

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan per blok sistem kemudian secara keseluruhan. Pengujian per blok dilakukan untuk mempermudah analisis apabila alat tidak bekerja sesuai dengan perencanaan.

Adapun pengujian yang dilakukan sebagai berikut:

- Pengujian sensor *rotary encoder*
- Pengujian motor dan *driver* motor
- Pengujian keseluruhan sistem
 - Pengujian pengontrolan mesin tanpa beban
 - Pengujian pengontrolan mesin pemutar es dengan kecepatan yang berbeda- beda.

5.1 Pengujian Sensor *Rotary Encoder*

Pengujian ini bertujuan untuk mengetahui level tegangan sinyal listrik oleh rangkaian *optocoupler* dan untuk mengetahui keberhasilan rangkaian sensor *rotary encoder* dalam mendeteksi putaran. Pengujian respon rangkaian sensor *rotary encoder* menggunakan *oscilloscope* untuk melihat karakteristik perubahan tingkat logika dari rangkaian sensor. Motor arus searah atau *direct current* (DC) terhubung dengan piringan berlubang (32 lubang) diberi sumber 12 V agar berputar.

Berdasarkan *datasheet*, *Optical IC* GP1A25LC produksi *SHARP Corporation* masih dapat membaca putaran dengan frekuensi maksimum 16000 Hz. Dengan menggunakan nilai frekuensi maksimum tersebut dapat dihitung kecepatan putaran motor maksimum yang dapat dideteksi oleh rangkaian sensor rotari menggunakan persamaan berikut.

$$\check{S}_{maks} = \frac{F_{maks}}{32} \quad (5.1)$$

$$\check{S}_{maks} = \frac{64000Hz}{32}$$

$$\check{S}_{maks} = 2000 \text{ rpm}$$

Berdasarkan persamaan 5.1 diketahui bahwa kecepatan putaran motor maksimum yang dapat dideteksi oleh sensor rotari adalah 2000 rpm.

5.1.1 Peralatan Pengujian

Dalam pengujian sensor *rotary encoder* ini digunakan beberapa peralatan, diantaranya:

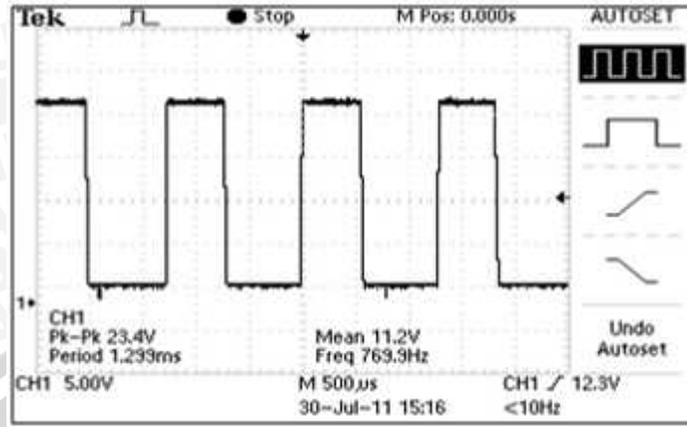
1. Motor arus searah atau *direct current* (DC)
2. Osiloskop dan *probe* osiloskop
3. Rangkaian pengkondisi sinyal *Rotary Encoder*
4. Sensor *Rotary Encoder*

5.1.2 Prosedur Pengujian

Prosedur pengujian dilakukan dengan menghubungkan *probe oscilloscope* pada rangkaian pengkondisi sinyal rotari. *Channel 1* pada osiloskop dihubungkan dengan pin *output* LM741 dan *channel 2* pada osiloskop dihubungkan dengan *ground* rangkaian. Selanjutnya motor DC yang terhubung dengan sensor *Rotary Encoder* diberi tegangan maksimal agar berputar.

5.1.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka pada osiloskop akan menunjukkan tampilan seperti dalam Gambar 5.1.



