

# Pengendalian Kursi Bioskop 4D Menggunakan Pengendali Diskrit

Yefta Devian Permana, Handry Khoswanto, Felix Pasila  
Program Studi Teknik Elektro, Universitas Kristen Petra  
Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia

E-Mail: yeftadp@gmail.com; handry@peter.petra.ac.id; felix@peter.petra.ac.id

**Abstrak**— Pembuatan pengendalian kursi bioskop 4D menggunakan pengendali diskrit bertujuan untuk mengendalikan gerakan kursi bioskop 4D menggunakan pengendali diskrit. Umumnya, pengendalian menggunakan sistem analog.

Pengendalian kursi bioskop 4D menggunakan pengendali diskrit menggunakan MATLAB sebagai pengendali utama, perangkat lunak cinema4D untuk mendesain film dan mendapatkan data input, Arduino sebagai pengumpan data, control valve sebagai katup elektrik untuk membuka dan menutup jalur angin, stewart platform sebagai kursi, pneumatik sebagai kaki dari kursi.

Dari pengujian sistem didapatkan nilai kesalahan dari sistem dibawah 10% yaitu pada 7,4%. Hal ini menunjukkan bahwa sistem yang dibuat berjalan dengan baik dan dapat menyesuaikan dengan film yang digunakan.

**Kata Kunci**— Arduino, diskrit, MATLAB, pneumatik, stewart platform.

## I. PENDAHULUAN

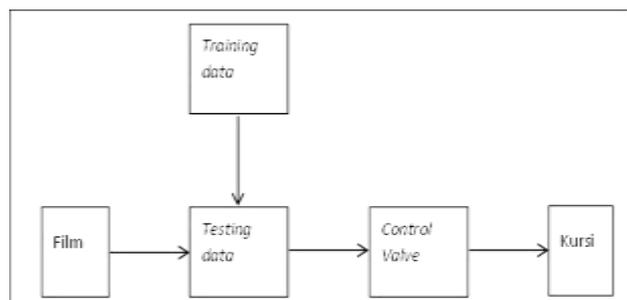
Kebutuhan dalam dunia hiburan (*entertainment*) semakin lama semakin besar, salah satu contohnya berkembangnya bioskop 3D. Awalnya, bioskop 3D hanya untuk film – film animasi tertentu dengan durasi yang singkat, tetapi sekarang sudah mulai banyak film – film *box office* yang berdurasi panjang tersedia dalam tampilan 3D. Hal – hal ini mengakibatkan film – film animasi 3D tadi dikembangkan menjadi animasi 4D yaitu ditambah dengan goyangan kursi dan semburan – semburan angin dan air sesuai animasi tersebut.

Desain kursi yang digunakan pun berbeda dengan kursi bioskop pada umumnya. Ada tambahan mesin penggerak di bawah kursi untuk dapat membuat kursi miring, naik, turun, dll. Manipulator mekanik dari kursi bergerak ini yang cukup populer adalah menggunakan *stewart platform*. *Stewart Platform* diusulkan oleh Stewart pada tahun 1965 sebagai simulator penerbangan [1]. *Platform* ini terdiri dari pelat atas (*moving-platform*), pelat dasar (*fixed-base*), dan enam kaki yang menghubungkan pelat atas dan pelat bawah.

## II. PERANCANGAN SISTEM

### A. Gambaran Umum Sistem

Sistem secara umum bekerja digambarkan seperti pada gambar 1. Pengguna akan memutar film yang digunakan. Film ini sebelumnya sudah diproses yaitu dengan merekam data posisi dari objek yang diinginkan.



Gambar. 1. Blok Diagram Sistem

*Input* yang digunakan untuk tugas akhir ini adalah data posisi objek yang sudah direkam di awal. Kemudian dicari persamaan antara posisi objek dari film dengan posisi gerak kursi atau sudut gerak kursi.

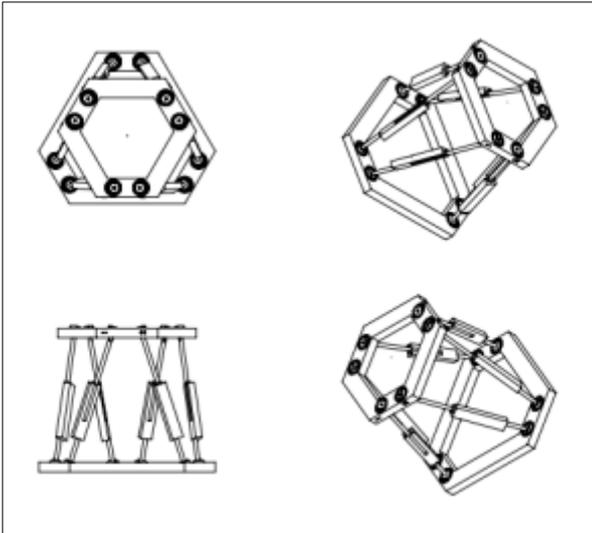
Proses penyamaan ini tidak menggunakan *database* tetapi menggunakan *training data* karena bila menggunakan *database* maka proses akan berjalan lambat karena banyaknya data. Berbeda dengan sistem *database*, proses *training data* adalah proses penyamaan dengan menggunakan beberapa data saja yang diasumsikan dapat mewakili semua data. Setelah mendapatkan persamaan antara kedua hal tersebut, maka hasil dari *training data* ini yang digunakan saat proses eksekusi (*testing data*) seperti pada gambar 1. Proses *training data* dilakukan menggunakan *toolbox* yang telah dibuat oleh Felix Pasila dalam penelitian *Designing the 6-DoF Massive Paraller Arrays with Artificial Intelligence Control* [2].

