# 3.4 Rancangan Penelitian

Dalam merancang dan pembuatan alat aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA ini, dilakukan secara bertahap.

Tahap pertama adalah perancangan dan pembuatan *hardware* alat tangan robot. Perancangan dan pembuatan hardware alat dibagi menjadi 2 (dua) tahap yaitu proses perancangan dan pembuatan *hardware* sistem mekanik dan perancangan dan pembuatan hardware sistem elektrik.

Perancangan dan pembuatan bagian mekanik terdiri dari beberapa proses tahapan yang dilakukan yaitu :

- 1. Merancang rangka bentuk tangan robot.
- 2. Menggambar bagian-bagian rangka tangan robot pada bahan akrilik.
- 3. Menggergaji dan mengamplas bagian-bagian rangka tangan robot.
- 4. Mengebor dudukan baut dan sambungan pada motor.
- 5. Merakit keseluruhan bagian-bagian tangan robot.
- 6. Mengecat semua bagian tangan robot.

Perancangan dan pembuatan bagian elektrik terdiri dari beberapa proses tahapan yang dilakukan yaitu :

- 1. Merancang gambar skematik rangkaian elektrik.
- 2. Menginventaris semua komponen yang diperlukan.
- 3. Menguji komponen yang hendak dipakai.
- 4. Melakukan percobaan rangkaian pada papan protoboard.

- 5. Membuat *lay out* rangkaian pada papan PCB kemudian dilakukan pengecingan.
- 6. Pengeboran papan PCB dan pemasangan komponen.
- 7. Pengujian rangkaian.

Tahap kedua adalah proses perancangan dan pembuatan software atau program kendali yang digunakan untuk pengendalian alat.

Gambar 3.1 menunjukkan konstruksi aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA.



Gambar 3.1 Konstruksi alat keseluruhan.

Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, terdiri dari 3 bagian penting, diantaranya adalah :

a. Komputer

Komputer berfungsi sebagai pengendali utama yang akan mengatur pergerakan tangan robot melalui pheripheral antarmuka yang dipasangkan pada slot ISA. Komputer akan menerima dan mengirim sinyal dari dan ke alat terkendali. Adapun bagian-bagian penting yang mendukung kerja komputer yaitu :

- 1. Power Suply 250 Watt
- 2. Main Board + Processor Pentium II
- 3. Memory SD Ram 32 MB
- 4. Hardisk 6 GB
- 5. Mouse, Keyboard dan monitor
- 6. Sistem Operasi Windows 98
- 7. Borland Delphi 7.0
- b. Pheripheral Antarmuka

Pheripheral antarmuka berfungsi sebagai media penghubung antara bagian pengendali yaitu komputer dengan alat terkendali yaitu tangan robot. Kerja daripada pheripheral antarmuka dapat dikendalikan oleh komputer dengan cara menerima dan mengirim data ke komputer. Data yang dikirimkan untuk pengontrolan sebanyak 8 bit (D0-D7). Adapun fasilitas pheripheral antarmuka adalah :

- 1. ISA PIO PPI 8255 Card 24 bit
- 2. Konektor DB25
- c. Objek Kendali

Objek kendali adalah objek yang akan dikendalikan oleh komputer dalam sistem pengendali. Adapun objek kendali yang hendak dikendalikan adalah :

1. Motor DC gearbox = 2 Buah

2.	Motor Servo 180 <sup>°</sup>	= 3 Buah
3.	Motor Servo 360 <sup>0</sup>	= 1 Buah

# 3.4.1. Bagian Mekanik

Gambar 3.2 menunjukkan konstruksi mekanik tangan robot.



Gambar 3.2 Konstuksi Mekanik Tangan Robot.

- Keterangan : Motor 1 Motor DC gearkotak 12 V
  Motor 2 Motor DC gearkotak 12 V
  Motor 3 Motor servo standar 180<sup>0</sup>
  Motor 4 Motor servo standar 180<sup>0</sup>
  Motor 5 Motor servo continous 360<sup>0</sup>
  Motor 6 Motor servo standar 180<sup>0</sup>
- 1. Bahan

Bahan yang digunakan pada aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA ini adalah akrilik yaitu sejenis plastik padat yang bentuknya menyerupai kaca. Pembuatan alat digunakan akrilik dengan ketebalan 8 mm dan 5 mm.

2. Pembuatan rangka tangan robot

a. Pembuatan kotak yang difungsikan untuk meletakkan hardware elektrik yaitu power suplay, driver motor DC, driver motor servo dan juga digunakan untuk meletakkan motor DC 1 pada posisi tengah kotak agar tangan robot ini dapat bergerak 360<sup>0</sup> pada porosnya. Digunakan akrilik ukuran 8 mm untuk sisi-sisinya dan ukuran 5 mm untuk dasar dan tutup kotak. Gambar 3.3 menunjukkan bentuk kotak dasar tangan robot.



Gambar 3.3 Kotak dasar tangan robot.

b. Pembuatan dudukan motor

Perancangan dan pembuatan dudukan motor disesuaikan dengan bentuk dudukan motor itu sendiri. Sebelum dibuat dudukan terlebih dahulu mengukur berapa besar luas sisi-sisi motor dan peletakan lubang untuk baut dudukan motor dan as gear motor. Untuk dudukan pada motor DC 2 ditambahkan gear/roda gigi tambahan. Untuk posisi sambungan antar lengan, pada rangka lengan belakang dibaut pada gear motor servo. Untuk pembuatan rangka dudukan motor digunakan akrilik ukuran 5 mm. Gambar 3.4 menunjukkan rangka dudukan motor.



Gambar 3.4 Rangka dudukan motor.

Keterangan : a. Dudukan Motor DC 2
b. Dudukan Motor Servo 3
c. Dudukan Motor Servo 4
d. Dudukan Motor Servo 5 dan 6

c. Pembuatan jari tangan

Perancangan dan pembuatan jari tangan robot menggunakan pegas yang dipasang pada bagian dalam ruas jari untuk mendapatkan posisi jari yang tertekuk/tertutup, sedangkan untuk membuat agar ruas jari dapat terbuka maka pada tiap ruas jari bagian luar diikatkan tali yang dihubungkan pada gear motor servo. Pada saat motor servo berputar searah jarum jam maka tali yang diikat pada ruas jari akan mengencang dan jari-jari tangan robot akan terbuka. Pada saat motor servo berputar berlawanan arah jarum jam maka, tali akan mengendur dan jari-jari tangan robot akan tertutup. Gambar 3.5 menunjukkan bentuk mekanik jari tangan robot.



Gambar 3.5 Jari tangan robot.

