

Implementasi DOF (*Degree Of Freedom*) Pada Pergerakan Motor *Stepper Smart Inventory 3 Axis*

Ali Ripaldo¹, Yordan Hasan^{2*}, Johansyah Al Rasyid³

1,2,3 Prodi Teknik Elektro Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Indonesia

e-mail: Alirivaldo@gmail.com, yordan.hasan@gmail.com, johansyahalrasyid01@gmail.com

ABSTRAK

Perkembangan teknologi industri saat ini memang tidak bisa diragukan lagi semenjak adanya *E-Commerce* yang sangat efisien dan efektif dalam menunjang perekonomian masyarakat Indonesia. Hal ini semakin kuat mendorong setiap *E-Commerce* untuk lebih berkembang lebih baik lagi agar dapat bersaing kedepannya untuk mendapatkan kepuasan pelanggan. Oleh sebab itu dibutuhkan system yang dapat mengurangi pemakaian tenaga manusia dan daya ingat manusia untuk mengontrol setiap barang yang tersedia di dalam *inventory* dari setiap masing-masing *E-Commerce*. *Smart Inventory* merupakan sebuah system yang kita biasa sebut Gudang untuk menyimpan barang agar dapat tersimpan dengan rapi dan baik, dengan mengurangi resiko nya data ataupun stok barang yang sering terjadi karena kelalaian manusia dalam menginput barang yang masuk ke dalam Gudang, Penelitian ini bertujuan untuk merancang suatu system yang dapat membantu pengolahan data dan penempatan barang secara akurat dan efisien dengan menggunakan aplikasi Python dibantu oleh sensor RFID maka pendataan dan masuknya barang kedalam Gudang lebih efisien dan efektif, terdapat 4 motor stepper sebagai *transporter* barang yang akan di masukkan kedalam rak yang sudah ditentukan oleh kartu member masing-masing dan dibantu proses nya oleh Raspberry Pi sebagai mikroprosesor dalam pengolahan data base.

Kata Kunci: *Smart Inventory, Python, Motor Stepper*

Implementation of DOF (Degree Of Freedom) in Smart Inventory 3 Axis Stepper Motor Movement

ABSTRACT

The development of industrial technology at this time cannot be doubted since the existence of *E-Commerce* which is very efficient and effective in supporting the economy of the Indonesian people. This is increasingly encouraging every *E-Commerce* to develop even better so that it can compete in the future to get customer satisfaction. Therefore a system is needed that can reduce the use of human power and human memory to control every item available in the inventory of each *E-Commerce*. *Smart Inventory* is a system that we usually call Warehouse for storing goods so that they can be stored neatly and properly, by reducing the risk of missing data or stock items that often occur due to human negligence in inputting goods into the Warehouse. This study aims to design a system that can help manage data and place goods accurately and efficiently by using Python applications assisted by RFID sensors so that data collection and entry of goods into the Warehouse is more efficient and effective, there are 4 stepper motors as goods transporters that will be put into predetermined shelves by each member card and assisted by the Raspberry Pi as a microprocessor in processing the data base.

Keywords: *Smart Inventory, Python, Motor Stepper*

Correspondence author : Yordan Hasan, Prodi Teknik Elektro Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Indonesia.
E-Mail: yordan.hasan@gmail.com

PENDAHULUAN

Perkembangan teknologi semakin pesat didalam kehidupan manusia untuk meningkatkan efisiensi dan akurasi, *Smart Inventory* adalah sebuah system yang menggunakan teknologi canggih seperti Robot, Sensor RFID, dan lain-lain untuk mengumpulkan dan menganalisis data *real-time* tentang lokasi, stok, dan tempat penyimpanan barang itu sendiri [5]. Sistem ini dapat mengotomatisasi tugas-tugas manual seperti pelacakan, perhitungan barang, dan pembaruan persediaan barang, yang biasanya dilakukan manual oleh manusia atau operator di dalam Gudang tersebut, oleh sebab itu untuk meningkatkan efisiensi dan meningkatkan kepuasan pelanggan banyak *E-commerce* menggunakan smart inventory sebagai system penyimpanan barang secara otomatis pada Gudang mereka [14]. Sebagai mode *transport* atau pengangkut barang digunakan motor stepper prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor stepper mempunyai magnet tetap pada rotor [6][10]. Motor stepper memiliki torsi yang besar sehingga kuat dan presisi. Dengan menggunakan Motor stepper yang memiliki torsi besar maka motor stepper dapat digunakan sebagai penarik van belt yang terdapat pada Project *Smart Inventory* kali ini [8].

