

## DESAIN SISTEM PENGENDALIAN KECEPATAN MOTOR DC PADA TRAINER DIGIAC 1750 MENGGUNAKAN KONTROLER LQR BERBASIS LABVIEW

**Fihan Jaka Ananta**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : fihanananta@mhs.unesa.ac.id

**Endryansyah**

Dosen Teknik Elektro, Fakultas Teknik, Universitas negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : endryansyah@unesa.ac.id

### Abstrak

Dengan semakin pesatnya perkembangan teknologi pada saat ini, berbagai macam teknologi banyak bermunculan, salah satu hasil dari perkembangan teknologi adalah motor. Motor DC banyak digunakan sebelum motor AC dikenal yaitu untuk menghasilkan tenaga mekanik pada mesin-mesin industri. Tujuan pada penelitian ini yaitu mengimplementasikan pengendalian kecepatan motor DC pada Trainer Digiac 1750 agar dapat mengikuti setpoint. Pengendali yang digunakan adalah Linear Quadratic Regulator (LQR). Hasil penelitian menunjukkan nilai parameter LQR  $Q=10$ ,  $R=1$  memiliki respon yang baik dengan nilai parameter tersebut mampu menghasilkan respon sistem yang stabil pada setpoint 2,5V dengan nilai  $t_d$  sebesar 0,153 detik, nilai  $tr(5\%-95\%)$  sebesar 0,653 detik, nilai  $tr(10\%-90\%)$  sebesar 0,487 detik, nilai  $ts(5\%)$  sebesar 0,666 detik, nilai  $ts(2\%)$  sebesar 0,888 detik, nilai  $ts(0,5\%)$  sebesar 1,11 detik, nilai  $Ess$  sebesar 0,36%. Sedangkan, untuk pengujian dengan beban, nilai  $Q=90$  memiliki hasil yang lebih baik, dengan nilai  $Ess$  sebesar 1,52%.

**Kata Kunci:** Trainer Digiac 1750, Motor DC, LQR, LabView, *State-Space*.

### Abstract

With the rapid development of technology at this time, many kinds of technology have emerged, one of the results of technological developments is motor. DC motors are widely used before AC motors are known, namely to produce mechanical power in industrial. The purpose of this study is to implement speed control DC Motor on a Trainer Digiac 1750 to be able to follow the setpoint. The controller used is a Linear Quadratic Regulator (LQR). The results showed the value of the LQR parameter  $Q=10$ ,  $R=1$  had a good response with the parameter value capable of producing a stable system response at a 2.5V setpoint with a  $t_d$  of 0.153 seconds, a  $tr(5\%-95\%)$  of 0.653 seconds,  $tr(10\%-90\%)$  of 0.487 seconds,  $ts(5\%)$  of 0.666 seconds,  $ts(2\%)$  of 0.888 seconds,  $ts(0.5\%)$  of 1.11 seconds,  $Ess$  value is 0.36%. For testing with loads, the value of  $Q=90$  has better results, with  $Ess$  values of 1.52%.

**Keywords:** Trainer Digiac 1750, Motor DC, LQR, LabView, *State-Space*.

### PENDAHULUAN

Manusia menggunakan teknologi karena memiliki akal. Perkembangan teknologi terjadi karena seseorang menggunakan akalnya untuk menyelesaikan setiap masalah yang dihadapinya. Salah satu hasil dari perkembangan teknologi adalah motor. Perkembangan motor sangat penting dalam aplikasi industri maupun aplikasi lainnya seperti mesin bubut, lift, konveyor, kereta listrik, penerbangan, robotika dan lain-lain (J. Zhang., dkk, 2004). Motor DC banyak digunakan sebelum motor AC dikenal yaitu untuk menghasilkan tenaga mekanik pada mesin-mesin industry di pabrik, dan lain sebagainya. Dalam penelitian ini bertujuan untuk merancang pengendalian kecepatan motor DC berbasis

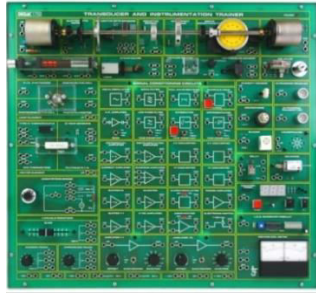
LabView. Dimana LabView digunakan sebagai program dan antarmuka dari grafik dan juga media pembuatan diagram blok dan menampilkan hasil pengendalian kecepatan motor DC menggunakan kontroler LQR (*Linear Quadratic Regulator*).

### KAJIAN PUSTAKA

#### Trainer Digiac 1750

Trainer Digiac 1750 adalah suatu alat *transducer*, instrumentasi dan kontrol yang digunakan oleh pelajar untuk praktikum mengenai *input* sensor, *output* aktuatur, sirkuit pengkondisi sinyal, dan perangkat tampilan. Trainer ini juga menyediakan buku pedoman kurikulum terperinci yang memberikan teori latar belakang kegiatan praktis, dan pertanyaan bagi pelajar (LJCREATE, 2010). Berikut Bentuk fisik Trainer Digiac seperti pada Gambar

1. dan spesifikasi Trainer Digiac 1750 pada Tabel 1. seperti dibawah ini.



