



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

**PENGENDALIAN POSISI SUDUT *JOINT* ROBOT
MANIPULATOR 4 DOF (*DEGREE OF FREEDOM*)**

Diana Yufika Anshori
NRP 2214030021

Dosen Pembimbing
Slamet Budiprayitno, S.T., M.T.
Yunafi'atul Aniroh, S.T., M.Sc.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

CONTROL POSITION OF JOINT 4 DOF (DEGREE OF FREEDOM) MANIPULATOR ROBOT

Diana Yufika Anshori
NRP 2214030021

Advisor
Slamet Budiprayitno, S.T., M.T.
Yunafi'atul Aniroh, S.T., M.Sc.

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pengendalian Posisi Sudut Joint Robot Manipulator 4 Degree of Freedom**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017



Diana Yufika Anshori
NRP 2214030021

-----Halaman ini sengaja dikosongkan-----

**PENGENDALIAN POSISI SUDUT *JOINT ROBOT*
MANIPULATOR 4 DOF (*DEGREE OF FREEDOM*)**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik
Pada
Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing 1

Dosen Pembimbing 2


Slamet Budipravitno, S.T., M.T.


Yunani Atul Aniroh, S.T., M.Sc.

NIP. 19781113 201012 1 002 DEPARTEMEN NIP. 2200201405001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

PENGENDALIAN POSISI SUDUT *JOINT* ROBOT MANIPULATOR 4 DOF (*DEGREE OF FREEDOM*)

Nama : Diana Yufika Anshori
Pembimbing : Slamet Budiprayitno, S.T., M.T.
Yunafi'atul Aniroh, S.T., M.Sc.

ABSTRAK

Tugas Akhir ini membahas masalah kontrol posisi pada *joint* robot manipulator 4 *Degree of Freedom*. Aktuator pada *joint-joint* tersebut berupa dua motor DC dan tiga motor servo. Motor servo digunakan untuk menggerakkan *joint* pada bagian *elbow*, *wrist*, dan *end effector*. Sedangkan motor DC digunakan untuk menggerakkan *joint* pada bagian *base* dan *shoulder*. Motor DC diberi *feedback* berupa sensor *rotary encoder* yang dapat mendeteksi perubahan sudut pada *joint* robot manipulator. Pada Tugas Akhir ini digunakan kontroler PID agar motor DC dapat berputar dan berhenti pada posisi sudut yang diinginkan. Untuk memudahkan *user* dalam memonitor serta mengontrol pergerakan *joint* robot manipulator, maka dibuatlah *Human Machine Interface* (HMI) menggunakan LabVIEW 2015. Hasil simulasi dan implementasi menunjukkan bahwa pada pengujian sensor *rotary encoder* didapatkan *error* rata-rata sebesar 0,074423% pada bagian *base* dan 0,005181% pada bagian *shoulder*. Pada pengujian *joint* ketika tanpa kontroler, respon perubahan sudut *joint* pada robot manipulator terjadi *overshoot*, *osilasi*, dan tidak pernah mencapai keadaan *steady*. Namun, pada pengujian ketika diberi kontroler, respon perubahan sudut *joint* pada robot manipulator mampu mencapai keadaan *steady*.

Kata Kunci : Robot Manipulator, Sensor *Rotary Encoder*, Kontroler PID, LabVIEW 2015

-----Halaman ini sengaja dikosongkan-----

CONTROL POSITION OF JOINT 4 DOF (DEGREE OF FREEDOM) MANIPULATOR ROBOT

Name : Diana Yufika Anshori
Advisor : Slamet Budiprayitno, S.T., M.T.
Yunafi'atul Aniroh, S.T., M.Sc.

ABSTRACT

This final project discusses the problem of control position in joint robot manipulator 4 DOF (Degree of Freedom). Actuators in the joints are two DC motors and three servo motors. Servo motors are used to drive joints on the elbow, wrist, and end effector. While the DC motor is used to move the joint on the base and shoulder. DC motors are given feedback in the form of rotary encoder sensors that can detect angular changes in the joint robot manipulator. In this final project, a PID controller is used to allow the DC motor to rotate and stop at the desired angle position. To facilitate the user in monitoring and controlling the movement of joint robot manipulator, Human Machine Interface (HMI) was made using LabVIEW 2015. The results of simulation and implementation showed that in the rotary encoder sensor test, the average error was 0,074423% on the base and 0,005181% on the shoulder. At joint testing when no controller, the joint angular change response on the robot manipulator occurs overshoot, oscillation, and never reaches steady state. However, on testing when given a controller, the joint angular change response on the robot manipulator is able to reach steady state.

Keywords : Manipulator Robot, Rotary Encoder Sensor, PID Controller, LabVIEW 2015

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

PENGENDALIAN POSISI SUDUT *JOINT ROBOT* MANIPULATOR 4 DOF (*DEGREE OF FREEDOM*)

Dalam Tugas Akhir ini dirancang sebuah kontroler untuk mengendalikan posisi sudut pada *joint robot* manipulator 4 DOF (*Degree of Freedom*) dengan menggunakan kontroler PID (*Proportional + Integral + Derivative*).

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Slamet Budiprayitno, S.T., M.T. dan Ibu Yunafi'atul Aniroh, S.T., M.Sc. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 19 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
BAB II DASAR TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Robot	5
2.3 Robot Manipulator	7
2.4 <i>Rotary Encoder</i>	8
2.5 LabVIEW	9
2.6 NI myRIO	9
2.7 Motor DC	11
2.8 Kontroler PID.....	12
BAB III PERANCANGAN DAN PEMASANGAN ALAT.....	15
3.1 Perancangan Robot Manipulator	15
3.1.1 Spesifikasi Robot Manipulator	15
3.1.2 Pemasangan Sensor <i>Rotary Encoder</i>	15

3.1.3	Pengkabelan NI myRIO dengan Motor DC dan Sensor Posisi.....	17
3.1.4	Pengkabelan NI myRIO	18
3.2	Cara Kerja Sistem	19
3.3	Blok Diagram Sistem	19
3.4	Perancangan <i>Software</i>	20
3.5	Linearisasi Sensor <i>Rotary Encoder</i>	24
3.6	Perancangan Pengujian <i>Driver Motor L298N</i>	25
3.7	Perancangan Pengujian Posisi Motor Servo	26
3.8	Perancangan Kontroler PID	27
BAB IV HASIL SIMULASI DAN IMPLEMENTASI		29
4.1	Pengujian <i>Driver Motor L298N</i>	29
4.2	Pengujian Sensor <i>Rotary Encoder</i>	33
4.2.1	Pengujian Sensor <i>Rotary Encoder</i> pada <i>Joint Base</i>	33
4.2.2	Pengujian Sensor <i>Rotary Encoder</i> pada <i>Joint Shoulder</i> ..	36
4.3	Pengujian Sudut pada Motor Servo	37
4.4	Pengujian <i>Joint</i> pada <i>Base</i>	41
4.4.1	Respon Perubahan Posisi pada <i>Joint Base</i> Tanpa Kontroler.....	41
4.4.2	Respon Perubahan Posisi pada <i>Joint Base</i> Dengan Kontroler	43
4.4.3	Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada <i>Joint Base</i>	48
4.5	Pengujian <i>Joint</i> pada <i>Shoulder</i>	50
4.5.1	Respon Perubahan Posisi pada <i>Joint Shoulder</i> Tanpa Kontroler.....	50
4.5.2	Respon Perubahan Posisi pada <i>Joint Shoulder</i> Dengan Kontroler	52
4.5.3	Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada <i>Joint Shoulder</i>	57
BAB V PENUTUP		59
DAFTAR PUSTAKA		61
LAMPIRAN A DATA PENGUJIAN.....		63
A.1	Gambar Gelombang pada Pengujian <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 1-1	63
A.2	Respon Perubahan Sudut <i>Joint Base</i> dengan Nilai K_p 2, T_i 0, T_d 0	70

A.3 Respon Perubahan Sudut <i>Joint Base</i> dengan Nilai K_p 1, T_i 0, T_d 0	72
A.4 Respon Perubahan Sudut <i>Joint Base</i> dengan Nilai K_p 1, T_i 0,02, T_d 0	73
A.5 Respon Perubahan Sudut <i>Joint Base</i> dengan Nilai K_p 1, T_i 0,09, T_d 0	74
A.6 Respon Perubahan Sudut <i>Joint Base</i> dengan Nilai K_p 1, T_i 0,09, T_d 0,001	76
A.7 Respon Perubahan Sudut <i>Joint Shoulder</i> dengan Nilai K_p 2, T_i 0, T_d 0.....	77
A.8 Respon Perubahan Sudut <i>Joint Shoulder</i> dengan Nilai K_p 1, T_i 0, T_d 0.....	78
A.9 Respon Perubahan Sudut <i>Joint Shoulder</i> dengan Nilai K_p 1, T_i 0,02, T_d 0.....	80
LAMPIRAN B <i>LISTING PROGRAM</i>	83
B.1 <i>Listing Program</i> Pengujian Driver Motor	83
B.2 <i>Listing Program</i> Pengujian Motor Servo	83
B.3 <i>Listing Program</i> Pengujian <i>Joint</i> pada <i>Base</i>	84
B.4 <i>Listing Program</i> Pengujian <i>Joint</i> pada <i>Shoulder</i>	85
LAMPIRAN C <i>DATASHEET</i>	87
C.1 <i>Datasheet</i> NI myRIO 1900	87
C.2 <i>Datasheet</i> Driver Motor L298N	90
C.3 <i>Datasheet</i> Motor Servo Hitec HS-485HB.....	92
C.4 <i>Datasheet</i> Motor Servo Hitec HS-331	93
C.5 <i>Datasheet</i> Motor Servo MG995	94
LAMPIRAN D DOKUMENTASI ALAT	95
RIWAYAT HIDUP PENULIS	97

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Robot Nao	6
Gambar 2.2	<i>Degree of Freedom</i>	7
Gambar 2.3	Prinsip Kerja <i>Rotary Encoder</i>	8
Gambar 2.4	<i>Incremental Rotary Encoder</i>	8
Gambar 2.5	Bentuk Fisik NI myRIO	10
Gambar 2.6	Motor DC	11
Gambar 2.7	Kontrol Robot <i>Loop</i> Terbuka	12
Gambar 2.8	Kontrol Robot <i>Loop</i> Tertutup	12
Gambar 3.1	Ilustrasi Perbandingan <i>Gear</i> Pada Motor DC dan <i>Rotary Encoder</i> Bagian <i>Base</i>	16
Gambar 3.2	Ilustrasi Perbandingan <i>Gear</i> Pada Motor DC dan <i>Rotary Encoder</i> Bagian <i>Shoulder</i>	16
Gambar 3.3	Pengkabelan NI myRIO dengan Motor DC dan <i>Rotary Encoder</i>	18
Gambar 3.4	<i>Hardware</i> NI myRIO	18
Gambar 3.5	Blok Diagram Sistem pada Motor Servo	20
Gambar 3.6	Blok Diagram Sistem pada Motor DC	20
Gambar 3.7	Rancangan HMI pada LabVIEW 2015	20
Gambar 3.8	Bagian A pada HMI di LabVIEW 2015	21
Gambar 3.9	Bagian B pada HMI di LabVIEW 2015	21
Gambar 3.10	Bagian C pada HMI di LabVIEW 2015	22
Gambar 3.11	Blok Diagram Perancangan HMI pada LabVIEW 2015	22
Gambar 3.12	Bagian A Blok Diagram Perancangan HMI	23
Gambar 3.13	Bagian B Blok Diagram Perancangan HMI	23
Gambar 3.14	Bagian C Blok Diagram Perancangan HMI	23
Gambar 3.15	Bagian D Blok Diagram Perancangan HMI	24
Gambar 3.16	Bagian E Blok Diagram Perancangan HMI	24
Gambar 3.17	Rangkaian Pengujian <i>Driver Motor</i> L298N	25
Gambar 3.18	<i>Front Panel</i> LabVIEW untuk Pengujian <i>Driver Motor</i>	25
Gambar 3.19	<i>Wiring</i> Motor Servo dan NI myRIO	26
Gambar 3.20	<i>Front Panel</i> LabVIEW untuk Pengujian Motor Servo	27
Gambar 4.1	Grafik Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 1-0	30

Gambar 4.2	Grafik Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 0-1.....	31
Gambar 4.3	Grafik Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 1-1.....	32
Gambar 4.4	Grafik Hasil Linearisasi Sensor <i>Rotary Encoder</i> pada <i>Joint Base</i>	34
Gambar 4.5	Grafik Hasil Linearisasi Sensor <i>Rotary Encoder</i> pada <i>Joint Shoulder</i>	36
Gambar 4.6	Grafik Pembacaan <i>Duty Cycle</i> pada Motor Servo sebagai <i>End Effector</i>	38
Gambar 4.7	Grafik Pembacaan <i>Duty Cycle</i> pada Motor Servo sebagai <i>Wrist</i>	39
Gambar 4.8	Grafik Pembacaan <i>Duty Cycle</i> pada Motor Servo sebagai <i>Elbow</i>	41
Gambar 4.9	Respon Perubahan Posisi Sudut <i>Joint Base</i> Tanpa Kontroler pada Sudut 45 Derajat	41
Gambar 4.10	Respon Perubahan Posisi Sudut <i>Joint Base</i> Tanpa Kontroler pada Sudut 90 Derajat	42
Gambar 4.11	Respon Perubahan Posisi Sudut <i>Joint Base</i> Tanpa Kontroler pada Sudut 180 Derajat	42
Gambar 4.12	Respon Perubahan Posisi Sudut <i>Joint Base</i> Tanpa Kontroler pada Sudut 315 Derajat	43
Gambar 4.13	Respon Perubahan Posisi Sudut <i>Joint Base</i> Tanpa Kontroler pada Sudut 360 Derajat	43
Gambar 4.14	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 2, T_i 0, T_d 0.....	44
Gambar 4.15	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0, T_d 0.....	44
Gambar 4.16	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0.....	45
Gambar 4.17	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0.....	45
Gambar 4.18	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,001	46
Gambar 4.19	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005	46
Gambar 4.20	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 90 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005	47

Gambar 4.21	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 180 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005	47
Gambar 4.22	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 315 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005	48
Gambar 4.23	Respon Perubahan Sudut <i>Joint Base</i> saat Diberi <i>Input</i> 360 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005	48
Gambar 4.24	Respon Perubahan Posisi <i>Joint Shoulder</i> Tanpa Kontroler pada Sudut 30 Derajat	50
Gambar 4.25	Respon Perubahan Posisi <i>Joint Shoulder</i> Tanpa Kontroler pada Sudut 45 Derajat	50
Gambar 4.26	Respon Perubahan Posisi <i>Joint Shoulder</i> Tanpa Kontroler pada Sudut 60 Derajat	51
Gambar 4.27	Respon Perubahan Posisi <i>Joint Shoulder</i> Tanpa Kontroler pada Sudut 80 Derajat	51
Gambar 4.28	Respon Perubahan Posisi <i>Joint Shoulder</i> Tanpa Kontroler pada Sudut 90 Derajat	52
Gambar 4.29	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 2, T_i 0, T_d 0.....	52
Gambar 4.30	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0, T_d 0.....	53
Gambar 4.31	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,01, T_d 0.....	53
Gambar 4.32	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0.....	54
Gambar 4.33	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,000009	54
Gambar 4.34	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 30 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008...55	55
Gambar 4.35	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008...55	55
Gambar 4.36	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 60 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008...56	56
Gambar 4.37	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 80 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008...56	56
Gambar 4.38	Respon Perubahan Sudut <i>Joint Shoulder</i> saat Diberi <i>Input</i> 90 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008...57	57

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

HALAMAN

Tabel 3.1	Spesifikasi Robot Manipulator	15
Tabel 3.2	Tabel Perencanaan Koneksi Pin NI myRIO	19
Tabel 4.1	Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 0-0	29
Tabel 4.2	Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 1-0	30
Tabel 4.3	Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 0-1	31
Tabel 4.4	Data <i>Driver Motor</i> dengan IN1 dan IN2 berlogika 1-1	32
Tabel 4.5	Data Hubungan Sudut dengan Pulsa pada <i>Joint Base</i>	33
Tabel 4.6	Data Setelah Linearisasi Sensor <i>Rotary Encoder</i> pada <i>Joint Base</i>	35
Tabel 4.7	Data Hubungan Sudut dengan Pulsa pada <i>Joint Shoulder</i>	36
Tabel 4.8	Data Setelah Linearisasi Sensor <i>Rotary Encoder</i> pada <i>Joint Shoulder</i> Tanpa Beban	37
Tabel 4.9	Hasil Percobaan Motor Servo HITEC HS 485 HB sebagai <i>End Effector</i>	38
Tabel 4.10	Hasil Percobaan Motor Servo HITEC HS-331 sebagai <i>Wrist</i>	39
Tabel 4.11	Hasil Percobaan Motor Servo MG-995 sebagai <i>Elbow</i>	40
Tabel 4.12	Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada <i>Joint Base</i>	49
Tabel 4.13	Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada <i>Joint Shoulder</i>	57

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Robot manipulator merupakan lengan yang memberikan gerakan robot untuk memutar, melipat, dan menjangkau objek. Gerakan ini disebut dengan derajat kebebasan robot atau jumlah sumbu yang ada pada robot. Manipulator terdiri dari beberapa *segment* dan sambungan (*joint*). Manipulator memiliki 2 bagian, yaitu bagian dasar dan bagian tambahan. Bagian dasar manipulator bisa kaku terpasang pada lantai area kerja. Bagian tambahan merupakan perluasan dari bagian dasar, biasa disebut juga lengan atau *arm*. Bagian ujungnya terpasang efektor yang berfungsi untuk fungsi kerja dari robot. Manipulator digerakan oleh aktuator atau disebut sistem *drive*. Aktuator atau sistem *drive* menyebabkan gerakan yang bervariasi dari manipulator.

Fungsi robot manipulator dalam bidang industri pada umumnya ditujukan untuk menggantikan peran manusia dalam melaksanakan tugas-tugas yang membutuhkan tenaga yang besar atau kuat, yang memerlukan ketelitian tinggi dan mengandung resiko tinggi terhadap keselamatan manusia. Atau dengan kata lain, robot manipulator dapat meminimalisasi *human error* (kesalahan manusia) pada proses pengindustrian serta dapat meningkatkan produktivitas produksi secara efisien yang memberikan keuntungan lebih. Robot manipulator juga dapat digunakan secara terus menerus tanpa merasakan adanya kebosanan dalam melakukan pemroduksian dan dapat juga digunakan pada lingkungan yang sangat berbahaya. Sebagai contoh dalam industri nuklir, robot manipulator digunakan karena radiasi nuklir yang sangat berbahaya bagi manusia apabila manusia berada dalam radius atau jarak yang cukup dekat dengan area tersebut. Robot manipulator juga digunakan pada industri perakitan, pengelasan, peleburan, pengecatan dan permesinan.

Menurut gobekasi.pojoksatu.id (27 Februari 2016) produsen kamera asal Jepang, Canon telah menerapkan sistem otomatisasi atau *manmachine* (perpaduan manusia dan mesin). Dengan menggunakan robot manipulator, proses produksi dapat diperpendek. Sebab, robot tersebut mampu bergerak konstan dan terukur dengan waktu yang lama. Sebagai perbandingan, pada 2003 Canon membutuhkan 42 pekerja untuk membuat 540 kamera. Pada 2016, dengan sistem otomatisasi, Canon hanya memerlukan 5 orang untuk memproduksi 600 kamera.

Oleh karena itu pada Tugas Akhir ini akan dibuat pengendalian robot manipulator yang dapat bergerak sesuai dengan posisi yang diinginkan. Untuk mengendalikan posisi robot manipulator, maka dirancang sebuah kontroler, yaitu kontroler PID. Kontroler PID adalah kontroler yang memiliki mekanisme umpan balik berulang tertutup. Kontrol PID digunakan untuk mengoreksi *error* dari hasil pengukuran sensor agar *output* sistem sesuai dengan nilai *set point* melalui perhitungan parameter *Proportional* (P) + *Integral* (I) + *Derivative* (D). Sensor yang digunakan adalah sensor *rotary encoder*. Sensor *rotary encoder* mendeteksi posisi sudut dari setiap *joint* pada robot manipulator. Jumlah *joint* yang akan dikontrol ada 2 *joint*, yaitu pada bagian *base* dan *shoulder*. *Joint* tersebut digerakkan oleh motor DC. Pada penelitian ini juga akan dibuat suatu *interface* yang mudah dipakai oleh pengguna. *Software* yang digunakan untuk membuat *interface* tersebut adalah LabVIEW.

1.2 Rumusan Masalah

Joint pada robot manipulator dapat digerakkan menggunakan motor. Salah satunya adalah motor DC. Permasalahannya adalah bagaimana caranya motor DC dapat berputar dan berhenti pada posisi yang diinginkan. Selain itu, diperlukan sensor yang baik untuk mendeteksi perubahan posisi motor DC. Hal ini dikarenakan motor DC tidak memiliki *feedback*.