3. Pemasangan motor

Setelah kotak dan rangka dudukan motor telah selesai dibuat maka proses selanjutnya adalah pemasangan motor-motor yang telah disiapkan. Posisi motor yang telah terpasang akan memperlihatkan bentuk lengan robot yang sesungguhnya seperti terlihat pada gambar 3.6.



Gambar 3.6 Perakitan mekanik tangan robot.

# 3.4.2. Bagian Elektrik

Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, dijalankan melalui suatu program menggunakan pemrograman Delphi 7.0 pada PC (*Personal Computer*) yang kemudian dari program tersebut akan mengendalikan perangkat luar yaitu card PPI 8255. Dengan mengendalikan card PPI 8255 maka port-port yang tersedia pada PPI 8255 dapat digunakan PC untuk memberikan data yang diperlukan pada tiap driver motor.

## 1. Rangkaian card PPI 8255

Rangkaian card PPI 8255 berfungsi untuk menerima data dari komputer kemudian mengirimkannya ke perangkat luar seperti driver motor dan sebaliknya dapat menerima data dari sensor yang kemudian dikirimkan menuju komputer. Rangkaian card PPI 8255 memiliki 24 bit jalur input/output yang dapat diprogram untuk keperluan tertentu. Gambar blok diagram rangkaian card PPI 8255 dapat dilihat pada gambar 3.7.



Gambar 3.7 Diagram Blok Rangkaian card PPI 8255.

Rangkaian card PPI 8255 dipasang pada slot ISA yang tersedia di *main board* komputer dengan penggunaan port sebagai berikut :

1. Bus Address

Bus Address yang digunakan sebanyak 10 bit yaitu A0 s.d A9. Untuk data 2 bit alamat bus yaitu A0 s.d A1 digunakan untuk mengendalikan penggunaan port A, port B atau port C dan kendali port pada PPI 8255. Sedangkan data 8 bit alamat bus yaitu A2-A9 digunakan untuk mengendalikan komposisi saklar sebagai alamat decoder yang berfungsi untuk menentukan alamat pengaktif PPI 8255 karena terhubung dengan CS (Chip Select) dari PPI 8255. CS (Chip Select) pada PPI 8255 bersifat aktif rendah yaitu PPI 8255 dapat aktif apabila CS diberi logika "0". Jika pada pin CS diberi logika "1" berarti PPI 8255 tidak aktif atau tidak berfungsi. Pemilihan untuk pengaktifan port pada PPI 8255 dapat dilihat pada tabel 3.1.

Tabel 3.1 Pengaktifan port PPI 8255

A1	A0	Aktif
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Kontrol register

Tabel 3.2 menunjukkan konfigurasi alamat card PPI 8255.

Tabel 3.2 Konfigurasi alamat
------------------------------

Jumper switch			Alamat PortA, Port B, Port C	
С	В	А	dan Kontrol Register	
0	0	0	200-203 Hex	
0	0	1	280-283 hex	
0	1	0	284-287 Hex	
0	1	1	2E0-2E3 Hex	
1	0	0	300-303 Hex	
1	0	1	380-383 Hex	
1	1	0	384-387 Hex	
1	1	1	3E0-3E3 Hex	
Keterangan : 1 = Jumper OFF dan 0 = Jumper ON				

Perincian pengalamatan untuk pemilihan tiap-tiap port pada PPI 8255 dapat dilihat pada tabel 3.3.

Tabel 3.3. Detail Pengalamatan Bus Address

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Hex	Nama Port
1	0	0	0	0	0	0	0	0	0	200	Port A
1	0	0	0	0	0	0	0	0	1	201	Port B
1	0	0	0	0	0	0	0	1	0	202	Port C
1	0	0	0	0	0	0	0	1	1	203	Port Kontrol
1	0	1	0	0	0	0	0	0	0	280	Port A
1	0	1	0	0	0	0	0	0	1	281	Port B
1	0	1	0	0	0	0	0	1	0	282	Port C
1	0	1	0	0	0	0	0	1	1	283	Port Kontrol
1	0	1	0	0	0	0	1	0	0	284	Port A
1	0	1	0	0	0	0	1	0	1	285	Port B
1	0	1	0	0	0	0	1	1	0	286	Port C
1	0	1	0	0	0	0	1	1	1	287	Port Kontrol
1	0	1	1	1	0	0	0	0	0	2E0	Port A
1	0	1	1	1	0	0	0	0	1	2E1	Port B
1	0	1	1	1	0	0	0	1	0	2E2	Port C
1	0	1	1	1	0	0	0	1	1	2E3	Port Kontrol
1	1	0	0	0	0	0	0	0	0	300	Port A
1	1	0	0	0	0	0	0	0	1	301	Port B
1	1	0	0	0	0	0	0	1	0	302	Port C
1	1	0	0	0	0	0	0	1	1	303	Port Kontrol
1	1	1	0	0	0	0	0	0	0	380	Port A
1	1	1	0	0	0	0	0	0	1	381	Port B
1	1	1	0	0	0	0	0	1	0	382	Port C
1	1	1	0	0	0	0	0	1	1	383	Port Kontrol
1	1	1	0	0	0	0	1	0	0	384	Port A
1	1	1	0	0	0	0	1	0	1	385	Port B
1	1	1	0	0	0	0	1	1	0	386	Port C
1	1	1	0	0	0	0	1	1	1	387	Port Kontrol
1	1	1	1	1	0	0	0	0	0	3E0	Port A
1	1	1	1	1	0	0	0	0	1	3E1	Port B
1	1	1	1	1	0	0	0	1	0	3E2	Port C
1	1	1	1	1	0	0	0	1	1	3E3	Port Kontrol

Pada Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, digunakan alamat 300 s.d 303 hex. 2. Bus Kontrol

Bus kontrol yang digunakan sebanyak 2 bit yaitu pin Read (RD) dan pin Write (WR). Pin Read (RD) aktif low (logika "0") yang berarti agar PPI 8255 mengirimkan data atau informasi status ke data bus, memerintahkan CPU untuk membaca data dari input PPI 8255 sedangkan pin Write (WR) aktif low (logika "0") yang berarti CPU mengirimkan data ke PPI 8255. Dengan mengatur kondisi dari pin Read (RD) dan Write (WR) maka masing-masing port pada PPI 8255 dapat dikendalikan, apakah port tersebut digunakan sebagai input atau sebagai output. Pengaturan pin Read (RD) dan Write (WR) harus juga dikombinasikan dengan pin-pin kontrol lainnya yaitu pin A0, A1 dan CS.