Saat eksekusi, *input data* adalah data posisi dari objek yang diamati di awal tadi. Kemudian nilai dari posisi ini akan dimasukkan ke dalam persamaan yang sudah didapatkan melalui proses *training data*. Kemudian sistem menghasilkan *output* sesuai dengan hasil persamaan. *Output* ini berupa *state* dari pneumatik. Pneumatik ini digunakan sebagai kaki kursi yang menentukan posisi kursi. Ada 3 *state* yang dihasilkan yaitu (0) adalah saat pneumatik menarik piston ke bawah, (1) adalah saat pneumatik diam, mengikuti posisi dari pneumatik yang lainnya, dan (2) adalah saat pneumatik mendorong piston keluar.

Sistem yang digunakan pada tugas akhir ini adalah pengendalian gerak kursi secara otomatis dengan menggunakan pengendali diskrit. Sistem ini menggunakan perangkat lunak MATLAB sebagai media pemrograman. Program pada MATLAB akan mengolah data sudut objek sebagai *input* dan menghasilkan data diskrit yang menjadi instruksi untuk menggerakkan pneumatik pada kursi.

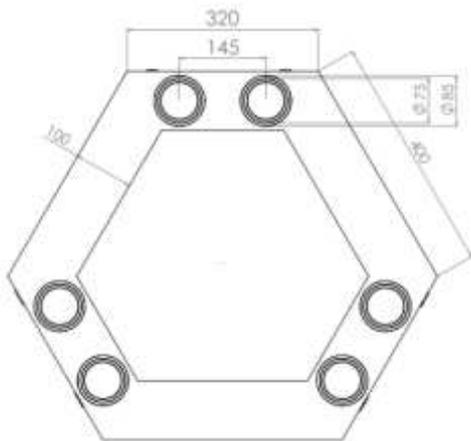
B. Desain Stewart Platform

Bagian – bagian dari *stewart platform* yang didesain adalah platform bagian atas, platform bagian bawah, silinder pneumatik, *spherical joint*. Gambaran besar dari desain *stewart platform* dapat dilihat pada gambar 2.



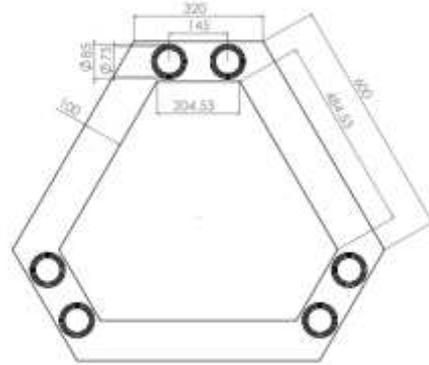
Gambar 2. Gambaran besar desain *stewart platform*

Bagian pertama adalah *platform* atas. Material yang digunakan adalah besi berongga. *Platform* ini dibentuk menjadi bangun segienam dengan 3 sisi panjang dan 3 sisi pendek. Sisi panjang memiliki panjang 400mm, sedangkan sisi pendek memiliki panjang 320mm. Lebar untuk masing – masing sisi adalah 100 mm. Pada sisi yang pendek terdapat lubang dengan diameter 85mm sebagai tempat untuk sendi dari pneumatik. Untuk lebih jelas dapat dilihat pada gambar 3.



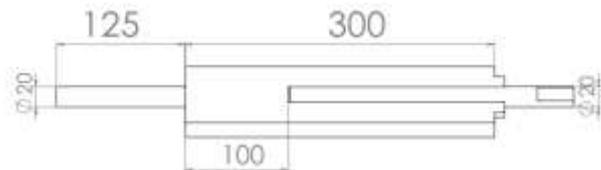
Gambar II. Skema *platform* atas

Bagian kedua adalah *platform* bawah, bentuk dari *platform* bawah identik dengan *platform* atas yaitu berbentuk bangun segienam, memiliki 3 sisi panjang dan 3 sisi pendek. Sisi panjang dari *platform* bawah memiliki panjang 600 mm, sisi pendek dari *platform* bawah memiliki panjang 320 mm. Pada sisi yang pendek juga tersedia lubang soket dengan diameter 85 mm untuk sendi dari pneumatik. untuk lebih jelas dapat dilihat pada gambar 4.



