

# Model Sistem Pengendalian Arus pada HHO Generator Menggunakan Model *Linear Quadratic Regulator*

Ajat Sudrajat<sup>1\*</sup>, Fitri Rahmah<sup>1</sup>, M. Irham Hudaya<sup>1</sup>, V. Vekky R. Repi<sup>1</sup>

<sup>1</sup>Program Studi Teknik Fisika, Universitas Nasional  
JL. Sawo Manila, Pasar Minggu, Jakarta 12520

\*Korespondensi penulis: ajatsudrajatr2@yahoo.com

**ABSTRAK.** Telah dibuat sistem kontrol arus HHO generator menggunakan arduino dan menggunakan model *linear quadratic regulator*. HHO generator merupakan alat yang mengubah air menjadi gas hidrogen dan oksigen yang biasa digunakan untuk keperluan efisiensi mesin berbahan bakar minyak. Untuk menghasilkan gas hidrogen dan oksigen yang stabil, dibutuhkan sistem kontrol parameter arus. Kontrol terhadap arus dilakukan menggunakan arduino dengan sensor arus ACS712, sensor tegangan INA219, dan aktuator MOSFET IRF530n. Sedangkan pemodelan sistem ruang keadaan dan desain kontrol LQR menggunakan MATLAB. Dari hasil uji didapatkan model sistem yang observable, kontrolabel, dan stabil. Pada respon step sistem kontrol menunjukkan pengontrol memiliki kecenderungan yang kurang stabil.

**Kata kunci:** Sistem kontrol, *state space*, *linear quadratic regulator*, HHO generator

## PENDAHULUAN

Meningkatnya kebutuhan energi pada saat ini membuat adanya banyak peningkatan efisiensi. Banyaknya mesin berbasis Bahan Bakar Minyak (BBM) dan maraknya penggunaan BBM membuat peningkatan efisiensi pada mesin BBM menjadi salah satu strategi yang tepat. HHO Generator merupakan alat yang dapat meningkatkan efisiensi pembakaran BBM dengan mengubah air menjadi gas hydrogen dan oksigen. Terdapat beberapa strategi untuk meningkatkan efisiensi pembakaran mesin, salah satunya adalah dengan memberikan input oksigen pada mesin sesuai pada prinsip segitiga api [1].

Selain itu faktor terjadinya ledakan juga mempengaruhi efisiensi mesin. Penggunaan metode kontrol ruang-keadaan masih sangat minim dan kurang lazim untuk digunakan. Metode control ruang-keadaan merupakan salah satu disiplin ilmu dalam bidang control yang cukup muda. Penggunaan metode ini sangat baik untuk mengontrol sistem MIMO (Multi Input Multi Output), dan lebih tegas dibandingkan dengan metode pengontrolan konvensional. Metode ini sangat mudah untuk diimplementasikan pada pengontrol digital mikrokontroler [2,3]

Disamping itu, salah satu disiplin kontrol yang kurang populer di Indonesia merupakan Kontrol Optimal. Kontrol Optimal merupakan strategi kontrol yang menggabungkan hukum-hukum dan teori kontrol dengan optimasi. Kontrol Optimal biasanya mempunyai batasan-batasan yang ditentukan oleh desainer. Batasan yang paling sering digunakan oleh desainer adalah batasan biaya dan penggunaan energi. Salah satu pemecahan kontrol optimal yang paling mudah yaitu adalah LQR atau *Linear Quadratic Regulator* [4-7].

Tujuan penelitian untuk merancang sistem kontrol arus listrik pada HHO Generator menggunakan Arduino dan Matlab. Dimana lingkup kegiatan dari penelitian ini adalah pemodelan HHO Generator, desain sistem kontrol dengan metode ruang-keadaan, desain pengontrol arus menggunakan mikrokontroler dan MOSFET, desain pengontrol menggunakan pendekatan LQR, kontrolabilitas, dan observabilitas pengontrol pada metode ruang-keadaan. Penelitian ini tidak membahas tentang kebutuhan gas HHO Generator pada mesin, desain HHO Generator, maupun pengaruh variabel proses pada sistem elektrik.