1.3 Tujuan

Pada Tugas Akhir ini membahas pengaturan posisi *joint* robot manipulator yang menggunakan motor DC sebagai aktuatornya. Untuk mengatur perubahan posisi *joint* robot manipulator tersebut, dirancang kontroler PID untuk mengolah data *error* yang berupa perubahan dari posisi sudut yang diinginkan terhadap posisi sudut aktual. Sebagai *feedback*-nya digunakan sensor *rotary encoder* yang dapat mendeteksi perubahan sudut dari motor DC tersebut.

1.4 Batasan Masalah

Robot manipulator yang digunakan merupakan robot manipulator 4 DOF (*Degree of Freedom*). Yang terdiri dari 4 *joint* dan 5 *link*. Posisi sudut pada *joint* robot yang akan dikontrol adalah *joint* pada *base* dan *shoulder*. *Joint* tersebut digerakkan menggunakan motor DC tipe TETRIX® MAX DC Gear Motor dan sensor yang digunakan adalah

rotary encoder tipe H38S400B. HMI (*Human Machine Interface*) yang dirancang menggunakan LabVIEW 2015.

1.5 Metodologi Penelitian

Tugas Akhir ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap identifikasi dan pemodelan sistem, tahap perancangan, tahap pembuatan alat, tahap pengujian dan analisa, dan yang terakhir adalah tahap penyusunan laporan.

Pada tahap persiapan dilakukan studi literatur mengenai identifikasi fisik robot lengan, dan mempelajari karakteristik sensor *rotary encoder* yang digunakan sebagai sensor posisi.

Pada tahap identifikasi dan pemodelan sistem akan dilakukan identifikasi dari sistem alat sesuai data yang telah didapatkan dari studi literatur serta dilakukan pemodelan dari alat yang akan dikerjakan.

Pada tahap perancangan dilakukan sebuah perancangan dan pemodelan alat tersebut. Pada tahap ini digunakan sensor *rotary encoder* untuk mendeteksi adanya perubahan posisi pada lengan robot. *Rotary encoder* ini dikopel dengan *shaft* motor DC dengan menggunakan *gear*. Sehingga apabila motor DC berputar, *rotary encoder* juga ikut berputar. Keluaran *rotary encoder* yang berupa pulsa selanjutnya akan dibaca oleh NI myRIO yang kemudian diolah pada LabVIEW 2015 dan perubahan sudutnya akan ditampilkan pada *front panel* LabVIEW 2015.

Pada tahap pembuatan alat, perancangan *hardware* dilakukan dengan merancang rangkaian-rangkaian elektronika dan rancangan mekanik. Komponen-komponen elektronika yang dibuat atau digunakan meliputi konfigurasi NI myRIO, rangkaian *driver motor*, dan rancangan mekanik robot manipulator.

Pada tahap selanjutnya dilakukan pengujian alat dan analisa kesalahan atau kegagalan pada alat dan mengatasi permasalahan tersebut. Pada tahap ini menganalisa faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi *error*. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan.

Tahap akhir penelitian adalah penyusunan laporan penelitian berupa buku Tugas Akhir.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Dasar Teori

Bab ini menjelaskan tentang teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

Bab III Perancangan dan Pembuatan Sistem

Bab ini membahas perancangan *hardware* yang meliputi pengkabelan pada setiap perangkat yang digunakan dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

Bab IV Hasil Simulasi dan Implementasi

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan akan tercipta beberapa manfaat yaitu:

1. Menghasilkan robot manipulator 4 DOF (*Degree of Freedom*) yang mampu bergerak secara efektif dan efisien sesuai dengan posisi yang diinginkan.
2. Menghasilkan *software* yang dapat memonitor serta mengontrol robot manipulator yang dapat bergerak sesuai dengan posisi yang diinginkan.

BAB II

DASAR TEORI

Beberapa teori penunjang yang dipaparkan dalam buku Tugas Akhir ini adalah teori dasar mengenai pengendalian posisi *joint* robot manipulator.

2.1 Tinjauan Pustaka

Sejauh ini sudah banyak dilakukan penelitian mengenai kontrol robot manipulator. Sensor yang digunakan pada kontrol robot manipulator juga bervariasi. Mulai dari sensor jarak sampai dengan *image processing*.

Pada penelitian yang dilakukan oleh Duli Ridlo Istriantono dan Trihastuti Agustinah telah dirancang pengaturan posisi robot manipulator berbasis PD *Fuzzy Mamdani Computed Torque Control* (PD *Fuzzy CTC*) di mana pada penelitian ini digunakan *Computed Torque Controller* (CTC) untuk mengontrol robot untuk mencapai posisi yang diinginkan.

Nanang Budi Hartono, dkk. telah melakukan penelitian tentang pengaturan posisi motor servo DC dengan metode P, PI, dan PID menggunakan potensiometer sebagai sensor posisi. Namun dalam penerapannya diperlukan penggunaan potensiometer yang benar-benar linear dan presisi.

Pada Tugas Akhir ini akan dirancang pengendalian posisi *joint* robot manipulator menggunakan sensor *rotary encoder*. Di mana posisi sudut dari motor DC tersebut dibaca oleh sensor *rotary encoder*. Untuk memastikan posisi robot manipulator bergerak sesuai dengan posisi yang diinginkan, maka diberikan kontroler PID (*Proportional + Integral + Derivative*) sebagai kontrol posisi motor DC pada *joint* robot manipulator.

2.2 Robot[7]

Robot adalah seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Istilah robot berawal bahasa Ceko “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Saat ini hampir tidak ada orang yang tidak mengenal robot, namun pengertian robot tidaklah sama bagi setiap orang. Pada kamus Webster pengertian

robot adalah sebuah alat otomatis yang melakukan fungsi berdasarkan kebutuhan manusia. Dari kamus Oxford diperoleh pengertian robot adalah Sebuah mesin yang mampu melakukan serangkaian tugas rumit secara otomatis, terutama yang diprogram oleh komputer. Beberapa organisasi di bidang robot membuat definisi tersendiri. Misalnya Robot Institute of America memberikan definisi robot sebagai sebuah manipulator multifungsi yang mampu diprogram, didesain untuk memindahkan material, komponen, alat, atau benda khusus lainnya melalui serangkaian gerakan terprogram untuk melakukan berbagai tugas. Sedangkan International Organization for Standardization (ISO 8373) mendefinisikan robot sebagai sebuah manipulator yang terkendali, multifungsi, dan mampu diprogram untuk bergerak dalam tiga aksis atau lebih, yang tetap berada di tempat atau bergerak untuk digunakan dalam aplikasi otomasi industri. Gambar 2.1 merupakan salah satu contoh *humanoid robot*.



Gambar 2.1 Robot Nao[12]

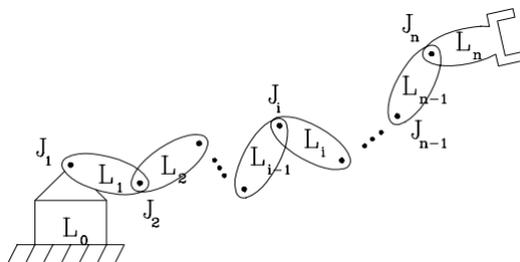
Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya yaitu untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput. Robot mulai menjadi populer ketika seorang penulis berbangsa Czech (Ceko), Karl Capek, membuat pertunjukan pertunjukan dari lakon komedi yang ditulisnya pada tahun 1921 yang berjudul RUR (Rossum's Universal Robot). Menurut Fu, et al. (1987) penelitian dan

pengembangan pertama yang berbuah produk robotik dapat dilacak mulai dari tahun 1940-an ketika Argonne National Laboratories di Oak Ridge, Amerika, memperkenalkan sebuah mekanisme robotik yang dinamai *master-slave* manipulator.

2.3 Robot Manipulator

Robot manipulator adalah bagian mekanik yang dapat difungsikan untuk memindah, mengangkat, dan memanipulasi benda kerja. Robot manipulator merupakan jenis robot lengan dengan material lengan pembawa beban. Robot manipulator dapat membawa beban dengan lokasi yang berpindah-pindah, berbahan material tipis, lebih ringan, lebih hemat dalam konsumsi daya, hanya memerlukan aktuator yang kecil, lebih mudah dioperasikan, serta lebih murah dalam proses *manufacturing* (Subudhi and Morris, 2002).[3]

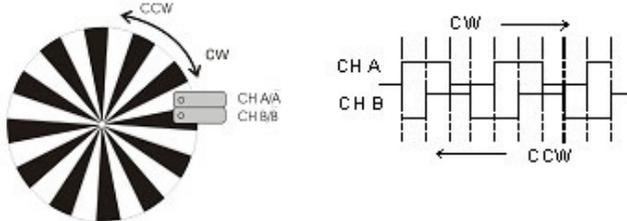
Pada robot manipulator *base*-nya tetap dan tak dapat berpindah. Robot lengan memiliki *link* dan *joint*. *Link* pada robot lengan merupakan penghubung antar *joint*. Sedangkan *joint* merupakan penghubung antar *link*. Jumlah dari *joint* suatu manipulator biasanya menentukan jumlah *degrees of freedom*. *Degrees of freedom* adalah jumlah arah independen *end-effector* dari sebuah robot yang dapat bergerak. *Joint* robot lengan mirip dengan sendi dalam tubuh manusia. Namun, setiap *joint* lengan robot hanya memiliki 1 derajat kebebasan sehingga hampir semua kasus, jumlah *joint* yang dimiliki robot lengan sama dengan jumlah DOF-nya. Setiap *joint* dihubungkan dengan dua *link*, yaitu *link* masukan dan keluaran. Gambar 2.2 adalah gambaran dari *degree of freedom*. [8]



Gambar 2.2 Degree of Freedom [1]

2.4 Rotary Encoder

Untuk pengukuran posisi putaran yang lebih presisi dapat menggunakan *rotary encoder*. Secara umum prinsip kerja *rotary encoder* ini dapat diilustrasikan seperti Gambar 2.3.[7]



Gambar 2.3 Prinsip Kerja *Rotary Encoder*[7]

Dua buah sensor optis (*channel A/A* dan *channel B/B*) pendeteksi “hitam dan putih” digunakan sebagai acuan untuk menentukan arah gerakan. Searah jarum jam (*clock-wise, CW*) atau berlawanan arah jarum jam (*counter clock-wise, CCW*). Sedangkan jumlah pulsa (baik A atau B) dapat dihitung (menggunakan prinsip *counter*) sebagai banyak langkah yang ditempuh. Dengan demikian arah gerakan dan posisi dapat dideteksi dengan baik oleh *rotary encoder*. [7]

Biasanya *encoder* dipasang segaris dengan poros (*shaft*) motor, *gearbox*, sendi atau bagian berputar lainnya. Beberapa tipe *encoder* memiliki poros berlubang (*hollow shaft encoder*) yang didesain untuk sistem sambungan langsung ke poros obyek dideteksi. [7]

Rotary encoder umumnya digunakan pada pengendalian robot, *motor drive*, dan sebagainya. Terdapat dua jenis *rotary encoder*, yaitu *absolut rotary encoder* dan *incremental rotary encoder*. Gambar 2.4 merupakan *incremental rotary encoder*.



Gambar 2.4 *Incremental Rotary Encoder*[12]

2.5 LabVIEW[6]

LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*) adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri. LabVIEW menggunakan gambar atau grafis sebagai fungsi (*Graphical Programming Language* atau *Visual Programming Language*). Jika pada pemrograman lainnya menggunakan teks untuk membuat sebuah fungsi, bahasa pemrograman ini menginterpretasikan sebuah grafis sebagai suatu fungsi. Program LabVIEW disebut dengan Virtual Instrumen (VI) karena beberapa tampilan dan operasi pada program LabVIEW menyerupai suatu instrumen seperti osiloskop dan multimeter. Setiap VI menggunakan fungsi-fungsi yang memanipulasi *input* dari *user interface* atau sumber lain dan menampilkan informasi tersebut atau memindahkan informasi tersebut ke *file* atau komputer lain.

Terdapat tiga desain antarmuka yang digunakan. Yang pertama adalah *Front Panel* adalah desain antarmuka yang dapat digunakan untuk memberi masukan atau sebagai keluaran. Sebagai contoh *numeric control* adalah sebuah *control* yang dapat memberi masukan berupa angka atau *image display* yang dapat menampilkan gambar. Ada beberapa grafis fungsi yang tidak dapat dipakai pada *front panel*.

Kedua, *Block Diagram* yaitu desain antarmuka yang berisi grafis fungsi-fungsi perhitungan yang digunakan dalam LabVIEW *programming*. Grafis fungsi yang ada kita hubungkan untuk membuat persamaan atau untuk menghasilkan nilai - nilai yang kita inginkan.

Ketiga, *Controls and Functions Palettes*. Pada *functions and Controls palettes* terdapat semua grafis fungsi yang dimiliki LabVIEW yang dapat kita gunakan pada *front panel* maupun *block diagram*.

2.6 NI myRIO[6]

NI myRIO adalah sebuah perangkat keras di mana pengguna dapat memanipulasi fungsi-fungsinya untuk membuat berbagai sistem. myRIO menggunakan ARM prosesor yang juga sebuah FPGA prosesor. Dengan menggunakan fitur-fitur yang ada dapat dibuat sistem yang lebih kompleks. myRIO menggunakan LabVIEW sebagai IDE. LabVIEW digunakan untuk membuat aplikasi yang akan menjalankan perangkat keras atau fitur-fitur yang ada pada myRIO. Pada *board* myRIO terdapat beberapa peripheral yang umum yang dapat digunakan seperti *bluetooth*, *wifi*, LED, *accelerometer*, *push button*, analog *input* dan *output*, RAM.

Dapat juga dihubungkan ke sebuah *integrated circuit* atau IC untuk membuat sistem yang lebih kompleks. Atau dihubungkan ke *board* lain yang memang diperlukan dalam membuat sebuah sistem. Gambar 2.5 merupakan bentuk fisik dari NI myRIO.



Gambar 2.5 Bentuk Fisik NI myRIO[11]

Sebuah *embedded processor* adalah mikroprosesor yang digunakan dalam sistem *embedded*. Prosesor ini biasanya memiliki ukuran lebih kecil dan mengkonsumsi daya yang lebih kecil. *Embedded processor* dapat dibagi menjadi dua kategori yaitu mikroprosesor biasa dan mikrokontroler. Mikrokontroler memiliki peripheral yang lebih banyak pada *chip*-nya. Pada dasarnya, *embedded processor* adalah *chip* CPU yang digunakan dalam sistem yang bukan merupakan *general purpose workstation* seperti laptop atau desktop.

Embedded processor hanyalah sebuah perangkat komputasi yang ditempatkan di dalam sistem yang dikontrolnya. Sebuah *embedded processor* yang ada pada sistem menangani semua perhitungan dan operasi logis dari komputer. *Embedded processor* juga menangani tugas-tugas seperti menyimpan dan mengambil data dari memori, dan pengolahan data dari *input* atau *output*. *Embedded processor* sering bekerja sebagai bagian dari sistem komputer, bersama memori dan perangkat *input* atau *output*.

Embedded Processor dapat ditemukan dalam perangkat portabel seperti jam tangan digital, PDA, kamera digital, unit GPS dan pemutar MP3. Mereka juga dapat ditemukan dalam sistem yang lebih besar seperti lampu lalu lintas, sistem pengendalian pembangkit listrik dan pengendali pabrik. Sistem telekomunikasi juga menggunakan beberapa *embedded system* dari *switch* telepon ke ponsel. *Embedded Processor*

juga digunakan dalam elektronik konsumen termasuk konsol video *game*, pemutar DVD dan printer. Beberapa peralatan rumah tangga termasuk *oven microwave*, mesin cuci dan mesin pencuci piring menggunakan *embedded processor* juga.

2.7 Motor DC

Motor DC (*direct current*) adalah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang desain awalnya diperkenalkan oleh Michael Faraday lebih dari seabad yang lalu. Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Ada berbagai jenis motor DC, diantaranya motor DC magnet permanen, motor DC stepper, motor DC *brushless*, motor DC servo, dan masih banyak lagi.[7]

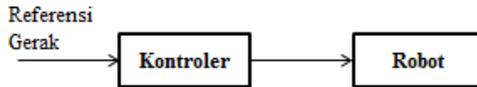
Agar motor DC dapat berputar dan berhenti di poisisi sudut yang diinginkan, maka diperlukan pengaturan posisi motor DC. Pada Tugas Akhir ini digunakan *rotary encoder* sebagai sensor posisi. *Rotary encoder* dipasang sejajar dan dikopel menggunakan *gear* dengan motor DC. Sehingga ketika motor DC berputar *rotary encoder* juga ikut berputar. Keluaran dari *rotary encoder* ini kemudian akan dibaca oleh NI myRIO yang selanjutnya diproses pada LabVIEW dan ditampilkan pada *front panel* LabVIEW. Gambar 2.6 merupakan bentuk fisik dari motor DC.



Gambar 2.6 Motor DC[13]

2.8 Kontroler PID

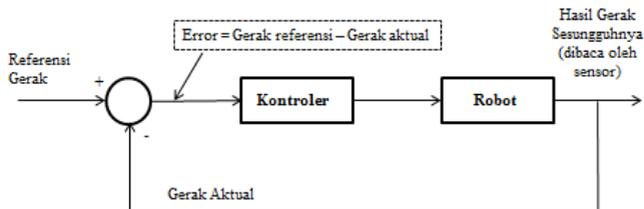
Sistem kontrol robotik pada dasarnya terbagi dua kelompok, yaitu sistem kontrol *loop* terbuka (*open loop*) dan *loop* tertutup (*close loop*). Diagram kontrol *loop* terbuka pada sistem robot dapat dinyatakan dalam Gambar 2.7.[7]



Gambar 2.7 Kontrol Robot *Loop* Terbuka[7]

Kontrol *loop* terbuka atau umpan maju (*feedforward control*) dapat dinyatakan sebagai sistem kontrol yang *outputnya* tidak diperhitungkan ulang oleh kontroler.[7]

Kontrol robot *loop* tertutup dapat dinyatakan seperti dalam Gambar 2.8.[7]



Gambar 2.8 Kontrol Robot *Loop* Tertutup[7]

Pada Gambar 2.8, jika hasil gerak aktual telah sama dengan referensi maka *input* kontroler akan nol. Artinya kontroler tidak lagi memberikan sinyal aktuasi kepada robot karena target akhir perintah gerak telah diperoleh. Makin kecil *error* terhitung maka makin kecil pula sinyal pengemudian kontroler terhadap robot, sampai akhirnya mencapai kondisi tenang (*steady state*).[7]

Referensi gerak dan gerak aktual dapat berupa posisi (biasanya didefinisikan melalui kedudukan ujung lengan terakhir (*end of effector*), kecepatan, akselerasi atau gabungan di antaranya).[7]

Salah satu jenis kontroler berdasarkan aksi kontrolnya adalah kontroler PID (*Proportional + Integral + Derivative*). Untuk kontroler PID, sinyal kesalahan $e(t)$ merupakan masukan kontroler sedangkan

keluaran kontroler adalah sinyal kontrol $u(t)$. Hubungan antara masukan kontroler $e(t)$ dan keluaran kontroler $u(t)$ adalah[2]

$$u(t) = K_p \left\{ e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{d_e(t)}{dt} \right\}$$

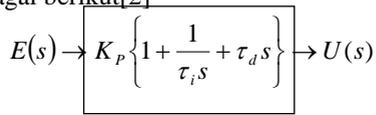
atau dalam besaran transformasi Laplace

$$U(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s)$$

dimana K_p adalah penguatan proporsional, τ_i adalah waktu integral dan τ_d adalah waktu derivatif. Parameter K_p , τ_i , dan τ_d ketiganya dapat ditentukan. Sehingga fungsi alih kontroler PID adalah[2]

$$\frac{U(s)}{E(s)} = K_p \left\{ 1 + \frac{1}{\tau_i s} + \tau_d s \right\}$$

Diagram blok kontroler proporsional ditambah differensial adalah sebagai berikut[2]



-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMASANGAN ALAT

Pada bab ini akan dibahas tahapan dalam perancangan pemasangan alat dalam pengendalian posisi sudut *joint* pada robot manipulator 4 DOF (*Degree of Freedom*).

3.1 Perancangan Robot Manipulator

Robot manipulator yang digunakan merupakan robot manipulator yang telah digunakan pada Tugas Akhir tahun sebelumnya. Pada Tugas Akhir sebelumnya digunakan potensiometer sebagai sensor untuk pengaturan posisi *joint* pada *base* dan *shoulder*. Pada Tugas Akhir kali ini, sensor yang digunakan untuk pengaturan posisi adalah sensor *rotary encoder*. Bentuk fisik dari robot manipulator yang digunakan dapat dilihat pada lampiran D.