Motor *stepper 3 axis* dapat menggerakkan *Belt Conveyor* yang dikendalikan kecepatan dan perputaran nya oleh DOF (*Degree Of Freedom*). DOF (*Degree Of Freedom*) sendiri adalah jumlah parameter yang diperlukan untuk menentukan posisi dan orientasi motor secara lengkap [9]. DOF (*Degree Of Freedom*) pada motor dapat berupa translasi (Perpindahan) atau rotasi (Putaran) pada sumbu-sumbu tertentu. Misalnya, motor roda dua memiliki dua derajat kebebasan translasi (Maju-Mundur dan Kiri-Kanan) dan satu derajat kebebasan rotasi (Miring) sedangkan pada *Motor Stepper* hanya memiliki satu derajat kebebasan rotasi (berputar pada sumbu poros) (*Clock Wise*) atau CCW (*Counter Clock Wise*) *Clock Wise* dan *Counter*. *Clock Wise* adalah istilah yang digunakan untuk menentukan arah putaran motor itu sendiri, dengan menentukan nilai dan arah putaran tersebut kita dapat mengatur motor sesuai arah dan jarak yang kita inginkan [3][11]

Sistem *Inventory* adalah system pengaturan data persediaan barang yang berkaitan dengan aktivitas logistic sebuah perusahaan. Barang-barang tersebut disimpan dengan tujuan tertentu dan bergantung pada skala perusahaan dan kebijakan management Gudang. System ini memungkinkan perusahaan untuk mengontrol dan memantau persediaan barang mereka dengan teratur dan efisien [7]

Driver Motor adalah perangkat penting yang menyediakan tegangan dan arus yang diperlukan ke motor *stepper* sehingga mendapat operasi yang lancar. *Driver Motor* pada motor *stepper* sendiri berfungsi untuk mengatur jumlah step yang dikirimkan dari driver ke motor *stepper*. *Driver A4988* digunakan pada motor *stepper* karena mempunyai arus maksimum sebesar 5A [2]



Gambar 1. Driver A4988

Motor *Stepper* adalah motor yang dikendalikan dengan input digital menghasilkan *step-step* yang digunakan untuk mengontrol sudut dari motor. Motor *Stepper* memiliki torsi yang besar sehingga kuat dan presisi. Untuk menunjang itu semua motor *stepper* memerlukan driver untuk mengatur tegangan dan arus yang dibutuhkan oleh motor *stepper* tersebut. Motor *Stepper* NEMA 17 adalah jenis motor *stepper* dengan sudut Langkah 1.8° (200 langkah/revolusi). Setiap fase menggambar arus sebesar 1.2A pada 4V, memungkinkan untuk torsi holding sebesar 3.2kg-cm. NEMA 17 umumnya digunakan pada printer, mesin CNC dan *laser cutter* [12].



Gambar 2. Motor Stepper NEMA 17

Limit switch merupakan suatu jenis saklar dan dilengkapi katup yang berfungsi sebagai pengganti tombol, yang hanya menghubungkan arus saat katup ditekan pada batas dan memutuskan arus saat katup tidak ditekan. *Limit switch* termasuk kategori sensor mekanis, di mana akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor [13].