**Gambar 1.** Trainer Digiac 1750  
(Sumber : LJCREATE, 2010)

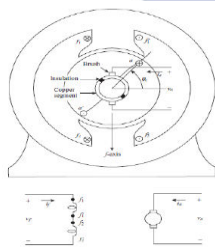
**Tabel 1.** Spesifikasi Trainer Digiac 1750

Power Supply	-5V, +5V 1A precision supply dan -12V, +12V 1A regulated supply.
Dimensi	500 x 450 x 150 mm
Supply Voltage	50-60Hz, 110-120V AC atau 220-240V AC
Packed Volume	0,109 m <sup>3</sup>
Packed Weight	14,9Kg

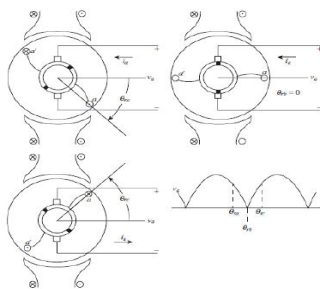
(Sumber: Ogata, 1985)

### Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor tersebut disebut stator, dan kumparan jangkar disebut rotor. Konstruksi dasar dari motor DC ditunjukkan pada Gambar 2. mesin dasar dua kutub ini dilengkapi dengan lilitan medan di kutub stator, sebuah koil rotor, dan komutator (Paul C. Krause Edisi Kedua, 2002).



**Gambar 2.** Dasar Motor DC 2 Kutub  
(Sumber : Paul C. Krause Edisi Kedua, 2002)



**Gambar 3.** Pergantian elemen motor DC  
(Sumber : Paul C. Krause Edisi Kedua, 2002)

Motor DC terdiri dari berbagai jenis, salah satunya adalah motor DC magnet permanen (Basuki, 2003). Pada tugas akhir ini motor yang digunakan adalah motor DC magnet permanen dengan tegangan maksimum 12 V. Penggunaan motor DC magnet permanen ini karena memiliki karakteristik torsi awal yang besar dan mudah dikendalikan. Ketika penggunaannya untuk memutar benda kerja, kecepatan putar motor belum tentu sesuai dengan yang diharapkan sehingga memerlukan pengaturan agar tegangan jangkar tetap terjaga stabil (Bambang Siswoyo, 2012).

### Tachogenerator

Sensor yang sering digunakan untuk sensor kecepatan angular adalah tachogenerator. Tachogenerator adalah sebuah generator kecil yang membangkitkan tegangan DC ataupun tegangan AC. Dari segi eksitasi tachogenerator dapat dibangkitkan dengan eksitasi dari luar atau imbas elektromagnet dari magnet permanen. Fungsi tachogenerator DC dapat membangkitkan tegangan DC yang langsung dapat menghasilkan informasi kecepatan, sensitivitas tachogenerator DC cukup baik terutama pada daerah kecepatan tinggi (Ibrahim., dkk, 2016).

### LabView

LabView adalah sebuah software pemrograman yang diproduksi oleh National Instruments dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, Java, matlab atau visual basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa LabVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text (National Instruments, 2018).

### NI Elvis

National Instruments Educational Laboratory Virtual Instrumentation Suite (NI ELVIS) adalah suatu module instrumentasi dan prototyping berbasis LabVIEW dan computer. NI ELVIS terdiri dari prototyping board, piranti data acquisition (DAQ) multifungsi dan instrument virtual berbasis LabVIEW (Muruganandhan, 2013).

### Analisis Karakteristik Orde Satu

Dari model matematis sebuah sistem, orde dari suatu sistem dapat dilihat dari besar pangkat variabel  $s$  (dalam transformasi Laplace). Suatu sistem dikatakan ber-orde satu jika fungsi alihnya mempunyai variabel  $s$  dengan pangkat tertinggi satu. Bentuk fisisnya bisa berupa rangkaian listrik RC, sistem termal, atau sistem lainnya (Ogata, 2010). Model sistem orde satu secara matematis dapat dituliskan sebagai berikut :

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1} \quad (1)$$

Keterangan :

$C(s)$  = Output sistem .

$R(s)$  = Input sistem.

$K$  = Gain Overall.

$\tau s + 1$  = Waktu yang dibutuhkan untuk mencapai 63,2% (detik) pada orde satu.

Untuk menentukan parameter  $K$  (*Gain Overall*) jika sistem linier, maka hubungan  $Y_{ss}$  dengan  $X_{ss}$  dapat dituliskan sebagai berikut :

$$K = \frac{Y_{ss}}{X_{ss}} \quad (2)$$

Keterangan :

$X_{ss}$  = *Setpoint*.

$Y_{ss}$  = Hasil Respon.

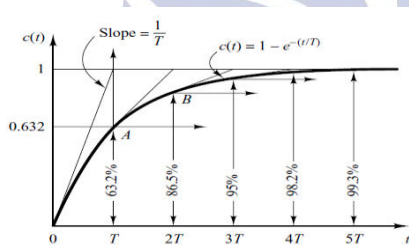
Untuk menentukan konstanta waktu ( $\tau$ ) dapat dihitung melalui respon keluaran sistem ketika mencapai 63,2% dari hasil akhirnya  $C(\tau)$  dapat dihitung sebagai berikut :

$$C(\tau) = 0.632 \times Y_{ss} \quad (3)$$

Keterangan :

$C(\tau)$  = Keluaran sistem ketika 63,2% dari hasil akhir.

Bentuk kurva tanggapan eksponensial orde satu ditunjukkan pada Gambar 4.