3.1.1 Spesifikasi Robot Manipulator

Robot manipulator yang digunakan memiliki empat derajat kebebasan (4 DOF) yaitu *base*, *shoulder*, *elbow*, dan *wrist*. Kemudian pada *end effector* ditambahkan *tool* berupa *gripper*. Spesifikasi dari robot manipulator yang digunakan terlihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Robot Manipulator[8]

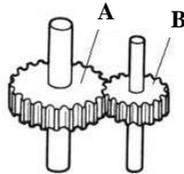
Model		Robot Manipulator
Koordinat Robot		4 derajat kebebasan (4 DOF) + <i>Gripper</i>
Batasan Gerak	<i>Base</i>	0° s/d 360°
	<i>Shoulder</i>	0° s/d 90°
	<i>Elbow</i>	0° s/d 90°
	<i>Wrist</i>	0° s/d 90°
	<i>Gripper</i>	0° s/d 90°
<i>Gripper</i>	<i>Open</i>	90°
	<i>Closed</i>	0°
Material		Alumunium

3.1.2 Pemasangan Sensor *Rotary Encoder*

Bagian *base* dan *shoulder* pada robot manipulator dipasang sebuah sensor *rotary encoder* sebagai sensor posisi, yang keluarannya berupa pulsa. Sensor *rotary encoder* yang digunakan menghasilkan 1600 pulsa setiap putarannya. Sensor ini dikopel dengan motor DC menggunakan

gear. Sehingga ketika motor berputar, *rotary encoder* juga berputar. Pemasangan sensor *rotary encoder* dapat dilihat pada lampiran.

Pada bagian *base*, perbandingan *gear* motor DC dengan *gear rotary encoder* adalah 2:1. Di mana *gear* pada motor DC memiliki 80 gigi dan *gear* pada *rotary encoder* memiliki 40 gigi. Ilustrasinya terlihat pada Gambar 3.1. pada Gambar 3.1, A merupakan *gear* pada motor DC dan B merupakan *gear* pada *rotary encoder*. Gambar *real*-nya dapat dilihat pada lampiran D.



Gambar 3.1 Ilustrasi Perbandingan *Gear* Pada Motor DC dan *Rotary Encoder* Bagian *Base*[17]

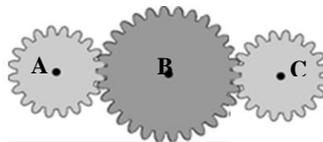
Dengan pemasangan seperti Gambar 3.1, maka satu kali putaran motor DC menghasilkan $2 \times 1600 = 3200$ pulsa. Sehingga 1 pulsa mewakili $0,1125^\circ$. Perhitungannya seperti berikut.

$$\frac{3200 \text{ pulsa}}{1 \text{ pulsa}} = \frac{360^\circ}{x}$$

$$x = \frac{360^\circ}{3200 \text{ pulsa}}$$

$$x = 0,1125^\circ$$

Pada bagian *shoulder*, perbandingan *gear rotary encoder* dengan *gear* motor DC adalah 1:1. Ilustrasinya terlihat pada Gambar 3.2. Di mana A merupakan *gear* pada motor DC, B adalah *gear* penggerak *joint* dan C merupakan *gear* pada *rotary encoder*. *Gear* pada motor DC dan *rotary encoder* memiliki 40 gigi. Sedangkan *gear* penggerak *joint* memiliki 80 gigi. Gambar *real*-nya dapat dilihat pada lampiran D.



Gambar 3.2 Ilustrasi Perbandingan *Gear* Pada Motor DC dan *Rotary Encoder* Bagian *Shoulder*[17]

Dengan pemasanga seperti Gambar 3.2, maka perhitungan sudut pada *joint shoulder* seperti berikut.

$$\begin{aligned} \frac{N_1}{N_2} &= \frac{\theta_2}{\theta_1} \\ \theta_2 &= \frac{N_1 \times \theta_1}{N_2} \\ \theta_2 &= \frac{40 \times \theta_1}{80} \\ \theta_2 &= \frac{1}{2} \theta_1 \end{aligned}$$

Di mana :

N_1 = Jumlah gigi *gear* pada motor DC

N_2 = Jumlah gigi *gear* pada *rotary encoder* bagian *shoulder*

θ_1 = Sudut yang ditempuh motor DC

θ_2 = Sudut yang ditempuh *joint shoulder*

Dari perhitungan di atas, sudut yang ditempuh oleh *joint shoulder* merupakan setengah dari sudut yang ditempuh oleh motor DC.

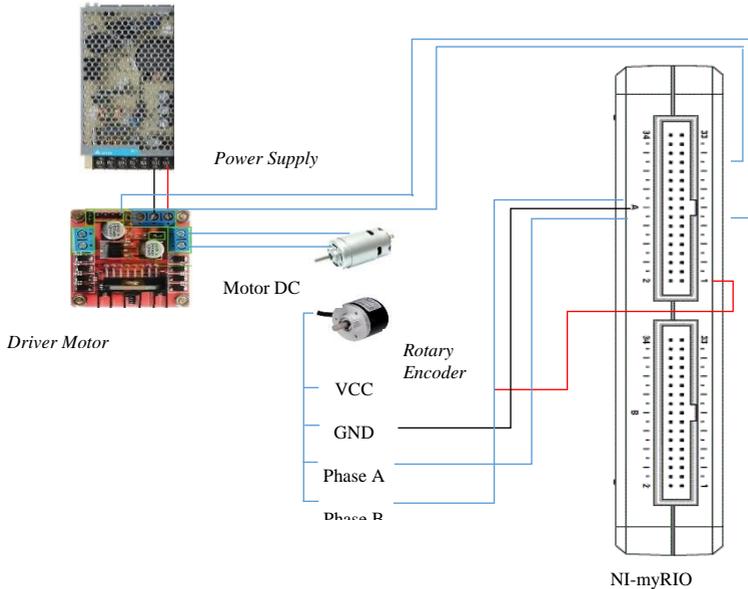
3.1.3 Pengkabelan NI myRIO dengan Motor DC dan Sensor Posisi

Pada *hardware*, terdapat beberapa rangkaian dan komponen elektronik yang harus disusun untuk dapat menggerakkan motor DC dengan baik. Motor DC ini berfungsi sebagai aktuator untuk menggerakkan sendi-sendi robot lengan, yaitu *base* dan *shoulder*. Dalam menggerakkan aktuator motor DC diperlukan sebuah *driver* yang akan mengatur arah putaran.

Pada Tugas Akhir ini digunakan *driver* motor L298N. Dimana IC L298N merupakan *driver* tipe dual *H-bridge* yang dapat menghasilkan *drive* dua arah dengan arus kontinyu sampai dengan 2 A pada tegangan 4,8 volt sampai 46 volt. Tiap *H-bridge* dilengkapi dengan sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. *Driver* Motor L298 mampu memutar dua buah motor.

Pengaturan posisi putar motor DC ini menggunakan *rotary encoder* sebagai sensor posisi. *Rotary encoder* ini terpasang dengan motor DC dan terhubung dengan dua *gear*. Saat motor berputar, *rotary encoder* juga berputar dan akan memberikan keluaran berupa pulsa. Keluaran *rotary encoder* yang berupa pulsa selanjutnya akan dibaca oleh NI myRIO yang kemudian diolah pada LabVIEW 2015 dan perubahan sudutnya akan ditampilkan pada *front panel* LabVIEW 2015. Gambar

3.3 merupakan skematik konfigurasi dari NI myRIO, motor DC dan *rotary encoder*.



Gambar 3.3 Pengkabelan NI myRIO dengan Motor DC dan *Rotary Encoder*

3.1.4 Pengkabelan NI myRIO

NI myRIO merupakan bagian terpenting dari sistem robot manipulator ini. NI myRIO merupakan pusat kontrol yang mengatur sensor posisi (*rotary encoder*) pada motor DC sehingga robot manipulator mampu bergerak sesuai dengan yang diinginkan. Gambar 3.4 merupakan rangkaian dalam *hardware* NI myRIO.



Gambar 3.4 *Hardware* NI myRIO[10]

Perancangan koneksi NI myRIO dengan *driver motor* dan sensor *rotary encoder* terlihat pada Tabel 3.2.

Tabel 3.2 Tabel Perencanaan Koneksi Pin NI myRIO

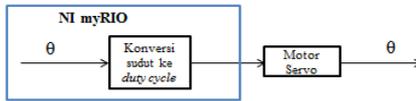
KONEKSI PIN NI myRIO				
NO PIN	MXP A		MXP B	
1	VCC Rotary Encoder (Base)		VCC Rotary Encoder (shoulder)	
8	Driver Motor	GND	Driver Motor	GND
15		Pin IN1		Pin IN3
17		Pin IN2		Pin IN4
18	Rotary Encoder (Base)	Channel A	Rotary Encoder (shoulder)	Channel A
20		GND		GND
22		Channel B		Channel B
27	Pin ENA (Driver Motor)		Pin ENB (Driver Motor)	

3.2 Cara Kerja Sistem

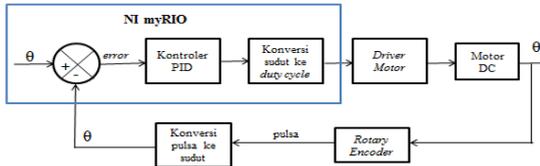
Cara kerja sistem pada penelitian ini yaitu, *user* dapat memberikan perintah untuk menggerakkan *joint* pada robot manipulator melalui HMI yang dirancang menggunakan LabVIEW 2015. Perintah yang diberikan berupa sudut yang harus ditempuh oleh *joint* robot manipulator. Aktuator pada *joint* robot ini berupa tiga motor servo dan dua motor DC. Motor servo digunakan sebagai aktuator pada *joint elbow*, *wrist*, dan *end effector*. Sedangkan motor DC digunakan sebagai aktuator pada *joint base* dan *shoulder*. Motor DC ini dikopel dengan sensor posisi *rotary encoder*. Ketika motor DC berputar, pergerakannya akan dideteksi oleh sensor *rotary encoder*. Data yang dibaca oleh sensor ini kemudian diolah oleh NI myRIO. Hasil pembacaan sensor akan ditampilkan pada HMI.

3.3 Blok Diagram Sistem

Terdapat dua sistem kontrol yang dirancang pada penelitian ini, yaitu *open loop* dan *close loop*. Sistem *open loop* seperti Gambar 3.5 dirancang untuk menggerakkan motor servo yang digunakan sebagai aktuator *joint* pada *end effector*, *wrist*, dan *elbow*. Di mana *user* dapat memberikan *input* berupa posisi sudut yang diinginkan. Kemudian sudut dikonversi ke *duty cycle* sehingga motor servo dapat bergerak.



Gambar 3.5 Blok Diagram Sistem pada Motor Servo

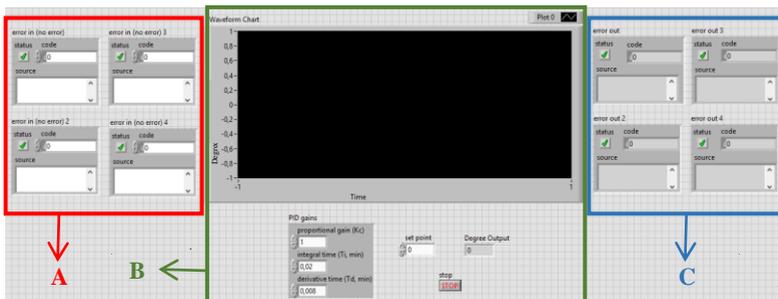


Gambar 3.6 Blok Diagram Sistem pada Motor DC

Sistem *close loop*, seperti pada Gambar 3.6 dirancang untuk menggerakkan motor DC yang digunakan sebagai aktuator *joint* pada *base* dan *shoulder*. Di mana *user* bisa memberikan *input* berupa posisi sudut yang diinginkan, kemudian diproses oleh kontrol motor sehingga motor bergerak ke posisi yang diinginkan. Pada poros motor dipasang sebuah sensor posisi yaitu *rotary encoder* untuk memberikan *feedback* posisi aktual dari motor. Keluaran dari sensor *rotary encoder* ini berupa pulsa, sehingga perlu dikonversikan dahulu ke sudut sebelum masuk pada proses kontrol. Hasil dari kontroler PID akan dikonversi ke dalam *duty cycle* untuk menghentikan pergerakan motor DC apabila telah mencapai posisi yang diinginkan.

3.4 Perancangan *Software*

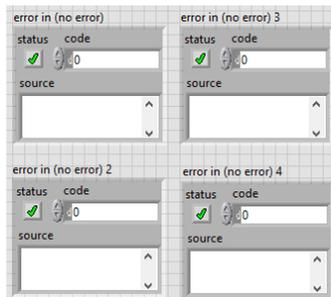
Untuk mempermudah pengendalian robot manipulator maka dibuatlah sebuah *Human Machine Interface* (HMI).



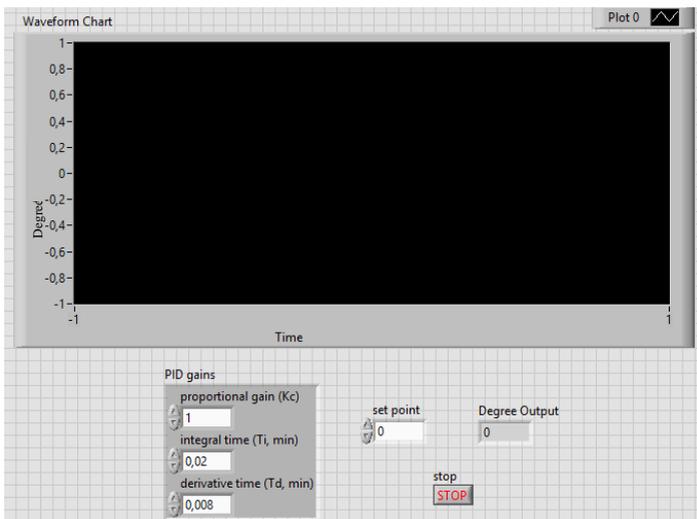
Gambar 3.7 Rancangan HMI pada LabVIEW 2015

Gambar 3.7 merupakan rancangan HMI yang digunakan untuk pengujian motor DC. *Software* yang digunakan adalah LabVIEW 2015 dengan bahasa pemrograman berbasis grafis atau blok diagram.

Dalam rancangan HMI pada Gambar 3.7 terdapat beberapa bagian, yaitu A, B, dan C. Bagian A apabila diperbesar maka akan tampak seperti pada Gambar 3.8. *Error in* pada Gambar 3.8 dapat menerima informasi mengenai *wiring* dari VI yang dipanggil sebelumnya. Informasi ini digunakan untuk memutuskan apakah ada fungsi yang harus dilewati jika terjadi kesalahan dari VI lainnya.

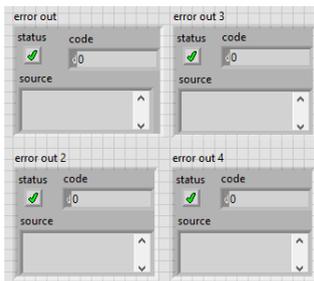


Gambar 3.8 Bagian A pada HMI di LabVIEW 2015



Gambar 3.9 Bagian B pada HMI di LabVIEW 2015

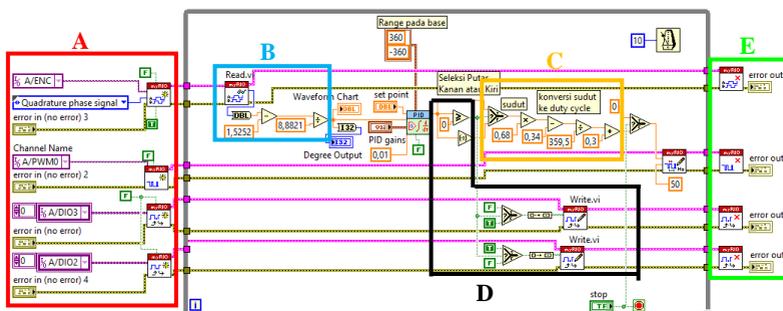
Bagian B HMI terlihat pada Gambar 3.9, didalamnya terdapat *waveform chart* yang dapat menunjukkan grafik respon perubahan posisi pada *joint* robot. Parameter untuk menentukan nilai K_p, τ_i, τ_d dapat diisikan pada bagian *PID Gains*. Untuk memasukkan nilai sudut yang diinginkan dapat diisikan pada kolom *set point*. Hasil dari pergerakan *joint* robot dapat dilihat pada kolom *Degree output*. Tombol stop berfungsi untuk mengakhiri program.



Gambar 3.10 Bagian C pada HMI di LabVIEW 2015

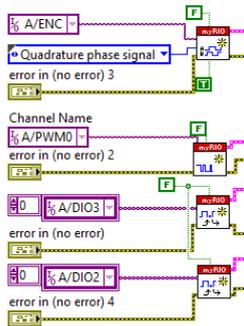
Bagian C HMI terlihat pada Gambar 3.10. Bagian tersebut mengandung informasi ketika terjadi *error*.

HMI pada Gambar 3.7 memiliki blok diagram yang tampak seperti Gambar 3.11.



Gambar 3.11 Blok Diagram Perancangan HMI pada LabVIEW 2015

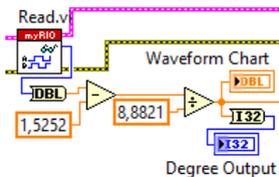
Blok diagram pada Gambar 3.11 memiliki beberapa bagian yaitu A, B, C, D, dan E.



Gambar 3.12 Bagian A Blok Diagram Perancangan HMI

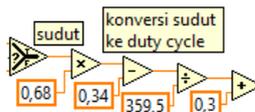
Bagian A apabila diperbesar akan tampak seperti Gambar 3.12. Bagian A merupakan inisialisasi dari *device* dan *port* yang digunakan.

Gambar 3.13 merupakan bagian B dari blok diagram perencanaan HMI pada LabVIEW 2015. Pada bagian ini pembacaan pulsa yang dideteksi oleh sensor *rotary encoder* dikonversi ke dalam sudut. Rumus untuk konversi ini didapatkan dari hasil linearisasi sensor *rotary encoder* yang telah dilakukan.

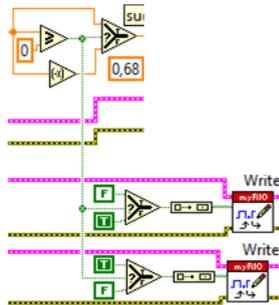


Gambar 3.13 Bagian B Blok Diagram Perencanaan HMI

Bagian C dari blok diagram perencanaan HMI pada LabVIEW 2015 terlihat pada Gambar 3.14. Bagian tersebut merupakan konversi dari sinyal kontrol yang berupa sudut ke *duty cycle*. Konversi ini bertujuan untuk menghentikan pergerakan *joint* robot apabila telah mencapai posisi yang diinginkan.



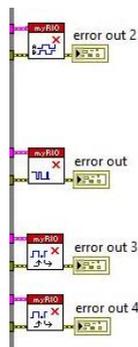
Gambar 3.14 Bagian C Blok Diagram Perencanaan HMI



Gambar 3.15 Bagian D Blok Diagram Perencanaan HMI

Gambar 3.15 merupakan bagian D dari blok diagram perencanaan HMI pada LabVIEW 2015. Bagian ini menyeleksi motor harus berputar ke kanan atau ke kiri dengan kondisi yang telah ditentukan.

Bagian E dari blok diagram perencanaan HMI terlihat pada Gambar 3.16.



Gambar 3.16 Bagian E Blok Diagram Perencanaan HMI

Gambar 3.16 dapat memberikan informasi apabila terjadi *error* pada saat program dijalankan.

3.5 Linearisasi Sensor *Rotary Encoder*

Tujuan dari linearisasi adalah untuk mengkalibrasi sensor *rotary encoder* yang digunakan pada *joint* robot manipulator. Berikut merupakan langkah-langkah yang dilakukan dalam linearisasi sensor *rotary encoder* pada bagian *base* dan *shoulder* robot manipulator.

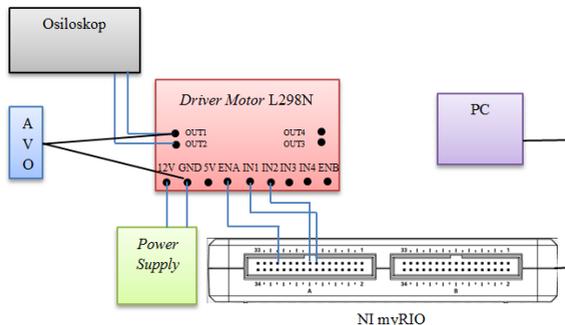
1. Memutar *joint* dengan sudut yang berbeda-beda.
2. Dari perputaran *joint* akan diketahui berapa banyak pulsa yang terbaca oleh sensor *rotary encoder*.
3. Mencatat sudut masukan dan pulsa yang terbaca oleh sensor *rotary encoder*.
4. Membuat grafik dari data yang didapatkan menggunakan Microsoft Excel.
5. Mencari persamaan dari grafik yang dibuat pada langkah nomor 4.

3.6 Perancangan Pengujian Driver Motor L298N

Langkah-langkah yang dilakukan dalam pengujian *driver motor* L298N adalah sebagai berikut.

1. *Wiring* seperti pada Gambar 3.17.

Pada Gambar 3.17, *driver motor* diberi tegangan 12V dari *power supply*. Pin ENA, IN1, dan IN2 pada *driver motor* dihubungkan ke pin DIO₈ (pin 27), DIO₂ (pin 15), dan DIO₃ (pin 17) pada NI myRIO. Dan pin OUT1 dan OUT2 dihubungkan ke osiloskop.