Gambar 5.1 Hasil pengujian *Rotary Encoder* menggunakan osiloskop

Berdasarkan Gambar 5.1 didapatkan *rise time* dan *fall time* pulsa saat kecepatan motor maksimal:

Peak to peak (Tegangan puncak ke puncak) = 23,4V

Period (Waktu yang diperlukan membentuk satu gelombang penuh) = 1,299ms

Frequency (Banyak gelombang yang terbentuk dalam satuan waktu) = 769,9Hz

Berikut akan ditampilkan pula hasil pengujian sensor *rotary encoder* dengan nilai *Counter* dan rpm (*radiant per minute*) yang ditunjukkan pada Tabel 5.1

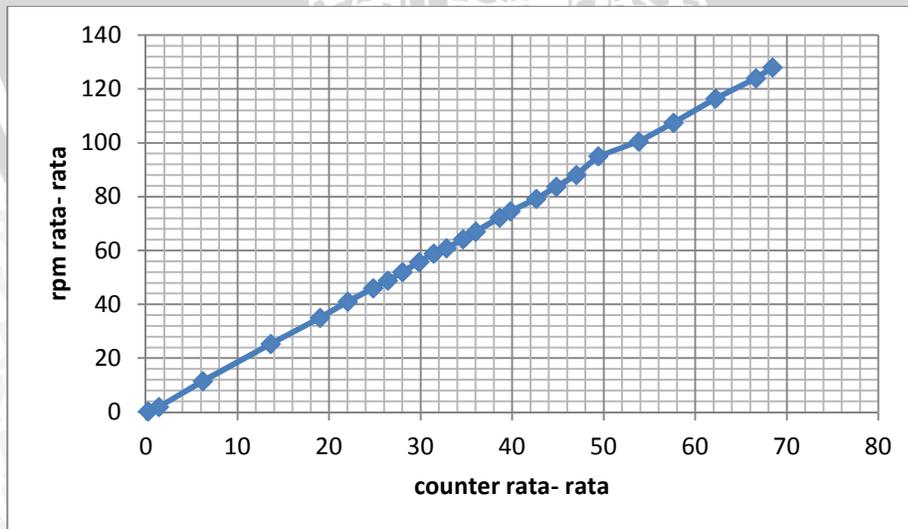
Tabel 5.1 Hasil Pengujian *Counter* dan Rpm

| rpm rata- rata | <i>counter</i> rata- rata |
|----------------|---------------------------|
| 0,2 | 0,2 |
| 1,8 | 1,4 |
| 11,4 | 6,2 |
| 25,2 | 13,6 |
| 35 | 19 |
| 41 | 22 |
| 46 | 24,8 |
| 48,8 | 26,4 |
| 52 | 28 |
| 55,6 | 29,8 |

Tabel 5.1 Hasil Pengujian *Counter* dan Rpm

| | |
|-------|------|
| 58,8 | 31,4 |
| 60,8 | 32,8 |
| 64,2 | 34,6 |
| 67 | 36 |
| 72,2 | 38,6 |
| 74,6 | 39,8 |
| 79,2 | 42,6 |
| 83,6 | 44,8 |
| 88 | 47 |
| 95 | 49,4 |
| 100,4 | 53,8 |
| 107,4 | 57,6 |
| 116,4 | 62,2 |
| 124 | 66,6 |
| 128 | 68,4 |

Berikut merupakan grafik dari pengujian *counter* dan rpm pada Gambar 5.2



Gambar 5.2 Grafik Hubungan Antara Rpm dan Counter

Pada Gambar 5.2 menunjukkan nilai *Counter* dapat mendeteksi kecepatan rata-rata (rpm) dari motor DC

5.2 Pengujian Motor dan Driver Motor

Tujuan dari pengujian ini adalah untuk mengetahui *output* dari *driver* motor apabila diberi *input* yang berbeda-beda.

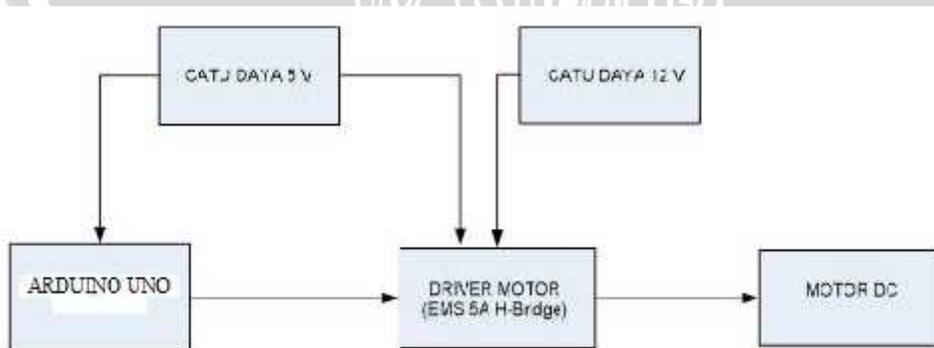
5.2.1 Peralatan yang digunakan

Dalam pengujian sensor *rotary encoder* ini digunakan beberapa peralatan, diantaranya:

1. Minimum sistem Arduino Uno
2. *Driver* motor
3. Motor DC
4. Catu daya 5V dan 12V

5.2.2 Prosedur pengujian

1. Merangkai peralatan seperti pada gambar 5.3
2. Mengaktifkan catu daya 5 volt dan 12 volt
3. Menghitung besar PWM dan tegangan dari mikrokontroler
4. Menghitung besar tegangan keluaran dari *driver* motor yang diperlukan untuk motor
5. Mencatat pergerakan motor DC
6. Mencatat nilai Rpm yang dihasilkan oleh motor

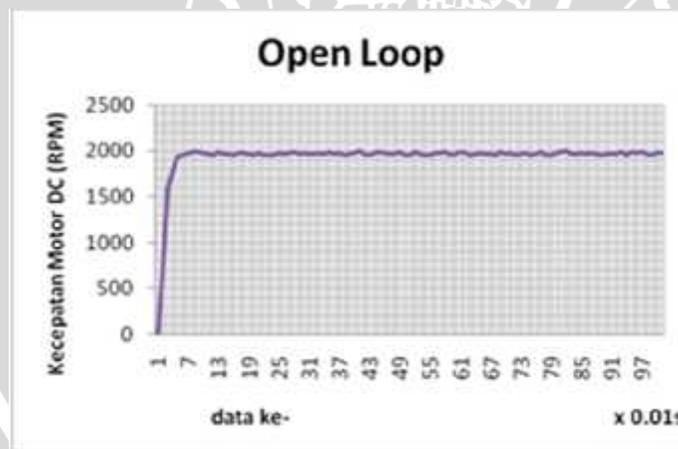


Gambar 5.3 Diagram Blok Pengujian *Driver* Motor DC

5.2.3 Hasil Pengujian

Hasil pengujian kecepatan motor dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan ditunjukkan Pada pengujian sistem secara *openloop* ini *plant* diberi PWM maksimal 255 dan tanpa beban untuk mengetahui bagaimana respon kecepatan motor DC. Pada system *openloop* tidak ada *feedback* yang dikembalikan ke dalam *plant* sehingga tidak ada *Error* selisih antara masukan dan keluaran. Di pengujian ini *plant* juga diberi tegangan maksimal agar nantinya bisa didapatkan permodelan dari *plant*. Nantinya *output* dari sensor kecepatan akan dikirim melalui serial ke PC dan ditampilkan grafiknya dalam *software visual basic* maupun *excel*. Kecenderungan dari system *openloop* adalah nilai *output* tidak stabil.

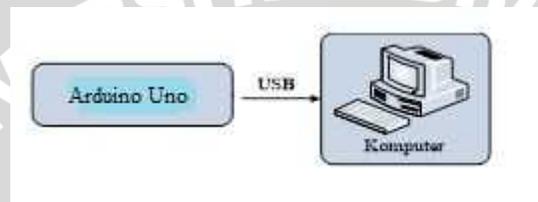
Pada Gambar 5.2 Nampak bahwa *output* system mencapai kondisi *steady state* pada kecepatan mendekati 2000 rpm dan detail datanya tepat adalah 1950 rpm. Waktu untuk mencapai kondisi *steady state* masih kurang cepat sehingga perlu adanya *controller* agar *plant* bisa mendapatkan respon yang lebih baik.



Tabel 5.4 Hasil Pengujian *Pulse Width Modulation* (PWM)

5.3 Pengujian Komunikasi ke PC

Pengujian ini dilakukan untuk mengetahui apakah rangkaian arduino dapat mengirimkan data melalui komunikasi. Untuk pengujian ini, arduino mengirimkan data pada terminal komputer menggunakan konfigurasi *baudrate* 9600 bps, 8 bit data, tanpa paritas dan 1 stop bit. Data yang dikirimkan merupakan paket data yang berisi pembacaan program. Prosedur pengujian dilakukan dengan menghubungkan arduino uno melalui usb, dan Komputer sesuai Gambar 5.4



Gambar 5.5 Diagram Pengujian Komunikasi arduino Uno

5.4 Pengujian Sistem Secara Keseluruhan

Pengujian ini bertujuan untuk mengetahui kemampuan kerja dari perangkat lunak LQR dan perangkat keras saat diintegrasikan secara bersama-sama. Pengujian dilakukan dengan merangkai rangkaian kemudian mengaktifkan semua alat yang telah dirakit, setelah itu mengamati kinerja dari sistem tersebut.