Gambar 3. Limit Switch

Pada penelitian ini menggunakan *Raspberry Pi* sebagai mikroprosesornya, *Raspberry Pi* adalah computer mini yang dapat digunakan untuk berbagai keperluan seperti membuat server, media center, dan banyak lagi. *Raspberry Pi* sangat mudah digunakan dan dipraktikkan. *Raspberry pi 4* terdiri dari RAM (*random access memory*), *Radio Module*, USB (*universal radio bus*), PMIC (*power management integrated circuit*), *Ethernet Port*, SOC (*system on chip*), GPIO (*general purpose input output*), dan *Micro SD*. *Raspberry Pi* menggunakan aplikasi *Python* sebagai bahasa programnya [4].



Gambar 4. RaspBerry Pi

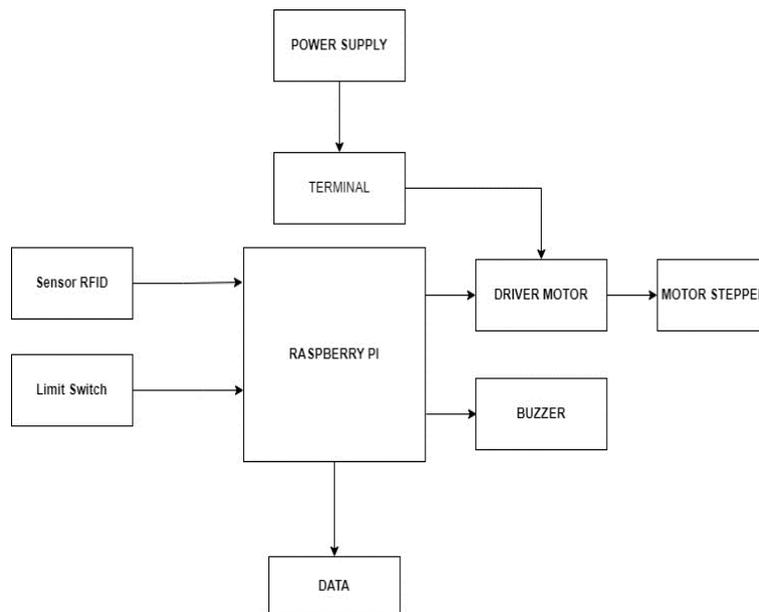
Python adalah bahasa pemrograman yang multifungsi dan serbaguna. *Python* dapat digunakan untuk membangun situs web, menganalisis data, penerapan algoritma *machine learning*, membuat aplikasi, otomatisasi dan banyak lagi. *Python* merupakan bahasa pemrograman tingkat tinggi yang lebih mudah dipahami manusia, di mana *python* memiliki beberapa kelebihan dibanding bahasa pemrograman lain. *Python* memiliki struktur sintaksis yang sederhana dan mudah dipahami [1]. *Python* lebih berorientasi kepada objeknya dan sangat mudah dikembangkan dengan cara menciptakan modul-modul yang baru, di mana *python* dapat menjalankan program yang rumit menggunakan sintaks yang sederhana. Selain itu, *python* juga bersifat *open source* dan memiliki banyak *library*.



Gambar 5. Logo Python

METODE PENELITIAN

Pada penelitian kali ini, perancangan alat yang dibuat berupa pembuatan robot *Smart Inventory* yang bergerak ke dalam rak yang telah disesuaikan oleh sensor RFID untuk menaruh barang ke dalam rak. Proses pengecekan kartu *member* dilakukan oleh sensor RFID yang akan menggerakkan *output* motor menuju rak yang telah disesuaikan oleh kartu *member* tersebut. Masing-masing kartu *member* dapat meletakkan barang sebanyak 3x dengan slot yang sama dan tidak dapat meletakkan barang pada slot yang berbeda. Percobaan ini dilakukan sebanyak 3x dan apabila sensor RFID telah mendeteksi bahwa *member* sudah meletakkan barang sebanyak 3x, maka akan ada penolakan dari RFID yang ditandai oleh *buzzer* berbunyi sebanyak 2x (*Beep-Beep*). Berikut gambar diagram *Blog Smart Inventory* pada penelitian yang dilakukan.