Gambar 3.17 Rangkaian Pengujian *Driver Motor* L298N

2. Memasukkan nilai *duty cycle* dan logika untuk pin IN1 dan IN2 pada *front panel* LabVIEW seperti pada Gambar 3.18.



Gambar 3.18 *Front Panel* LabVIEW untuk Pengujian *Driver Motor*

3. Mengukur *duty cycle* pada pin ENA, OUT1, dan OUT2 yang ada pada *driver motor* L298N menggunakan avometer.
4. Mencatat hasil yang didapatkan.
5. Menempatkan *probe* osiloskop pada pin OUT yang ada di *driver motor*.
6. Melihat gelombang yang dihasilkan pada osiloskop.
7. Menyimpan gambar gelombang pada osiloskop dengan cara difoto.

3.7 Perancangan Pengujian Posisi Motor Servo

Berikut ini adalah langkah-langkah dalam pengukuran posisi motor servo.

1. Menyalakan PC
2. Menghubungkan NI myRIO dengan sumber, kemudian mengkoneksikan NI myRIO dengan PC (melalui Wi-Fi ataupun USB)
3. *Wiring* motor servo dan myRIO seperti pada Gambar 3.19.



Gambar 3.19 *Wiring* Motor Servo dan NI myRIO

4. Menghubungkan port NI myRIO dengan kaki yang terdapat pada motor servo
5. Gambar 3.20 merupakan tampilan halaman HMI (*Human Machine Interface*) pada LabVIEW.

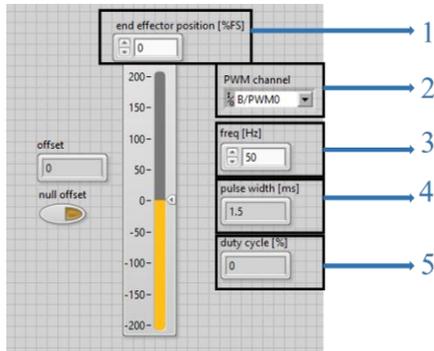
Angka 1 : Memasukan nilai derajat atau posisi yang ingin dicapai oleh motor servo

Angka 2 : Inisialisasi pin PWM antara motor servo dan NI myRIO

Angka 3 : Nilai frekuensi dari motor servo

Angka 4 : Menampilkan besarnya nilai PWM

Angka 5 : Menampilkan besarnya nilai *duty cycle*



Gambar 3.20 Front Panel LabVIEW untuk Pengujian Motor Servo

3.8 Perancangan Kontroler PID

Pada penelitian ini, dirancang sebuah kontroler untuk memastikan pergerakan *joint* pada robot manipulator sesuai dengan posisi yang diinginkan. Kontroler yang digunakan yaitu kontroler PID. Metode yang digunakan untuk mencari nilai parameter K_p , τ_i , τ_d yaitu dengan *trial and error*. Berikut merupakan langkah-langkah dalam mencari nilai parameter K_p , τ_i , dan τ_d pada bagian *base* dan *shoulder* robot manipulator.

1. Menentukan parameter *gain* proporsional (P) terlebih dahulu, di mana *gain* proporsional (P) menguatkan keluaran agar responnya mendekati masukan.
2. Menentukan parameter integral (I), di mana fungsi dari integral (I) adalah memaksa sistem untuk tetap berada pada *steady state* dan juga mengurangi *error* atau kesalahan.
3. Menentukan parameter derivative (D). Derivative (D) berfungsi untuk mengurangi osilasi.

-----Halaman ini sengaja dikosongkan-----

BAB IV

HASIL SIMULASI DAN IMPLEMENTASI

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah sesuai dengan yang diharapkan atau tidak, maka dilakukan pengujian dan analisa berdasarkan perencanaan dari sistem yang telah dibuat.

4.1 Pengujian *Driver Motor L298N*

Pengujian ini bertujuan untuk mengetahui apakah karakteristik dari *driver motor L298N* yang digunakan sesuai dengan *datasheet* atau tidak. Dari pengambilan data yang dilakukan diperoleh beberapa data yang diperoleh dengan kondisi IN1 dan IN2 yang berbeda-beda. Data tersebut terlihat pada Tabel 4.1 sampai Tabel 4.4.

Tabel 4.1 merupakan hasil pengambilan data *driver motor L298N* ketika IN1 diberi logika 0 dan IN2 diberi logika 0. Pada tabel tersebut terlihat bahwa nilai yang terbaca pada pin ENA hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar 0,05126%. Nilai yang terbaca pada pin OUT1 dan OUT2 0, dikarenakan logika yang diberikan pada pin IN1 dan IN2 juga 0. Pada kondisi ini motor tidak berputar.

Tabel 4.1 Data *Driver Motor* dengan IN1 dan IN2 berlogika 0-0

<i>Duty Cycle</i>	Logika		Pengukuran dengan AVO (%)		
	IN1	IN2	ENA	OUT1	OUT2
0,1	0	0	10	0	0
0,2	0	0	20	0	0
0,3	0	0	29,9	0	0
0,4	0	0	39,9	0	0
0,5	0	0	49,9	0	0
0,6	0	0	59,9	0	0
0,7	0	0	69,9	0	0
0,8	0	0	79,9	0	0
0,9	0	0	89,9	0	0
1 (0,98)	0	0	98,8	0	0

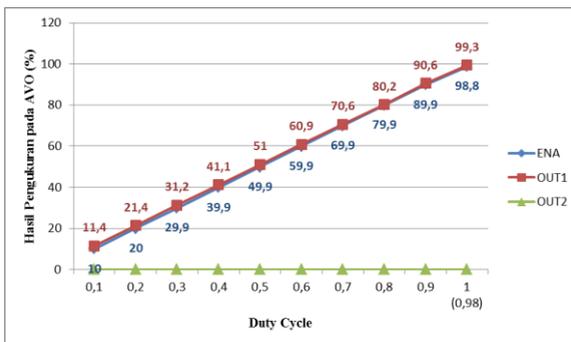
Tabel 4.2 merupakan hasil pengambilan data *driver motor L298N* ketika IN1 diberi logika 1 dan IN2 diberi logika 0. Pada tabel tersebut

terlihat bahwa nilai yang terbaca pada pin ENA hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar 0,05126%. Nilai yang terbaca pada pin OUT1 juga hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar 3,43503%. Nilai yang terbaca pada pin OUT2 0, dikarenakan logika yang diberikan pada pin IN2 juga 0. Pada kondisi ini motor berputar ke kiri.

Data pada Tabel 4.2 apabila dibuat grafik maka terlihat seperti Gambar 4.3. Pada gambar tersebut terlihat bahwa grafik pin ENA dengan OUT1 merupakan garis lurus dan kedua grafik tersebut hampir berhimpit yang menandakan bahwa nilai dari kedua pin tersebut hampir sama. Sedangkan grafik pin OUT2 berada pada nilai 0

Tabel 4.2 Data *Driver Motor* dengan IN1 dan IN2 berlogika 1-0

<i>Duty Cycle</i>	Logika		Pengukuran dengan AVO (%)		
	IN1	IN2	ENA	OUT1	OUT2
0,1	1	0	10	11,4	0
0,2	1	0	20	21,4	0
0,3	1	0	29,9	31,2	0
0,4	1	0	39,9	41,1	0
0,5	1	0	49,9	51	0
0,6	1	0	59,9	60,9	0
0,7	1	0	69,9	70,6	0
0,8	1	0	79,9	80,2	0
0,9	1	0	89,9	90,6	0
1 (0,98)	1	0	98,8	99,3	0

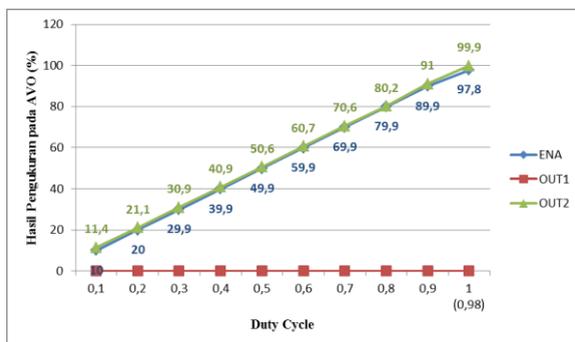


Gambar 4.1 Grafik Data *Driver Motor* dengan IN1 dan IN2 berlogika 1-0

Tabel 4.3 merupakan hasil pengambilan data *driver motor* L298N ketika IN1 diberi logika 0 dan IN2 diberi logika 1. Pada tabel tersebut terlihat bahwa nilai yang terbaca pada pin ENA hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar -0,15330%. Nilai yang terbaca pada pin OUT1 0, dikarenakan logika yang diberikan pada pin IN1 juga 0. Nilai yang terbaca pada pin OUT2 juga hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar 3,12736%. Pada kondisi ini motor berputar ke kanan.

Tabel 4.3 Data *Driver Motor* dengan IN1 dan IN2 berlogika 0-1

<i>Duty Cycle</i>	Logika		Pengukuran dengan AVO (%)		
	IN1	IN2	ENA	OUT1	OUT2
0,1	0	1	10	0	11,4
0,2	0	1	20	0	21,1
0,3	0	1	29,9	0	30,9
0,4	0	1	39,9	0	40,9
0,5	0	1	49,9	0	50,6
0,6	0	1	59,9	0	60,7
0,7	0	1	69,9	0	70,6
0,8	0	1	79,9	0	80,2
0,9	0	1	89,9	0	91
1 (0,98)	0	1	97,8	0	99,9



Gambar 4.2 Grafik Data *Driver Motor* dengan IN1 dan IN2 berlogika 0-1

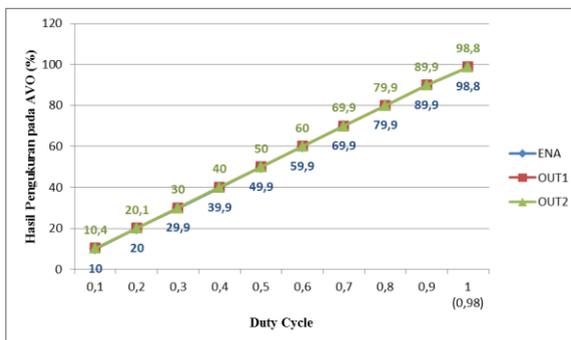
Data pada Tabel 4.3 apabila dibuat grafik maka terlihat seperti Gambar 4.4. Pada gambar tersebut terlihat bahwa grafik pin ENA

dengan OUT2 merupakan garis lurus dan kedua grafik tersebut hampir berhimpit yang menandakan bahwa nilai dari kedua pin tersebut hampir sama. Sedangkan grafik pin OUT1 berada pada nilai 0

Tabel 4.4 merupakan hasil pengambilan data *driver motor* L298N ketika IN1 diberi logika 1 dan IN2 diberi logika 1. Pada tabel tersebut terlihat bahwa nilai yang terbaca pada pin ENA, OUT1, dan OUT2 hampir mendekati sama seperti nilai *duty cycle* yang dimasukkan dengan *error* rata-rata sebesar 0,05126% pada pin ENA, 0,48353% pada pin OUT1, dan 0,49373% pada pin OUT2. Pada kondisi ini motor tidak berputar.

Tabel 4.4 Data *Driver Motor* dengan IN1 dan IN2 berlogika 1-1

Duty Cycle	Logika		Pengukuran dengan AVO (%)		
	IN1	IN2	ENA	OUT1	OUT2
0,1	1	1	10	10,4	10,4
0,2	1	1	20	20,1	20,1
0,3	1	1	29,9	30	30
0,4	1	1	39,9	40	40
0,5	1	1	49,9	50	50
0,6	1	1	59,9	60	60
0,7	1	1	69,9	69,9	69,9
0,8	1	1	79,9	79,9	79,9
0,9	1	1	89,9	89,9	89,9
1 (0,98)	1	1	98,8	98,7	98,8



Gambar 4.3 Grafik Data *Driver Motor* dengan IN1 dan IN2 berlogika 1-1

Data pada Tabel 4.4 apabila dibuat grafik maka terlihat seperti Gambar 4.5. Pada gambar tersebut terlihat bahwa grafik pin ENA , OUT1 dan OUT2 merupakan garis lurus dan ketiga grafik tersebut hampir berhimpit yang menandakan bahwa nilai dari ketiga pin tersebut hampir sama. Gambar gelombang pada osiloskop dapat dilihat pada lampiran.

4.2 Pengujian Sensor *Rotary Encoder*

Setelah dilakukan linearisasi sensor *rotary encoder* seperti yang dijelaskan pada bab 3, kemudian dilakukan pengujian terhadap sensor tersebut.

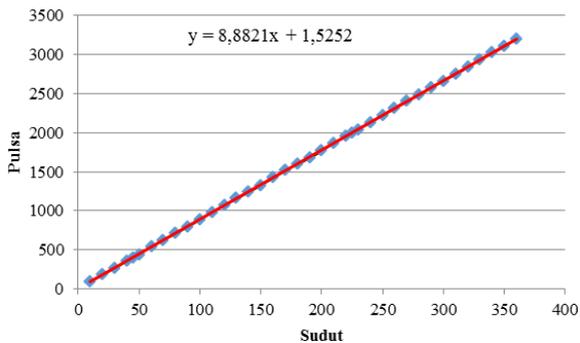
4.2.1 Pengujian Sensor *Rotary Encoder* pada *Joint Base*

Data yang diperoleh dari linearisasi pada *joint base* terlihat pada Tabel 4.5. Pada tabel tersebut terdapat kolom x dan y. Di mana x merupakan sudut yang dimasukkan dan y adalah jumlah pulsa yang terbaca.

Tabel 4.5 Data Hubungan Sudut dengan Pulsa pada *Joint Base*

No	x (°)	y (pulsa)
1	10	94
2	20	183
3	30	271
4	40	358
5	45	395
6	50	444
7	60	537
8	70	627
9	80	710
10	90	795
11	100	886
12	110	983
13	120	1072
14	130	1159
15	140	1244
16	150	1328
17	160	1427
18	170	1517

19	180	1595
20	190	1679
21	200	1773
22	210	1866
23	220	1957
24	225	1998
25	230	2039
26	240	2128
27	250	2227
28	260	2316
29	270	2408
30	280	2485
31	290	2583
32	300	2662
33	310	2750
34	320	2839
35	330	2938
36	340	3027
37	350	3108
38	360	3203



Gambar 4.4 Grafik Hasil Linearisasi Sensor *Rotary Encoder* pada *Joint Base*

Data pada Tabel 4.5 apabila dibuat grafik maka akan terlihat seperti Gambar 4.4. Grafik tersebut dibuat menggunakan Microsoft Excel dan didapatkan sebuah persamaan $y = 8,8821x + 1,5252$ yang selanjutnya diimplementasikan pada program untuk konversi dari pulsa

ke sudut. Pada Gambar 4.4 terlihat bahwa grafik yang terbentuk merupakan garis lurus. Hal ini menandakan sensor *rotary encoder* tersebut sudah linear.

Hasil pengambilan data setelah dilakukan linearisasi sensor *rotary encoder* pada *joint base* tampak pada Tabel 4.6. Pada tabel tersebut terdapat kolom *degree input*, *degree output* dan *% error*. Di mana *degree input* merupakan sudut yang diinginkan dan *degree output* adalah sudut hasil pembacaan sensor *rotary encoder*. *% error* merupakan selisih antara *degree output* dan *degree input*. Dalam tabel tersebut terlihat bahwa selisih antara sudut yang diinginkan dengan sudut yang terbaca oleh sensor *rotary encoder* tidak terlalu banyak. Dengan *error* rata-rata sebesar 0,074423%.

Tabel 4.6 Data Setelah Linearisasi Sensor *Rotary Encoder* pada *Joint Base*

No	<i>Degree Input</i>	<i>Degree Output</i>	<i>% Error</i>
1	10	7	-0,3
2	20	18	-0,1
3	30	28	-0,066666667
4	40	38	-0,05
5	50	53	0,06
6	60	77	0,283333333
7	70	75	0,071428571
8	80	77	-0,0375
9	90	118	0,311111111
10	100	99	-0,01
11	120	147	0,225
12	140	166	0,185714286
13	160	188	0,175
14	180	215	0,194444444
15	200	233	0,165
16	240	243	0,0125
17	260	291	0,119230769
18	300	331	0,103333333
19	320	350	0,09375
20	360	379	0,052777778
Rata-Rata			0,074423

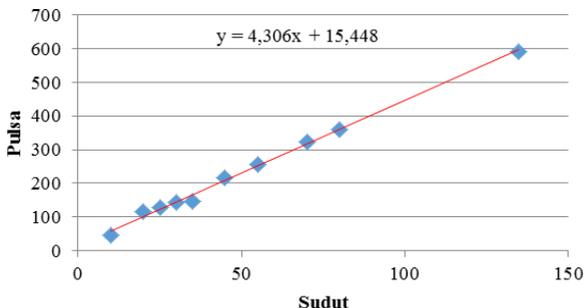
4.2.2 Pengujian Sensor *Rotary Encoder* pada *Joint Shoulder*

Data yang diperoleh dari linearisasi pada *joint shoulder* terlihat pada Tabel 4.7. Pada tabel tersebut terdapat kolom x dan y. Di mana x merupakan sudut yang dimasukkan dan y adalah jumlah pulsa yang terbaca.

Data pada Tabel 4.7 apabila dibuat grafik maka akan terlihat seperti Gambar 4.5. Grafik tersebut dibuat menggunakan Microsoft Excel dan didapatkan sebuah persamaan $y = 4,306x + 15,448$ yang selanjutnya diimplementasikan pada program untuk konversi dari pulsa ke sudut. Pada Gambar 4.6 terlihat bahwa grafik yang terbentuk merupakan garis lurus. Hal ini menandakan sensor *rotary encoder* tersebut sudah linear.

Tabel 4.7 Data Hubungan Sudut dengan Pulsa pada *Joint Shoulder*

No	x (°)	y (pulsa)
1	10	46
2	20	115
3	25	129
4	30	145
5	35	146
6	45	217
7	55	256
8	70	323
9	80	359
10	135	593



Gambar 4.5 Grafik Hasil Linearisasi Sensor *Rotary Encoder* pada *Joint Shoulder*

Tabel 4.8 Data Setelah Linearisasi Sensor *Rotary Encoder* pada *Joint Shoulder* Tanpa Beban

No.	<i>Degree Input</i>	<i>Degree Output</i>	% <i>Error</i>
1	10	11	0,1
2	20	18	-0,1
3	30	28	-0,066666667
4	40	39	-0,025
5	50	55	0,1
6	60	60	0
7	70	72	0,028571429
8	80	79	-0,0125
9	90	92	0,022222222
Rata-Rata			0,005180776

Hasil pengambilan data setelah dilakukan linearisasi sensor *rotary encoder* pada *joint shoulder* tanpa beban tampak pada Tabel 4.8. Pada tabel tersebut terdapat kolom *degree input*, *degree output* dan % *error*. Di mana *degree input* merupakan sudut yang diinginkan dan *degree output* adalah sudut hasil pembacaan sensor *rotary encoder*. % *error* merupakan selisih antara *degree output* dan *degree input*. Dalam tabel tersebut terlihat bahwa selisih antara sudut yang diinginkan dengan sudut yang terbaca oleh sensor *rotary encoder* tidak terlalu banyak. Dengan *error* rata-rata sebesar 0,005180776%.

4.3 Pengujian Sudut pada Motor Servo

Data yang diperoleh dari pengujian ini terlihat pada Tabel 4.9. Dari Tabel 4.9 hasil pengukuran motor servo menghasilkan posisi yang sama dengan *input* yang diberikan. Hal ini sesuai dengan rumus perhitungan rumus *duty cycle* di bawah ini.

$$\text{duty cycle} = \frac{t_{on}}{t_{total}} \cdot 100\%$$

$$\text{duty cycle} = \frac{PWM}{periode} \cdot 100\%$$

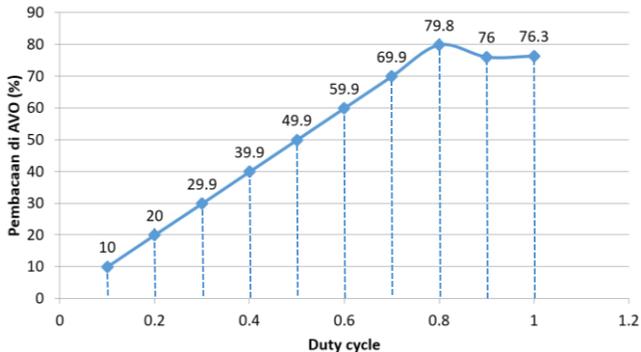
$$\text{duty cycle} = \frac{PWM}{20} \cdot 100\%$$

$$\text{duty cycle} = 5PWM$$

Motor servo tipe HITEC HS 485 HB merupakan jenis *standard* servo yang artinya mampu bergerak dalam sudut -90° sampai dengan 90° . Sehingga nilai *neutral* pada motor servo adalah 0° .