## METODE PENELITIAN

### Alur Perancangan

Pertama-tama desain hardware untuk pengontrolan arus dengan menggunakan MOSFET IRF530n dan sensor arus ACS712. Desain *hardware* ini juga menggunakan Arduino sebagai pengontrolnya. Lalu dilakukan pengambilan data dengan memasukkan nilai-nilai tertentu untuk mendapatkan pemodelan sistem. Data-data tersebut diproses di dalam matlab dan didapat model sistem pengontrol beserta HHO generator. Langkah selanjutnya yaitu mendesain model kontrol dengan menggunakan matlab. Setelah parameter kontrol didapat, lalu dilakukan *programming* pada arduino untuk fungsi kontrol dapat dijalankan.

### Desain Hardware

Dalam pengontrolan tegangan, Arduino memiliki fitur PWM. Pengontrolan tersebut tidak dapat dilakukan pada sistem ini dikarenakan sistem PWM akan memperlakukan MOSFET sebagai switch dengan frekuensi tinggi. Penggunaan MOSFET sebagai switch pengukuran pada arus tidak bias dilakukan. Untuk pengontrolan tegangan pada MOSFET diperlukan sebuah DAC. Penulis merancang Digital-to-Analog sederhana dengan menggunakan tangga resistor yang dihubungkan pada pin-pin arduino.

Hasil keluaran dari rangkaian DAC ini lalu dihubungkan ke opamp 741 pertama. Op-Amp ini berguna sebagai *voltage follower*. Keluaran dari Op-Amp *voltage follower* lalu dihubungkan ke Op-Amp 741 kedua yang berguna sebagai penguat tegangan. Untuk penelitian ini, penulis menggunakan penguatan sebesar 5/3 kali. Penguatan ini didapat dengan memasang resistor  $2 \times 22k$  Ohm dan  $3 \times 22K$  Ohm. Tegangan yang keluar dari *voltage follower* mempunyai nilai sebesar 1.2V untuk nilai minimumnya, yang menyebabkan kurangnya bit yang bisa digunakan pada arduino. Penulis menemukan kekurangan ini sewaktu penulis mengetes respon alat dengan memberikan sinyal sinusoidal pada keluaran DAC.

Daerah kerja pada MOSFET yaitu berkisar 2-20V. namun berdasarkan hasil yang didapat, pengontrolan MOSFET tidak dibutuhkan untuk nilai maksimum sampai 20V dikarenakan arus yang mengalir kurang dari 5A. keterbatasan alat catudaya dan sensor arus juga menjadi pertimbangan penulis untuk mendesain pengontrol pada rentang 0-5 Ampere. Dalam rangkaian terdapat resistor yang menghubungkan antara terminal *source* dan *gate* pada MOSFET. Penggunaan resistor ini berguna sebagai "*pull-down resistor*", yaitu agar tegangan menjadi lebih turun. Penggunaan ini dibutuhkan agar pengontrolan arus menjadi lebih mudah. Pada sistem pengontrol terdapat komponen-komponen catu daya 12V, MOSFET IRF530n, arduino, resistor, sensor arus ACS712, sensor tegangan INA219, OpAmp LM741 dan *plant* HHO generator. Sensor arus dipasang secara seri pada kedua ujung dari *plant* HHO agar didapat hasil yang lebih akurat [7].

### Pengambilan Data

Pengambilan data hubungan input dan output dilakukan dengan menggunakan arduino melalui kanal komunikasi Serial. Arduino diprogram untuk mengirim sinyal tegangan sebesar 2, 3, dan 4 V dsb. ke terminal *gate* pada MOSFET. MOSFET lalu mengubah tahanan berdasarkan nilai tegangan yang dikirim oleh Arduino. Perubahan tegangan pada *gate* menyebabkan perubahan resistansi pada MOSFET. Perubahan resistansi tersebut mengakibatkan adanya perubahan arus pada terminal *drain* ke *source*. Arus terminal *source* dialirkan ke *plant* HHO generator lalu diukur dengan menggunakan sensor arus ACS712. Pengambilan data ini dilakukan untuk mencari hubungan antara tegangan pada terminal *gate* MOSFET, arus pada *plant* HHO generator, dan tahanan pada *plant* HHO generator. Komunikasi serial mengembalikan data yang dibaca oleh sensor arus dan juga mengirim tegangan ke terminal *gate* MOSFET.