3. Bus Data

Bus Data yang digunakan sebanyak 8 bit yaitu pin D0-D7 yang berfungsi sebagai kontrol word atau kata kontrol yang dapat mengendalikan grup A dan grup B. Pada pembuatan alat Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA ini menggunakan 24 bit output untuk mengendalikan driver motor. Untuk mode operasi yang digunakan adalah mode 0 dengan rincian bit sebagai berikut :

- 1. Motor DC 1 = 2 bit (output)
- 2. Motor DC 2 = 2 bit (output)
- 3. Motor Servo 3 = 8 bit (output)

- 4. Motor Servo 4 = 8 bit (output)
- 5. Motor servo 5 = 2 bit (output)
- 6. Motor Servo 6 = 2 bit (output)

Penggunaan port PPI 8255 pada aplikasi tangan robot, dapat dilihat pada tabel 3.4.

Bit PPI 8255	Port	Kontrol	Status	Ket
2 bit	Port C (PC4-PC5)		Output	Motor DC 1
2 bit	Port C (PC6-PC7)	Grup A	Output	Motor DC 2
8 bit	Port A		Output	Motor Servo 3
8 bit	Port B		Output	Motor Servo 4
2 bit	Port C (PC0-PC1)	Grup B	Output	Motor Servo 5
2 bit	Port C (PC2-PC3)		Output	Motor Servo 6

Tabel 3.4 Tabel penggunaan port PPI 8255

Karena pada pembuatan alat Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA ini menggunakan 24 bit yaitu port A, port B dan port C sebagai output maka mode kontrol input/output menggunakan kata kontrol 80 hex atau 128 desimal. Format kontrol input/output PPI 8255 dapat dilihat pada tebel 3.5.

Tabel 3.5 Format kontrol input/output PPI 8255

Kata Kontrol (Hex)	Kata Kontrol (Desimal)	Port A	Port B	Port C
80	128	OUT	OUT	OUT
82	130	OUT	IN	OUT
89	137	OUT	OUT	IN
8B	139	OUT	IN	IN
90	144	IN	OUT	OUT
92	146	IN	IN	OUT
99	153	IN	OUT	IN
9B	155	IN	IN	IN



Gambar 3.8 menunjukkan gambar rangkaian card PPI 8255.

Gambar 3.8 Skematik rangkaian card PPI 8255.

# 2. Konektor DB 25

Pada pembuatan alat aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, memerlukan konektor untuk menghubungkan antara pheripheral antarmuka dengan alat yang dikendalikan. Karena aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA ini mengeluarkan 24 bit output, maka konektor yang tepat digunakan adalah DB 25 yang mempunyai 25 pin. Adapun konfigurasi pin pada konektor DB 25 dapat dilihat pada tabel 3.6.

PIN DB25	Card PPI 8255	Driver Motor	
1	VCC +5V	VCC +5V	
2	Port A0		
3	Port A1		
4	Port A2		
5	Port A3	Motor Some 2	
6	Port A4	Motor Servo 5	
7	Port A5		
8	Port A6		
9	Port A7		
10	Port C0	Motor Somo 5	
11	Port C1	wow serve 3	
12	Port C2	Motor Somio 6	
13	Port C3	Motor Servo o	
14	Port B0		
15	Port B1		
16	Port B2		
17	Port B3	Motor Some 4	
18	Port B4	Motor Servo 4	
19	Port B5		
20	Port B6		
21	Port B7		
22	Port C4	Motor DC 1	
23	Port C5	MOIOT DC 1	
24	Port C6	Motor DC 2	
25	Port C7	MOIOI DC 2	

Tabel 3.6 Konfigurasi Pin koneksi DB25



Gambar 3.9 menunjukkan jalur rangkaian konektor DB 25.

Gambar 3.9 Jalur Rangkaian pada DB25.

#### 3. Driver Motor DC

Rangkaian driver motor DC berfungsi untuk mengendalikan motor DC 1 dan DC 2. Untuk rangkaian pengendali motor DC digunakan IC L293D sehingga pengontrolan motor DC menjadi lebih mudah. Dengan memberikan data sebesar +5V pada tiap com maka PC melalui PPI 8255, dapat dengan mudah mengendalikan perputaran motor. IC L293D digunakan untuk melayani motor DC1 dan motor DC 2 secara bersamaan. Motor DC 1 menggunakan motor DC gearbox +24V dengan torsi 8 Kg dan motor DC 2 menggunakan motor DC gearbox +24 V dengan torsi 25 Kg. Jika kedua motor diberikan tegangan +24 V maka rpm yang keluar akan terlalu cepat sehingga dari percobaan diperoleh tegangan sebesar +12 V DC sudah cukup untuk menggerakkan motor.

Berikut tabel 3.7 menunjukkan koneksi driver motor DC dengan card PPI 8255.

Motor	Driver	Pin	PPI	Kot	
DC	motor	DB25	8255	Ket	
1	com 1	22	PC4	Putar Kiri	
T	com 2	23	PC5	Putar Kanan	
2	com 3	24	PC6	Putar Kiri	
	com 4	25	PC7	Putar Kanan	

Tabel 3.7 Koneksi Driver Motor DC dengan PPI 8255

Fungsi S1 dan S2 adalah untuk membatasi pergerakan motor DC 1 agar pergerakannya tidak melebihi 360<sup>0</sup>. Fungsi S3 dan S4 adalah untuk membatasi pergerakan as motor DC 2 agar pergerakannya tidak melebihi 60<sup>0</sup>. Gambar 3.10 menunjukkan rangkaian driver motor DC 1 dan motor DC 2.



Gambar 3.10 Rangkaian Driver Motor DC 1 dan 2.

### 4. Driver Motor Servo Standar 180<sup>o</sup>

Rangkaian driver motor servo standar berfungsi untuk mengendalikan motor servo 3 dan motor servo 4. Prinsip dasar rangkaian driver motor servo standar adalah mengubah input rangkaian yang berupa data digital menjadi output rangkaian berupa pulsa PWM, dimana untuk menggerakkan motor servo 180<sup>0</sup> harus mengatur besar duty cycle pulsa yang dikeluarkan. Rangkaian ini menggunakan beberapa komponen utama yaitu DAC 0808, LM 353 dan IC 555.

DAC 0808 berfungsi sebagai converter dari input berupa data digital 8 bit (pin 5-12) menjadi output berupa arus (pin 4). Dengan mengatur pemberian input data digital (C0-FF hex) maka tegangan output (pin 4) dapat dikendalikan. Berikut adalah rumus perhitungan konversi.