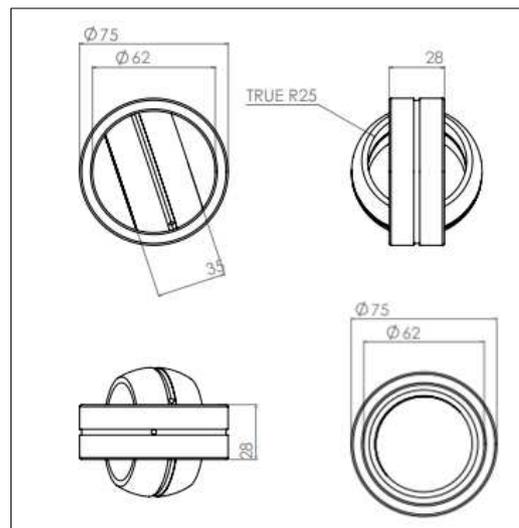
Gambar 4. Skema *platform* bawah

Pneumatik yang digunakan memiliki diameter tabung 50 mm, piston sepanjang 200 mm, dan diameter piston 20 mm. Gambar dari pneumatik dapat lebih jelas dilihat pada gambar 5. Pada piston terdapat diameter dalam 14 mm dengan ulir dalam yang dihubungkan dengan *shaft bearing*.



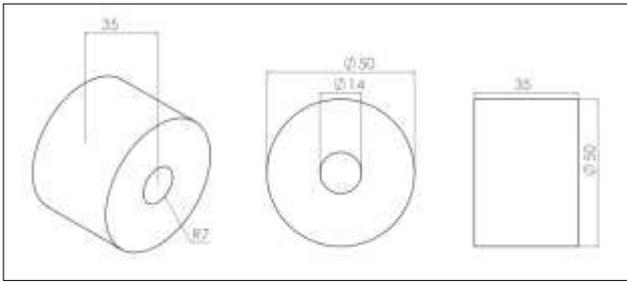
Gambar 5. Skema pneumatik

Bagian berikutnya adalah *spherical joint* yang merupakan sendi yang digunakan oleh *stewart platform*. *Spherical joint* yang digunakan dalam sistem ini terbuat dari *bearing* dan *shaft*. Skema dari *bearing* yang digunakan dapat dilihat dari gambar 6. *Bearing* ini berdiameter dalam 50 mm sesuai dengan diameter pneumatik yaitu 50 mm.



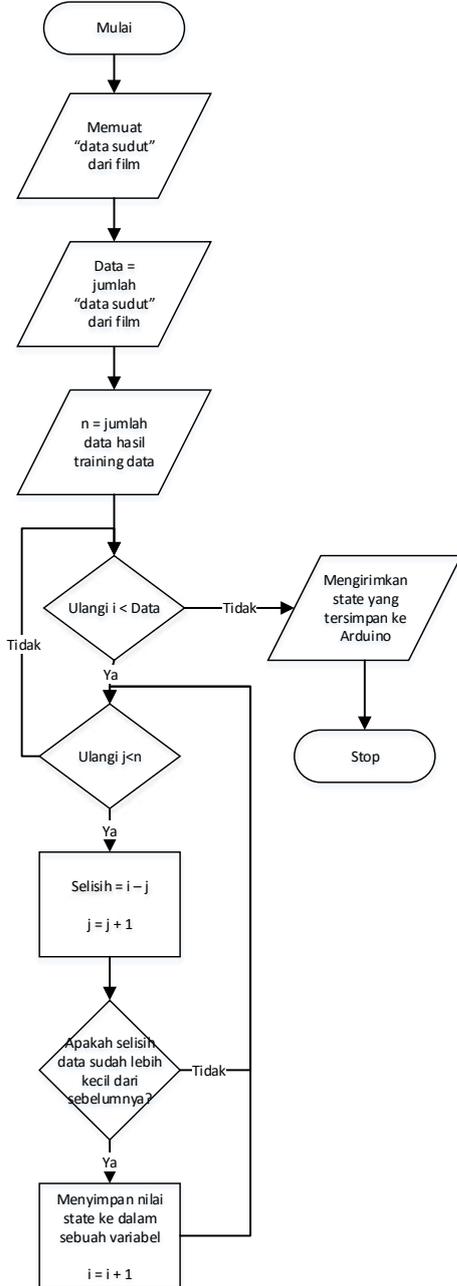
Gambar 6. Skema *bearing*

*Shaft bearing* yang digunakan terbuat dari besi pejal. Diameter dari besi ini adalah 50 mm sesuai dengan diameter dalam dari bearing. Contoh dari *shaft bearing* yang digunakan dapat dilihat pada gambar 7.