5.4.1 Peralatan Pengujian

1. Sensor *Rotary Encoder*
2. Variasi nilai Q dan rpm .
3. Mikrokontroler Arduino UNO
4. Rangkaian Driver Motor
5. Motor DC
6. USB
7. Perangkat *Writer*
8. Software *Arduino*

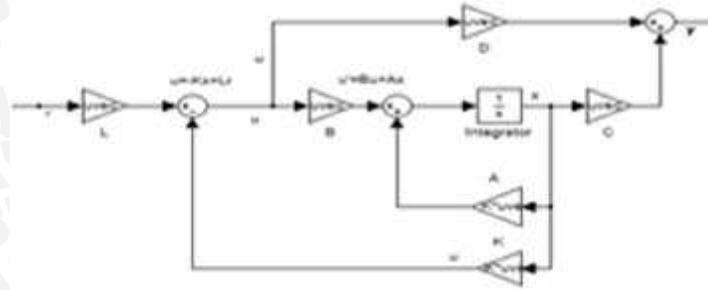
5.4.2 Prosedur Pengujian

Pada pengujian LQR sebagai kontroler, nilai bervariasi sedangkan nilai R ditetapkan satu. Dari variasi nilai Q ini didapatkan nilai K dan L yang bervariasi pula. Dari Tabel 4.1, 4.2 telah dicoba untuk diimplementasikan dalam mikrokontroler namun ada beberapa nilai Q yang bisa diimplementasikan. Nilai Q yang dibawah 1 yaitu antara 0,1-0,00001 nilai K dan L nya tidak dapat diimplementasikan pada mikrokontroler. Dan data perubahan nilai Q pada Tabel 5.2

Tabel 5.2 Variasi nilai Q

| no | nilai Q | Nilai K | Nilai L |
|----|---------|----------|---------|
| 1 | 0.0001 | 5.40E-05 | 0.926 |
| 2 | 0.001 | 5.40E-04 | 0.9265 |
| 3 | 0.01 | 0.0054 | 0.9313 |
| 4 | 0.1 | 0.0525 | 0.9784 |
| 5 | 1 | 0.4369 | 1.3628 |
| 6 | 2 | 0.7644 | 1.6904 |
| 7 | 3 | 1.0381 | 1.964 |
| 8 | 10 | 2.3691 | 3.295 |

Sesuai dengan judul kontrol optimal maka sebisa mungkin mengoptimalkan nilai Q yang ada dengan batasan-batasan yang dimiliki. Antara nilai 1-10 maka nilai optimal didapatkan pada nilai Q=1. Pada nilai tersebut LQR berjalan sesuai dengan *setpoint* yang diberikan. Namun hanya saja penerapan metode LQR ini tidak berjalan dengan baik. Logika dari kontrolernya sebagai berikut jika dilihat dari simulasi pada simulink:



Gambar 5.6 blok simulink LQR pada Matlab

Dari blok diagram tersebut dapat disederhanakan bentuk *state* nya agar lebih mudah dipahami, dan *state* diagram bloknnya adalah demikian:

$$u = Lr - Kx$$

$$u' = Bu + Ax$$

$$x = \int u'$$

nilai C = 1 dan D = 0.

Jika dijadikan logika dalam bahasa C pada mikrokontroler maka logikanya seperti dibawah ini:

```
Error=(L*Target)-(K*Rotary);
```

```
ErrorI= (int)Error * BB - (Rotary*AA);
```

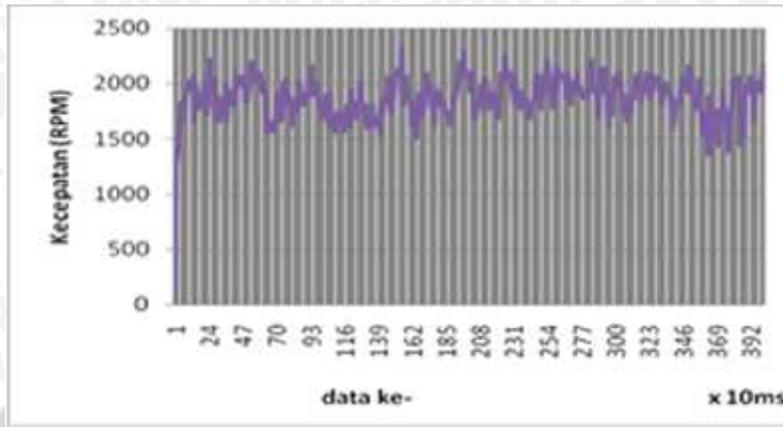
```
iError+=ErrorI;
```

```
U+=(int)iError * I; >> setelah blok integrator
```