**Gambar 4.** Kurva Tanggapan Eksponensial Orde 1  
(Sumber : Ogata, 2010)

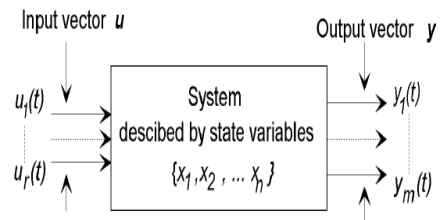
Rumus analitik untuk perhitungan parameter respon dari orde satu sebagai berikut :

1. Waktu tunda ( $td$ )  
 $td = \tau \ln 2$  (4)
2. Waktu naik ( $tr$ )  
  - a. Waktu naik (5%-95%)  
 $tr = \tau \ln 19$  (5)
  - b. Waktu naik (10%-90%)  
 $tr = \tau \ln 9$  (6)
3. Waktu Tunak ( $ts$ )  
  - a. Waktu tunak (0.5%)  
 $tr = 5\tau$  (7)
  - b. Waktu naik (2%)  
 $tr = 4\tau$  (8)
  - c. Waktu naik (5%)  
 $tr = 3\tau$  (9)
4. Persentasi *Error Steady State* ( $Ess$ )

$$Ess = \left| \frac{Y_{ss} - X_{ss}}{X_{ss}} \right| \times 100\% \quad (10)$$

## State-Space

*State-space* merupakan metode analisis untuk sebuah kendali yang kompleks. Metode digunakan untuk menganalisa sistem kendali dengan *input* banyak dan *output* banyak atau disebut MIMO (*Multiple Inputs and Multiple Outputs*). Pada model *state-space* keadaan internal ( $x$ ) dari sistem digunakan untuk memprediksi output  $y(t)$ . sehingga  $y$  tidak lagi bergantung hanya pada input sistem tetapi juga pada keadaan internal sistem. Sistem *input* dan *output* pada *state-space* ditunjukkan pada Gambar 5.



**Gambar 5.** Sistem *Input* dan *Output* pada *State-Space*  
(Sumber : Ogata, 2010)

Persamaan model *state-space* dituliskan sebagai berikut (Ogata, 2010) :

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (11)$$

Keterangan :

$A(t)$  = *Matrix state*

$B(t)$  = *Matrix input*

$C(t)$  = *Matrix output*

$D(t)$  = *Matrix direct-transmission*

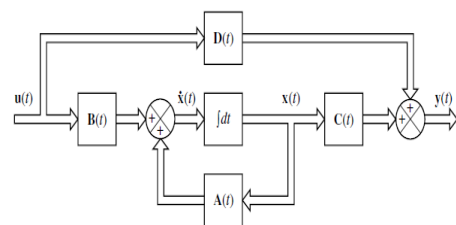
$u(t)$  = *Input system*

$x(t)$  = *State system*

$y(t)$  = *Output System*

$\dot{x}(t)$  = *Differential x*

Berikut adalah diagram blok *state-space* ditunjukkan pada Gambar 6.



**Gambar 6.** Diagram blok *State-Space*  
(Sumber : Ogata, 2010)

## Linear Quadratic Regulator (LQR)

Pada desain LQR dengan menggunakan persamaan matematis metode ini memiliki rumusan *cost function* sebagai berikut :

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (12)$$



Keterangan :

$J$  = Cost Function

$Q$  = Faktor pembobotan state (matrix semidefinite positive)

$R$  = Bobot faktor variabel kontrol (matrix definite positive)

$[\ ]^T$  = Transpose

$X$  = State System

$U$  = Input System

Untuk desain pengendali LQR, Langkah pertama adalah memilih matriks bobot nilai  $Q$  dan  $R$ . Masukan  $R$  lebih berat daripada state sementara ketika nilai bobot  $Q$  state lebih dari input. Kemudian umpan balik  $K$  dapat dihitung dan respon loop sistem tertutup dapat ditemukan dengan simulasi.

Rumus pengendali LQR sebagai berikut :

$$u = -Kx \quad (13)$$

Keterangan :

$K$  = Gain Feedback

Dari indeks persamaan state space system dan indeks performansi didapat nilai matriks  $K$  yang optimal untuk indeks performansi dengan rumus sebagai berikut :

$$K = R^{-1} B^T P \quad (14)$$

Keterangan :

$P$  = Variable Algebraic Riccati Equation (ARE)

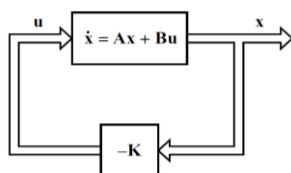
$[\ ]^{-1}$  = Invers

$B$  = Matrix State-Space

Persamaan Algebraic Riccati Equation (ARE) untuk mendapatkan nilai  $P$  sebagai berikut :

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (15)$$

Dalam perancangan teknik kontrol optimal Linear Quadratic Regulator (LQR), setelah matriks  $P$  diketahui maka nilai  $P$  tersebut disubstitusikan kedalam Persamaan 14 sehingga didapatkan nilai matriks optimal  $K$ . Diagram blok pengendali LQR ditunjukkan pada Gambar 7.