Gambar 7. Skema shaft bearing

C. Desain Software



Gambar. 8. Flowchart Sistem Keseluruhan

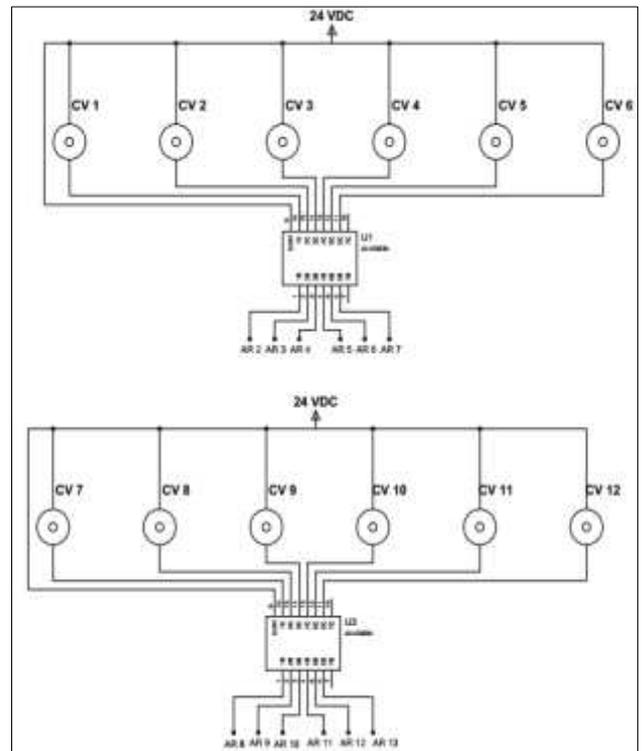
Program berjalan dengan memuat data sudut sebagai input. Jumlah data input adalah 1351 buah terambil dari frame 0 hingga frame 1350 dari desain film pada perangkat lunak cinema4D. Data input kemudian dihitung satu per satu baris dengan data sudut dari training data. Data masukkan dikurangkan dengan data sudut dari training data untuk mengetahui selisihnya. Perhitungan diulang

sebanyak 60 kali yaitu jumlah baris dari tabel training data. Selisih sudut yang paling kecil yang dipilih menjadi sudut dari kursi, kemudian dari sudut kursi ditentukan state pneumatik yang digunakan.

Proses perhitungan sampai pencarian state pneumatik diulang hingga semua data input selesai dihitung yaitu sebanyak 1350 kali. Setelah proses ini selesai maka semua state pneumatik yang terpilih dikirim satu per satu menuju Arduino untuk diproses menuju tahap berikutnya.

D. Desain Rangkaian Driver

Driver dibuat untuk mengakomodasi arus yang dibutuhkan oleh sebuah control valve karena arus yang dikeluarkan dari Arduino tidak mampu untuk menyuplai control valve. Sebuah control valve memiliki 2 relay yang masing – masing membutuhkan tegangan 24 V untuk aktif. Arus kerja dari sebuah relay dari control valve adalah 145mA – 200mA, sedangkan arus dari output Arduino adalah 40mA. Kedua relay ini tidak berhubungan satu dengan yang lain, sehingga sebuah control valve membutuhkan 2 output driver.



Gambar. 9. Rangkaian Driver

Rangkaian driver dibuat menggunakan 12 rangkaian identik seperti yang terlihat pada gambar 9. Driver memiliki 12 pasang kanal yaitu 12 kanal input dan 12 kanal output. Masing-masing kanal mengakomodasi 1 gerakan pneumatik, sehingga untuk 1 buah pneumatik menggunakan 2 kanal driver. Rangkaian driver terdiri dari 2 buah IC ULN 2003 serta catu daya 24 VDC.

Kaki basis (pin 1 – pin 6) dari ULN 2003 terhubung pada pin out arduino. Kaki kolektor dari ULN 2003 (pin 11 – pin 16) terhubung pada pin negatif dari control valve (CV). Pin positif dari CV terhubung pada kutub positif catu daya 24 Volt. Kaki emitor dari ULN 2003 (pin 8) terhubung pada kutub negatif dari catu daya (GND) dan pin COM (pin 9) terhubung pada kutub positif dari catu daya.

### III. PENGUJIAN SISTEM

#### A. Pengujian Arus Driver

Pengujian *driver* bertujuan mengetahui apakah rangkaian yang dibuat dapat mengakomodasi kebutuhan arus untuk *control valve*. *Driver* terdiri dari 12 kanal *output*. Pengujian ini dilakukan dengan cara memasang *amperemeter* pada pin negatif dari *control valve* dan kaki *output* dari *driver*. Pengujian ini dilakukan pada masing – masing kanal.

Tabel 1. Tabel pengukuran arus rangkaian *driver*

Kanal yang diuji	Arus yang dibaca (mA)
Kanal 1	171,8
Kanal 2	171,7
Kanal 3	169,3
Kanal 4	170,6
Kanal 5	170,7
Kanal 6	172,6
Kanal 7	178,8
Kanal 8	174,6
Kanal 9	177,4
Kanal 10	175,5
Kanal 11	177,2
Kanal 12	176,1

Pengujian arus *driver* yang dilakukan terbukti dapat mengaktifkan *relay* dari *control valve* yang digunakan. Dari table 1, diketahui arus yang dihasilkan dari pengujian adalah rata – rata 173,85, arus ini mampu mengaktifkan *relay* dari *control valve* karena arus kerja dari *relay* tersebut adalah 145mA – 200mA.