5.4.2.1 Pengujian Dengan Variasi Kecepatan

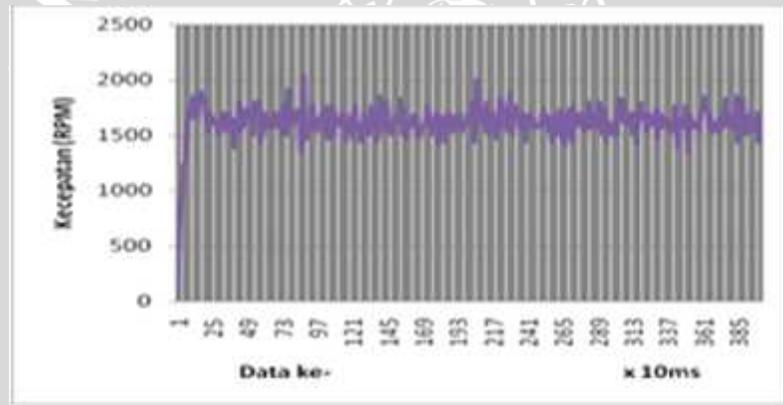
Pengujian ini dilakukan dengan memilih nilai Q=1 untuk semua kecepatan referensi. Dari hasil pengujian system dengan variasi kecepatan didapat hasil seperti pada Gambar 5.7 - 5.10.

Kecepatan 1200 rpm



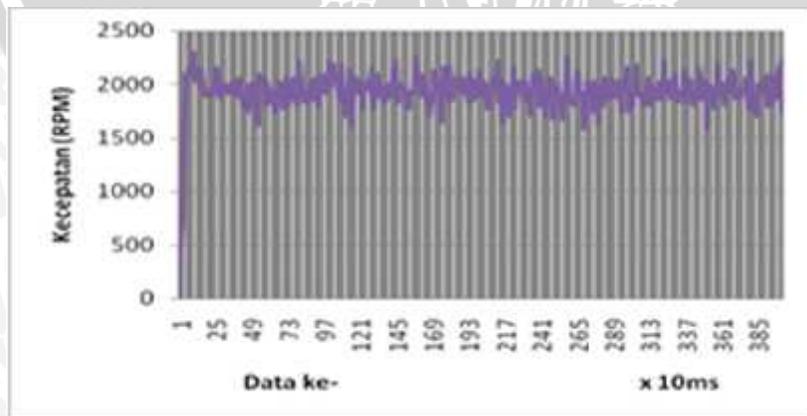
Gambar 5.7 Grafik respon transien dengan $Q=1$ dan kecepatan 1200 rpm

Kecepatan 1500 rpm



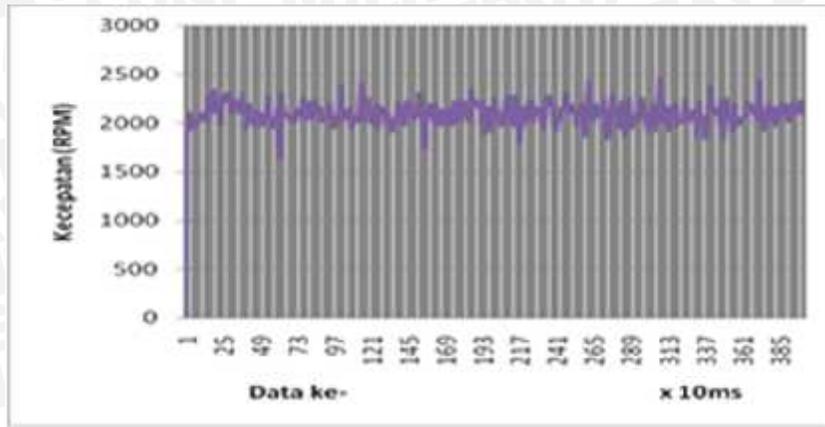
Gambar 5.8 Grafik respon transien dengan $Q=1$ dan kecepatan 1500 rpm

