbanding lurus dengan nilai kecepatan motor DC. Pada bagian motor DC telah dipasang sensor kecepatan yang dapat membaca besar nilai kecepatan. Nilai besar kecepatan digunakan sebagai nilai referensi untuk *looping* kembali ke kontroler agar besar nilai kecepatan dapat menuju *set point*.



Gambar 3.14 Perancangan Sistem Pengaturan Kecepatan Motor DC

Secara umum sistem penegndalian kecepatan motor DC dibagi menjadi 3 proses yaitu masukan, proses dan keluaran. Berikut ini uraian singkat fungsi dari masing-masing bagian utama blok diagram sistem pengaturan kecapata motor DC dengan menggunakan kontroler PI dan PID.

3.6.1 Masukan Sistem

Pada bagia masukan terdapat niali *set point*, *feed back* pembacaan sensor kecepatan berfungsi untuk masukan bagi kontroler PI dan PID serta nilai PI dan PID yang telah di tentukan.

A. *Set Point*

Nilai *set point* adalah nilai kecepatan motor yang diinginkan. Pada proses penelitian nilai *set point* yang diberikan adalah 80 rpm, 100 rpm dan 120 rpm untuk kontroler PI dan PID. Nilai *set point* dimasukkan di dalam pemrograman mirokontroler Arduino at-Mega 2560.

B. *Feed Back Motor*

Sensor yang digunakan dalam penelitian adalah sensor kecepatan yang telah terpasang pada motor DC. Sensor kecepatan berfungsi untuk membaca besar nilai kecepatan saat motor digerakan. Hasil pembacaan sensor digunakan untuk referensi dan menjadi masukan di dalam program.



Gambar 3.15 Motor PMDC JGA 25-370 dengan Sensor Kecepatan

C. *Nilai PI dan PID*

Nilai PI dan PID yang telah di dapat dalam sub bab sistem penambahan dengan kontrol PI dan sistem penambagan dengan kontrol PID. Nilai PI dan PID digunakan sebagai parameter dalam program yang digerakan.

3.6.2 Proses Sistem

Arduino mega 2560 digunakan sebagai perangkat kontrol utama perancangan sistem penegndalian kecepatan motor DC dengan menggunakan kontroler PID pada penelitian ini. *Microcontroller* memproses setiap masukan dengan PI dan PID serta mengeksekusi perangkat keluaran sesuai dengan instruksi program yang telah diatur. Hasil proses dari *microcontroller* mengasilakn nilai PWM yang digunakan untuk menentukan kecepatan motor DC dengan nilai 0-255. Nilai tersebut akan dikonversi menjadi nilai tegangan keluaran 0-12 volt yang di dihasilkan oleh motor *driver*. Nilai PWM akan berosilasi agar mengeluarkan nilai tegangan yang sesuai untuk mencapai kecepatan sesuai dengan *set point*.

3.6.3 Keluaran Sistem

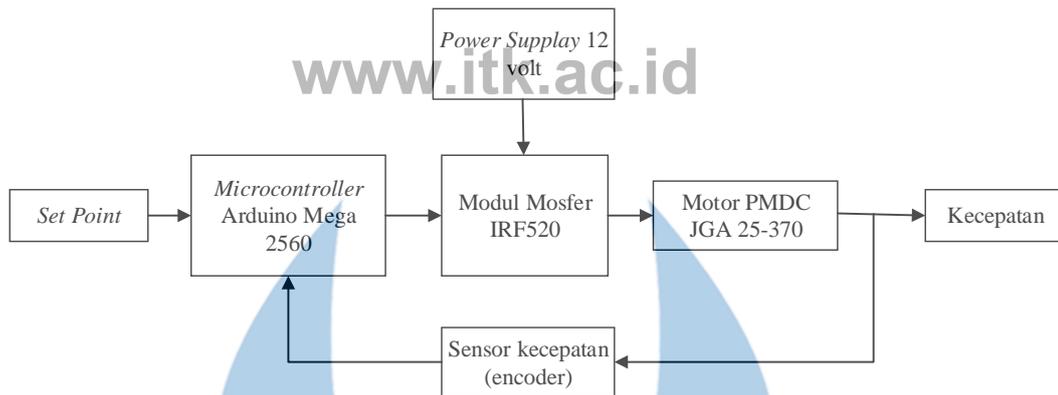
Keluaran berfungsi sebagai aktuator dari hasil proses pengolahan masukan yang dilakukan oleh PI dan PID yang terintegrasi dengan *microcontroller*. Perangkat keluaran yang digunakan dalam perancangan sistem penegndaliakn kecepatan motor DC adalah motor *driver*. Motor *driver* yang digunakan dalam penelitian ini adalah modul mosfet IRF 520. Modul mosfer IRF 520 merupakan sebuah perangkat semionduktor yang secara luas di gunakan sebagai *switching*. Keuntungan menggunakan modul mosfet IRF 520 adalah dapat langsung diintegrasikan dengan *microcontroller* Arduino. Gambar mosfet IRF 520 ditunjukkan pada gambar 2.14.

3.7 Pembuatan *Hardware* Sistem Pengendalian Kecepatan Motor DC

Pembuatan *hardware* pada sistem pengendalian kecepatan motor DC, dilakukan dengan melakukan perancangan dan pembuatan rangkaian mekanis. Dilakukan pada *prototype* pengendalian motor DC dan sistem kontrol yang meliputi pembuatan rangkaian-rangkaian elektronik yang saling terintegrasi membentuk suatu sistem kendali dengan tujuan mengendalikan kecepatan motor DC agar dapat bekerja secara optimal.