Hasil pengambilan data tersebut berupa data waktu cuplik, tegangan *gate* MOSFET, dan arus setelah melewati *plant* HHO generator. Data tersebut lalu dirapihkan dan dimasukkan pada program matlab untuk mengetahui resistansi pada waktu, tegangan, dan arus tertentu. Waktu cuplik yang digunakan adalah sebesar lalala milisekon. Waktu cuplik tersebut disimpan untuk digunakan dalam proses pemodelan sistem pada matlab. Data tegangan,

arus, dan tahanan lalu digunakan untuk dilakukan pemodelan dengan menggunakan perintah *ssest* pada matlab. Data tersebut lalu diubah menjadi objek data *iddata* menggunakan perintah *iddata*. Pengubahan tipe data ini diperlukan agar data tersebut dapat diproses menggunakan perintah *ssest* pada matlab.

### Pemodelan Sistem dengan MATLAB

Dari data hubungan input-output yang didapat dari arduino, data ini lalu diubah dalam bentuk objek data *iddata*. Hasil pengukuran dimasukkan pada objek data *iddata* yang berbeda yang nantinya akan digunakan untuk pemrosesan data pada *ssest*. Data tegangan, arus, dan waktu yang sudah dalam bentuk objek data *iddata*, lalu diproses dengan menggunakan perintah *ssest*. Perintah ini lalu mengembalikan nilai A, B, C dan D berdasarkan input dan output pada *iddata*. Hasil pemrosesan data pada matlab akan mengembalikan sebuah plot dengan kemungkinan orde sistem yang paling tinggi [8].

Pemilihan orde sistem dapat berdasarkan dari jumlah nilai singular yang didapat. Pada gambar dijelaskan bahwa jumlah singular pada sistem. Pertimbangan dalam pemilihan orde sistem juga perlu melihat seberapa dekat bentuk pemodelan tersebut. Pada penelitian ini di pilih untuk menggunakan orde 3 (tiga) dikarenakan nilai pemodelan yang lebih tinggi dibandingkan dengan orde lainnya. Hasil dari estimasi pemodelan ini lalu digunakan kembali untuk iterasi selanjutnya. Iterasi selanjutnya juga menggunakan perintah *ssest*. Berikut *source code* yang penulis gunakan untuk mendapatkan data pemodelan [8].

```
u=[];u1=[]; //input untuk kedua sampling data dimasukkan secara manual
y=[];y1=[]; //output untuk kedua sampling data dimasukkan secara manual
data1=iddata(y,u,0.1);
data2=iddata(y1,u1,0.075);
sys1=ssest(data1,1:5);
sys2=ssest(data2,sys1);
Co=ctrb(sys2);
Ob=obsv(sys2);
Sys3=c2d(sys2,0.01);
```

### Desain Kontrol LQR

Hasil pemodelan sistem pada matlab mengembalikan nilai A,B,C dan D yang merupakan model sistem pengontrol dalam bentuk ruang-keadaan. Tahap selanjutnya adalah mencari pengontrol agar proses tersebut dapat berlangsung dengan lebih baik. Pencarian parameter pengontrol dilakukan dengan menggunakan perintah *dlqr*. Berikut *source code* yang digunakan [9]:

```
Q=Co' * Co;
[K,S,E]=dlqr(sys3.a,sys3.b,Q,1);
An=sys3.a-(sys3.b*(K.*sys3.c));
nano=sys3; nano.a=An
eig(An)
```

Perintah *dlqr* pada matlab merupakan perintah untuk mencari parameter pengontrol yang optimal. Perintah ini membutuhkan parameter A, B, C, D, Q, R dan Ts. Parameter A, B, C dan D telah didapat pada proses sebelumnya, namun parameter Q dan R harus dicari terlebih dulu. Setelah nilai Q dan R didapatkan, penulis memproses data Q dan R dengan perintah *dlqr* pada matlab. Perintah tersebut mengembalikan nilai K, S, dan E. Nilai K digunakan sebagai gain pengontrol umpan balik.