Kecepatan 1800 rpm



Gambar 5.9 Grafik respon transien dengan $Q=1$ dan kecepatan 1800 rpm

Kecepatan 1950 rpm



Gambar 5.10 Grafik respon transien dengan $Q=1$ dan kecepatan 1950 rpm

Berdasarkan Grafik pengujian *system* dengan menggunakan Linear Quadratic Regulator (LQR) dengan variasi kecepatan pada Gambar 4.14 sampai Gambar 4.17 terlihat bahwa pada kecepatan 1200 rpm, *controller* tidak mampu mencapai *setpoint* yang diinginkan. Hal ini bisa jadi disebabkan karena rendahnya nilai *setpoint* yang diberikan sehingga motor tidak mampu menjangkau referensi tersebut. Sedangkan pada kecepatan 1500, 1800 dan 1950 rpm, motor mampu berputar berdasarkan *setpoint* yang diberikan walaupun ada sedikit *Error*.

$$\% e = \frac{(r_l \ o \ - \ r_l \ s \ p)}{r_l \ s \ p} \times 100$$

Tabel % *Error* antara *setpoint* dengan output kecepatan motor dapat dilihat dalam Tabel berikut:

Tabel 5.3. % *Error* antara *setpoint* dengan output kecepatan

| Set point (RPM) | Rata-rata kecepatan (RPM) | % <i>Error</i> |
|-----------------|---------------------------|----------------|
| 1200 | 1867 | 55.6 |
| 1500 | 1613 | 7.5 |
| 1800 | 1936 | 7.5 |
| 1950 | 2090 | 7.2 |

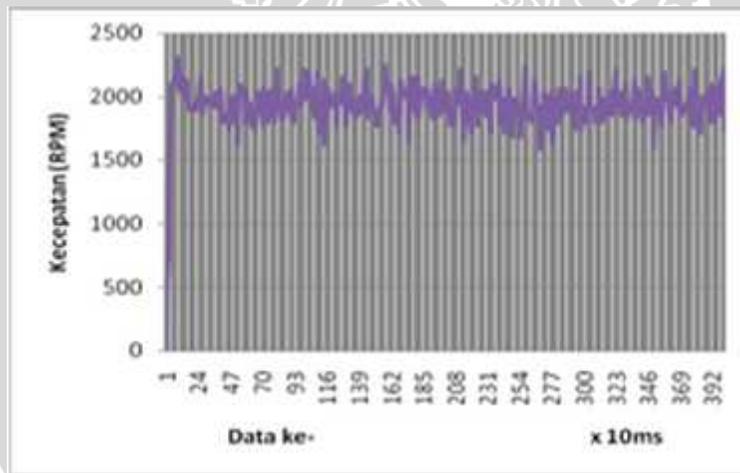
Dari hasil Tabel 4.2 dapat dilihat bahwa *error* terbesar pada set point kecepatan 1200 rpm. Sedangkan untuk kecepatan yang lain, *error*

tidak lebih dari 10%. Itu artinya bahwa dari beberapa pengujian yang dilakukan dengan variasi kecepatan bisa dikatakan berhasil karena perbandingan antara yang sesuai dengan target dan yang tidak maka lebih besar yang telah memenuhi target. Toleransi *error* dalam pengujian adalah 10 persen. Dimana hasil dari pengujian ini pada kecepatan 1500-1950 rpm, *error* yang didapatkan adalah rata-rata 7.4 %.