LM 353 mempunyai 2 buah penguat. LM 353 dalam rangkaian driver motor servo berfungsi sebagai op-amp yaitu penguat inverting maupun non-inverting. Pada LM 353 tegangan output dari DAC 0808 akan mengalami 2 kali proses penguatan yaitu penguatan inverting dan non inverting. Tegangan tersebut dimasukkan ke rangkaian penguatan inverting melalui (pin 2) LM 353 untuk dikuatkan kemudian keluar melalui (pin 1). Kemudian tegangan tersebut masuk ke (pin 5) lalu mengalami proses penguatan non inverting kemudian tegangan tersebut keluar (pin7) berupa tegangan (+) positiv. Besar kecilnya tegangan yang keluar dari (pin 7) dapat tergantung pula oleh besar kecilnya hambatan pada potensio. Hal ini berlaku rumus :

$$V_0 = 1 + (R2/R1) V_{id} \dots (5)$$

Setelah tegangan keluar dari (pin 7) IC LM 353, seterusnya tegangan masuk ke bagian timer IC 555. Pada IC 555 tegangan input akan diolah yang kemudian pada keluaran output pin 3 IC 555 akan menghasilkan sinyal PWM (*Pulse Width Modulation*) dengan periode 20 mS (50Hz).

Perincian koneksi driver motor servo standar 180<sup>0</sup> dengan card PPI 8255 dapat dilihat pada tabel 3.8.

servo	Driver	Pin	PPI	
	motor	DB25	8255	ĸet
	A0	2	Port A0	
	A1	3	Port A1	
	A2	4	Port A2	
2	A3	5	Port A3	9 hit
5	A4	6	Port A4	o Dit
	A5	7	Port A5	
	A6		Port A6	
	A7	9	Port A7	
	A0	14	Port B0	
	A1	15	Port B1	
	A2	16	Port B2	
Λ	A3	17	Port B3	9 hit
4	A4	18	Port B4	o Dit
	A5	19	Port B5	
	A6	20	Port B6	
	A7	21	Port B7	

Tabel 3.8 Koneksi Driver Motor Servo 3 dan servo 4 dengan PPI 8255

Gambar 3.11 menunjukkan rangkaian driver motor servo.



Gambar 3.11 Rangkaian Driver Motor Servo 3 dan 4.

#### 5. Driver Motor Servo Continous 360<sup>o</sup>

Rangkaian driver motor servo continous berfungsi untuk mengendalikan motor servo 5 dan motor servo 6. Rangkaian driver ini menggunakan IC ULN 2003 sebagai pengganti rangkaian konvensional. Rangkaian ini dilengkapi dengan 2 buah potensio sebagai pengatur berapa besar nilai PWM yang diperlukan untuk menjalankan motor servo 5 dan servo 6. Motor servo 5 adalah motor servo continous yang dapat berputar 360<sup>0</sup>. Motor servo 5 hanya memerlukan 2 jenis PWM yaitu pulsa PWM dengan T ON = 1,3 mS (motor servo putar kanan) dan T ON = 1.7 mS (motor servo putar kiri). Untuk menggerakkan motor agar dapat berputar kanan maka pada com 1 diberikan tegangan high (+5V) maka relay 1 akan bekerja dan merubah kontak menjadi NO (Normal Open) sehingga tegangan akan mengalir ke IC timer 555 dan pada pin 3 akan menghasilkan PWM dengan T ON = 1,3 mS. Begitu sebaliknya apabila com 2 diberikan tegangan high (+5V) maka motor akan berputar kiri. Motor servo 6 adalah motor servo standar yang dapat bergerak sebesar 180<sup>0</sup>. Karena motor ini berfungsi sebagai penggerak jari pada tangan robot, maka motor servo 6 hanya membutuhkan pergerakan minimal dan maksimal yang berarti jari-jari tangan robot akan terbuka (180<sup>0</sup>) atau tertutup  $(0^{0})$ . Motor servo 6 hanya memerlukan 2 jenis PWM yaitu pulsa PWM dengan T ON = 2 mS (motor servo tertutup /  $0^0$ ) dan T ON = 1 mS (motor servo terbuka / 180<sup>°</sup>). Untuk menggerakkan motor servo 6 agar jari tangan robot dapat terbuka, maka com 2 diberikan tegangan high (+5V) sehingga relay 1 akan bekerja dan kontak akan berubah menjadi NO (Normal Open) dan tegangan akan masuk ke IC 555 sehingga pada (pin 3) IC 555 menghasilkan PWM dengan T ON sebesar 1 mS. Begitu sebaliknya apabila com 1 diberikan tegangan high (+5V) maka motor akan bergerak ke kiri sehingga motor pada posisi 180<sup>0</sup> yang artinya jari-jari tangan robot tertutup.

(Melalui percobaan, penentuan besarnya nilai potensio 1 dan potensio 2 disesuaikan dengan besarnya PWM yang dibutuhkan untuk menggerakkan motor)

Perincian koneksi driver motor servo 5 dan servo 6 dengan card PPI 8255 dapat dilihat pada tabel 3.9.

servo	Driver motor	Pin DB25	PPI 8255	Ket
		DDZJ	0255	
5	com 1	10	PC0	Putar Kiri
	com 2	11	PC1	Putar Kanan
6	com 1	12	PC2	Tutup
	com 2	13	PC3	Buka

Tabel 3.9 Koneksi Driver Motor Servo dengan PPI 8255

Gambar 3.12 menunjukkan rangkaian pengendali motor servo 5 dan motor servo 6.



Gambar 3.12. Rangkaian Driver Motor servo 5 dan 6.

# 3.4.3. Pembuatan program kendali

Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, menggunakan program Delphi 7.0 sebagai bahasa pemrograman. Untuk mengaktifkan PPI 8255 maka hal-hal yang harus diperhatikan adalah :

- a. Menentukan mode operasi yang digunakan dari tiap-tiap port (port
   A, port B dan port C). Setelah mode operasi ditetapkan maka
   berikutnya adalah menentukan *control word* dan base alamat
   portnya.
- b. Membuat prosedur umum atau inisialisasi.
- c. Setelah PPI 8255 di inisialisasi sesuai dengan mode yang diinginkan, proses selanjutnya adalah menulis atau membaca data ke port yang diinginkan sesuai dengan alamatnya portnya.

Flowchart program dapat dilihat pada lampiran 1. Gambar 3.13 menunjukkan tampilan modul kendali (PC).

😿 Form1	
Kendali Motor 1	Kendali Servo 6
KIRI STOP KANAN	TUTUP BUKA
Kendali Motor 2	Kendali Servo 5
TURUN STOP NAIK	KIRI
Kendali Servo 3 Center 225	Kendali Servo 4 Bytes Center 225
Naik Turun	Turun Naik
0 45 90 135 180	0 45 90 135 180
<u>R</u> eset PA <u>R</u> e	eset PB <u>R</u> eset PC
St	and by

Gambar 3.13. Tampilan modul kendali (PC).