### Programming Arduino

Setelah didapat nilai K dengan menggunakan perintah *lqrd* pada matlab, memproses model sistem ruang keadaan dengan hubungan persamaan (1) [11].

$$\dot{x}(t) = (A + BK(I - DK)^{-1}C)x(t) \quad (1)$$

$$y(t) = (I - DK)^{-1}Cx(t)$$

Dalam banyak kasus, nilai D mempunyai nilai nol. Dalam perancangan sistem ini tidak ada pengaruh umpan maju dari nilai input ke nilai output. Maka, nilai D dapat diasumsikan nol. Matriks A, B, C dan K diproses melalui persamaan (1) diatas di ubah menjadi persamaan (2) dan (3).

$$\dot{x}(t) = Gx(t) \tag{2}$$

$$y(t) = Cx(t)$$

$$G = (A + BKC) \tag{3}$$

Persamaan (2) dan (3) diubah dalam bentuk persamaan diskrit menjadi (4) dan (5).

$$x(k + 1) = Gx(k) \tag{4}$$

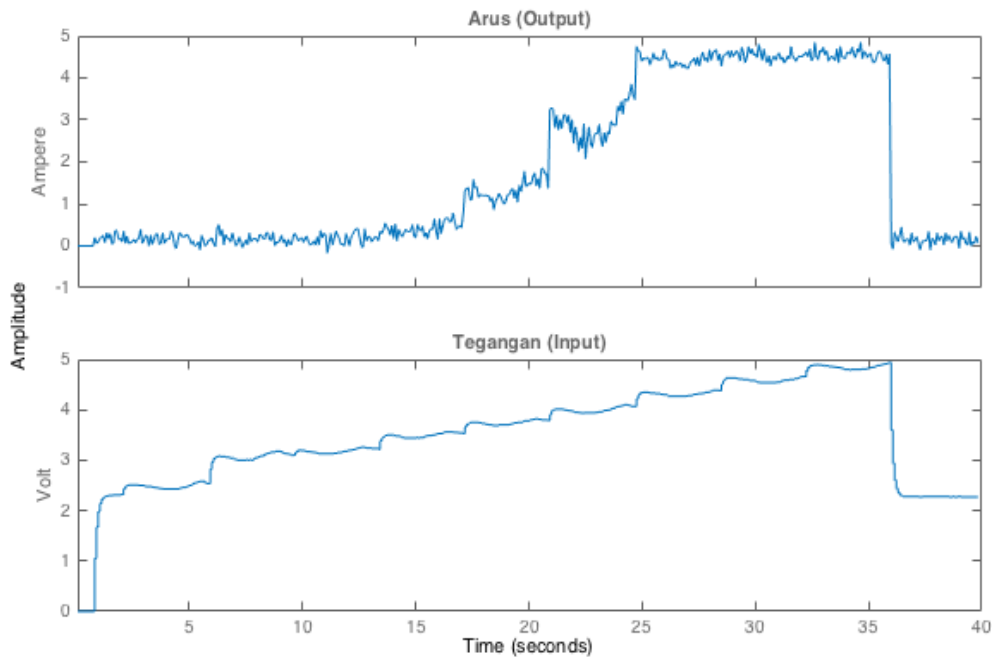
$$y(k) = Cx(k) \tag{5}$$

Setelah parameter didapat, lalu dilakukan pemograman pada Arduino mengikuti persamaan diskrit.

## HASIL DAN PEMBAHASAN

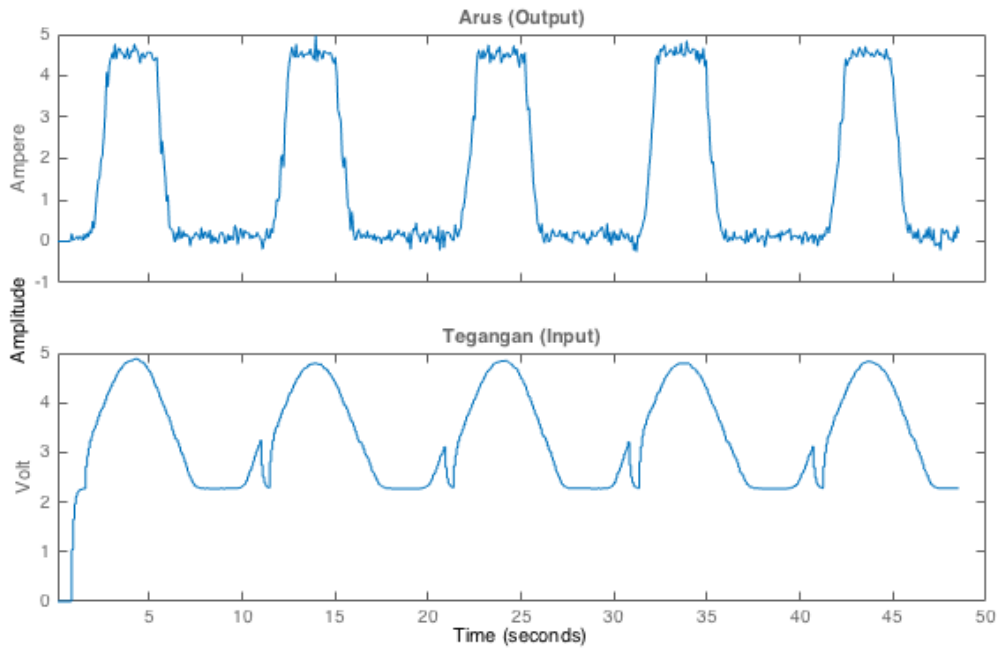
### Hasil Perancangan Sistem Hubungan Tegangan gate MOSFET dengan Arus