Tabel 4.9 Hasil Percobaan Motor Servo HITEC HS 485 HB sebagai *End Effector*

Degree (input)	Degree (output)	Duty Cycle (%)	PWM (Pulse Width Modulation)
-90	-90	3	0,6
-80	-80	3,5	0,7
-70	-70	4	0,8
-60	-60	4,5	0,9
-50	-50	5	1
-40	-40	5,5	1,1
-30	-30	6	1,2
-20	-20	6,5	1,3
-10	-10	7	1,4
0	0	7,5	1,5
10	10	8	1,6
20	20	8,5	1,7
30	30	9	1,8
40	40	9,5	1,9
50	50	10	2
60	60	10,5	2,1
70	70	11	2,2
80	80	11,5	2,3
90	90	12	2,4



Gambar 4.6 Grafik Pembacaan *Duty Cycle* pada Motor Servo sebagai *End Effector*

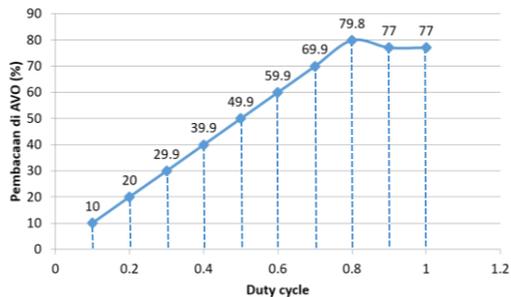
Gambar 4.6 merupakan respon dari pin PWM pada motor servo tipe HITEC HS-485 HB. Dari hasil grafik diketahui jika *duty cycle* maksimal

yaitu 0,8 karena ketika nilai *duty cycle* 0,8 hingga 1, nilai yang terbaca di AVOMeter mengalami penurunan.

Tabel 4.10 merupakan data hasil percobaan pada motor servo tipe HITEC HS-331 yang digunakan sebagai aktuator pada *joint wrist* robot manipulator. Dari Tabel 4.10 tampak *degree output* dan *degree input* adalah sama, sehingga tidak ada *error*.

Tabel 4.10 Hasil Percobaan Motor Servo HITEC HS-331 sebagai *Wrist*

<i>Degree (input)</i>	<i>Degree (output)</i>	<i>Duty Cycle (%)</i>	<i>PWM (Pulse Width Modulation)</i>
-90	-90	3	0,6
-80	-80	3,5	0,7
-70	-70	4	0,8
-60	-60	4,5	0,9
-50	-50	5	1
-40	-40	5,5	1,1
-30	-30	6	1,2
-20	-20	6,5	1,3
-10	-10	7	1,4
0	0	7,5	1,5
10	10	8	1,6
20	20	8,5	1,7
30	30	9	1,8
40	40	9,5	1,9
50	50	10	2
60	60	10,5	2,1
70	70	11	2,2
80	80	11,5	2,3
90	90	22	2,4



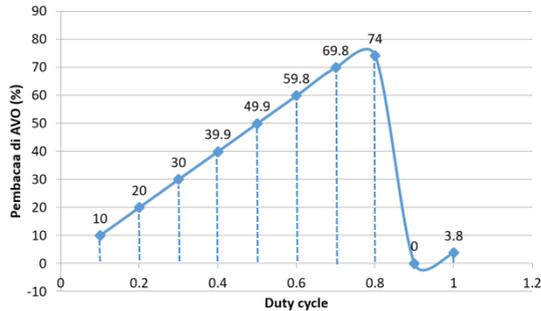
Gambar 4.7 Grafik Pembacaan *Duty Cycle* pada Motor Servo sebagai *Wrist*

Gambar 4.7 merupakan respon dari pin PWM pada motor servo tipe HITEC HS-331 yang digunakan pada *joint wrist*. Dari hasil grafik diketahui jika *duty cycle* maksimal yaitu 0,8 karena ketika nilai *duty cycle* 0,9 hingga 1, nilai yang terbaca di AVOMeter mengalami penurunan.

Tabel 4.11 merupakan data hasil percobaan pada motor servo tipe MG-995 yang digunakan pada *joint elbow*. Dari tabel tersebut terlihat bahwa motor servo tipe MG-995 memiliki rata-rata *error* sebesar 11,8421%. hal ini dikarenakan beban yang dimiliki motor pada *elbow* lebih berat dibandingkan motor-motor sebelumnya.

Tabel 4.11 Hasil Percobaan Motor Servo MG-995 sebagai *Elbow*

<i>Degree (input)</i>	<i>Degree (output)</i>	<i>Error (%)</i>	<i>Duty Cycle (%)</i>	<i>PWM (Pulse Width Modulation)</i>
-90	-90	0	3	0,6
-80	-80	0	3,5	0,7
-70	-70	0	4	0,8
-60	-65	8,33	4,5	0,9
-50	-50	0	5	1
-40	-40	0	5,5	1,1
-30	-25	-16,7	6	1,2
-20	-10	-50	6,5	1,3
-10	-5	-50	7	1,4
0	0	0	7,5	1,5
10	5	-50	8	1,6
20	10	-50	8,5	1,7
30	25	-16,7	9	1,8
40	40	0	9,5	1,9
50	50	0	10	2
60	60	0	10,5	2,1
70	70	0	11	2,2
80	80	0	11,5	2,3
90	90	0	12	2,4



Gambar 4.8 Grafik Pembacaan *Duty Cycle* pada Motor Servo sebagai *Elbow*

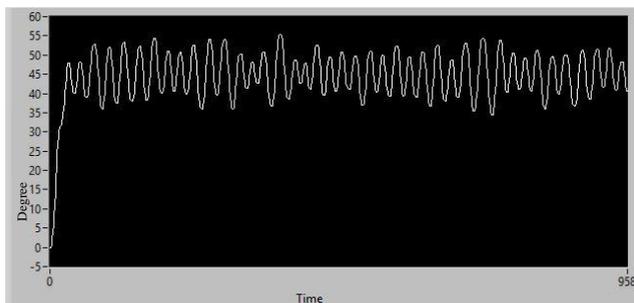
Gambar 4.8 merupakan respon dari pin PWM pada motor servo tipe MG-995 yang digunakan pada *joint elbow*. Dari hasil grafik diketahui jika *duty cycle* maksimal yaitu 0,8 karena ketika nilai *duty cycle* 0,9 hingga 1, nilai yang terbaca di AVOMeter mengalami penurunan bahkan sampai bernilai 0.

4.4 Pengujian *Joint* pada *Base*

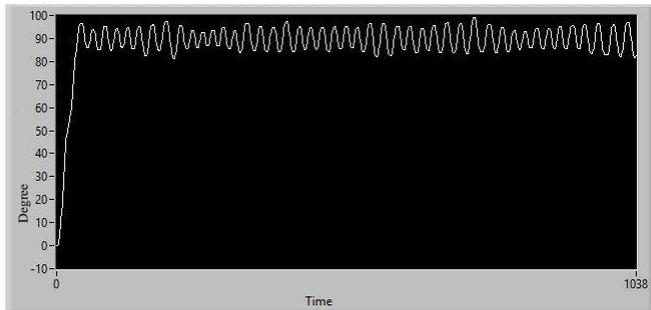
Pengujian ini bertujuan untuk mengetahui respon perubahan posisi pada *joint base* ketika diberi kontroler dan tidak diberi kontroler.

4.4.1 Respon Perubahan Posisi pada *Joint Base* Tanpa Kontroler

Gambar 4.9 merupakan respon perubahan posisi sudut *joint base* pada robot manipulator tanpa kontroler dengan *input* sudut 45 derajat. Pada gambar tersebut terlihat bahwa respon mengalami osilasi sebesar 10 derajat.



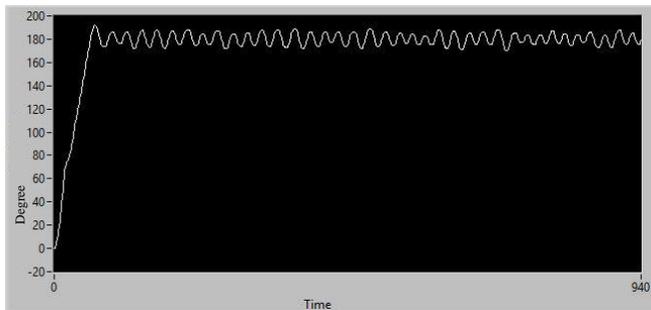
Gambar 4.9 Respon Perubahan Posisi Sudut *Joint Base* Tanpa Kontroler pada Sudut 45 Derajat



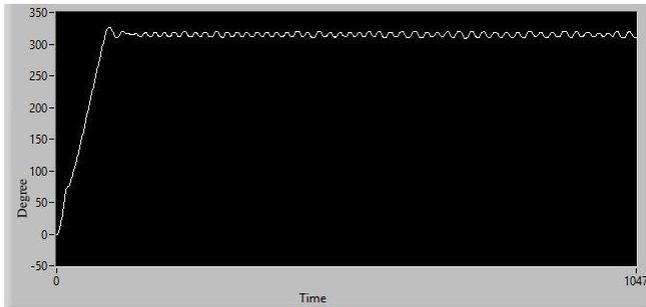
Gambar 4.10 Respon Perubahan Posisi Sudut *Joint Base* Tanpa Kontroler pada Sudut 90 Derajat

Gambar 4.10 merupakan respon perubahan posisi sudut *joint base* pada robot manipulator tanpa kontroler dengan *input* sudut 90 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan osilasi sebesar 10 derajat.

Gambar 4.11 merupakan respon perubahan posisi sudut *joint base* pada robot manipulator tanpa kontroler dengan *input* sudut 180 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan osilasi sebesar 10 derajat.



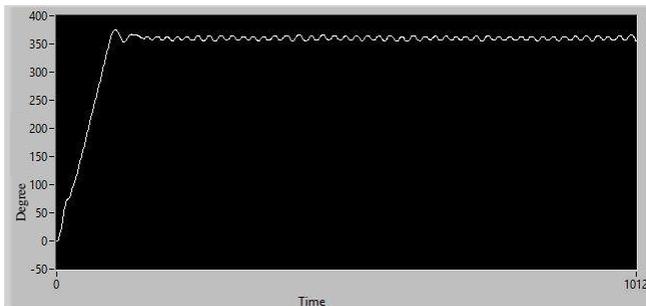
Gambar 4.11 Respon Perubahan Posisi Sudut *Joint Base* Tanpa Kontroler pada Sudut 180 Derajat



Gambar 4.12 Respon Perubahan Posisi Sudut *Joint Base* Tanpa Kontroler pada Sudut 315 Derajat

Gambar 4.12 merupakan respon perubahan posisi sudut *joint base* pada robot manipulator tanpa kontroler dengan *input* sudut 315 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan osilasi sebesar 10 derajat.

Gambar 4.13 merupakan respon perubahan posisi sudut *joint base* pada robot manipulator tanpa kontroler dengan *input* sudut 360 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan osilasi sebesar 10 derajat.



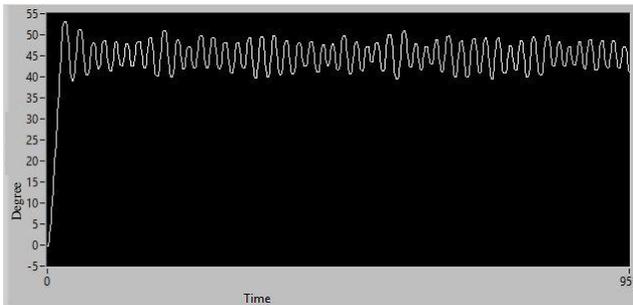
Gambar 4.13 Respon Perubahan Posisi Sudut *Joint Base* Tanpa Kontroler pada Sudut 360 Derajat

4.4.2 Respon Perubahan Posisi pada *Joint Base* Dengan Kontroler

Terdapat beberapa variasi parameter yang digunakan pada pengujian ini.

1. Parameter $K_p = 2, \tau_i = 0, \tau_d = 0$

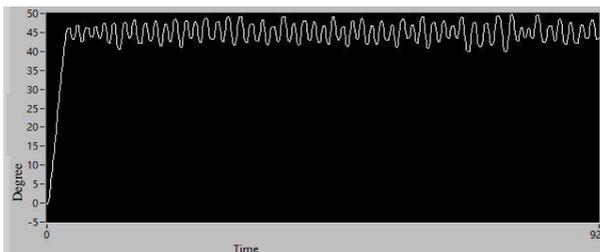
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.14. Gambar 4.14 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mencapai lebih dari 50 derajat dan mengalami osilasi sebesar 6 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.14 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 2, T_i 0, T_d 0

2. Parameter $K_p = 1, \tau_i = 0, \tau_d = 0$

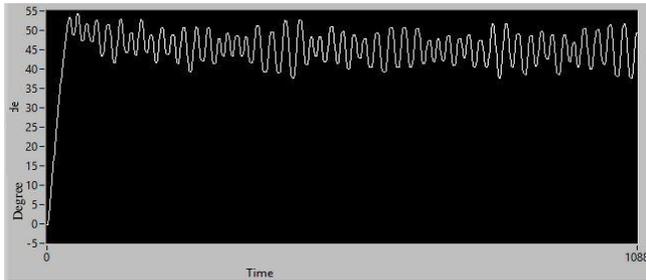
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.15. Gambar 4.15 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan mengalami osilasi sebesar 5 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.15 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

3. Parameter $K_p = 1$, $\tau_i = 0,02$, $\tau_d = 0$

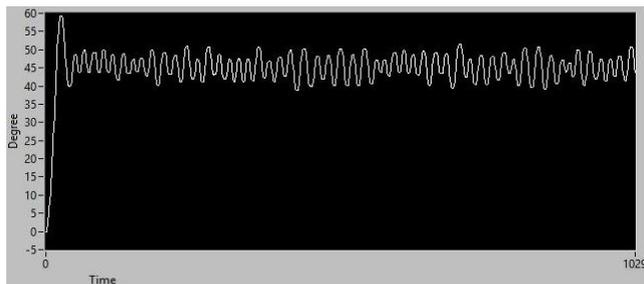
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.16. Gambar 4.16 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan mengalami osilasi sebesar 7 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.16 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0

4. Parameter $K_p = 1$, $\tau_i = 0,09$, $\tau_d = 0$

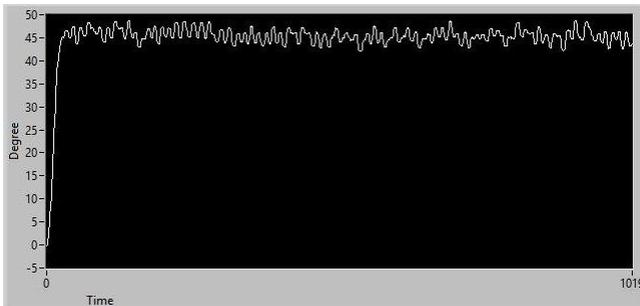
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.17. Gambar 4.17 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mencapai posisi 60 derajat dan mengalami osilasi sebesar 5 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.17 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0

5. Parameter $K_p = 1$, $\tau_i = 0,09$, $\tau_d = 0,001$

Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.18. Gambar 4.18 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan mengalami osilasi sebesar 4 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



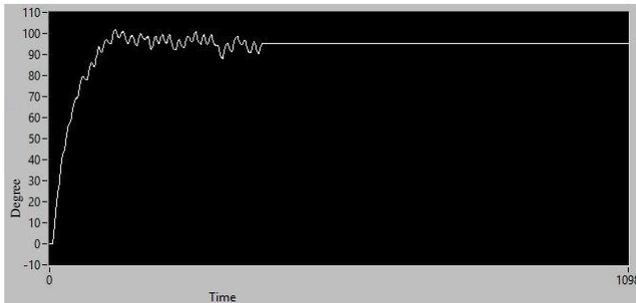
Gambar 4.18 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,001

6. Parameter $K_p = 1$, $\tau_i = 0,09$, $\tau_d = 0,005$

Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.19. Gambar 4.19 menunjukkan respon perubahan sudut *joint base* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan tidak mengalami osilasi.



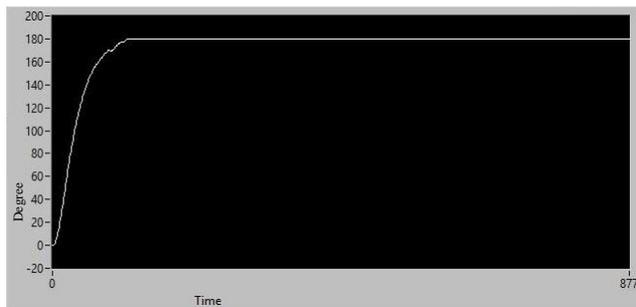
Gambar 4.19 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005



Gambar 4.20 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005

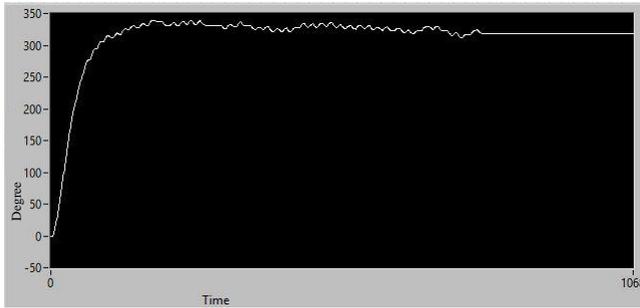
Gambar 4.20 merupakan respon perubahan posisi sudut *joint base* ketika diberi *input* sudut 90 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mencapai posisi 100 derajat dan sempat berosilasi namun pada akhirnya dapat mencapai keadaan *steady* pada posisi 90 derajat.

Gambar 4.21 merupakan respon perubahan posisi sudut *joint base* ketika diberi *input* sudut 180 derajat. Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan tidak berosilasi. Respon mampu mencapai *set point* dan *steady state* pada posisi 180 derajat.

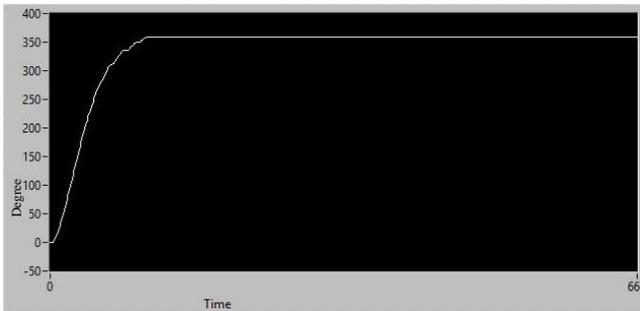


Gambar 4.21 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005

Gambar 4.22 merupakan respon perubahan posisi sudut *joint base* ketika diberi *input* sudut 315 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* dan sempat berosilasi namun pada akhirnya dapat mencapai keadaan *steady*.



Gambar 4.22 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005



Gambar 4.23 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,005

Gambar 4.23 merupakan respon perubahan posisi sudut *joint base* ketika diberi *input* sudut 360 derajat. Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan tidak berosilasi. Respon mampu mencapai *set point* dan *steady state* pada posisi 180 derajat.

4.4.3 Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada *Joint Base*

Dari pengujian *joint* pada *base* dengan menggunakan kontroler PID dan tanpa kontroler, didapatkan hasil perbandingan seperti pada Tabel 4.12. Dari tabel tersebut dapat dilihat bahwa respon tanpa kontroler selalu mengalami osilasi dan *overshoot*. Sehingga tidak pernah mencapai keadaan *steady*. Sedangkan respon ketika diberi kontroler mampu mencapai keadaan *steady*.

Tabel 4.12 Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada *Joint Base*

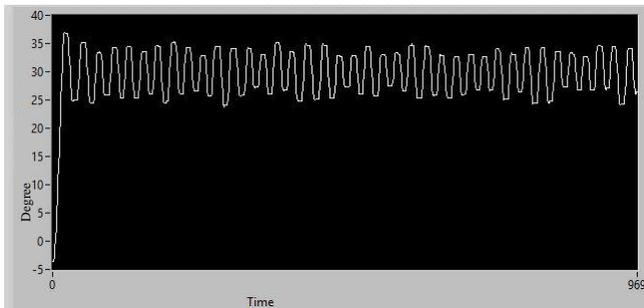
Input (°)	Tanpa Kontroler (°)	Dengan Kontroler [K_P 1, T_i 0,09, T_d 0,005] (°)	Keterangan	
			Tanpa Kontroler	Dengan Kontroler
45	45	45	<ul style="list-style-type: none"> Mengalami osilasi 	<ul style="list-style-type: none"> Mencapai <i>set point</i> pada posisi 45° Tidak mengalami osilasi dan <i>overshoot</i>.
90	90	95	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Melebihi <i>set point</i> pada posisi 95° Tidak mengalami osilasi dan <i>overshoot</i>
180	180	180	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Mampu mencapai <i>set point</i>, <i>Steady</i> pada posisi 180°, Tidak mengalami osilasi dan <i>overshoot</i>.
315	315	319	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Melebihi <i>set point</i>, <i>Steady</i> pada posisi 319° Tidak mengalami osilasi dan <i>overshoot</i>
360	360	359	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Tidak mencapai <i>set point</i>, <i>Steady</i> pada posisi 359° Tidak mengalami osilasi dan <i>overshoot</i>.