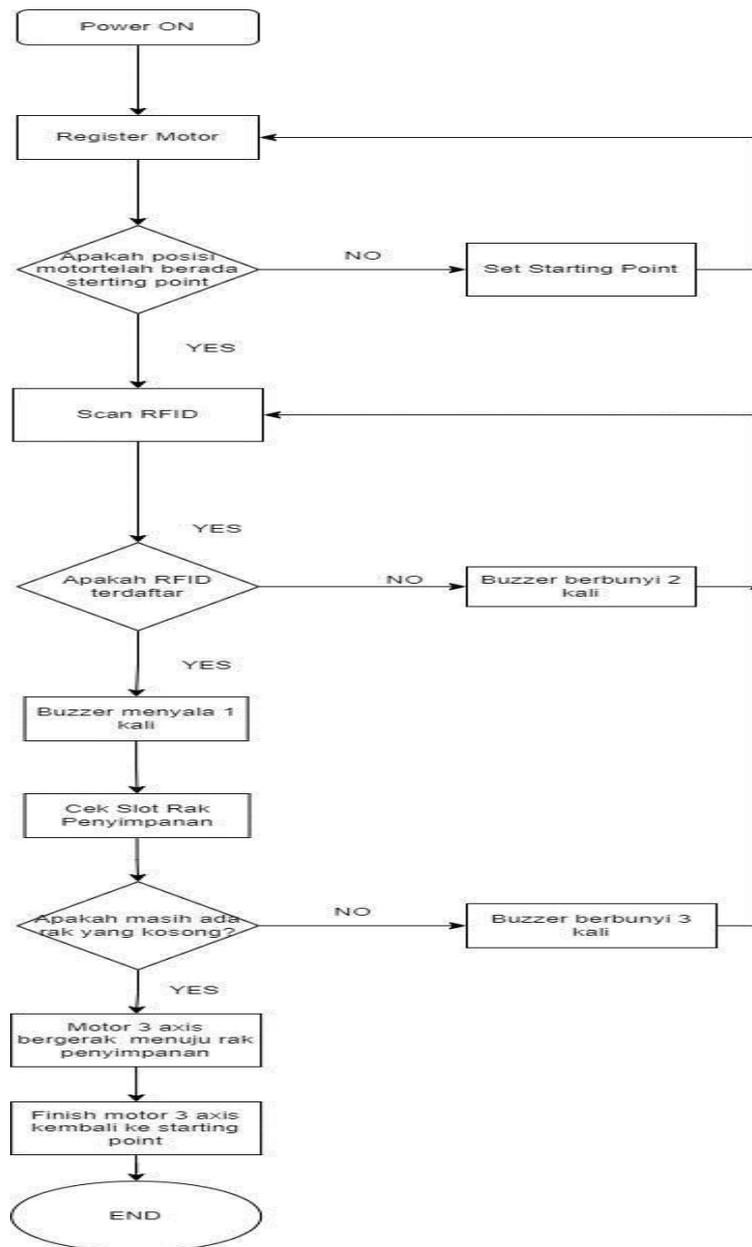


Gambar 6. Diagram *Blog Smart Inventory*

Flow Chart Smart Inventory Memiliki tahapan proses-proses kerja agar *conveyor* dapat mengantarkan barang ke pada rak yang telah ditentukan, yaitu:

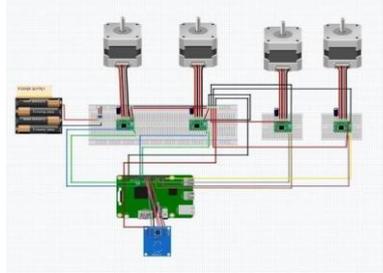
1. Proses awalan atau *Home Position* adalah ketika alat dihidupkan maka motor akan kembali pada *set point* nya masing-masing, yang diawali dengan X dan Y lalu disusul oleh Z.
2. Proses kedua adalah ketika ditempelkan kartu membernya pada RFID dan RFID memproses kartu tersebut, apabila kartu tersebut terdeteksi adalah member dari *Smart Inventory*, maka RFID akan mengirim data ke motor, apabila kartu yang ditempelkan bukan member dari *Smart*