Gambar 7. Diagram blok Pengendali LQR (Sumber : Ogata, 2010)

Permasalahan yang ditemui dalam bidang kendali yaitu tidak hanya menyetabilkan system, tetapi bagaimana keluaran system mengikuti perubahan setpoint atau referensi yang ditentukan. Dalam hal ini jika diinginkan keluaran plant ( $x$ ) sama dengan input referensi maka perlu dirancang sistemnya menggunakan setpoint tracking (Khijwania, 2012). Dari persamaan 13 tanpa pengendali setpoint, maka didapatkan bentuk umum rumus kendali setpoint tracking sebagai berikut:

$$u = -Kx + Lr \quad (16)$$

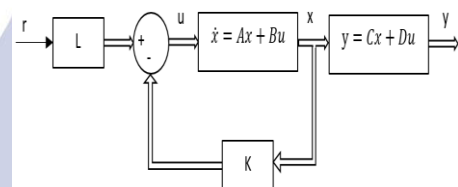
Keterangan :

$L$  = Gain setpoint tracking

Dengan rumus  $L$  sebagai berikut :

$$L = (-C(A - BK)^{-1}B)^{-1} \quad (17)$$

Diagram blok pengendali LQR ditunjukkan pada Gambar 8.



Gambar 8. Diagram blok Pengendali LQR dengan Setpoint Tracking (Sumber : Ogata, 2010)

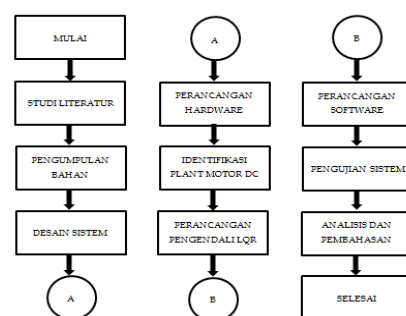
## METODE PENELITIAN

### Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Yaitu melibatkan perhitungan angka atau kuantifikasi data. Pengumpulan data dilakukan melalui pengukuran dengan menggunakan alat yang objektif dan baku (Sugiyono, 2015).

### Rancangan Penelitian

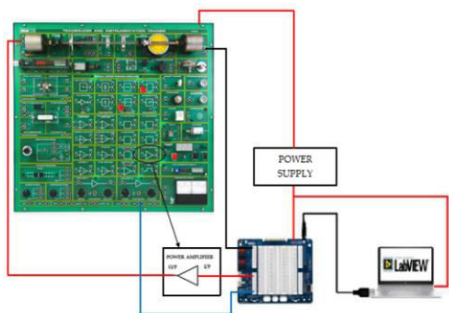
Tahapan perancangan penelitian ini secara garis besar dijelaskan dalam Gambar 9.



Gambar 9. Bagan Rancangan Penelitian (Sumber : Dokumen Pribadi, 2019)

### Rancang bangun Hardware

Desain sistem pengaturan kecepatan motor DC ditunjukkan pada Gambar 10.



**Gambar 10.** Skema Rangkaian Hardware Desain Sistem Pengaturan Kecepatan Motor DC pada Trainer Digiac 1750 Menggunakan Kontroler LQR (Sumber : Dokumen Pribadi, 2019)

Catu daya digunakan untuk menyediakan tegangan listrik ke semua rangkain dengan sumber listrik sebesar 220 VAC. Tachogenerator adalah sensor kecepatan yang akan menerima pembacaan data kecepatan yang berubah pada laju motor DC. NI Elvis akan menerima hasil pembacaan sensor yang telah dikirim. NI Elvis sebagai piranti akuisisi data perantara antara komputer dan aktuator pada proses pengendalian dengan tujuan membaca nilai input dan mengeset output pada digital atau analog. Driver motor DC berfungsi sebagai aktuator yang digunakan untuk menggerakkan motor DC (*plant*) pada kecepatan tertentu untuk menyesuaikan dengan setpoint nya.

Bentuk fisik dari sistem pengaturan kecepatan motor DC ditunjukkan pada Gambar 11.

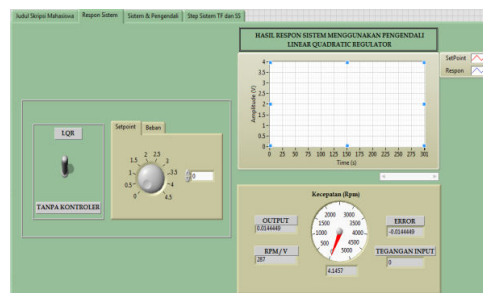


**Gambar 11.** Bentuk Fisik Sistem Pengaturan Kecepatan Motor DC pada Trainer Digiac 1750 (Sumber : Dokumen Pribadi, 2019)

### Rancang bangun Software

Software yang digunakan pada penelitian ini adalah LabView 2014. LabView 2014 berfungsi sebagai pusat pengendali *Linear Quadratic Regulator*. Berikut adalah

tampilan GUI LabView 2014 yang ditunjukkan pada Gambar 12.



**Gambar 12.** Tampilan Front Panel pada LabView 2014 (Sumber : Dokumen Pribadi, 2019)

### HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas proses kalibrasi sensor *tachogenerator*, pemodelan sistem, validasi model, perancangan kontroler LQR, dan pengujian respon sistem.

#### Kalibrasi Sensor Tachogenerator

Proses kalibrasi sensor *tachogenerator* dilakukan agar didapatkan nilai RPM motor DC dalam setiap satuan *volt* yang dikeluarkan oleh sensor *tachogenerator*. Alat bantu kalibrasi menggunakan komponen yang ada pada Trainer Digiac 1750 yaitu *Slotted Opto Sensor* ang ditampilkan pada 7-Segmen *Counter*. Tabel kalibrasi ditunjukkan pada Tabel 2.