### LISTING PROGRAM

```
unit Unit1;
interface
uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
   GroupBox1: TGroupBox;
   Button1: TButton;
   Button2: TButton;
   Button3: TButton;
    GroupBox2: TGroupBox;
   Button4: TButton;
    Button5: TButton;
   Button6: TButton;
    GroupBox3: TGroupBox;
   Label1: TLabel;
    Label3: TLabel;
   Label4: TLabel;
```

```
ScrollBar1: TScrollBar;
Edit1: TEdit;
Button7: TButton;
Button8: TButton;
Button9: TButton;
Button10: TButton;
Button12: TButton;
GroupBox4: TGroupBox;
Label5: TLabel;
Label6: TLabel;
ScrollBar2: TScrollBar;
Edit2: TEdit;
Button11: TButton;
Button13: TButton;
Button14: TButton;
Button15: TButton;
Button16: TButton;
GroupBox5: TGroupBox;
GroupBox6: TGroupBox;
Button17: TButton;
Button18: TButton;
Button19: TButton;
Button20: TButton;
Label2: TLabel;
Timer1: TTimer;
Button21: TButton;
Button22: TButton;
Button23: TButton;
Button24: TButton;
Button25: TButton;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure ScrollBar1Change(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure ScrollBar2Change(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure Button16Click(Sender: TObject);
procedure Button17Click(Sender: TObject);
procedure Button18Click(Sender: TObject);
```

```
procedure Button19Click(Sender: TObject);
    procedure Button20Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button21Click(Sender: TObject);
    procedure Button22Click(Sender: TObject);
    procedure Button23Click(Sender: TObject);
    procedure Button24Click(Sender: TObject);
    procedure Button25Click(Sender: TObject);
    procedure Button26Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  ol: byte;
  pl: byte;
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
begin
asm
  mov dx, $303
  mov al, $80
  out dx, al
end;
end;
{motor dc1---kanan}
procedure TForm1.Button1Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
  mov al, $20
  out dx, al
end;
end;
{motor dc1---kiri}
procedure TForm1.Button2Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
  mov al, $10
```

```
out dx, al
end;
end;
{motor dc1---stop}
procedure TForm1.Button3Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
 mov al, $00
  out dx, al
end;
end;
{motor dc2---naik}
procedure TForm1.Button4Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
 mov dx, $302
 mov al, $80
  out dx, al
end;
end;
{motor dc2---turun}
procedure TForm1.Button5Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
 mov al, $40
  out dx, al
end;
end;
{motor dc2---stop}
procedure TForm1.Button6Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
 mov al, $00
  out dx, al
end;
end;
{Servo 3 --- fungsi scrollbar1}
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
Edit1.Text := IntToStr(ScrollBar1.Position);
```

```
ol:= StrToInt(Edit1.Text);
asm
 mov dx, $300
 mov al, ol
  out dx, al
end;
end;
{tombol 0 derajat}
procedure TForm1.Button7Click(Sender: TObject);
begin
Scrollbar1.Position:=195;
end;
{tombol 45 derajat}
procedure TForm1.Button19Click(Sender: TObject);
begin
scrollbar1.Position:=207;
end;
{tombol 90 derajat}
procedure TForm1.Button8Click(Sender: TObject);
begin
scrollbar1.Position:=213;
end;
{tombol 135 derajat}
procedure TForm1.Button9Click(Sender: TObject);
begin
scrollbar1.Position:=229;
end;
{tombol 180 derajat}
procedure TForm1.Button10Click(Sender: TObject);
begin
scrollbar1.Position:=255;
end;
{Tombol Center--- posisi center dari scrollbar}
procedure TForm1.Button12Click(Sender: TObject);
begin
scrollbar1.Position:=223;
end;
{Servo 4 --- fungsi scrollbar2}
procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
Edit2.Text := IntToStr(ScrollBar2.Position);
pl:= StrToInt(Edit2.Text);
asm
 mov dx, $301
```

```
mov al, pl
  out dx, al
end;
end;
{0 derajat}
procedure TForm1.Button11Click(Sender: TObject);
begin
scrollbar2.Position:=195;
end;
{45 derajat}
procedure TForm1.Button20Click(Sender: TObject);
begin
scrollbar2.Position:=207;
end;
{90 derajat}
procedure TForm1.Button13Click(Sender: TObject);
begin
scrollbar2.Position:=213;
end;
{135 derajat}
procedure TForm1.Button14Click(Sender: TObject);
begin
scrollbar2.Position:=229;
end;
{180 derajat}
procedure TForm1.Button15Click(Sender: TObject);
begin
scrollbar2.Position:=255;
end;
{Center}
procedure TForm1.Button16Click(Sender: TObject);
begin
scrollbar2.Position:=223;
end;
{kendali servo5---Putar Kiri ~Timer}
procedure TForm1.Button17Click(Sender: TObject);
begin
Timer1.Enabled:=True;
asm
  mov dx, $302
 mov al, $01
  out dx, al
end;
end;
```

```
{kendali servo5---Putar Kanan ~Timer}
procedure TForm1.Button18Click(Sender: TObject);
begin
Timer1.Enabled:=True;
asm
 mov dx, $302
 mov al, $02
  out dx, al
end;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
asm
 mov dx, $302
 mov al, $00
  out dx, al
end;
end;
{Servo6---Tutup}
procedure TForm1.Button21Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
 mov dx, $302
 mov al, $04
  out dx, al
end;
end;
{Servo6---Buka}
procedure TForm1.Button22Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
 mov al, $08
  out dx, al
end;
end;
{reset Port A}
procedure TForm1.Button23Click(Sender: TObject);
begin
Timer1.Enabled:=False;
ScrollBar1.Position:=225;
asm
 mov dx, $300
 mov al, $00
  out dx, al
end;
end;
```

```
{Reset Port B}
procedure TForm1.Button24Click(Sender: TObject);
begin
Timer1.Enabled:=False;
ScrollBar2.Position:=225;
asm
 mov dx, $301
 mov al, $00
  out dx, al
end;
end;
{Reset Port C}
procedure TForm1.Button25Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
 mov dx, $302
 mov al, $00
 out dx, al
end;
end;
{stand by}
procedure TForm1.Button26Click(Sender: TObject);
begin
Timer1.Enabled:=False;
ScrollBar1.Position:=233;
asm
  mov dx, $300
  mov al, $E9
  out dx, al
end;
procedure TForm1.Button26Click(Sender: TObject);
begin
Timer1.Enabled:=False;
ScrollBar1.Position:=200;
asm
  mov dx, $301
 mov al, $C8
  out dx, al
end;
procedure TForm1.Button26Click(Sender: TObject);
begin
Timer1.Enabled:=False;
asm
  mov dx, $302
 mov al, $90
  out dx, al
```

end; end.