- Inventory* maka RFID akan menghidupkan *buzzer* sebanyak 2x (*Beep-Beep*).
3. Proses ketiga adalah setelah pengecekan kartu yang dilakukan RFID, maka motor akan bergerak menuju rak yang telah ditentukan oleh kartu member tersebut. Motor akan kembali ke posisinya awal (*set point*) setelah menaruh barang sesuai rak yang di tentukan.
 4. Proses yang terakhir adalah pada saat member tersebut sudah meletakkan barang nya sebanyak 3 barang, maka sensor RFID akan menolak dengan indikasi *buzzer* berbunyi sebanyak 3x (*Beep-Beep-Beep*)



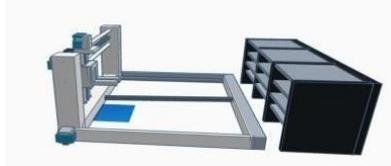
Gambar 7. Flow Chart SmartInventory

Skema rangkaian ini menampilkan keseluruhan komponen yang digunakan. Skema rangkaian elektronik untuk sistem *Smart Inventory* dapat dilihat pada gambar dibawah ini



Gambar 8. Skema Rangkaian

Perancangan mekanik adalah proses merancang suatu produk atau system mekanik yang melibatkan pemilihan bahan, pemilihan ukuran dan bentuk komponen serta pemilihan metode pembuatan. Perancangan mekanik melibatkan prinsip fisika dan matematika untuk menentukan bagaimana suatu produk atau system akan berfungsi. Perancangan mekanik juga melibatkan penggunaan teknologi komputer untuk membantu dalam proses perancangan. Berikut rancangan mekanik alat *Smart Inventory* dibawah ini.



Gambar 9. Desain Mekanik Alat

HASIL DAN PEMBAHASAN

Pengujian ini dilakukan sebanyak 3x dengan masing-masing rak memiliki perbedaan *number of steps* nya masing-masing berdasarkan jarak dan tinggi nya yang dilakukan awal secara manual dengan cara mengukur satu persatu jarak antara rak dari rak A sampai rak C untuk menetapkan akurasi, maka motor digerakkan dimulai dari motor X yang bergerak kiri-kanan, dilanjutkan oleh motor Y yang bergerak turun-naik dan di akhiri oleh motor Z sebagai pengantar maju-mundur, dengan cara manual ini dapat kita lihat akurasi dan ketepatan dari system *smart inventory*. Pengukuran dan percobaan dapat dilihat menggunakan perhitungan dan data di bawah ini

Pengukuran Manual rak A

Dari pengujian pertama terhadap rak A maka didapat nilai *Number Of Steps* dari setiap motor XYZ

Tabel 1. Data Pengujian *Number Of Steps* pada Rak A

Rak A	Motor Y	Motor X	Motor Z	Motor Y-
	+ 300	+4000	+3200	-400

Dari data Tabel 1 diatas, pengujian dilakukan sebanyak 3x yaitu dari motor Y yang bergerak *clock wise* dengan *number of steps* sebesar +300, dilanjutkan motor X yang bergerak *clock wise*



dengan *number of steps* sebesar +4000, lalu motor Z yang bergerak *clock wise* dengan *number of steps* sebesar +3200, dan diakhiri oleh Motor Y yang bergerak *counter clock wise* dengan *number of steps* sebesar -400.