#### B. Pengujian Kursi

Pengujian kursi dilakukan untuk mengetahui sudut – sudut maksimal dan minimal dari kursi yang digunakan. Pengujian ini menggunakan pembacaan *gyroscope* dengan program yang dibuat oleh Jeff Rowberg [3]. Pengujian ini dilakukan dengan mengirimkan *state* yang diinginkan ke kursi, kemudian *gyroscope* membaca sudut dari kursi.

Pengujian ini menghasilkan sudut – sudut dari kursi yang digunakan. Pengambilan data ini dilakukan sebanyak 729 kali sesuai dengan kemungkinan kombinasi yang ada dari 6 buah pneumatik, dimana masing – masing memiliki 3 *state*. Dari kombinasi yang dikirim maka didapatkan 3 sudut yaitu *pitch*, *roll*, dan *yaw*. Contoh hasil dari pengujian adalah sebagai berikut. Data yang diberikan pada kursi adalah 0-1-2-0-1-2, maka nilai *pitch* yang didapatkan sebesar -11°, nilai *roll* 4°, dan nilai *yaw* -2°.

Dari pengujian ini didapatkan bahwa kisaran sudut untuk *pitch* adalah +40° sampai -42°. Kisaran sudut untuk *roll* adalah +40° sampai -40°, sedangkan kisaran sudut untuk *yaw* adalah 27° sampai -30°.

#### C. Pengujian Komunikasi dan Pemetaan Data Dari MATLAB Menuju Arduino

Pengujian ini bertujuan untuk mengetahui cara mengkomunikasikan antara MATLAB dan Arduino. MATLAB dan Arduino tersambung melalui serial. Caranya adalah dengan menginisialisasi serial pada MATLAB. Serial yang digunakan diberikan nama variabel untuk memudahkan penggunaan selanjutnya, selain itu

diatur juga *port* yang digunakan serta *baudrate* yang sesuai dengan perangkat yang dihubungkan.

Setelah menginisialisasi, kemudian yang dilakukan adalah membuka serial yang tersedia. Jika tidak ada kesalahan maka dapat dilakukan pengiriman data dari MATLAB menuju Arduino. Program lengkap komunikasi ini dapat dilihat pada gambar 10.

```

s1 =
serial('COM1','BaudRate',9600)//prose
s inisialisasi

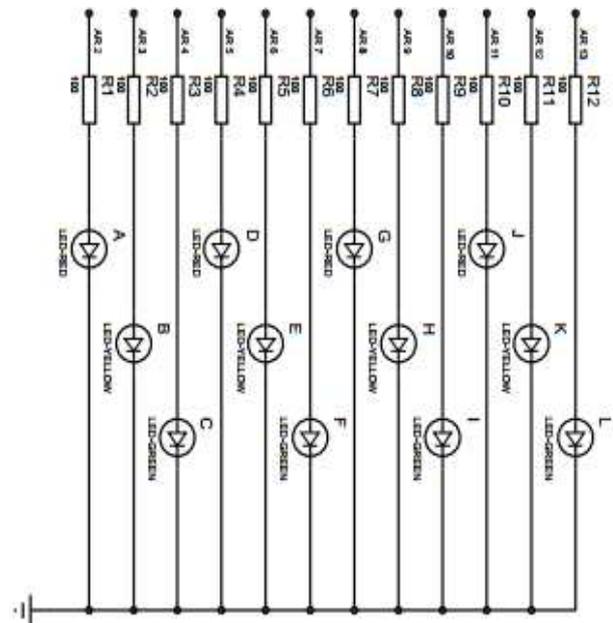
fopen(s1)//membuka port

fprintf(s1,'%d',state)//mengirim data

fclose(s1)//menutup port
    
```

Gambar 10. Program untuk mengirim data dari MATLAB

Pengujian dilakukan dengan mencoba mengirimkan kombinasi data buatan sendiri dari MATLAB dimana kombinasi yang dikirimkan sesuai dengan kombinasi data yang sebenarnya. Setelah data dikirimkan maka data akan dipetakan oleh Arduino. Program pemetaan menggunakan desain program pembacaan *array* per bagian. Setelah itu, Arduino akan mengirim data yang sudah dipetakan menjadi logika 0 atau 1 ke rangkaian LED yang sudah tersedia. Rangkaian LED dapat dilihat pada gambar 11.