4.5 Pengujian *Joint* pada *Shoulder*

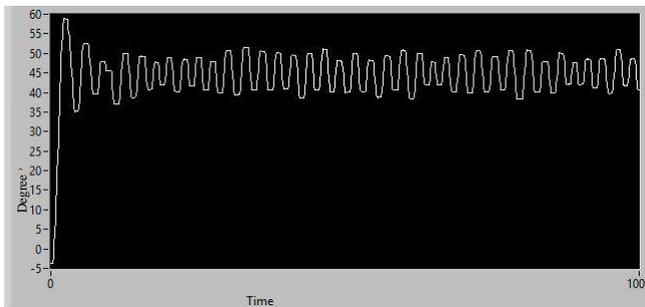
Pengujian ini bertujuan untuk mengetahui respon perubahan posisi pada *joint shoulder* ketika diberi kontroler dan tidak diberi kontroler.

4.5.1 Respon Perubahan Posisi pada *Joint Shoulder* Tanpa Kontroler

Gambar 4.24 merupakan respon perubahan posisi sudut *joint shoulder* pada robot manipulator tanpa kontroler dengan *input* sudut 30 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga melebihi posisi 35 derajat. Dan mengalami osilasi sebesar 5 derajat.



Gambar 4.24 Respon Perubahan Posisi *Joint Shoulder* Tanpa Kontroler pada Sudut 30 Derajat



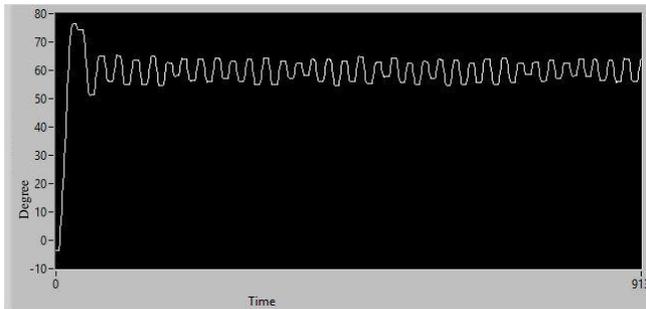
Gambar 4.25 Respon Perubahan Posisi *Joint Shoulder* Tanpa Kontroler pada Sudut 45 Derajat

Gambar 4.25 merupakan respon perubahan posisi sudut *joint shoulder* pada robot manipulator tanpa kontroler dengan *input* sudut 45

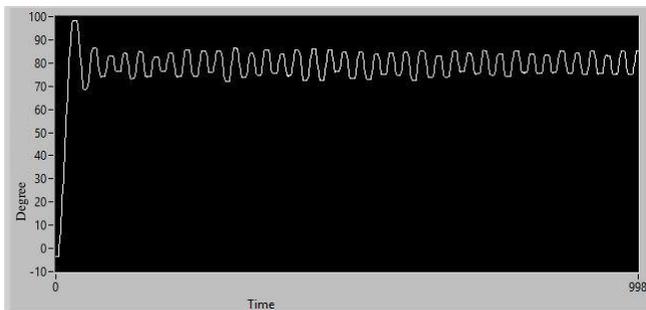
derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mendekati posisi 60 derajat. Dan mengalami osilasi sebesar 8 derajat.

Gambar 4.26 merupakan respon perubahan posisi sudut *joint shoulder* pada robot manipulator tanpa kontroler dengan *input* sudut 60 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mendekati posisi 80 derajat. Dan mengalami osilasi sebesar 5 derajat.

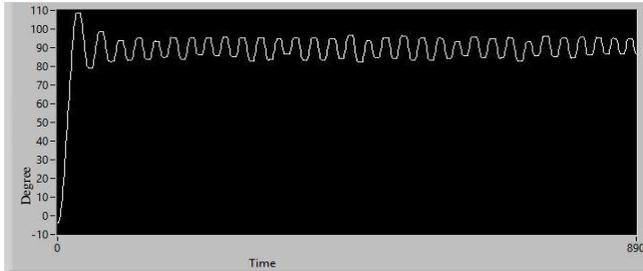
Gambar 4.27 merupakan respon perubahan posisi sudut *joint shoulder* pada robot manipulator tanpa kontroler dengan *input* sudut 80 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mendekati posisi 100 derajat. Dan mengalami osilasi sebesar 4 derajat.



Gambar 4.26 Respon Perubahan Posisi *Joint Shoulder* Tanpa Kontroler pada Sudut 60 Derajat



Gambar 4.27 Respon Perubahan Posisi *Joint Shoulder* Tanpa Kontroler pada Sudut 80 Derajat



Gambar 4.28 Respon Perubahan Posisi *Joint Shoulder* Tanpa Kontroler pada Sudut 90 Derajat

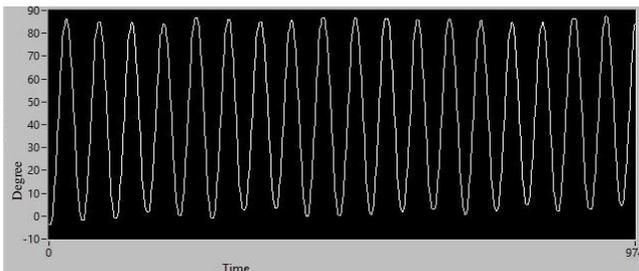
Gambar 4.28 merupakan respon perubahan posisi sudut *joint shoulder* pada robot manipulator tanpa kontroler dengan *input* sudut 90 derajat. Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mendekati posisi 110 derajat. Dan mengalami osilasi sebesar 5 derajat.

4.5.2 Respon Perubahan Posisi pada *Joint Shoulder* Dengan Kontroler

Terdapat beberapa variasi parameter yang digunakan pada pengujian ini.

1. Parameter $K_p = 2$, $\tau_i = 0$, $\tau_d = 0$

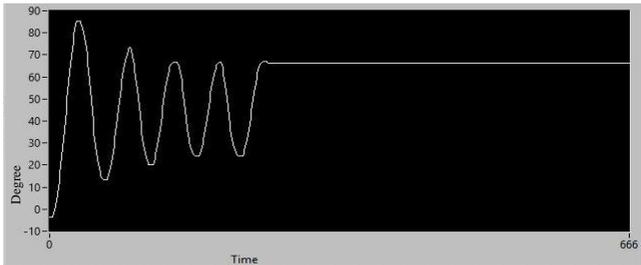
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.29. Gambar 4.29 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami osilasi sebesar 40 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.29 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 2, T_i 0, T_d 0

2. Parameter $K_p = 1, \tau_i = 0, \tau_d = 0$

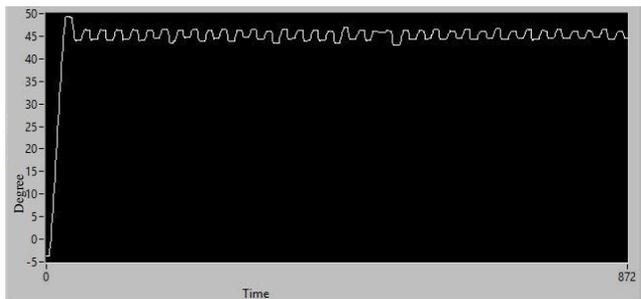
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.30. Gambar 4.30 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga melebihi posisi 80 derajat dan sempat mengalami osilasi namun pada akhirnya mencapai keadaan *steady* pada posisi 68 derajat. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.30 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

3. Parameter $K_p = 1, \tau_i = 0,01, \tau_d = 0$

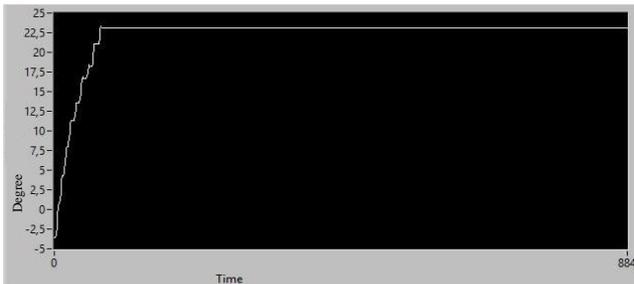
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.31. Gambar 4.31 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga mendekati posisi 50 derajat dan mengalami osilasi sebesar 3 derajat.



Gambar 4.31 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,01, T_d 0

4. Parameter $K_p = 1$, $\tau_i = 0,02$, $\tau_d = 0$

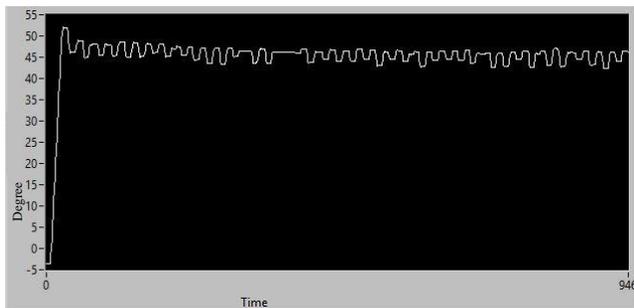
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.32. Gambar 4.32 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 23 derajat dan tidak mencapai *set point*. Untuk pengujian dengan sudut lain dapat dilihat pada lampiran A.



Gambar 4.32 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0

5. Parameter $K_p = 1$, $\tau_i = 0,02$, $\tau_d = 0,000009$

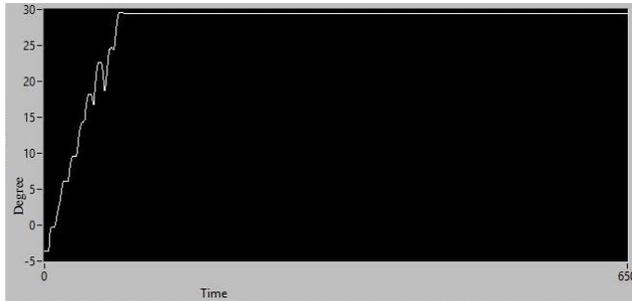
Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.33. Gambar 4.33 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon mengalami *overshoot* hingga melebihi posisi 50 derajat dan mengalami osilasi sebesar 3 derajat.



Gambar 4.33 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,000009

6. Parameter $K_p = 1$, $\tau_i = 0,02$, $\tau_d = 0,008$

Dari penggunaan parameter ini hasil yang didapatkan tampak seperti Gambar 4.34. Gambar 4.34 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 30 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 29 derajat.



Gambar 4.34 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 30 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008



Gambar 4.35 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 45 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008

Gambar 4.35 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 45 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 26 derajat.

Gambar 4.36 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 60 derajat Pada gambar tersebut terlihat bahwa

respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 57 derajat.



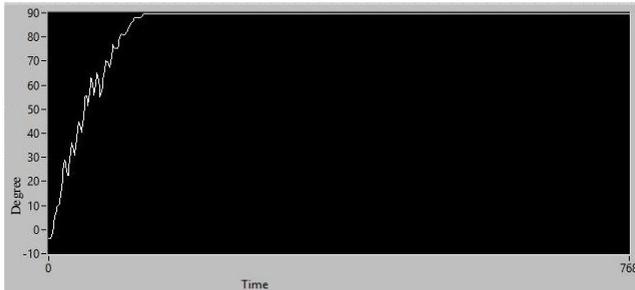
Gambar 4.36 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 60 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008



Gambar 4.37 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 80 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008

Gambar 4.37 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 80 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 80 derajat.

Gambar 4.38 menunjukkan respon perubahan sudut *joint shoulder* ketika diberi *input* 90 derajat Pada gambar tersebut terlihat bahwa respon tidak mengalami *overshoot* dan osilasi. Respon mencapai keadaan *steady* pada posisi 89 derajat.



Gambar 4.38 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0,008

4.5.3 Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada *Joint Shoulder*

Dari pengujian *joint* pada *shoulder* dengan menggunakan kontroler PID dan tanpa kontroler, didapatkan hasil perbandingan seperti pada Tabel 4.13. Dari Tabel 4.13 dapat dilihat bahwa respon tanpa kontroler selalu mengalami osilasi dan *overshoot*. Sehingga tidak pernah mencapai kondisi *steady*. Sedangkan respon ketika diberi kontroler mampu mencapai keadaan *steady*.

Tabel 4.13 Perbandingan Hasil Kontroler dengan Tanpa Kontroler pada *Joint Shoulder*

Input (°)	Tanpa Kontroler (°)	Dengan Kontroler [K_p 1, T_i 0,02, T_d 0,008] (°)	Keterangan	
			Tanpa Kontroler	Dengan Kontroler
30	30	29	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Tidak mencapai <i>set point</i> <i>Steady</i> pada posisi 29° Tidak mengalami osilasi dan <i>overshoot</i>
45	45	46	<ul style="list-style-type: none"> Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> Melebihi <i>set point</i> <i>Steady</i> pada posisi 46° Tidak mengalami osilasi dan <i>overshoot</i>

60	60	57	<ul style="list-style-type: none"> • Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> • Tidak mencapai <i>set point</i> • <i>Steady</i> pada posisi 57° • Tidak mengalami osilasi dan <i>overshoot</i>.
80	80	80	<ul style="list-style-type: none"> • Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> • Mampu mencapai <i>set point</i> • <i>Steady</i> pada posisi 80° • Tidak mengalami osilasi dan <i>overshoot</i>.
90	90	90	<ul style="list-style-type: none"> • Mengalami <i>overshoot</i> dan osilasi 	<ul style="list-style-type: none"> • Mampu mencapai <i>set point</i> • <i>Steady</i> pada posisi 90° • Tidak mengalami osilasi dan <i>overshoot</i>

BAB V PENUTUP

Setelah dilakukan pengujian alat, maka dapat diambil kesimpulan sebagai berikut :

1. Penggunaan kontroler PID dengan nilai $K_p = 1$, $\tau_i = 0,09$, $\tau_d = 0,005$ pada *joint base* dan $K_p = 1$, $\tau_i = 0,02$, $\tau_d = 0,008$ pada *joint shoulder*, hasil yang didapatkan adalah keluaran posisi sudut sesuai dengan posisi yang diinginkan. *Output* sudut sama dengan *input* sudut, tidak terjadi osilasi dan *overshoot*.
2. Sensor *rotary encoder* dapat mendeteksi perubahan sudut pada *joint robot manipulator* dengan prosentase kesalahan rata-rata sebesar 0,074423% pada bagian *base* dan 0,005181% pada bagian *shoulder*.
3. Pengujian yang dilakukan pada motor servo didapatkan prosentase kesalahan rata-rata sebesar 11,8421% pada bagian *elbow*, 0% pada bagian *wrist* dan *end effector*.

Untuk pengembangan penelitian selanjutnya sebaiknya dilakukan pemodelan terhadap respon motor DC terlebih dahulu untuk kontrol posisi, supaya didapatkan nilai parameter K_p , τ_i , dan τ_d yang sesuai dengan perhitungan.

-----Halaman ini sengaja dikosongkan-----

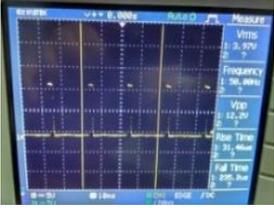
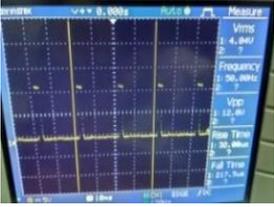
DAFTAR PUSTAKA

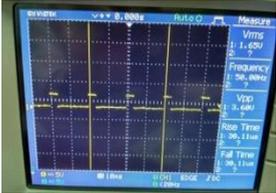
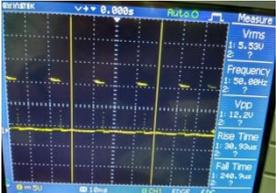
- [1] C. Melchiorri, "Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)," 2014. [Online]. Available: http://www-lar.deis.unibo.it/people/cmelchiorri/Files_Robotica/FIR_01_Industrial_Robotics.pdf. [Accessed 20 November 2016]
- [2] Gamayanti, Nurlita, "KONTROLER PROPORSIONAL DITAMBAH INTEGRAL DITAMBAH DIFFERENSIAL (PID)".
- [3] Hafid, 2009, "Manipulator", <https://hafid234.wordpress.com/2009/03/21/manipulator/> (Diakses 20 Mei 2017)
- [4] Hartono, N.B, dkk, "Pengaturan Posisi Motor Servo DC dengan Metode P, PI, dan PID", *Proyek Akhir Jurusan Teknik Elektronika-PENS*, Surabaya.
- [5] Istrianono, D.R. dan Agustinah, T., "Perancangan Pengaturan Posisi Robot Manipulator Berbasis PD Fuzzy Mamdani Computed Torque Control (PD Fuzzy CTC)", *Jurnal Teknik ITS Vol. 4, No. 1, (2015) ISSN: 2337-3539*, Surabaya, 2015.
- [6] Jurnal Binus, 2014, "Bab II Landasan Teori", <http://library.binus.ac.id/eColls/eThesidoc/Bab2/2014-2-00288-SK%20Bab2001.pdf> (Diakses 20 Mei 2017)
- [7] Pitowarno, Endra, "Robotika, Desain, Kontrol, dan Kecerdasan Buatan", Andi Offset, Yogyakarta, 2006.
- [8] S. Adi dan A. Faisal. 2016. "Perancangan *Human Machine Interface* (HMI) Untuk Robot Lengan Tetrix Menggunakan Visual Basic Berbasis Open-GL", *Laporan Tugas Akhir*, Program Studi D3 Teknik Elektro, ITS, Surabaya.
- [9] _____, "Hitec HS-485HB - Deluxe HD Ball Bearing Servo", <https://servodatabase.com/servo/hitec/hs-485hb> (Diakses 13 Juli 2017)
- [10] _____, "myRIO Student Embedded Device" <http://www.ni.com/en-id/shop/select/myrio-student-embedded-device> (Diakses 20 Mei 2017)

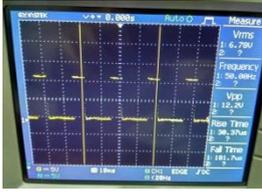
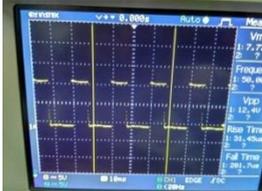
- [11] ____, “myRIO-1900”, <http://www.ni.com/en-us/support/model.myrio-1900.html> (Diakses 20 Mei 2017)
- [12] ____, “NAO”, <http://robotics.wikia.com/wiki/NAO> (Diakses 20 Mei 2017)
- [13] ____, “TETRIX[®] MAX DC Motor”, <https://www.pitsco.com/TETRIX-DC-Gear-Motor> (Diakses 20 Mei 2017)
- [14] ____, <http://users.ece.utexas.edu/~valvano/Datasheets/ServoHS311.pdf> (Diakses 13 Juli 2017)
- [15] ____, <http://www.directindustry.com/prod/hengstler/product-5789-873945.html> (Diakses 20 Mei 2017)
- [16] ____, http://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf (Diakses 13 Juli 2017)
- [17] ____, <https://www.etrailer.com/faq-hand-winch.aspx> (Diakses 13 Juli 2017)

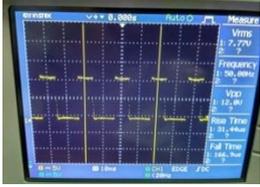
LAMPIRAN A DATA PENGUJIAN

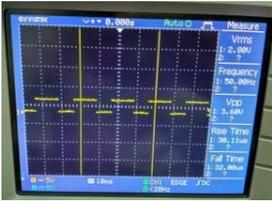
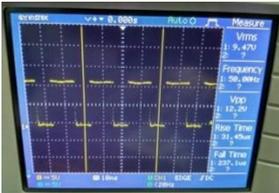
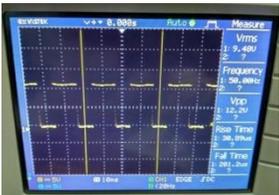
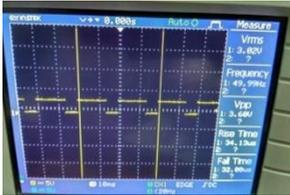
A.1 Gambar Gelombang pada Pengujian *Driver Motor* dengan IN1 dan IN2 berlogika 1-1

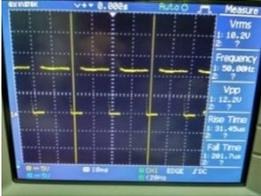
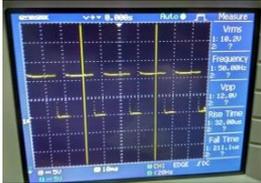
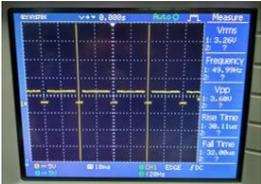
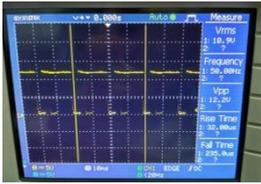
No.	<i>Duty Cycle</i>	Gambar Gelombang
1	0,1	<p data-bbox="544 411 645 437">Pin ENA :</p>  <p data-bbox="544 708 656 734">Pin OUT1 :</p>  <p data-bbox="544 1007 656 1032">Pin OUT2 :</p> 

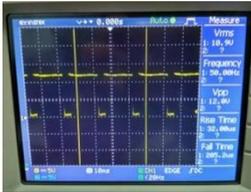
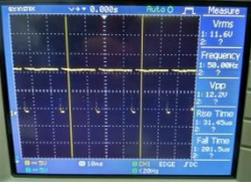
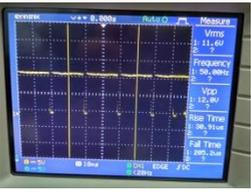
2	0,2	<p>Pin ENA :</p>  <p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 
3	0,3	<p>Pin ENA :</p> 