Gambar 10. Pengukuran Manual Rak A

DOF = N-K

```
File Edit Tabs Help
Motor type = A4988
Clockwise = True
Step Type = Full
Number of steps = 3200
Step Delay = 0.0005
Initial delay = 0.05
Size of turn in degrees = 5760.0
Steps count 400
RpiMotorLib, Motor Run finished, Details:.

Motor type = A4988
Clockwise = False
Step Type = Full
Number of steps = 400
Step Delay = 0.0005
Initial delay = 0.05
Size of turn in degrees = 720.0
Steps count 3700
RpiMotorLib, Motor Run finished, Details:.

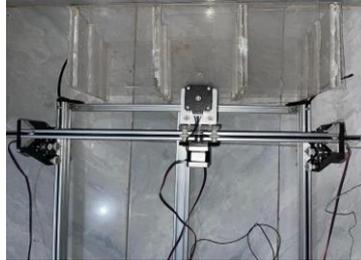
Motor type = A4988
Clockwise = False
Step Type = Full
Number of steps = 3700
Step Delay = 0.0005
Initial delay = 0.05
Size of turn in degrees = 6660.0
Steps count 3700
RpiMotorLib, Motor Run finished, Details:.

Motor type = A4988
Clockwise = True
Step Type = Full
Number of steps = 3700
Step Delay = 0.0005
Initial delay = 0.05
Size of turn in degrees = 6660.0
Stop Motor Interrupt : RpiMotorLib:
Stop Motor Interrupt : RpiMotorLib:
No. Kartu: 40901325783
No. Rak: 1
```

Gambar 11. Tampilan *Number Of Steps* Motor Rak A pada Aplikasi Python



Pengukuran Manual Rak B



Gambar 12. Pengukuran Manual Rak B

```

Motor type = A4988
Clockwise = True
Step Type = Full
Number of steps = 3400
Step Delay = 0.0005
Intial delay = 0.05
Size of turn in degrees = 6120.0
Stop Motor Interrupt : RpiMotorLib:
Stop Motor Interrupt : RpiMotorLib:
Steps count 400
RpiMotorLib, Motor Run finished, Details:.

Motor type = A4988
Clockwise = False
Step Type = Full
Number of steps = 400
Step Delay = 0.0005
Intial delay = 0.05
Size of turn in degrees = 720.0
Steps count 3700
RpiMotorLib, Motor Run finished, Details:.

Motor type = A4988
Clockwise = True
Step Type = Full
Number of steps = 3700
Step Delay = 0.0005
Intial delay = 0.05
Size of turn in degrees = 6660.0
Steps count 3700
RpiMotorLib, Motor Run finished, Details:.

Motor type = A4988
Clockwise = False
Step Type = Full
Number of steps = 3700
Step Delay = 0.0005
Intial delay = 0.05
Size of turn in degrees = 6660.0
Stop Motor Interrupt : RpiMotorLib:
Stop Motor Interrupt : RpiMotorLib:
No. Kartu: 803118258889
No. Rak: 2
Total item: 1
=====
Scan kartu member....
    
```

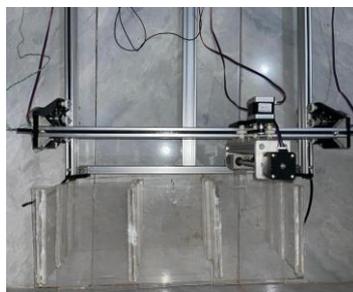
Gambar 13. Tampilan *Number Of Steps* Motor Rak B pada Aplikasi Python

Tabel 2. Data Pengujian Number Of Steps pada Rak B

Rak B	Motor Y	Motor X	Motor Z	Motor Y-
	+ 1600	+4000	+3800	-400

Dari data Tabel 2 pengujian dilakukan sebanyak 3x yaitu dari motor Y yang bergerak *clock wise* dengan *number of steps* +1600, dilanjutkan motor X yang bergerak *clock wise* dengan *number of steps* sebesar +4000, lalu motor Z yang bergerak *clock wise* dengan *number of steps* sebesar +3800, dan diakhiri oleh Motor Y yang bergerak *counter clock wise* dengan *number of steps* sebesar -400.