Pada gambar 1 adalah hasil pengambilan data pada arduino dalam bentuk grafik, merupakan data dengan uji masukan tegangan yang semakin naik. Dalam grafik dapat dilihat bahwa terdapat banyak noise yang terbaca pada sensor arus. Juga dapat dilihat bahwa hubungan antara arus dengan tegangan tidak linier. *Noise* yang terjadi dapat berasal dari piranti arduino dan/atau dari HHO generator yang tidak stabil dikarenakan proses kimia elektrolisis yang menyebabkan perubahan tahanan yang dapat mempengaruhi perubahan arus.



**GAMBAR 1.** Grafik hubungan arus dan tegangan dengan input step

Pada pemograman arduino, tegangan masukan ditahan pada nilai-nilai tertentu, namun pada saat pengambilan data, tegangan pada terminal *gate* MOSFET bergerak kearah atas. Hal ini mungkin terjadi dikarenakan adanya perubahan nilai tegangan pada terminal Source pada MOSFET. Hubungan antara tegangan dengan arus juga terlihat kurang baik pada arus 1.5-3A. hal ini mungkin disebabkan karena sirkuit MOSFET IRF530n yang kurang baik untuk bekerja dalam daerah ohmic. Pengontrolan arus pada tegangan 4V keatas semakin menjadi tidak handal. Hal ini mungkin dikarenakan arus maksimum yang bisa diterima oleh plant HHO hanya sebesar 4 A pada tegangan kerja 12V.



GAMBAR 2. Grafik Hubungan Arus dan Tegangan dengan input Sinusoidal

Pada gambar 2 adalah pengujian kedua dengan menggunakan bentuk sinyal yang berbeda, didapatkan bentuk respon sistem yang lebih linier. Hal ini juga dimungkinkan karena karakteristik MOSFET IRF530n yang lebih sesuai sebagai *switch*. Pada pengujian kedua Arduino diprogram untuk menghasilkan nilai sinusoidal dengan rentang 0-5 V.

**Analisis Pemodelan dan Sistem**

Dari hasil pemodelan, digunakan estimasi model dengan orde tiga menggunakan data input-output step. Dari estimasi tersebut lalu dimasukkan data tambahan dengan input-output sinusoidal untuk mendapatkan pemodelan yang jauh lebih akurat. Nilai-nilai parameter A, B, C dan D dapat dilihat pada persamaan (6).

$$\begin{aligned}
 A &= \begin{pmatrix} 0.9833 & -0.07367 & -0.02868 \\ 0.04836 & -0.189 & -0.08315 \\ 8.506e - 05 & 0.03215 & -0.2247 \end{pmatrix} \\
 B &= \begin{pmatrix} -0.007435 \\ -0.0693 \\ -0.1492 \end{pmatrix} \\
 C &= (-33.89 \quad -2.914 \quad -0.6738) \\
 D &= (0)
 \end{aligned}
 \tag{6}$$

Dari hasil pengujian kontrolabilitas dan observabilitas didapat bahwa sistem tersebut kontrolabel dan observabel. Pengujian kontrolabilitas menggunakan perintah *ctrb()* dan pengujian observabilitas menggunakan perintah *obsv()*. Hasil pengujian mengembalikan matriks kontrolabilitas dan observabilitas. Rank dari kedua matriks tersebut lalu dicocokkan pada orde sistem. Didapat rank yang sama pada kedua matriks. Pengujian stabilitas sistem dilakukan dengan mencari nilai eigenvalue. Hasil pengujian *eigenvalue* menggunakan perintah *eig()* dan didapat model sistem yang stabil [10].