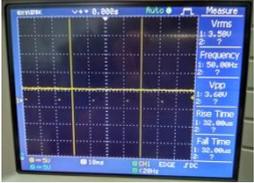
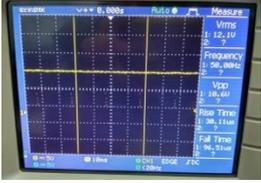
		<p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 
4	0,4	<p>Pin ENA :</p>  <p>Pin OUT1 :</p> 

		<p>Pin OUT2 :</p> 
5	0,5	<p>Pin ENA :</p>  <p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 

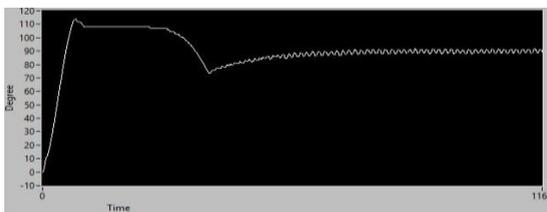
6	0,6	<p>Pin ENA :</p>  <p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 
7	0,7	<p>Pin ENA :</p> 

		<p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 
8	0,8	<p>Pin ENA :</p>  <p>Pin OUT1 :</p> 

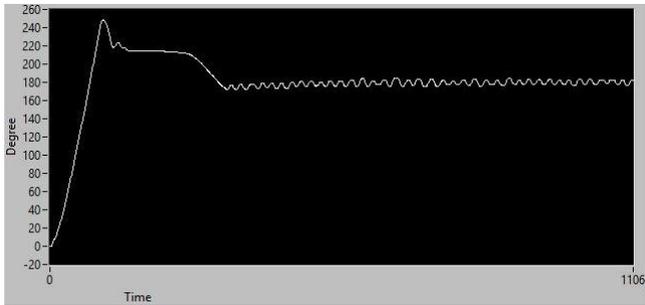
		<p>Pin OUT2 :</p> 
9	0,9	<p>Pin ENA :</p>  <p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 

10	0,99	<p>Pin ENA :</p>  <p>Pin OUT1 :</p>  <p>Pin OUT2 :</p> 
----	------	--

A.2 Respon Perubahan Sudut *Joint Base* dengan Nilai K_p 2, T_i 0, T_d 0



Gambar 1 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 2, T_i 0,02, T_d 0



Gambar 2 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 2, T_i 0,02, T_d 0

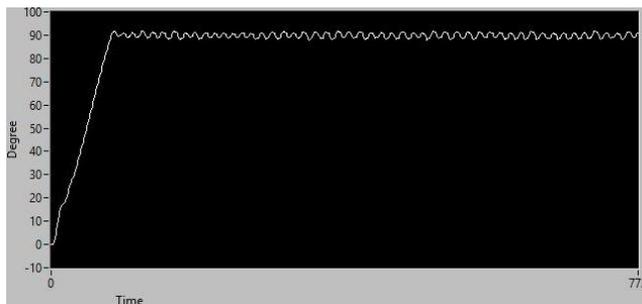


Gambar 3 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 2, T_i 0,02, T_d 0



Gambar 4 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_p 2, T_i 0,02, T_d 0

A.3 Respon Perubahan Sudut *Joint Base* dengan Nilai K_p 1, T_i 0, T_d 0



Gambar 5 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0, T_d 0



Gambar 6 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

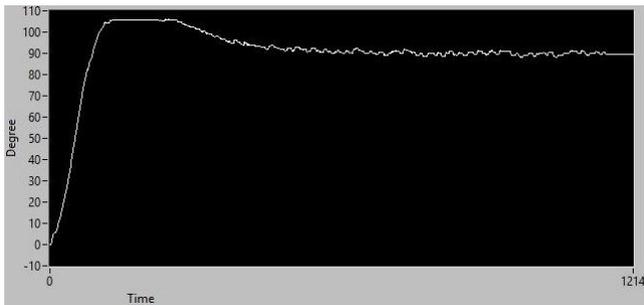


Gambar 7 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

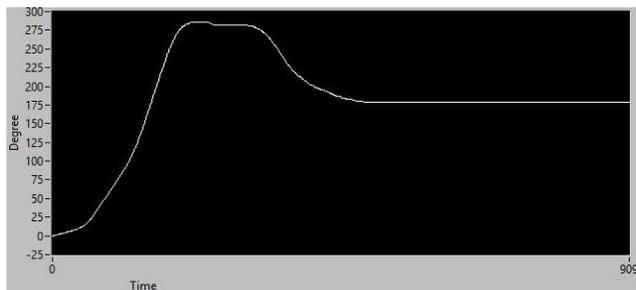


Gambar 8 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

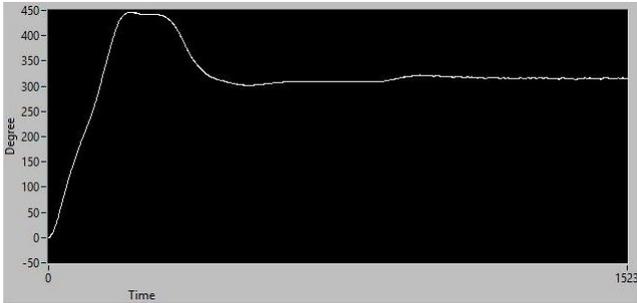
A.4 Respon Perubahan Sudut *Joint Base* dengan Nilai K_p 1, T_i 0,02, T_d 0



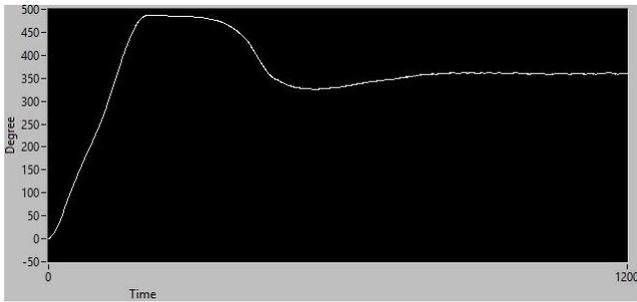
Gambar 9 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0



Gambar 10 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0

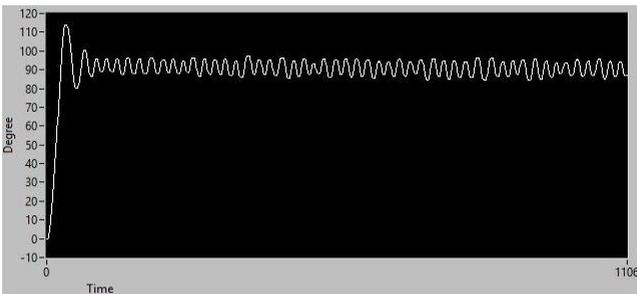


Gambar 11 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0

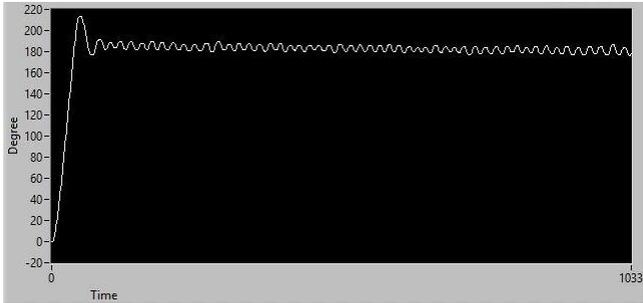


Gambar 12 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_p 1, T_i 0,02, T_d 0

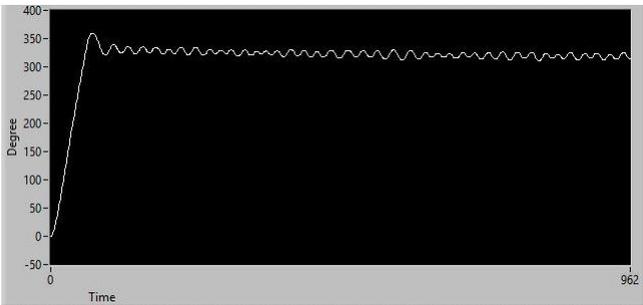
A.5 Respon Perubahan Sudut *Joint Base* dengan Nilai K_p 1, T_i 0,09, T_d 0



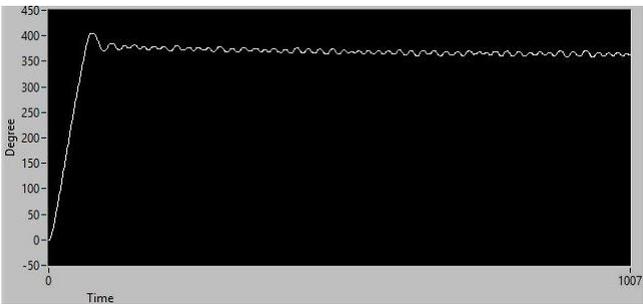
Gambar 13 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0



Gambar 14 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0

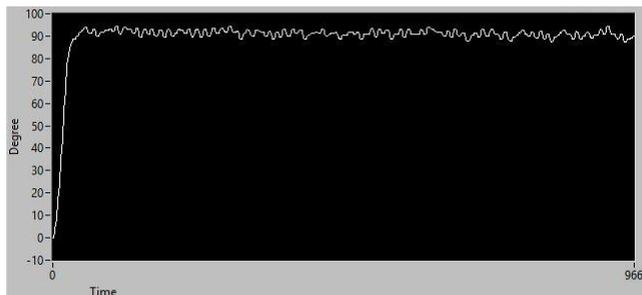


Gambar 15 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0

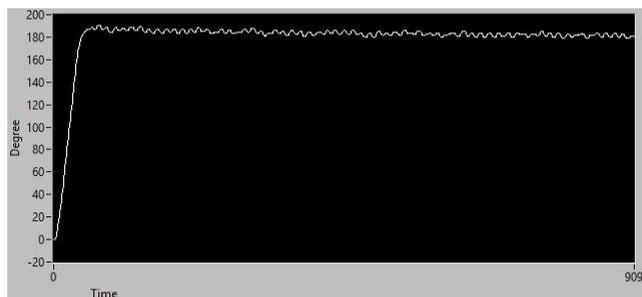


Gambar 16 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0

A.6 Respon Perubahan Sudut *Joint Base* dengan Nilai K_p 1, T_i 0,09, T_d 0,001



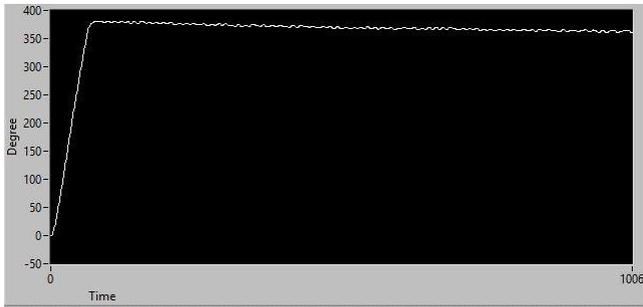
Gambar 17 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,001



Gambar 18 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 180 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,001

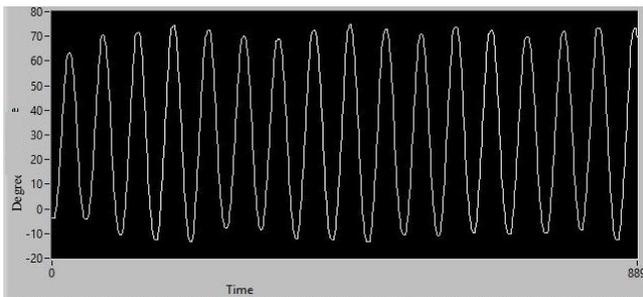


Gambar 19 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 315 Derajat dengan Nilai K_p 1, T_i 0,09, T_d 0,001

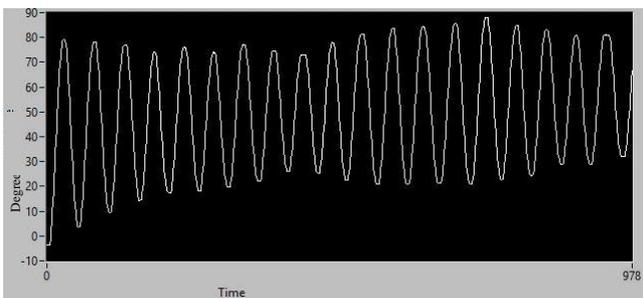


Gambar 20 Respon Perubahan Sudut *Joint Base* saat Diberi *Input* 360 Derajat dengan Nilai K_P 1, T_i 0,09, T_d 0,001

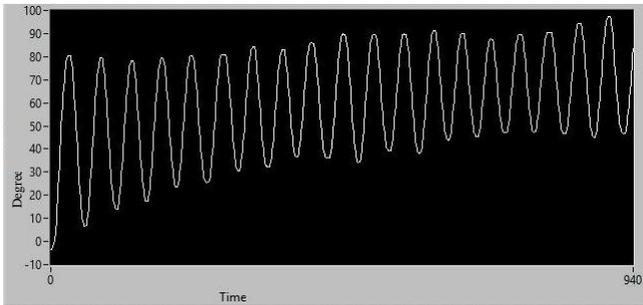
A.7 Respon Perubahan Sudut *Joint Shoulder* dengan Nilai K_P 2, T_i 0, T_d 0



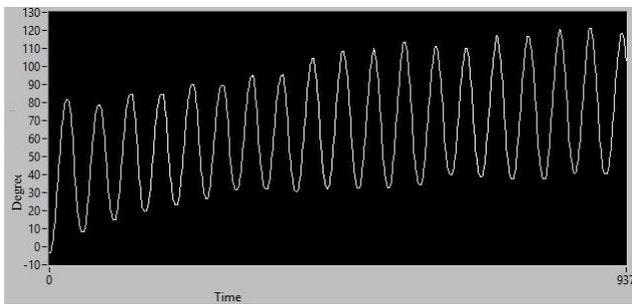
Gambar 21 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 30 Derajat dengan Nilai K_P 2, T_i 0,1, T_d 0



Gambar 22 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 60 Derajat dengan Nilai K_P 2, T_i 0,1, T_d 0

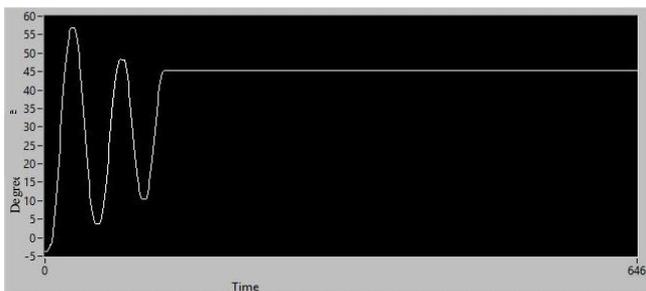


Gambar 23 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 80 Derajat dengan Nilai K_p 2, T_i 0,1, T_d 0



Gambar 24 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 90 Derajat dengan Nilai K_p 2, T_i 0,1, T_d 0

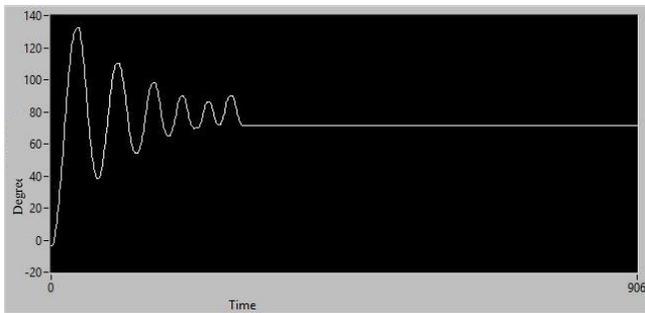
A.8 Respon Perubahan Sudut *Joint Shoulder* dengan Nilai K_p 1, T_i 0, T_d 0



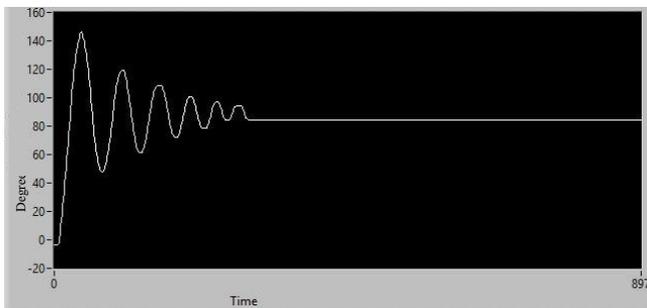
Gambar 25 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 30 Derajat dengan Nilai K_p 1, T_i 0, T_d 0



Gambar 26 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 60 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

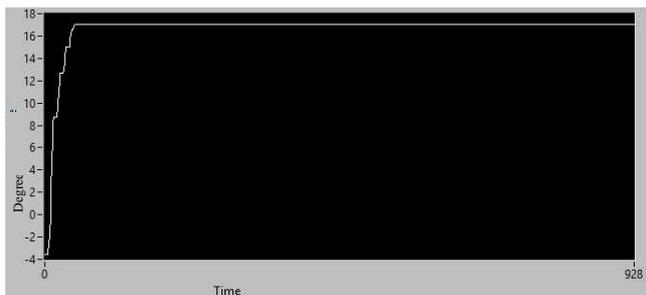


Gambar 27 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 80 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

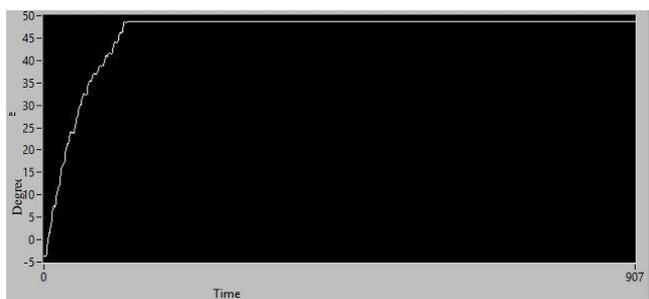


Gambar 28 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0, T_d 0

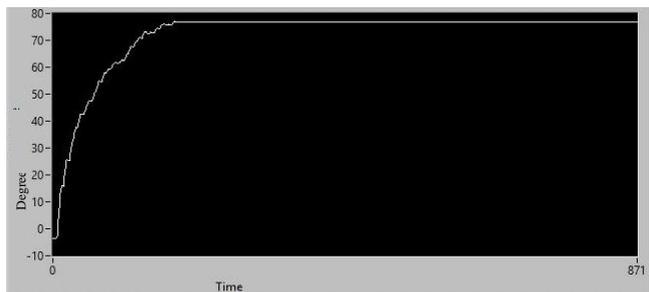
A.9 Respon Perubahan Sudut *Joint Shoulder* dengan Nilai K_p 1, T_i 0,02, T_d 0



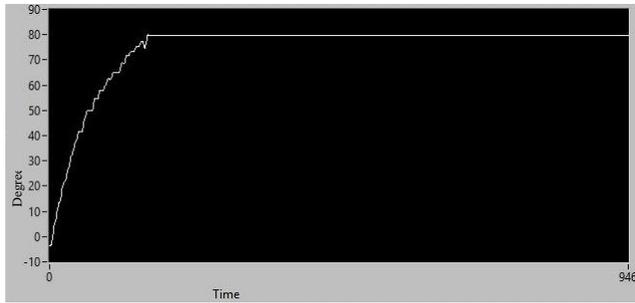
Gambar 33 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 30 Derajat dengan Nilai K_p 1, T_i 0,2, T_d 0,01



Gambar 34 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 60 Derajat dengan Nilai K_p 1, T_i 0,2, T_d 0,01



Gambar 35 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 80 Derajat dengan Nilai K_p 1, T_i 0,2, T_d 0,01

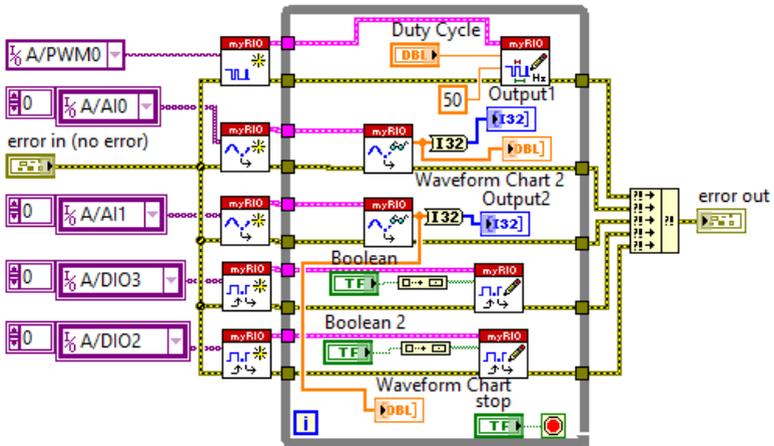


Gambar 36 Respon Perubahan Sudut *Joint Shoulder* saat Diberi *Input* 90 Derajat dengan Nilai K_p 1, T_i 0,2, T_d 0,01