**Tabel 2.** Data Kalibrasi Tachogenerator

No	Slotted Opto Sensor (RPM)	Tachogenerator (Volt)
1.	287	1
2.	332	1.25
3.	413	1.5
4.	575	2
5.	656	2.25
6.	735	2.5
7.	899	3
8.	982	3.25
9.	1063	3.5
10.	1226	4

(Sumber: Dokumen Pribadi, 2019)

Setelah mendapatkan data kalibrasi, dapat ditentukan nilai RPM setiap kenaikan tegangan pada *tachogenerator* yang telah dihitung yaitu rata-rata 287 RPM per 1V tegangan pada *tachogenerator*.

#### Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan metode pendekatan karakteristik sistem orde-1 yang bertujuan untuk mendapatkan model atau fungsi alih dari sistem. Fungsi alih tersebut digunakan sebagai acuan untuk merancang parameter dan simulasi untuk mengetahui respon sistem. Pada skripsi ini dilakukan pemodelan

dengan 5 setpoint berbeda, yaitu 1,25V, 2V, 2.5V, 3V dan 3,5V. Menggunakan persamaan 1 dengan setpoint 2.5V didapatkan fungsi alih sebagai berikut :

$$G(s) = \frac{0,879}{0,246s + 1}$$

Keterangan:

$G(s)$  = Fungsi alih pemodelan sistem

### Validasi Sistem

Validasi sistem dilakukan untuk mendapatkan model atau fungsi alih dari sistem yang memiliki respon dengan tingkat error dan untuk menentukan apakah fungsi alih yang telah didapatkan benar dan dapat digunakan sebagai model dari *plant* riil. Fungsi alih tiap *setpoint* beserta error rata-rata ditunjukkan pada Tabel 3.

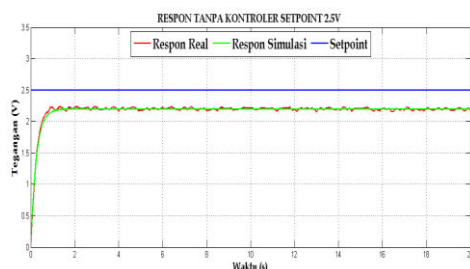
**Tabel 3.** Fungsi Alih tiap *Setpoint*

No	Setpoint	Fungsi Alih	Error Rerata
1.	1,25 V	$G(s) = \frac{0,744}{0,424s + 1}$	0,029
2.	2 V	$G(s) = \frac{0,851}{0,915s + 1}$	0,038
3.	2,5 V	$G(s) = \frac{0,879}{0,246s + 1}$	0,018
4.	3 V	$G(s) = \frac{0,828}{0,317s + 1}$	0,032
5.	3,5 V	$G(s) = \frac{0,846}{0,313s + 1}$	0,037

(Sumber: Dokumen Pribadi, 2019)

Dari data fungsi alih yang diperoleh dipilih fungsi alih dengan *error* rerata terkecil yaitu 0,018. Fungsi alih dengan *setpoint* 2,5 V dipilih sebagai acuan untuk mendesain parameter pada kondisi *open loop* riil dan simulasi.

Perbandingan respon dari fungsi alih dengan respon sistem riil dan simulasi ditunjukkan pada Gambar 13.



**Gambar 13.** Perbandingan Respon Sistem Riil dan Simulasi

(Sumber : Dokumen Pribadi, 2019)

Sebelum merancang parameter dari pengendali LQR, dihitung parameter dari respon sistem riil *plant* secara *open loop* pada *setpoint* 2,5 V dengan nilai  $\tau = 0,246$  detik. Parameter respon *plant* riil dengan *setpoint* 2,5 V secara *open loop* ditunjukkan pada Tabel 4.

**Tabel 4.** Parameter Respon Riil dengan *Setpoint* 2,5V

Parameter Respon	Nilai
Waktu Tunda (td)	0,17 detik
Waktu Naik (tr = 5%-95%)	0,724 detik
Waktu Naik (tr = 10%-90%)	0,54 detik
Waktu Tunak (ts = 5%)	0,738 detik
Waktu Tunak (ts = 2%)	0,984 detik
Waktu Tunak (ts = 0,5%)	1,23 detik
Error SteadyState (Ess)	12,04%
Maximum Overshoot (MO)	-

(Sumber: Dokumen Pribadi, 2019)

### Perancangan Pengendali LQR

Perancangan pengendali LQR dilakukan untuk mendapatkan parameter dari pengendali LQR yaitu K (LQR) dan L (setpoint tracking (feed forward)). Berikut adalah cara untuk mendapatkan nilai parameter untuk pengendali LQR :

### Mengubah fungsi alih ke bentuk *state-space*

Untuk merubah bentuk fungsi alih ke bentuk *state-space* salah satu cara yang mudah untuk mengubah bentuk dari fungsi alih adalah dengan menggunakan *mathscript* yang ada pada program LabVIEW. Menggunakan persamaan 1 dengan *setpoint* 2,5 V didapatkan fungsi alih sebagai berikut :

$$G(s) = \frac{0,879}{0,246s + 1}$$

Menggunakan fungsi alih diatas, maka didapatkan nilai num dan den sebagai berikut :

$$\begin{aligned} \text{num} &= 0,879 \\ \text{den} &= [0,246 \ 1] \end{aligned}$$