### 3.4.4. Deskripsi cara kerja alat

Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, merupakan alat yang berfungsi untuk memindahkan benda dari satu tempat ke tempat yang lain. Pemindahan benda tersebut dilakukan dengan mengendalikan pergerakan motor yang ada pada tangan robot. Pengendalian dilakukan pada modul kontrol yang ada pada tampilan monitor komputer yang telah di program agar dapat mengendalikan PPI 8255 yang kemudian memberikan input pada driver pengendali motor sehingga motor dapat bergerak. Gerakan motor dalam sistem alat aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, dijelaskan pada tabel 3.10.

<b>T</b> 1 1 <b>A</b> 1 A	D 1 '	•	1 .
Tabal 3 10	1 000 2 11	nci naraar	olzan motor
1 a D C I J I U	DESKIII	051 001201	акан шолог

Motor	Sudut Gerak	Arah Putaran
Motor DC 1	2600	Kanan
MOTOL DC 1	300	Kiri
Motor DC 2	600	Kanan
Motor DC 2	00	Kiri
Motor Some 2	1800	Kanan
Motor Servo 5	180°	Kiri
Mater Sama 4	1900	Kanan
Motor Servo 4	180°	Kiri
Matan Sama 5	2600	Kanan
Motor Servo 5	360°	Kiri
Matan Sama (	1900	Kanan/buka
wotor Servo 6	180°	Kiri/tutup

#### 1. Motor DC 1

Motor DC 1 akan bergerak ke kanan ketika tombol button "kanan" pada modul kontrol komputer ditekan. Inisialisasi data dikirim ke alamat \$303 untuk mengaktifkan kendali port pada PPI 8255 dan \$80 sebagai kata kontrol pengaktifan port A, port B dan port C berfungsi sebagai output. Output data pada alamat \$302 untuk pengiriman data ke port C. Untuk menggerakkan motor ke kanan maka com 2 pada driver motor diberikan logika "1" dan com 1 diberikan logika "0". Untuk itu data yang dikirim adalah \$20 dengan perincian data pada tabel 3.11.

Tabel 3.11 Data motor DC 1 putar kanan

	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0   0   1   0   0   0   0   0   \$20	0	0	1	0	0	0	0	0	\$20

Pada saat motor DC 1 berputar kanan sehingga limit switch S2 tertekan atau kondisi ON maka motor DC 1 tidak akan berputar atau diam walaupun tombol button "kanan" pada modul kendali pada PC ditekan. Hal ini disebabkan karena pada saat S2 tertekan maka relay 2 akan mendapatkan tegangan +5V sehingga kontak akan berubah menjadi NO (Normal Open). Pada saat itu kondisi masukan yang sebelumnya com 1 berlogika "0" dan com 2 berlogika "1" berubah menjadi com 1 berlogika "0" dan com 2 berlogika "0". Kondisi ini motor akan berhenti hingga com 1 diberikan logika "1" dan com 2 diberikan logika "0" maka motor akan berputar kiri. Caranya adalah

dengan menekan tombol button "kiri" pada modul kontrol PC. Untuk itu data yang dikirim adalah \$10 dengan perincian data pada tabel 3.12.

Tabel 3.12 Data motor DC 1 putar kiri

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	1	0	0	0	0	\$10

Motor DC 1 yang bergerak ke kanan atau ke kiri dapat dihentikan melalui modul kontrol pada komputer dengan menekan tombol button "stop". Data yang dikirim adalah \$00 dengan perincian data pada tabel 3.13 dibawah ini.

Tabel 3.13 Data motor DC 1 stop

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	0	0	0	0	0	\$00

# 2. Motor DC 2

Tuas pada motor DC 2 akan bergerak ke atas ketika tombol button "naik" pada modul kontrol komputer ditekan. Inisialisasi data dikirim ke alamat \$303 untuk mengaktifkan kendali port pada PPI 8255 dan \$80 sebagai kata kontrol pengaktifan port A, port B dan port C berfungsi sebagai output. Output data pada alamat \$302 untuk pengiriman data ke port C. Untuk menggerakkan tuas motor ke atas maka com 4 pada driver motor diberikan logika "1" dan com 3 diberikan logika "0". Untuk itu data yang dikirim adalah \$80 dengan perincian data pada tabel 3.14 dibawah ini.

Tabel 3.14 Data motor DC 2 naik

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
1	0	0	0	0	0	0	0	\$80

Pada saat tuas motor DC 2 bergerak keatas sehingga limit switch S3 tertekan atau kondisi ON maka motor DC 2 tidak akan berputar atau diam walaupun tombol button "naik" pada modul kendali pada PC ditekan. Hal ini disebabkan karena pada saat S3 tertekan maka relay 3 akan mendapatkan tegangan +5V sehingga kontak akan berubah menjadi NO (Normal Open). Pada saat itu kondisi masukan yang sebelumnya com 3 berlogika "0" dan com 4 berlogika "1" berubah menjadi com 3 berlogika "0" dan com 4 berlogika "0". Kondisi ini motor akan berhenti hingga com 3 diberikan logika "1" dan com 4 diberikan logika "0" maka motor akan berputar kiri. Caranya adalah dengan menekan tombol button "turun" pada modul kontrol PC. Untuk itu data yang dikirim adalah \$40 dengan perincian data pada tabel 3.15 dibawah ini.

Tabel 3.15 Data motor DC 2 turun

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	1	0	0	0	0	0	0	\$40

Motor DC 1 yang bergerak ke kanan atau ke kiri dapat dihentikan melalui modul kontrol pada komputer dengan menekan tombol button stop. Data yang dikirim adalah \$00 dengan perincian pada tabel 3.16.

Tabel 3.16 Data motor DC 2 stop

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	0	0	0	0	0	\$00

#### 3. Motor Servo 3

Untuk menggerakkan motor servo 3 adalah dengan menekan tombol button scroll bar maximal/minimal atau dengan menggeser posisi derajat pergerakan tombol scroll bar. Inisialisasi data dikirim ke alamat \$303 untuk mengaktifkan kendali port pada PPI 8255 dan \$80 sebagai kata kontrol pengaktifan port A, port B dan port C berfungsi sebagai output. Output data pada alamat \$300 untuk pengiriman data ke port A. Untuk menggerakkan tuas motor servo ke arah atas dengan cara menekan tombol scroll bar mimimal atau dengan menggeser posisi derajat pergerakan motor ke arah naik, dan sebaliknya untuk menggerakkan motor servo ke arah bawah dengan cara menekan tombol scroll bar maximal atau menggeser posisi derajat pergerakan motor ke arah turun. Karena motor servo hanya bergerak sebesar  $180^{\circ}$ maka pada posisi 0<sup>0</sup> dimulai dari pemberian data \$C0 hingga posisi 180<sup>0</sup> sebesar \$FF. Artinya adalah komputer akan memberikan data 8 bit sebanyak 64 jenis data yang berubah-ubah yaitu antara C0 sampai FF dengan mengatur scroll bar pada modul kontrol.