Pengukuran Manual Rak C



Gambar 14. Pengukuran Manual Rak C

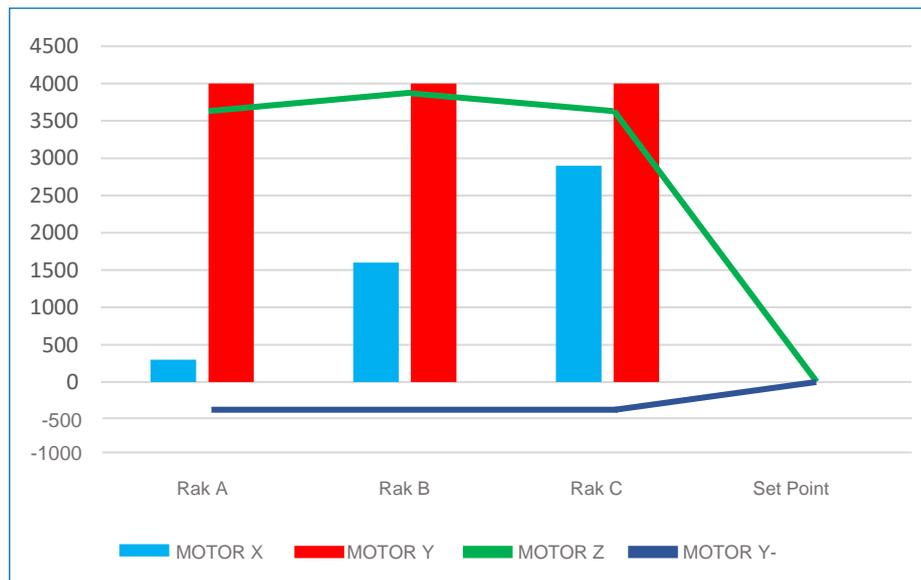
Dilanjutkan pengujian kedua terhadap rak C, maka di dapat nya nilai *Number Of Steps* dari setiap motor XYZ

Tabel 3. Data Pengujian Number Of Steps pada Rak C

Rak B	Motor Y	Motor X	Motor Z	Motor Y-
	+ 2900	+4000	+3200	-400

Dari data Tabel 3 diatas, pengujian dilakukan sebanyak 3x yaitu dari motor Y yang bergerak *clock wise* dengan *number of steps* sebesar +2900, dilanjutkan motor X yang bergerak *clock wise* dengan *number of steps* sebesar +4000, lalu motor Z yang bergerak *clock wise* dengan *number of steps* sebesar +3200, dan diakhiri oleh Motor Y yang bergerak *counter clock wise* dengan *number of steps* sebesar -400.

Dari Gambar 15 dapat dilihat bahwa *Number Of Steps* dari setiap rak banyak memiliki kesamaan nilai, di mana motor Y yang memiliki nilai *Number Of Steps* (+4000) dan motor Y- (-400). Hal ini dikarenakan tinggi yang sama antara rak A, B, dan C, sehingga motor Y tidak perlu dirubah lagi nilainya agar mendapat akurasi yang pas. Tetapi berbeda dengan motor X yang memiliki *Number Of Steps* yang berbeda, dikarenakan jarak antara set point menuju rak A, B, dan C berbeda, sehingga diperlukan pengukuran manual berkali-kali untuk mendapatkan titik yang pas, agar sesuai dengan tujuan rak masing-masing. Dan *Number Of Steps* terhadap motor Y- mendapatkan nilai minus dikarenakan motor Y- bergerak secara CWW (*counter clock wise*) yang artinya motor Y- bergerak mundur ketika motor XYZ mencapai *Number Of Steps* yang ditentukan sejak awal.



Gambar 15. Grafik Pergerakan *Number Of Steps* Motor XYZ dari rak A, B, dan C terhadap *Set Point*