5.4.2.2 Dengan Variasi Nilai Q

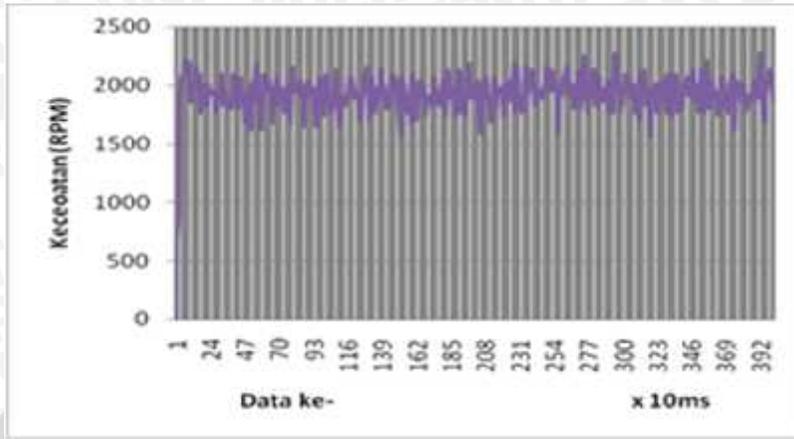
Pengujian ini dilakukan dengan memilih kecepatan 1800 rpm untuk semua nilai Q karena pada pengujian sebelumnya pada kecepatan 1800 rpm didapat nilai *Error* yang dibawah toleransi. Dari hasil pengujian *system* dengan variasi nilai Q dapat diperlihatkan pada Gambar 5.10 - 5.12. Dengan nilai Q yang digunakan adalah 1, 2, dan 3 karena jika digunakan nilai Q yang dibawah satu maka *system* tidak berjalan dan motorpun tidak dapat berputar.

Nilai Q = 1



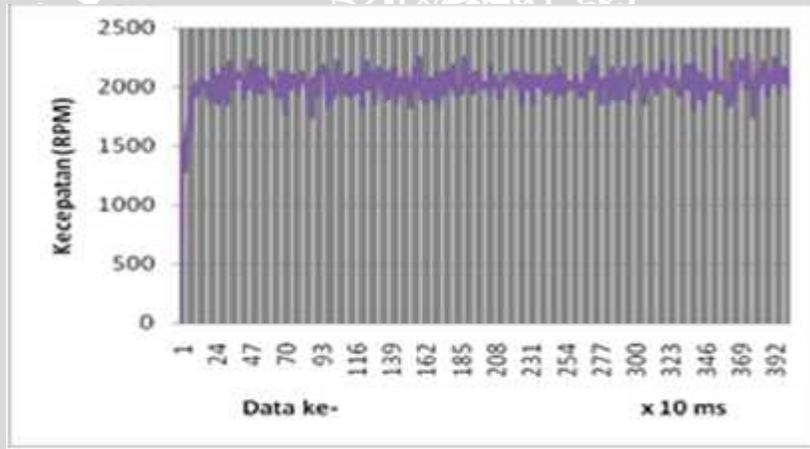
Gambar 5.11. Grafik respon transien dengan Q=1 dan kecepatan 1800 rpm

Nilai $Q = 2$



Gambar 5.12. Grafik respon transien dengan $Q=2$ dan kecepatan 1800 rpm

Nilai $Q = 3$



Gambar 5.13. Grafik respon transien dengan $Q=3$ dan kecepatan 1800 rpm

Dari grafik-grafik pengujian system dengan Linear Quadratic Regulator (LQR) untuk variasi nilai Q , terlihat bahwa kecepatan putar motor hampir sama namun sedikit saja perbedaan yang ditunjukkan dari variasi nilai Q . Nampak bahwa dengan diberikannya nilai Q yang semakin besar maka *overshoot* dari respon semakin berkurang akan tetapi tidak memenuhi *setpoint* yang ditentukan seperti gambar 5.12 ketika nilai $Q=3$. Nilai optimal dalam variasi nilai Q ini berada pada *range* $Q=1$ dan $Q=2$ dimana pada *range* antara nilai Q 1-2, *controller* mampu menjalankan motor agar berputar berdasarkan

setpoint yang diberikan. Dimana Tabel perbandingan % *Error* nilai *Q* dan output kecepatan dapat dilihat dalam Tabel 5.4

Tabel 5.4. % *Error* variasi nilai *Q* dengan rata-rata kecepatan motor

| Nilai <i>Q</i> | Rata-rata kecepatan (RPM) | % <i>Error</i> |
|----------------|---------------------------|----------------|
| 1 | 1936 | 7.6 |
| 2 | 1929 | 7.2 |
| 3 | 2021 | 12.3 |