Keterangan :

num = Pembilang dari fungsi alih

den = Penyebut dari fungsi alih

Dengan menuliskan *syntax* [A,B,C,D] = *tf\_to\_ss*(num,den), sehingga didapatkan parameter *matrix state-space* sebagai berikut :

$$\begin{aligned} A &= -4,0650 \\ B &= 1 \\ C &= 3,5732 \\ D &= 0 \end{aligned}$$



### Perhitungan parameter untuk pengendali LQR

Berikut perhitungan untuk mencari nilai  $K$  (*feed forward*) dan  $L$  (*Setpoint Tracking*). Diketahui nilai *matrix*  $Q$  dan  $R$  :

$$Q = 1, 10, \text{ dan } 90$$

$$R = 1 \text{ (konstan)}$$

Dengan menggunakan nilai state-space dan nilai *matrix*  $Q$  dan  $R$ . Selanjutnya perhitungan untuk mencari nilai  $P$  dengan menyelesaikan Persamaan 15. Sehingga didapatkan nilai  $P$  (dengan nilai  $Q=1$ ) sebesar :

$$P_1 = 0,12$$

$$P_2 = -8,25$$

Setelah itu menyubstitusikan nilai  $P$  ke Persamaan 14 (nilai  $P$  yang digunakan adalah 0,12, karena syarat nilai  $P$  adalah  $P$  memiliki nilai *semi-definite positive*), sehingga didapatkan hasil sebagai berikut :

$$K = 0,12$$

Selanjutnya mencari nilai  $L$  dengan menggunakan Persamaan 17, sebagai berikut :

$$L = 1,1715$$

Nilai  $P$ ,  $K$ , dan  $L$  pada setiap perubahan  $Q$  ditunjukkan pada Tabel 5.

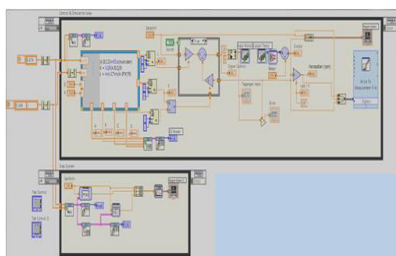
**Tabel 5.** Nilai  $P$ ,  $K$ , dan  $L$  pada setiap perubahan Nilai  $Q$

$Q$	$R$	$P$	$K$	$L$
1	1	0,1211	0,1211	1,1715
10	1	1,0851	1,0851	1,4413
90 (Untuk Pengujian Beban)	1	6,2560	6,2560	2,8884

(Sumber: Dokumen Pribadi, 2019)

### Implementasi Pengendali LQR

Setelah mendapatkan nilai  $K$  dan  $L$ , selanjutnya adalah mengimplementasikan pengendali LQR pada LabView 2014. Diagram Blok pada Labview 2014 dengan pengendali LQR ditunjukkan Gambar 14.



**Gambar 14.** Diagram Blok pada LabView 2014 dengan Pengendali LQR  
(Sumber : Dokumen Pribadi, 2019)

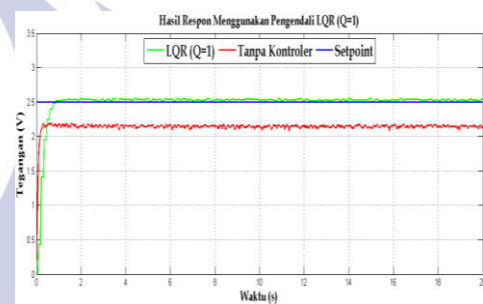
Untuk mengetahui perbedaan respon sistem sebelum dan sesudah ditambahkan pengendali LQR, maka dilakukan beberapa pengujian sebagai berikut :

### Pengujian Pengendali LQR dengan nilai $Q = 1$

Pengujian pengendali LQR dengan nilai  $Q = 1$  dilakukan sebanyak dua kali percobaan yaitu pengujian dengan setpoint tetap dan pengujian dengan beban. Setelah mendapatkan hasil dari masing-masing percobaan dapat dilanjutkan dengan menghitung parameter respon. Berikut adalah hasil pengujian dengan nilai  $Q = 1$  :

#### 1. Pengujian dengan Setpoint Tetap

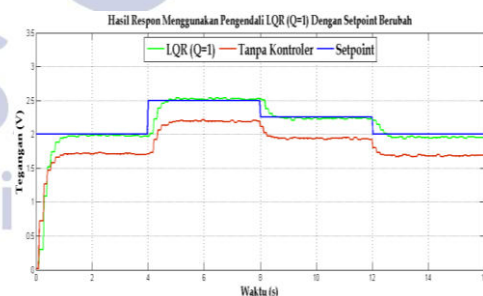
Dilakukan pengujian dengan setpoint tetap yaitu dengan setpoint 2,5V selama 20 detik. Hasil Respon menggunakan pengendali LQR dengan nilai  $Q = 1$  ditunjukkan pada Gambar 15.



**Gambar 15.** Hasil Respon Menggunakan Pengendali LQR dengan Nilai  $Q = 1$   
(Sumber : Dokumen Pribadi, 2019)

#### 2. Pengujian dengan Setpoint Berubah

Dilakukan pengujian dengan memberikan setpoint bervariasi yaitu 2 V, 2,5 V, dan 2,25V selama 16 detik. Berikut adalah hasil respon menggunakan pengendali LQR dengan nilai  $Q = 1$  dengan setpoint berubah ditunjukkan pada Gambar 16.