KESIMPULAN

Sistem *Smart Inventory* dengan menggunakan motor *stepper* sebagai *transporter* antara *set point* menuju rak A, B, dan C berfungsi dengan baik, dengan melakukan percobaan menggerakkan motor satu persatu menuju titik rak A, B, dan C didapatkanlah nilai *Number Of Stepper* dari setiap masing-masing motor, diperlukan ketelitian dan konsentrasi untuk mendapat nilai akurasi yang tinggi sebagai acuan motor bergerak menuju rak A, B, dan C untuk itu percobaan yang dilakukan harus berkali-kali agar mendapatkan titik yang diinginkan, ukuran rak dan jarak antara rak A, B, dan C juga harus menjadi acuan untuk kita menilai berapa *Number Of Stepper* dari masing-masing motor dan juga masing-masing rak. *Limit Switch* disini berperan sangat penting sebagai *interrupted* motor ketika sudah sampai di *set point* nya, penempatan *Limit Switch* juga sangat berpengaruh penting terhadap stop nya motor dikarenakan terjadi beberapa kali motor tidak menyentuh *Limit Switch* tapi berhenti dengan sendirinya oleh sebab itu penempatan *Limit Switch* juga sangat penting dalam lancarnya sistem *Smart Inventory* kali ini.

DAFTAR PUSTAKA

- [1] Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *KARIMAH TAUHID*, 2(1), 1-6.
- [2] Andika, R. J., Rusdinar, A., & Wibowo, A. S. (2018). Perancangan dan Implementasi Driver Motor Tiga Fasa untuk Pengendali kecepatan motor BLDC berbasis PWM pada mobil listrik. *eProceedings of Engineering*, 5(1).
- [3] Febriansyah, A. A., Candra, H., & Sulaiman, S. (2022). Implementasi Voice Recognition pada Pengendalian Pergerakan Lengan Robot. *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*, 9(1), 86-95

- [4] Hafidz, A. F. M., Sani, M. I., & Meisaroh, L. (2021). Perancangan Dan Implementasi Smart Home Menggunakan Raspberry Pi Dan Esp8266. *eProceedings of Applied Science*, 7(6)
- [5] Herysyaputra, M. S., Ripanti, E. F., & Muhardi, H. (2021) *Smart Inventory* system untuk Distribusi Vaksin dengan Metode *Economic Order Quantity*. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 7(3), 455-46
- [6] Nugroho, N., & Agustina, S. (2015). Analisa motor DC (Direct Current) sebagai penggerak mobil listrik. *Jurnal Mikrotiga*, 2(1), 28
- [7] Paryanto, P. (2022). Pengembangan Prototype Smart Inventory System berbasis teknologi RFID untuk Industri Garmen. *ROTASI*, 24(3), 50-56
- [8] Prianto, E. (2017). Proses permesinan cnc dalam pembelajaran simulasi CNC. *Jurnal Edukasi Elektro*, 1(1)
- [9] Rahman, M. R. F. (2018). *Sistem Kendali Robot Lengan Empat DOF Berbasis Pengolahan Citra Dengan Sensor Kamera Kinect* (Doctoral dissertation, Universitas Gadjah Mada).
- [10] Soedjarwanto, N., Nama, G. F., & Nugroho, R. A. (2021). Prototipe Smart door lock Menggunakan Motor Stepper Berbasis IoT (Internet of Things). *Electrician*, 15(2), 73-82.
- [11] Suhendra, A., Prebianto, N. F., & Asaad, N. S. (2021). Optimasi Pergerakan Motor Stepper 3 Axis Dengan Metode Microstepping Dan Pembuatan Unique Code Pada Lead Frame. *Journal of Applied Sciences, Electrical Engineering and Computer Technology*, 2(1), 24-30
- [12] Suhendro, B., Antoro, L. M., & Suroso, S. (2020). Sistem Kendali Penggerak Motor Stepper Pada Orbital Welding Menggunakan Perangkat Lunak LabVIEW. In *Prosiding SNFA (Seminar Nasional Fisika dan Aplikasinya)* (Vol. 5).
- [13] Umam, K., Haryanto, H., & Alfita, R. (2019). Rancang Bangun Robot Pembersih Kaca Otomatis Berbasis Mikrokontroler ARM STM32 Dengan Sensor Proximity. *Indonesian Journal of Engineering and Technology (INAJET)*, 2(1), 24-29
- [14] Wahyudin, W., & Bela, S. (2021). Rancang Bangun Sistem Informasi Inventory Stock Barang Berbasis Web. *Jurnal Teknik Komputer*, 7(2), 208-214