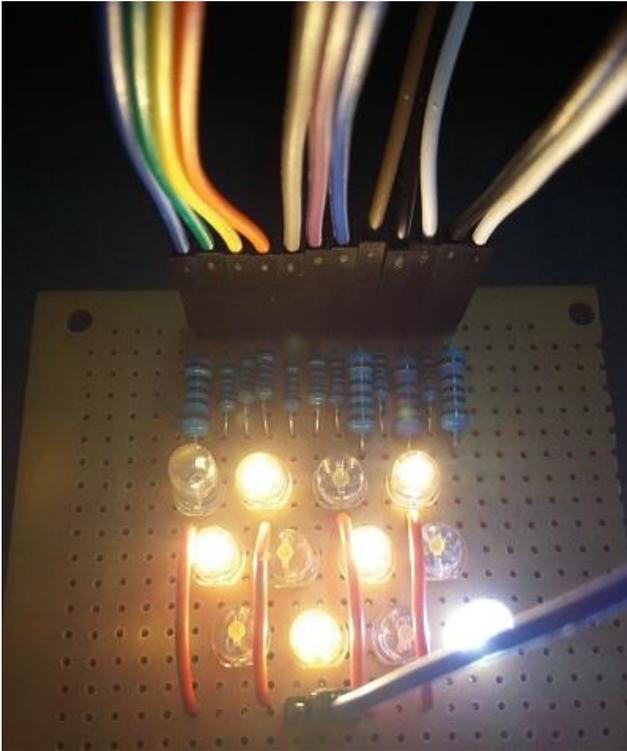


Gambar 11. Rangkaian LED untuk pengujian

Rangkaian LED terdiri dari 12 LED dimana sebuah LED merepresentasikan sebuah *relay* dari sebuah *control valve*. Kombinasi data yang dikirimkan dan kombinasi LED yang seharusnya aktif dapat dilihat pada tabel kebenaran pada lampiran 4.

Dari proses pengujian didapatkan hasil sesuai dengan tabel kebenaran yang ada. Pengujian dilakukan dengan mengikuti langkah – langkah yang sudah dijelaskan sebelumnya. Hal ini membuktikan bahwa langkah –

langkah untuk komunikasi dan pemetaan data dapat digunakan untuk mengirimkan data dari MATLAB menuju Arduino. Contoh dari pembuktian dapat dilihat dari gambar 12.

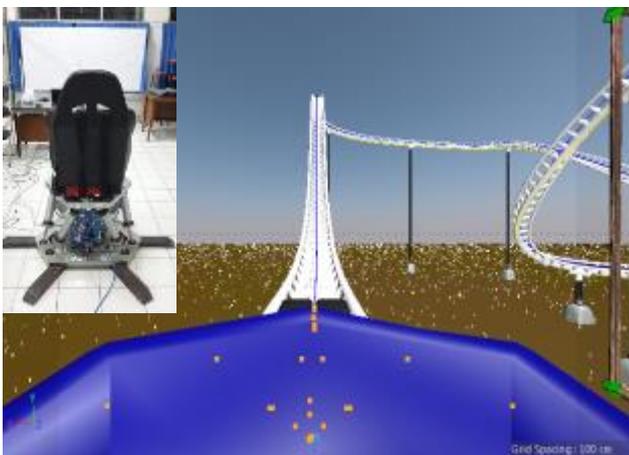


Gambar 12. Pengujian komunikasi MATLAB dan Arduino kombinasi (2,2,2,2,2,2).

**D. Pengujian Aktuator**

Pengujian aktuator ini dilakukan untuk mengetahui kecepatan dan ketepatan respon dari aktuator. Pengujian ini dilakukan dengan menjalankan sistem dari awal hingga akhir. Pengujian ini juga untuk menguji apakah kombinasi *state* yang didapatkan mirip secara visual dengan posisi dari *roller coaster* pada film.

Pengujian ini menggunakan 10 buah potongan gambar dari film yang dipilih acak kemudian dari pilihan 10 buah potongan gambar ditampilkan juga gerakan aktuatornya. Hasilnya dapat dilihat pada gambar 13 hingga gambar 22.



Gambar 13. Posisi awal dari *roller coaster*.



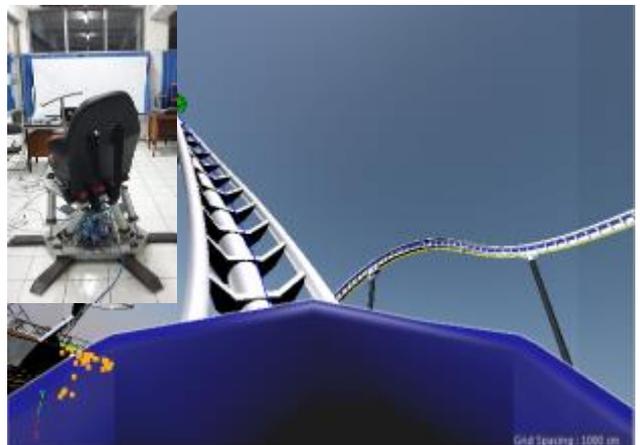
Gambar 14. *Roller coaster* pada posisi menanjak