**Gambar 16.** Hasil Respon Menggunakan Pengendali LQR Nilai  $Q = 1$  dengan Setpoint Berubah  
(Sumber : Dokumen Pribadi, 2019)

#### 3. Analisis Respon Sistem

Respon Sistem akan mengalami perubahan, setelah ditambahkan pengendali LQR. Maka dilakukanlah perhitungan parameter respon sistem dengan nilai setpoint 2,5 V, untuk mengetahui perubahan tersebut. Parameter yang dihitung adalah  $t_d$  (Time Delay),  $t_r$  (Rise Time),  $t_s$  (Settling Time),  $E_{ss}$  (Error Steady State) dengan nilai  $\tau = 0,322$  detik.

Parameter respon plant dengan setpoint 2,5 V menggunakan pengendali LQR tanpa beban ditunjukkan pada Tabel 6.

**Tabel 6.** Parameter Respon Riil Menggunakan Pengendali LQR ( $Q = 1$ )

Parameter Respon	Nilai
Waktu Tunda (td)	0,223 detik
Waktu Naik ( $tr = 5\%-95\%$ )	0,948 detik
Waktu Naik ( $tr = 10\%-90\%$ )	0,707 detik
Waktu Tunak ( $ts = 5\%$ )	0,966 detik
Waktu Tunak ( $ts = 2\%$ )	1,288 detik
Waktu Tunak ( $ts = 0,5\%$ )	1,61 detik
Error SteadyState (Ess)	1,09%
Maximum Overshoot (MO)	-

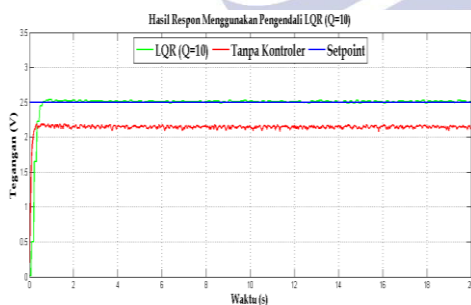
(Sumber: Dokumen Pribadi, 2019)

#### Pengujian Pengendali LQR dengan nilai $Q = 10$

Pengujian pengendali LQR dengan nilai  $Q = 10$  dilakukan sebanyak tiga kali percobaan yaitu pengujian dengan setpoint tetap dan pengujian dengan beban. Setelah mendapatkan hasil dari masing-masing percobaan dapat dilanjutkan dengan menghitung parameter respon. Berikut adalah hasil pengujian dengan nilai  $Q = 10$ :

##### 1. Pengujian dengan Setpoint Tetap

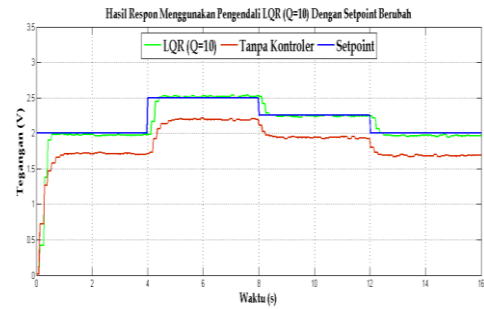
Dilakukan pengujian dengan setpoint tetap yaitu dengan setpoint 2,5V selama 20 detik. Hasil Respon menggunakan pengendali LQR dengan nilai  $Q = 10$  ditunjukkan pada Gambar 17.



**Gambar 17.** Hasil Respon Menggunakan Pengendali LQR dengan Nilai  $Q = 10$   
(Sumber : Dokumen Pribadi, 2019)

##### 2. Pengujian dengan Setpoint Berubah

Dilakukan pengujian dengan memberikan setpoint bervariasi yaitu 2 V, 2,5 V, dan 2,25V selama 16 detik. Berikut adalah hasil respon menggunakan pengendali LQR dengan nilai  $Q = 10$  dengan setpoint berubah ditunjukkan pada Gambar 18.



**Gambar 18.** Hasil Respon Menggunakan Pengendali LQR Nilai  $Q = 10$  dengan Setpoint Berubah  
(Sumber : Dokumen Pribadi, 2019)

##### 3. Analisis Respon Sistem

Respon Sistem akan mengalami perubahan, setelah ditambahkan pengendali LQR. Maka dilakukanlah perhitungan parameter respon sistem dengan nilai setpoint 2,5 V, untuk mengetahui perubahan tersebut. Parameter yang dihitung adalah td (Time Delay), tr (Rise Time), ts (Settling Time), Ess (Error Steady State) dengan nilai  $\tau = 0,222$  detik. Parameter respon plant dengan setpoint 2,5 V menggunakan pengendali LQR tanpa beban ditunjukkan pada Tabel 7.

**Tabel 7.** Parameter Respon Riil Menggunakan Pengendali LQR ( $Q = 10$ )

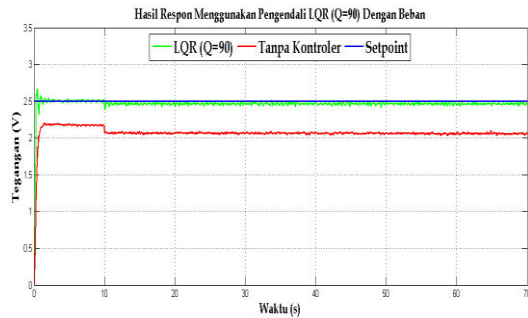
Parameter Respon	Nilai
Waktu Tunda (td)	0,153 detik
Waktu Naik ( $tr = 5\%-95\%$ )	0,653 detik
Waktu Naik ( $tr = 10\%-90\%$ )	0,487 detik
Waktu Tunak ( $ts = 5\%$ )	0,666 detik
Waktu Tunak ( $ts = 2\%$ )	0,888 detik
Waktu Tunak ( $ts = 0,5\%$ )	1,11 detik
Error SteadyState (Ess)	0,36%
Maximum Overshoot (MO)	-