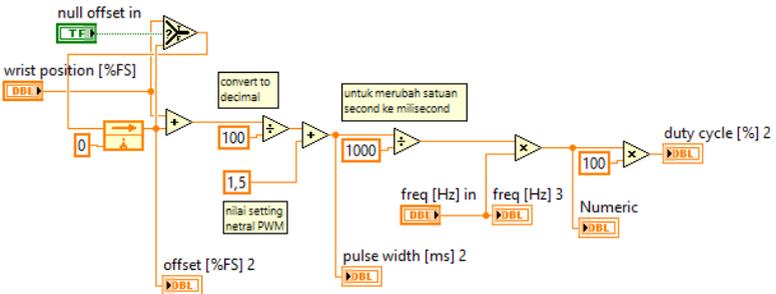
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B LISTING PROGRAM

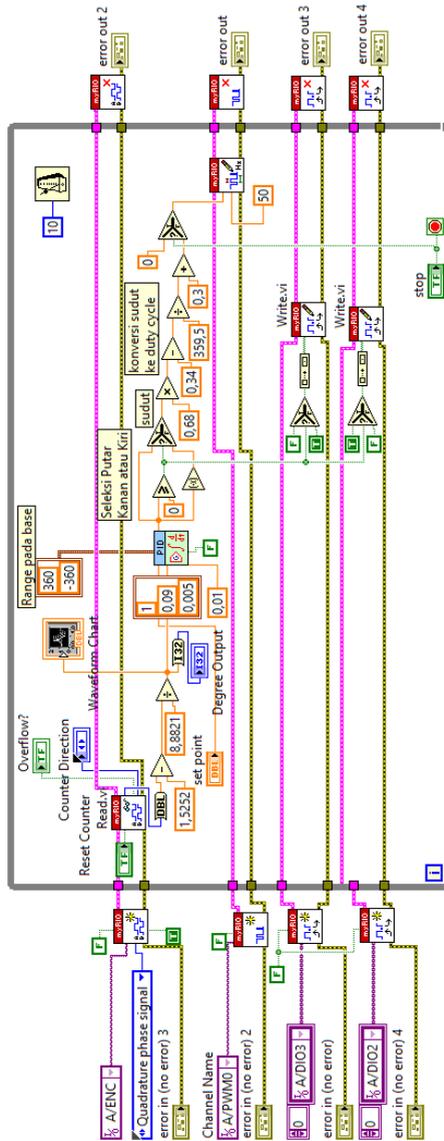
B.1 Listing Program Pengujian Driver Motor



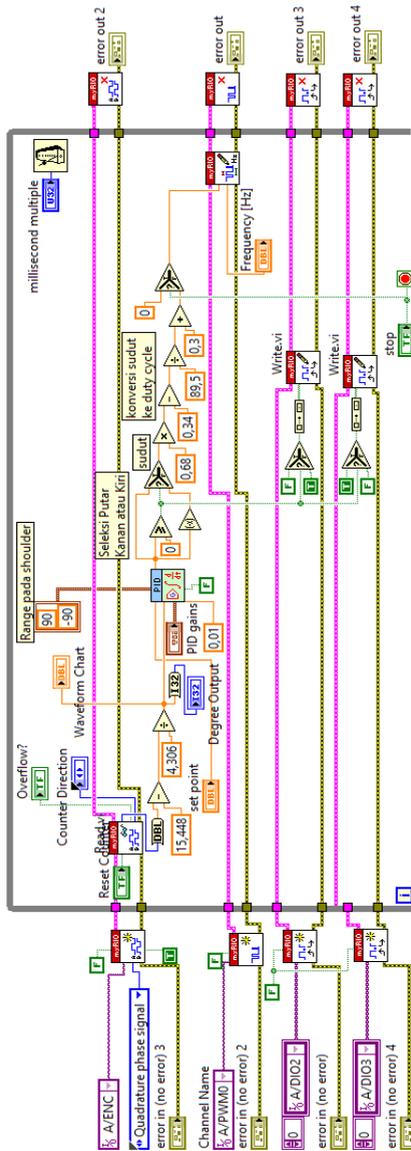
B.2 Listing Program Pengujian Motor Servo



B.3 Listing Program Pengujian Joint pada Base



B.4 Listing Program Pengujian Joint pada Shoulder



-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

DATASHEET

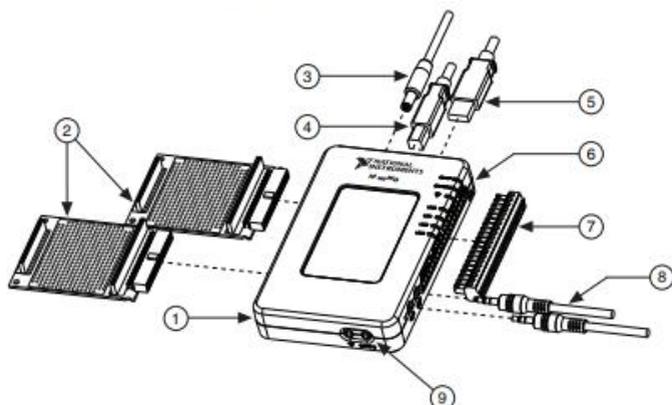
C.1 Datasheet NI myRIO 1900

USER GUIDE AND SPECIFICATIONS

NI myRIO-1900

The National Instruments myRIO-1900 is a portable reconfigurable I/O (RIO) device that students can use to design control, robotics, and mechatronics systems. This document contains pinouts, connectivity information, dimensions, mounting instructions, and specifications for the NI myRIO-1900.

Figure 1. NI myRIO-1900



- | | |
|--|---|
| 1 NI myRIO-1900 | 6 LEDs |
| 2 myRIO Expansion Port (MXP) Breakouts (One Included in Kit) | 7 Mini System Port (MSP) Screw-Terminal Connector |
| 3 Power Input Cable | 8 Audio In/Out Cables (One Included in Kit) |
| 4 USB Device Cable | 9 Button0 |
| 5 USB Host Cable (Not Included in Kit) | |

Connector Pinouts

NI myRIO-1900 Expansion Port (MXP) connectors A and B carry identical sets of signals. The signals are distinguished in software by the connector name, as in `ConnectorA/DIO1` and `ConnectorB/DIO1`. Refer to the software documentation for information about configuring and using signals. The following figure and table show the signals on MXP connectors A and B. Note that some pins carry secondary functions as well as primary functions.

Figure 3. Primary/Secondary Signals on MXP Connectors A and B

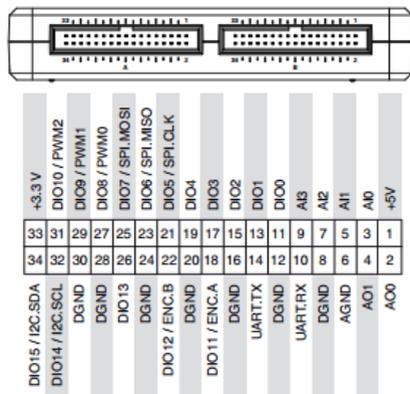


Table 1. Descriptions of Signals on MXP Connectors A and B

Signal Name	Reference	Direction	Description
+5V	DGND	Output	+5 V power output.
AI <0..3>	AGND	Input	0-5 V, referenced, single-ended analog input channels. Refer to the Analog Input Channels section for more information.
AO <0..1>	AGND	Output	0-5 V referenced, single-ended analog output. Refer to the Analog Output Channels section for more information.
AGND	N/A	N/A	Reference for analog input and output.
+3.3V	DGND	Output	+3.3 V power output.
DIO <0..15>	DGND	Input or Output	General-purpose digital lines with 3.3 V output, 3.3 V/5 V-compatible input. Refer to the DIO Lines section for more information.
UART.RX	DGND	Input	UART receive input. UART lines are electrically identical to DIO lines.
UART.TX	DGND	Output	UART transmit output. UART lines are electrically identical to DIO lines.
DGND	N/A	N/A	Reference for digital signals, +5 V, and +3.3 V.

DIO Lines

The NI myRIO-1900 has 3.3 V general-purpose DIO lines on the MXP and MSP connectors. MXP connectors A and B have 16 DIO lines per connector. On the MXP connectors, each DIO line from 0 to 13 has a 40 k Ω pullup resistor to 3.3 V, and DIO lines 14 and 15 have 2.2 k Ω pullup resistors to 3.3 V. MSP connector C has eight DIO lines. Each MSP DIO line has a 40 k Ω pulldown resistor to ground. DGND is the reference for all the DIO lines. You can program all the lines individually as inputs or outputs. Secondary digital functions include Serial Peripheral Interface Bus (SPI), I2C, pulse-width modulation (PWM), and quadrature encoder input. Refer to the NI myRIO software documentation for information about configuring the DIO lines.

Figure 7. DIO Lines <13..0> on MXP Connector A or B

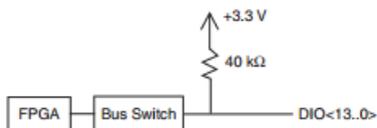


Figure 8. DIO Lines <15..14> on MXP Connector A or B

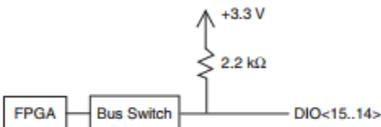
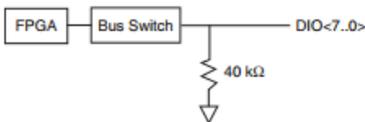


Figure 9. DIO Lines <7..0> on MSP Connector C



When a DIO line is floating, it floats in the direction of the pull resistor. A DIO line may be floating in any of the following conditions:

- when the myRIO device is starting up
- when the line is configured as an input
- when the myRIO device is powering down

You can add a stronger resistor to a DIO line to cause it to float in the opposite direction.

C.2 Datasheet Driver Motor L298N



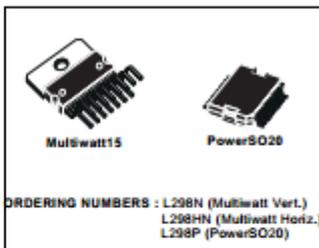
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

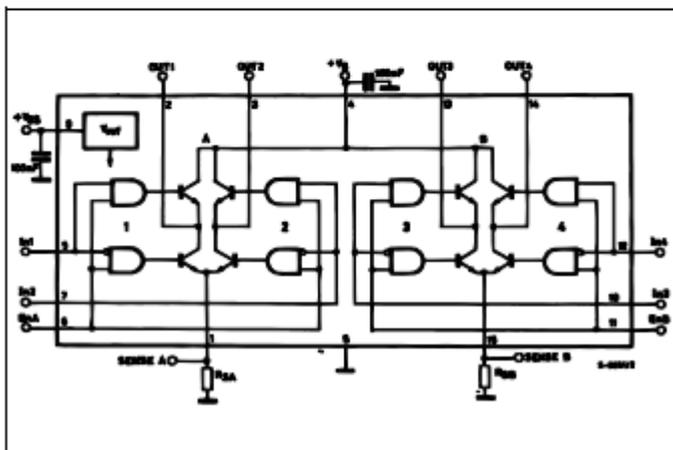


Figure 6 : Bidirectional DC Motor Control.

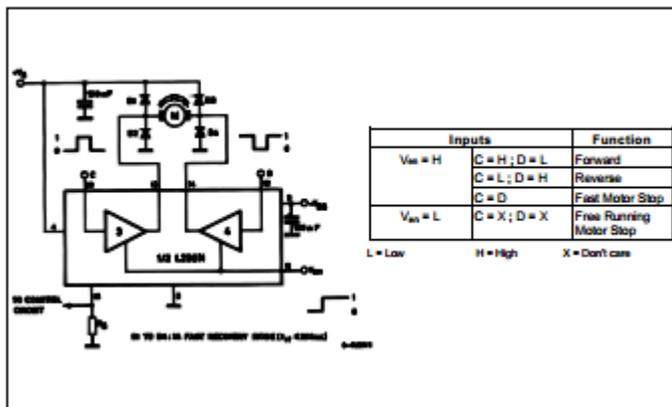
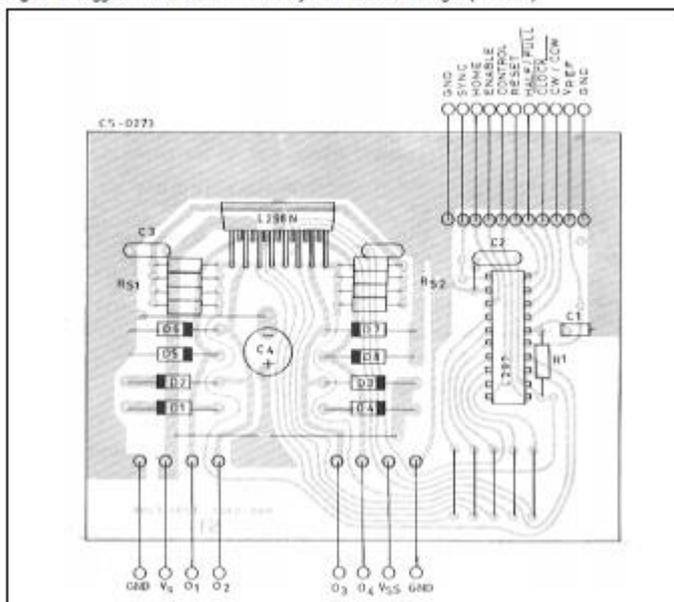


Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).



C.3 Datasheet Motor Servo Hitec HS-485HB

Hitec HS-485HB - Deluxe HD Ball Bearing Servo

Specifications

Modulation:	Analog
Torque:	4.8V: 67.00 oz-in (4.82 kg-cm) 6.0V: 83.00 oz-in (5.98 kg-cm)
Speed:	4.8V: 0.22 sec/60° 6.0V: 0.18 sec/60°
Weight:	1.59 oz (45.1 g)
Dimensions:	Length: 1.57 in (39.9 mm) Width: 0.78 in (19.8 mm) Height: 1.49 in (37.8 mm)
Motor Type:	3-pole
Gear Type:	Plastic
Rotation/Support:	Single Bearing
Rotational Range:	60°
Pulse Cycle:	20 ms
Pulse Width:	900-2100 μ s
Connector Type:	JR



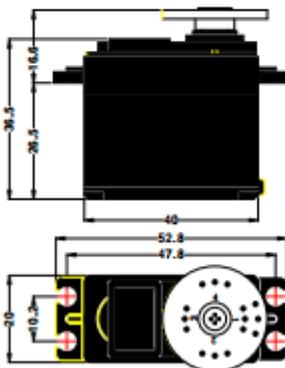
Brand:	HITEC
Product Number:	334855
Typical Price:	16.99 USD
Compare:	add+

C.4 Datasheet Motor Servo Hitec HS-331

PREPARED BY JUN HEE, LEE
UPDATE: APR 01, 2002

ANNOUNCED SPECIFICATION OF HS-311 STANDARD SERVO

1. TECHNICAL VALUE		
CONTROL SYSTEM	:-PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	:4.8V TO 6.0V	
TEST VOLTAGE	:AT 4.8V	AT 6.0V
OPERATING SPEED	:0.19sec/60□ AT NO LOAD	0.15sec/60□ AT NO LOAD
STALL TORQUE	:3.0kg.cm(42oz.in)	3.5kg.cm(48.60oz.in)
IDLE CURRENT	:7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	:160mA/60□ AT NO LOAD	180mA/60□ AT NO LOAD
STALL CURRENT	:700mA	800mA
DEAD BAND WIDTH	:5usec	5usec
OPERATING TRAVEL	:40□/ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	:CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	:CORED METAL BRUSH	
POTENTIOMETER TYPE	:4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	:ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	:40x20x36.5mm(1.57x0.78x1.43in)	
WEIGHT	:43g(1.51oz)	
BALL BEARING	:TOP/RESIN BUSHING	
GEAR MATERIAL	:RESIN	
HORN GEAR SPLINE	:24 SEGMENTS/□15.76	
SPLINED HORNS	:SUPER/R-XA	
CONNECTOR WIRE LENGTH	:300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	:40EA	
CONNECTOR WIRE GAUGE	:40EA	



2. FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING

3. APPLICATIONS

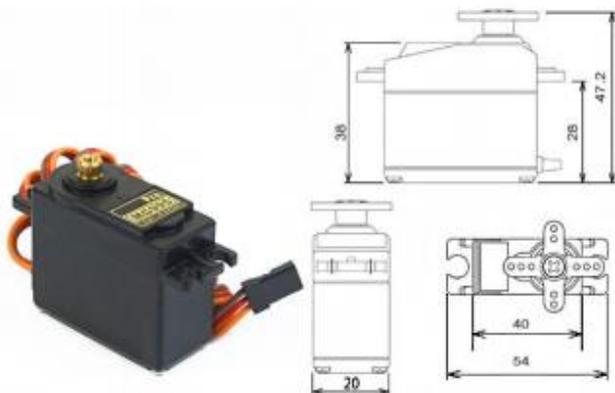
AIRCRAFT 20-40 SIZE, STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4. ACCESSORY & OPTION

CASE SET/	GEAR SET/	HORN SET/
HS322T:1EA	HS322G1:1EA	R-XA:1EA
HS322M:1EA	HS322G2:1EA	
HS322L:1EA	HS322G3:1EA	
PH/T-2 2x30 NI:4EA	HS322G4:1EA	
	HS300RB:1EA	

C.5 Datasheet Motor Servo MG995

MG995 High Speed Metal Gear Dual Ball Bearing Servo



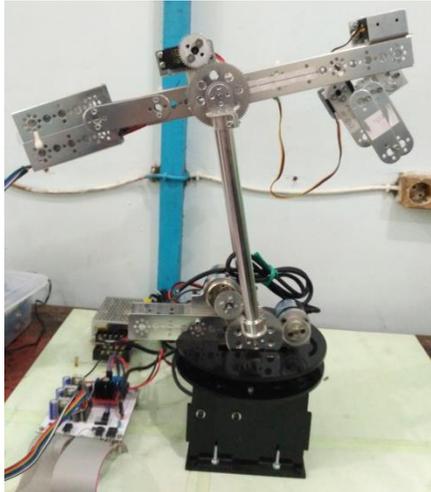
The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

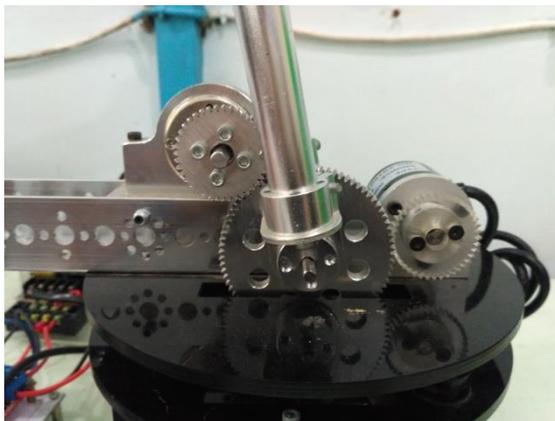
Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgF·cm (4.8 V), 10 kgF·cm (6 V)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

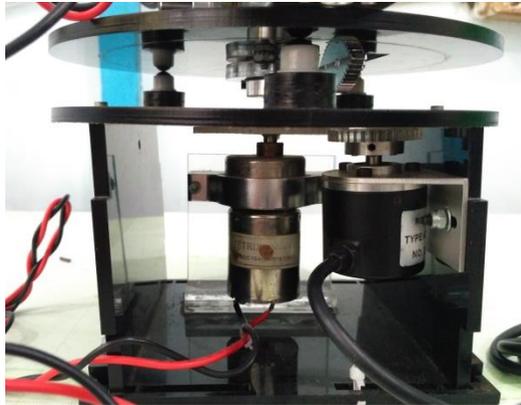
LAMPIRAN D DOKUMENTASI ALAT



Gambar 37 Bentuk Fisik Robot Manipulator yang Digunakan



Gambar 38 Pemasangan Sensor *Rotary Encoder* pada *Joint Shoulder*



Gambar 39 Pemasangan Sensor *Rotary Encoder* pada *Joint Base*

RIWAYAT HIDUP PENULIS



Nama : Diana Yufika Anshori
TTL : Tulungagung, 31 Maret 1995
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Dsn. Krajan Ds. Karangsono
¾ Ngunut Tulungagung
No. HP : 085730161669
Email : dianayufika@gmail.com
Hobi : Mendengarkan musik,
membaca

RIWAYAT PENDIDIKAN

- 2002-2008 : SDI Al Azhaar Tulungagung
- 2008-2011 : SMPN 1 Tulungagung
- 2011-2014 : SMAN 1 Kedungwaru
- 2014-sekarang : Program Studi Komputer Kontrol Departemen Teknik Elektro Otomasi ITS

PENGALAMAN KERJA

- Kerja Praktek di PT. PJB UPHT Gresik (2016)

PENGALAMAN ORGANISASI

- Staff Departemen Riset dan Teknologi Himpunan Mahasiswa D3 Teknik Elektro, ITS (2015-2016)

PENGALAMAN KEPANITIAAN

- Staff Sie Kestari FTI OLYMPIC GAMES (FOG)
- Staff Sie Konsumsi IARC (*Industrial Automatic Robot Competition*) 2015
- Staff Sie Konsumsi IARC (*Industrial Automatic Robot Competition*) 2016