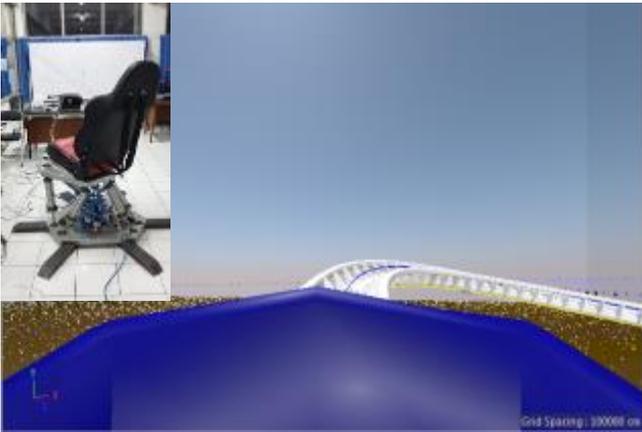
Gambar 15. *Roller coaster* pada posisi turun dan berbelok ke kanan



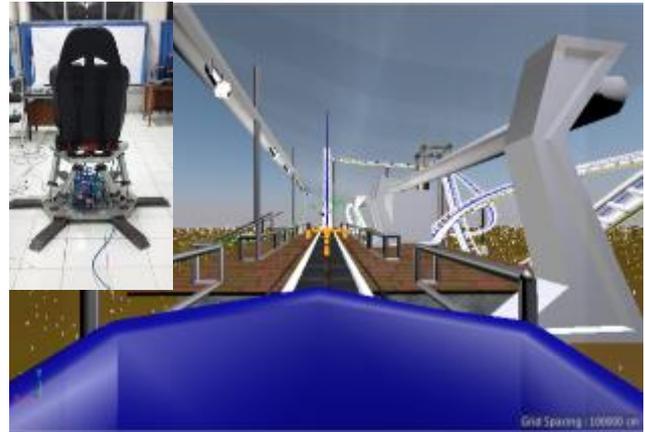
Gambar 16. *Roller coaster* pada posisi turun dan berbelok kiri.



Gambar 17. *Roller coaster* pada posisi menanjak dan berbelok ke kiri.



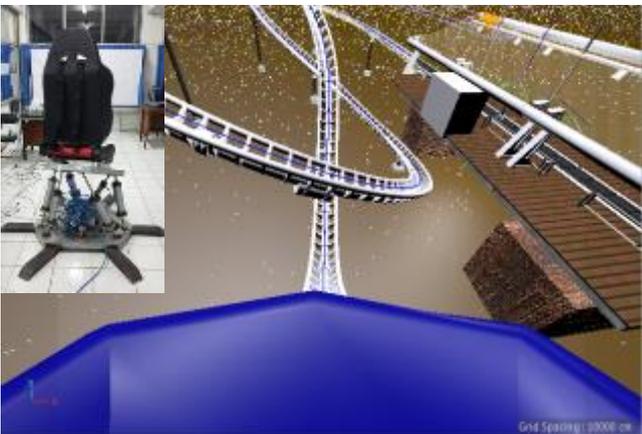
Gambar 18. *Roller coaster* pada posisi menanjak dan berbelok ke kanan



Gambar 22. *Roller coaster* pada posisi memasuki stasiun



Gambar 19. *Roller coaster* pada posisi berbelok ke kiri



Gambar 20. *Roller coaster* pada posisi turun



Gambar 21. *Roller coaster* pada posisi berbelok ke kanan.

Dari gambar – gambar tersebut diketahui bahwa sistem berjalan dengan baik. Tabel hasil dari *training data* dapat menyesuaikan gerakan – gerakan antara film dan aktuator. Terlihat pada Gambar sampai Gambar bahwa gerakan aktuator mirip dengan posisi *roller coaster* pada film.

Hasil dari pengujian juga dilihat dari perhitungan nilai kesalahan yang didapatkan. Perhitungan menggunakan persamaan sebagai berikut :

$$Error = \sqrt{\frac{\sum_{n=1}^k (Y_n - Y_{n'})^2}{k}} \quad (4.1)$$

Dimana :

k = jumlah data

$Y_n$  = data sudut input

$Y_{n'}$  = data sudut kursi

Persamaan 4.1 merupakan persamaan umum untuk menghitung kesalahan. Dari persamaan diatas didapatkan kesalahan sudut rata – rata. Nilai kesalahan sudut rata – rata tersebut kemudian dibagi dengan nilai sudut maksimal untuk mengetahui berapa persen kesalahan sudut yang didapat. Persamaan untuk menghitung persen kesalahan dapat dilihat pada persamaan 4.2.

$$\%Error = \frac{Error}{z} \times 100\% \quad (4.2)$$

Dimana :

Error = nilai error dari perhitungan pertama

z = sudut maksimal dari komponen *pitch*

atau *roll* atau *yaw*

Nilai kesalahan sudut yang didapat dari perhitungan, dihitung per komponen, baik itu komponen *pitch* atau *roll* atau *yaw*. Untuk menghitung kesalahan komponen *pitch*, maka nilai yang digunakan adalah nilai dari *pitch* saja. Demikian juga untuk perhitungan *roll* dan *yaw*. Perhitungan kesalahan untuk sistem menggunakan kesalahan komponen yang ada kemudian dibagi dengan jumlah komponen.