**Hasil Desain Pengontrol LQR**

Parameter Q dalam persamaan permasalahan Linear-Quadratic didapat dengan mengalikan inversi matriks kontrolabilitas dengan matriks kontrolabilitas. Cara tersebut biasa digunakan untuk dijadikan titik awal pencarian nilai Q dengan cara trial-and-error. Nilai R yang digunakan pada penelitian ini bernilai 1 (satu). Pengontrolan dengan menggunakan

perintah *dlqr* memberikan nilai kontrolabilitas, observabilitas serta respon sistem sebagai berikut:

```
>> Co1=ctrb(nano)
Co1 =
    -0.0074    0.0021   -0.0007
    -0.0693    0.0255   -0.0074
    -0.1492    0.0313   -0.0062
>> rank(Co1)
ans =
     3
>> Ob1=obsv(nano)
Ob1 =
   -33.8881   -2.9143   -0.6738
   -33.3351    3.0257    1.3658
   -32.7624    1.9280    0.3977
>> rank(Ob1)
ans = 3
```

Hasil penambahan pengontrol juga memberikan sistem yang kontrolabel dan observabel. Hal ini ditunjukkan dengan nilai rank matriks kontrolabilitas dan observabilitas sama dengan orde sistem pemodelan. Dari hasil pencarian nilai K, dibuat sistem baru dengan memasukkan matriks A berdasarkan Persamaan 13. Sistem nano merupakan realisasi sistem pengontrol diskrit dengan waktu sample sebesar 75ms. Sistem nano menggunakan matriks A hasil dari Persamaan 13 dan matriks B, C, D mengikuti sistem *sys2*. Berikut detail untuk sistem nano yang dijelaskan pada keluaran matlab [9,10].

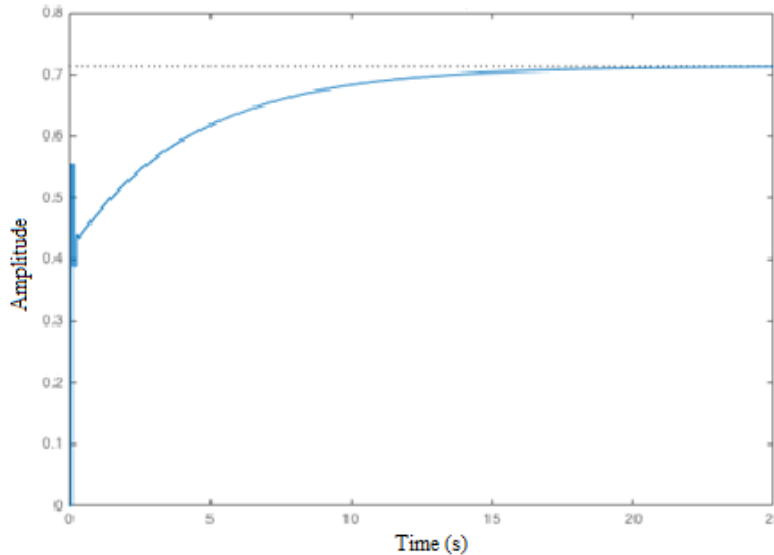
```
>> nano
nano =
Discrete-time identified state-space model:
x(t+Ts) = A x(t) + B u(t)
y(t) = C x(t) + D u(t)
A =
    x1    x2    x3
x1  0.9833  -0.07367  -0.02868
x2  0.004836  -0.189  -0.08315
x3  8.506e-05  0.03215  -0.2247
B =
    u1
x1 -0.007435
x2 -0.0693
x3 -0.1492
C =
    x1    x2    x3
y1 -33.89  -2.914  -0.6738
D =
    u1
y1 0
Sample time: 0.075 seconds
Parameterization:
FREE form (all coefficients in A, B, C free).
Feedthrough: none
Disturbance component: estimate
Number of free coefficients: 18
Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.
Status:
Created by direct construction or transformation. Not estimated.
>>
```

Dari keluaran matlab tersebut didapat matriks A, B, C, dan D pada sistem nano di persamaan (7).

$$A = \begin{pmatrix} 0.9833 & -0.07367 & -0.02868 \\ 0.04836 & -0.189 & -0.08315 \\ 8.506e-05 & 0.03215 & -0.2247 \end{pmatrix}$$

$$\begin{aligned}
 B &= \begin{pmatrix} -0.007435 \\ -0.0693 \\ -0.1492 \end{pmatrix} \\
 C &= (-33.89 \quad -2.914 \quad -0.6738) \\
 D &= (0)
 \end{aligned}
 \tag{7}$$

Dilakukan tes untuk respon step pada sistem nano untuk melihat karakteristik sistem pengontrol. respon step sistem nano dapat dilihat pada gambar 3.