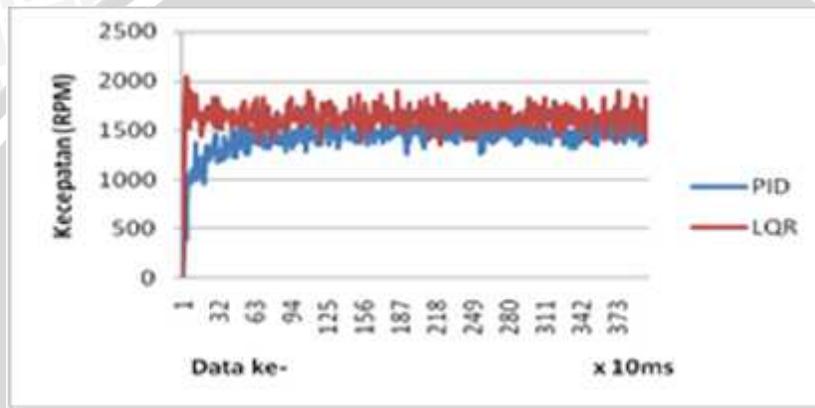
Pada nilai *Q*=3 didapat nilai *Error* yang melebihi toleransi sebesar 10% sedangkan pada *Q*=1 dan *Q*=2 nilai *Error* tidak lebih dari 10 % artinya *controller* dapat bekerja dengan baik. Dengan variasi nilai *Q* ini dapat diketahui bahwa pada plant ini *controller* bekerja dengan optimal pada range *Q* sebesar 1 dan 2. Sesuai dengan prinsip *control* optimal dimana dari range nilai *Q* yang ditentukan maka akan ada nilai terbaik yang akan menghasilkan respon yang diharapkan dan itu terjadi pada nilai *Q*=2 karena memiliki *Error* yang lebih kecil dari nilai *Q*=1.

Jika dilihat dari respon time pada variasi nilai *Q* ini maka pada *Q*=2 didapatkan nilai *overshoot* yang lebih rendah dari nilai *Q*=1. Agak sulit menentukan nilai respon yang lain dikarenakan gambar grafik yang kurang begitu baik. Hal ini disebabkan karena jumlah lubang pada *rotary encoder* yang hanya maksimal 40 lubang dan jika di *sampling* dengan waktu 10 *milisecond* maka akan terbaca antara 12-14 lubang namun lebih dominan pada jumlah 14 lubang. Semakin banyak lubang yang dimiliki piringan *rotary* maka semakin detail dan rinci pula pembacaan sensor kecepatan.

Dari semua pengujian pada implementasi *control optimal Linear Quadratic Regulator* ini telah dilakukan, sesuai dengan tujuan dan batasan masalah yang ditentukan dengan melihat jumlah dominan keberhasilan maka bisa dikatakan bahwa metode *Linear Quadratic Regulator* telah dapat diimplementasikan kedalam mikrocontroller dan mampu mengontrol kecepatan motor dc.

5.5 Dengan pembandingan PID

Jika dibandingkan dengan *controller* PID maka secara teori dengan simulasi pada matlab maka akan Nampak bahwa metode LQR ini jauh lebih baik dari metode PID karena memiliki respon yang lebih cepat dalam mencapai *setpoint* yang ditentukan. Dimana untuk menentukan nilai Kp, Ki dan Kd dengan menggunakan pendekatan metode Zigler Nichols seperti yang ditunjukkan pada buku ogata.



Gambar 5.14. Grafik respon antara PID dan LQR

Dapat diamati bahwa dalam pengujian ini respon LQR lebih baik dari pada respon PID. Pada PID sedikit lambat untuk mencapai *setpoint* yang ditentukan, selisihnya adalah $63 \times 10\text{ms} = 0.63$ detik. Jika dilihat *error* dari set point maka bisa dilihat pada Tabel 5.5.

Tabel 5.5. %Error *setpoint* antara PID dan LQR

| Kontroller | %Error |
|------------|--------|
| PID | 4.2 |
| LQR | 8.5 |

Dalam hal ketepatan pencapaian nilai *setpoint* dengan nilai *error* sebesar 4.2 % bisa dikatakan PID memang lebih baik namun dalam respon time maka PID jauh tertinggal dari LQR dengan selisih 0.6 detik.

Pada pengujian ini membuktikan bahwa dalam hal-hal tertentu dengan nilai-nilai tertentu maka LQR jauh lebih baik dari PID dalam hal respon time karena responnya yang cepat dalam mencapai *setpoint* namun *error* yang dihasilkan lebih besar dibandingkan PID.

Secara keseluruhan dari semua pengujian mulai dari variasi nilai kecepatan, variasi nilai Q dan perbandingan dengan *controller* PID, maka *Linear Quadratic Regulator* (LQR) bisa dikatakan mampu diimplementasikan kedalam mikrokontroler serta LQR mampu mengontrol kecepatan motor DC.