(Sumber: Dokumen Pribadi, 2019)

#### Pengujian Beban Pengendali LQR dengan nilai $Q = 90$

Pada saat melakukan pengujian menggunakan beban yaitu dengan menggunakan beban pada program yang berguna untuk mengganggu kecepatan motor DC. Pada pengujian beban menggunakan nilai  $Q = 90$ , agar respon bisa mengejar setpoint yang telah ditetapkan. Berikut adalah hasil respon menggunakan pengendali LQR dengan nilai  $Q = 90$  dengan beban ditunjukkan pada Gambar 19.





**Gambar 19.** Hasil Respon Menggunakan Pengendali LQR dengan Nilai  $Q = 90$  dengan Beban (Sumber : Dokumen Pribadi, 2019)

Pada percobaan plant menggunakan pengendali LQR ( $Q = 90$ ) dengan beban didapatkan hasil yang ditunjukkan pada Gambar 19. Kemudian, perlu dilakukan perhitungan mencari Ess saat plant diberi beban dengan nilai ( $Q = 90$ ). Ess (Error SteadyState) dengan beban menggunakan Persamaan 10 diperoleh nilai sebesar 1,52%.

## PENUTUP

### Simpulan

Berdasarkan penelitian yang dilakukan, maka didapatkan simpulan bahwa pengujian yang dilakukan menggunakan *software* LabView 2014 mampu mengolah data respon sistem riil dengan *real time*. Pada saat pengujian *setpoint* tetap dan berubah, nilai  $Q = 10$  memiliki hasil yang lebih baik dari seluruh pengendali yang telah dilakukan, masing-masing nilai dari parameter respon saat pengendalian LQR dengan nilai  $Q = 10$  yang terdapat pada Tabel 7 dengan nilai  $t_d$  sebesar 0,153 detik, nilai  $t_r$  (5%-95%) sebesar 0,653 detik, nilai  $t_r$  (10%-90%) sebesar 0,487 detik, nilai  $t_s$  (5%) sebesar 0,666 detik, nilai  $t_s$  (2%) sebesar 0,888 detik, nilai  $t_s$  (0,5%) sebesar 1,11 detik, nilai Ess sebesar 0,36%. Sedangkan, untuk pengujian dengan beban, nilai  $Q = 90$  memiliki hasil yang lebih baik, dengan nilai Ess sebesar 1,52%.

### Saran

Berdasarkan hasil penelitian yang telah dilakukan terdapat beberapa aspek yang dapat diperbaiki dan dikembangkan agar memberikan hasil yang lebih maksimal meliputi model atau fungsi alih yang didapatkan masih kurang baik, sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini. Adapun sistem ini masih dapat dikembangkan dengan menggunakan metode pengendali yang lain seperti Sliding Mode Control (SMC), Proportional Integral Derivative (PID) Menggunakan Genetic Algoritim (GA), Model Predictive Control (MPC) dan lain-lain untuk mendapatkan hasil respon kecepatan yang lebih baik.

## DAFTAR PUSTAKA

- Arifin, Fatchul. 2006. "Perancangan dan Simulasi Sistem Suspensi Mobil Berbasis Kendali Optimal". Yogyakarta: UNY.
- Darmawan, Aria. 2011. "Perancangan Embedded Kontroler LQR Adaptive menggunakan Mikrokontroller untuk Pengaturan Kecepatan Motor DC". Surabaya: ITS.
- Ibrahim, A.W., dkk. 2016. "Sistem Kontrol Torsi pada Motor DC". Yogyakarta: UGM.
- Instruments, National. 2018. "What is LabVIEW?". (Online). (<http://www.ni.com/labview/>, diakses pada 20 Februari 2018).
- Khijwania, Sunil. 2012. "Setpoint Tracking". NPTEL, INDIA : IIT Guwahati.
- Krause, Paul. 2002. "Analysis of Electric Machinery and Drive Systems". Second Edition. Purdue University. ISBN 0-471-14326-X.
- LJCREATE, 2010. "Product Information Sheet Transducer, Instrumentation & Control Trainer". United Kingdom: Norwich.
- Muruganandhan, S., dkk. 2013. "LabVIEW-NI ELVIS II based Speed Control of DC Motor". Tamilnadu, India: Kalaingar Karunanidhi Institute of Technology.
- Ogata, Katsuhiko. 2010. "Modern Control Engineering fifth edition". New York: Prentice-Hall, Inc.
- Prasetya, Nor. 2018. "Sistem Pengaturan Kecepatan Motor Brushless DC Menggunakan Pengendali Linear Quadratic Regulator". Surabaya: Universitas Negeri Surabaya.
- Siswoyo, Bambang. 2012. "Motor DC Model State Space". (Online). (<http://bsiswoyo.lecture.ub.ac.id/201/01/motor-dc-model-state-space/>, diakses pada tanggal 11 Januari 2019).
- Sugiyono. 2015. "Jenis-Jenis Metode Penelitian". Bandung: Universitas Pasundan
- Zhang, J., dkk. 2004 "A developed method of tuning PID controllers with fuzzy rules for integrating process". Boston: Proceedings of the American Control Conference.