**GAMBAR 3.** Grafik respon step sistem pengontrol

Kestabilan dari pengontrol dapat dilihat pada hasil keluaran matlab sebagai berikut:

```

>> eig(nano.a)
ans =
    0.9830 + 0.0000i
   -0.2067 + 0.0484i
   -0.2067 - 0.0484i
    
```

dari hasil *eigenvalue*, terlihat bahwa sistem nano mempunyai nilai pole yang dekat dengan 1. Hal ini menunjukkan bahwa pengontrol ini cenderung tidak stabil. Ketidakstabilan ini dimungkinkan kurang tepatnya pemilihan nilai Q dan R dalam pencarian nilai pengontrol. Pemilihan nilai Q dan R mengacu pada pengetahuan parameter sistem, sedangkan hubungan variabel keadaan sistem tidak diketahui dikarenakan hasil matriks merupakan hasil estimasi pemodelan.

### KESIMPULAN

Pada penelitian ini, telah dirancang alat pengontrol arus untuk HHO generator menggunakan sensor arus ACS 712 , sensor tegangan INA 219, serta aktuator MOSFET IRF530n. Pemodelan sistem kontrol dilakukan dengan menggunakan perintah *sst* pada matlab. Hasil pemodelan sistem mempunyai model sistem yang kontrolabel dan observabel. Estimasi pemodelan sistem mempunyai orde 3 (tiga) dengan kedekatan data sebesar 85.61%. Estimasi pemodelan sistem menghasilkan sistem yang kontrolabel dan observabel. Pencarian parameter pengontrol menggunakan perintah *dlqr* pada matlab yang menghasilkan pengontrol yang kontrolabel dan observabel. Kurangnya parameter hubungan variabel keadaan pada sistem menyebabkan parameter pengontrol yang kurang baik. Pembacaan arus sensor memiliki batasan 5A. Pembacaan arus mempunyai nilai *noise* yang tinggi. Pembacaan tegangan pada terminal *gate* MOSFET mempunyai range 2.2 – 7.4 volt.

**DAFTAR PUSTAKA**

- [1] University of South Carolina, "The Fire Triangle," 28 July 2016. [Online]. Available: [http://www.sc.edu/ehs/modules/Fire/01\\_triangle.htm](http://www.sc.edu/ehs/modules/Fire/01_triangle.htm). [Accessed 28 July 2016].
- [2] B. Friedland, Control System Design, Michigan: McGraw-Hill, 1986.
- [3] K. M. Hangos, R. Lakner and M. Gerszon, Intelligent Control Systems: An Introduction with Examples, New York: Springer, 2001, p. 254.
- [4] K. Ogata, Modern Control Engineering, 3rd Edition ed., Upper Saddle River: Prentice-Hall, 1997.
- [5] K. R. E., "On the General Theory of Control Systems," in *Congress of IFAC*, Moscow, 1960.
- [6] C. Schmid, "State Space vs. Transfer Function," 09 05 2005. [Online]. Available: <http://virtual.cvut.cz/course/syscontrol/node97.html>. [Accessed 28 15 2016].
- [7] IKALOGIC S.A.S, "8-bit Digital to Analog converter (DAC)," IKALOGIC S.A.S, 2016. [Online]. Available: <https://www.ikalogic.com/8-bit-digital-to-analog-converter-dac/>.
- [8] H. & S. R. Kwakernaak, Linear Optimal Control Systems, Wiley-Interscience, 1972.
- [9] MathWorks, "Matlab," MathWorks Inc., 2016. [Online]. Available: <http://www.mathworks.com/help/ident/ref/iddata.html>. [Accessed 2016].
- [10] MathWorks, "Estimation of State Space Model," MathWorks Inc., 2016. [Online]. Available: <http://www.mathworks.com/help/ident/ref/ssest.html>. [Accessed 2016].
- [11] MathWorks, "Linear Quadratic Regulator," MathWorks Inc., 2016. [Online]. Available: <http://www.mathworks.com/help/control/ref/dlqr.html>.