$$Error\ sistem = \frac{error\ pitch + error\ roll + error\ yaw}{x} \quad (4.3)$$

Dimana :

*Error pitch* = nilai kesalahan untuk komponen *pitch*

$Error\ pitch = \text{nilai kesalahan untuk komponen roll}$

$Error\ pitch = \text{nilai kesalahan untuk komponen yaw}$

$X = \text{jumlah komponen yang dihitung}$

Nilai kesalahan dari sistem adalah 0,0743 atau 7,4 %. Nilai kesalahan dari komponen *pitch* 0,0624 atau 6.2%. Nilai kesalahan dari komponen *roll* adalah 0,0401 atau 4,01%. Nilai kesalahan dari komponen *yaw* adalah 0,1205 atau 12,05%. Dari nilai kesalahan yang didapatkan, disimpulkan bahwa hasil dari *training data* dapat digunakan sebagai tabel acuan untuk proses eksekusi. Waktu yang diperlukan untuk memproses semua data *input* adalah 5,5 detik.

Sebagai pembandingan, maka dilakukan pemrosesan data dengan menggunakan tabel hasil pengujian kursi. Hasil pengujian dapat dilihat pada tabel 2. Proses perhitungan tidak dilakukan menggunakan *training data*.

Tabel 2. Perbandingan pengujian dengan dan tanpa *training data*

Perbandingan	Dengan <i>training data</i>	Tanpa <i>training data</i>
Nilai kesalahan <i>pitch</i>	6.2%	4.2%
Nilai kesalahan <i>roll</i>	4,01%	3,05%
Nilai kesalahan <i>yaw</i>	12,05%	7,08%
Nilai kesalahan sistem	7,4 %	4,8%
Waktu	5,5 detik	20,67 detik

Nilai kesalahan sistem yang didapat dari proses ini adalah 0,0479 atau 4,8%. Kesalahan yang didapat lebih kecil karena probabilitas lebih besar dibanding hasil dari *training data*. Nilai kesalahan dari komponen *pitch* 0,0422 atau 4.2%. Nilai kesalahan dari komponen *roll* adalah 0,0305 atau 3,05%. Nilai kesalahan dari komponen *yaw* adalah 0,0708 atau 7,08%. Waktu yang diperlukan untuk proses perhitungan dengan metode ini adalah 20,67 detik. Waktu yang diperlukan lebih lama karena jumlah data yang dibandingkan lebih banyak. Untuk pengujian ini, digunakan komputer dengan *processor* Core i3 2,30GHz, dengan RAM 6GB.

Komunikasi dari MATLAB menuju Arduino, kemudian menuju *driver*, berikutnya menuju *control valve* dan aktuator pneumatik berjalan dengan baik dengan mengetahui semua data terkirim pada *output* sesuai dengan *input*. Program pemetaan dari *state* pneumatik hingga menjadi logika biner sesuai dengan yang dibutuhkan, hal ini terlihat dari gerakan pada pneumatik sesuai dengan *state* yang diberikan oleh MATLAB.

#### IV. KESIMPULAN

Dari perancangan, pembuatan desain, dan pengujian sistem dapat ditarik kesimpulan sebagai berikut :

- Arus yang terukur pada driver bernilai rata – rata 173,85, nilai dari arus ini menurun bila dibandingkan dengan menghubungkan langsung control valve dan power supply yang memiliki rata – rata nilai 195mA. Hal ini terjadi karena arus yang diambil oleh IC yang digunakan.
- Nilai RMSE dari training data adalah 0,0474 atau 4,74 %, hal ini menunjukkan bahwa hasil training data dapat

digunakan untuk proses eksekusi dengan nilai kesalahan tidak lebih dari 10%.

- Nilai kesalahan dari komponen *pitch*, *roll*, dan *yaw* sangat bagus yaitu 6,2%, 4,01%, dan 12,05% hal ini menunjukkan bahwa tabel hasil dari training data sudah bagus dan mampu mengakomodasi gerakan *pitch*, *roll*, *yaw* dengan baik.
- Kombinasi yang dapat dihasilkan oleh pneumatik terbatas sehingga ada data yang akhirnya tidak terakomodasi sempurna dan pada akhirnya digunakan kombinasi untuk sudut yang lain.

#### DAFTAR PUSTAKA

[1] D. Stewart, "A Platform with Six Degrees of Freedom," *Proc. Inst. Mech. Eng.*, vol. 180, no. 15, 1965.

[2] F. Pasila, R. Alimin, and H. Natalius, "NEURO-FUZZY ARCHITECTURE OF THE 3D MODEL OF MASSIVE PARALLEL ACTUATORS," *ARPN J. Eng. Appl. Sci.*, vol. 9, no. 12, pp. 2900–2905, 2014.

[3] J. Rowberg, "jrowberg/i2cdevlib," *GitHub*, 2014. [Online]. Available: <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050/Examples>.