3.7.1 Perancangan Pembautaan *Hardware*

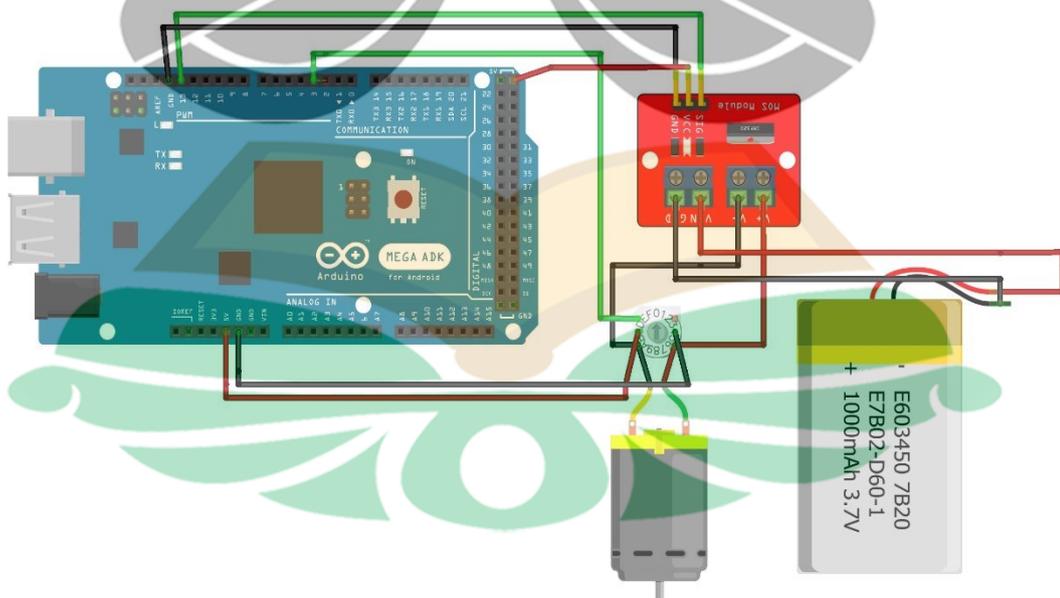
Perancangan pembuatan *hardware* dilakukan dengan tujuan memberikan gambaran rangkaian antar komponen yang digunakan dalam proses pembuatan *prototype*. Perancangan pembuatan *hardware* merujuk kepada perancangan sistem yang telah dilakukan. Sistem arduino mega 2560 adalah rangkaian yang dikhususkan untuk mengoperasikan IC (*Integrated Circuit*) *microcontroller*. *Microcontroller* inilah yang digunakan sebagai pusat sistem kendali pada kecepatan motor DC. Arduino mega 2560 memiliki *Port I/O* yang difungsikan untuk menerima masukan dari sensor kecepatan dan nilai PWM serta keluaran menuju modul mosfet IRF 520.



Gambar 3.16 Perancangan *Hardware Prototype* Sistem Pengendalian Kecepatan Motro DC

3.7.2 *Wiring Diagram* Rangkaian Sistem Penegndalian Kecepatan Motor DC

Wiring diagram berfungsi sebagai penentuan jalur peyambungan antar komponen pada distem pegendalian kecepatan motor DC untuk pembuatan *prototype*. Pemasangan *wiring diagram* sistem dilakukan dengan memperhatikan fungsi pin di Arduino Mega 2560 agar dapat memproses sinyal *input* dan mengeluarkan hasil sinyal *output* untuk sistem. Pemasangan konfigurasi *wiring* pin Arduino ditunjukan pada tabel 3.3 dan untuk *wiring diagram* pin Arduino ditunjukan pada gambar 3.17.



Gambar 3.17 Rangkaian *Wiring Diagram Hardware*

Tabel 3.3. Konfigurasi Pin Arduino

Pin Arduino	Keterangan
Pin 3	Sinyal PWM dari encoder
Pin 13	Sinyal PWM ke mosfet
Ground	Semua pin Ground
5 v	Vin mosfet dan encoder

3.7.3 Perancangan dan Pembuatan Perangkat Lunak

Perancangan dan pembuatan perangkat lunak dilakukan dengan melakukan perancangan dan pembuatan program pada sistem PID dan sistem *embedded* PID ke Arduino Mega 2560, perancangan perangkat lunak bertujuan untuk menentukan setiap alur kerja dari sistem pengaturan kecepatan motor DC. Setiap masukan yang diterima akan diatur oleh PID yang selanjutnya akan diproses untuk menentukan eksekusi pada bagian keluaran. Sistem *embedded* PID ke arduino mega 2560 ditunjukkan pada lampiran D.

3.7.4 Rangkaian Sistem Pengendalian Kecepatan Motor DC

Rangkaian sistem penegndalian kecepatan motor DC berfungsi sebagai aktuator dari hasil proses yang diolah oleh Arduino Mega 2560 dengan kontroler PI dan PID. Rangkaian ini terdiri dari motor PMDC sebagai *plant*, mosfet IRF 520 sebagai motor *driver* dan *microcontroller* Arduino Mega 2560 sebagai pengolah program dan penghasil sinyal PWM. Mekanisme pemasangan sistem pengendalian telah terpasang di *prototype conveyor*. *Prototype conveyor* dapat dilihat pada lampiran E.