(Melalui percobaan, penentuan besarnya nilai data CO sampai FF berdasarkan pada penyesuaian PWM yang dibutuhkan untuk menggerakkan motor servo yaitu sebesar 180<sup>0</sup> dengan nilai T ON sebesar 1 - 2 mS sedangkan tegangan output op-amp yang dibutuhkan adalah 3,78 -5,0 V sehingga data yang diperlukan sebanyak 64 jenis data antara CO sampai FF )

Data yang dikirim dengan perincian pada tabel 3.17 dibawah ini.

NO	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	Hex
1	1	1	0	0	0	0	0	0	C0
2	1	1	0	0	0	0	0	1	C1
3	1	1	0	0	0	0	1	0	C2
4	1	1	0	0	0	0	1	1	C3
5	1	1	0	0	0	1	0	0	C4
6	1	1	0	0	0	1	0	1	C5
Ļ									
64	1	1	1	1	1	1	1	1	FF

Tabel 3.17 Data untuk motor servo 3

Dengan mengatur data yang dikeluarkan dari PC melalui PPI 8255 maka kita dapat mengatur derajat pergerakan motor servo. Data dari PPI 8255 akan masuk ke IC DAC 0808 (pin 5-12) dan menghasilkan output pada (pin 4). Tegangan output yang keluar dari (pin 4) kemudian masuk ke (pin 2) IC LM 353. Tegangan tersebut dimasukkan ke rangkaian penguat inverting melalui (pin 2) LM 353 untuk dikuatkan kemudian keluar melalui (pin 1). Kemudian tegangan tersebut masuk ke (pin 5) lalu mengalami proses penguatan non inverting kemudian tegangan tersebut keluar (pin7).

Setelah mengalami penguatan, tegangan keluar dari (pin 7) IC LM 353, seterusnya tegangan masuk ke IC 555. Pada IC 555 tegangan input akan diolah yang kemudian pada keluaran output pin 3 IC 555 akan menghasilkan sinyal PWM (*Pulse Width Modulation*) dengan periode 18 mS. Dengan mengendalikan tegangan yang masuk ke IC 555 maka kita dapat mengendalikan besar *duty cycle* yang menentukan pergerakan motor servo.

# 4. Motor Servo 4

Untuk menggerakkan motor servo 4 sama halnya dengan cara menggerakkan motor servo 3. Yang membedakan adalah motor servo 4 menggunakan port B (8 bit) pada PPI 8255 sedangkan motor servo 3 menggunakan port A (8 bit). Output data dikirim melalui alamat \$301 untuk pengiriman data ke port B.

### 5. Motor Servo 5

Motor servo 5 akan bergerak ke kanan ketika tombol button "kanan" pada modul kontrol komputer ditekan. Inisialisasi data dikirim ke alamat \$303 untuk mengaktifkan kendali port pada PPI 8255 dan \$80 sebagai kata kontrol pengaktifan port A, port B dan port C berfungsi sebagai output. Output data pada alamat \$302 untuk pengiriman data ke port C. Untuk menggerakkan motor ke kanan maka com 2 pada driver motor diberikan logika "1" dan com 1 diberikan logika "0".

Untuk itu data yang dikirim adalah \$02 dengan perincian data pada tabel 3.18 dibawah ini.

Tabel 3.18 Data motor servo 5 putar kanan

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	0	0	0	1	0	\$02

Pada saat motor servo 5 berputar kanan maka untuk membalik arah putaran menjadi arah kiri adalah dengan cara menekan tombol button "kiri" pada modul kontrol. Tombol button "kiri" ditekan maka data yang dikirim adalah \$01 dengan perincian pada tabel 3.19 dibawah ini.

Tabel 3.19 Data motor servo 5 putar kiri

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	0	0	0	0	1	\$01

## 6. Motor Servo 6

Motor servo 6 akan bergerak (jari tangan robot terbuka) ketika tombol button "buka" pada modul kontrol komputer ditekan. Inisialisasi data dikirim ke alamat \$303 untuk mengaktifkan kendali port pada PPI 8255 dan \$80 sebagai kata kontrol pengaktifan port A, port B dan port C berfungsi sebagai output. Output data pada alamat \$302 untuk pengiriman data ke port C. Untuk menggerakkan motor (jari tangan robot terbuka) maka com 2 pada driver motor diberikan logika "1" dan com 1 diberikan logika "0". Untuk itu data yang dikirim adalah \$08 dengan perincian data pada tabel 3.20 dibawah ini.

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex

1

0

0

0

\$08

Tabel 3.20 Data jari tangan robot terbuka

Pada saat motor servo 6 (jari tangan robot terbuka) maka untuk membuat agar jari tangan robot dapat tertutup adalah dengan cara menekan tombol button "tutup" pada modul kontrol. Tombol button "tutup" ditekan maka data yang dikirim adalah \$04 dengan perincian data pada tabel 3.21 dibawah ini.

Tabel 3.21 Data jari tangan robot tertutup

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Hex
0	0	0	0	0	1	0	0	\$04

### 3.5 **Prosedur Penelitian**

0

0

0

0

Pada Aplikasi pengendali tangan robot berbasis komputer menggunakan PPI 8255 melalui slot ISA, dilakukan pengujian dengan melakukan pengukuran dari tiap-tiap bagian penting rangkaian sehingga dapat dipastikan rangkaian dapat berfungsi sebagaimana yang diharapkan.

# 3.5.1. Pengukuran Pada Rangkaian Pengendali Motor DC

Pada pengujian rangkaian pengendali motor DC digunakan instrumen yaitu AVO meter digital untuk mengukur besar tegangan dan arus. Prosedur pengukuran adalah sebagai berikut :

 Menyiapkan catu daya yang diperlukan untuk rangkaian pengendali motor DC yaitu sebesar +5V.

- Menyiapkan sistem pengendalian dari PC dengan program kendali yang disiapkan untuk pengaktifan PPI 8255 dan melalui DB25 data siap dikirim ke rangkaian pengendali motor DC.
- 3. Menyiapkan rangkaian pengendali motor DC yang sudah dihubungkan dengan sumber pengiriman data, catu daya dan 2 buah motor DC.
- 4. Melakukan pengukuran.

# 3.5.2. Pengukuran Pada Rangkaian Pengendali Motor Servo 180<sup>0</sup>

Pada pengujian rangkaian pengendali motor servo standar digunakan instrumen yaitu AVO meter digital untuk mengukur besar tegangan dan arus dan osciloscope untuk mengetahui bentuk sinyal PWM yang keluar. Prosedur pengukuran adalah sebagai berikut :

- Menyiapkan catu daya yang diperlukan untuk rangkaian pengendali motor DC yaitu sebesar +5V, -12V dan +12V.
- Menyiapkan sistem pengendalian dari PC dengan program kendali yang disiapkan untuk pengaktifan PPI 8255 dan melalui DB25 data siap dikirim ke rangkaian pengendali motor servo.
- 3. Menyiapkan rangkaian pengendali motor servo yang sudah dihubungkan dengan sumber pengiriman data dan catu daya.
- 4. Menghubungkan probe oscilloscope ke output rangkaian dan grounding.
- Menghubungkan rangkaian pengendali ke motor servo berupa jalur data, Vcc +5V dan grounding.

6. Melakukan pengukuran dan pantauan bentuk gelombang.

## 3.5.3. Pengukuran Pada Rangkaian Pengendali Motor Servo 360<sup>0</sup>

Pada pengujian rangkaian pengendali motor DC digunakan instrumen yaitu AVO meter digital untuk mengukur besar tegangan dan arus. Prosedur pengukuran adalah sebagai berikut :

- Menyiapkan catu daya yang diperlukan untuk rangkaian pengendali motor DC yaitu sebesar +5V.
- Menyiapkan sistem pengendalian dari PC dengan program kendali yang disiapkan untuk pengaktifan PPI 8255 dan melalui DB25 data siap dikirim ke rangkaian pengendali motor servo continous.
- Menyiapkan rangkaian pengendali motor servo yang sudah dihubungkan dengan sumber pengiriman data, catu daya dan motor servo.
- 4. Melakukan pengukuran.

# 3.6 Teknik Analisis Data

Teknik analisis data pada penelitian ini dibagi menjadi 3 (tiga) bagian yaitu pengujian rangkaian pengendali motor DC, rangkaian pengendali motor servo standar dan rangkaian pengendali motor servo kontinus.

# 3.6.1. Pengukuran Rangkaian Pengendali Motor DC

Pada rangkaian pengendali motor DC menggunakan 4 bit data yaitu PC4 untuk com 1, PC5 untuk com 2, PC6 untuk com 3 dan PC7 untuk com 4. Pengujian program dapat dilihat pada tabel 3.22.

Port	Data Hexadesimal	Data Biner	Data Desimal
Port C4	\$10		
Port C5	\$20		
Port C6	\$40		
Port C7	\$80		

Tabel 3.22 Pengujian program kendali dari PC ke PPI 8255

Data yang dikirim dari PC ke PPI 8255 akan dilanjutkan menuju rangkaian pengendali motor DC melalui konektor DB25. Pengujian level logika diukur pada port C dengan pengiriman data \$00 dan \$FF dapat dilihat pada tabel 3.23.

Port C	Pin DB25	Logika	Hasil ukur	Satuan
Deat CO	10	rendah		Volt
Pon Co	10	tinggi		Volt
Dort C1	11	rendah		Volt
Folt CI	11	tinggi		Volt
Port C2	12	rendah		Volt
Folt C2	12	tinggi		Volt
Port C3	13	rendah		Volt
PontCS	15	tinggi		Volt
Port C4	$\gamma\gamma$	rendah		Volt
ron C4	22	tinggi		Volt
Port C5	23	rendah		Volt
1011 C.5	23	tinggi		Volt
Port C6	24	rendah		Volt
FOILCO	24	tinggi		Volt
Port C7	25	rendah		Volt
1011 C7	23	tinggi		Volt

Tabel 3.23 Pengukuran level logika output PPI 8255

			Motor	DC 1		
Hex	pin 2	pin 7	pin	ı 3	pin	16
	Volt	Volt	Volt	А	Volt	А
10 H						
20 H						

Tabel 3.24 Pengukuran rangkaian pengendali motor DC 1

Tabel 3.25 Pengukuran rangkaian pengendali motor DC 2

			Motor	DC 2		
Hex	pin 15	pin 10	pin	11	pin	14
	Volt	Volt	Volt	А	Volt	А
40 H						
80 H						

# 3.6.2. Pengukuran Rangkaian Pengendali Motor Servo 180<sup>0</sup>

\_\_\_\_

Pada rangkaian pengendali motor servo standar 180<sup>0</sup> menggunakan port 8 bit data yaitu A0-A7. Data yang dikirim dapat berubah-ubah sebanyak 64 jenis data dari \$C0 sampai \$FF. Pengujian ini berlaku untuk motor servo 3 dan servo 4. Pengujian program dapat dilihat pada tabel 3.27.

Tabel 3.26 Pengujian program kendali dari PC ke PPI 8255

Data Hexadesimal	Data Biner	Data Desimal
C0		
C6		
C9		
CC		
CF		
D2		
D5		
DA		

E0		
E5		
EA		
F0		
F7		
FF		

Pengujian konversi dari data digital menjadi tegangan analog pada output DAC 0808 yang kemudian dikuatkan dan dirubah jenis tegangannya pada IC LM 353 dan masuk menuju input IC 555, dapat dilihat pada Tabel 3.27.

\_\_\_\_

Tabel 3.27 Pengujian rangkaian pengendali motor servo 180<sup>0</sup>

	IC 0808	lm 353	lm 353	IC 555	<b>T</b> 0.11
Hex 8 bit A0-A7	Io	Vo1	Vo2	Pin 3	T ON PWM
-	mA	Volt	Volt	Volt	
C0					
C6					
C9					
CC					
CF					
D2					
D5					
DA					
E0					
E5					
EA					
F0					
F7					
FF					

Hex 8 bit A0-A7	T ON	Frekuensi	Derajat
C6			
CC			
D5			
E5			
FF			

Tabel 3.28 Pengujian output IC 555

# 3.6.3. Pengukuran Rangkaian Pengendali Motor Servo 360<sup>o</sup>

Pada rangkaian pengendali motor servo kontinus menggunakan 2 bit data yaitu PC0 untuk com 1 dan PC1 untuk com 2.

Tabel 3.29 Pengujian program kendali motor servo 360<sup>0</sup>

Port	Data Hexadesimal	Data Biner	Data Desimal
Port C0	\$01		
Port C1	\$02		

Tabel 3.30 Pengujian rangkaian pengendali motor servo 360<sup>0</sup>

Motor	Output Relay	Data Motor	com 1	com 2
Putar Kiri				
Stop				
Putar Kanan				