PERANCANGAN *SMART TROLLEY* MENGGUNAKAN SENSOR IMU (*INERTIAL MEASUREMENT UNIT*) BERBASIS KENDALI PI

SMART TROLLEY DESIGN USING SENSOR IMU (INERTIAL MEASUREMENT UNIT) BASED ON PI

Vitriyani[1], Ir. Porman Pangaribuan, M.T.[2], Agung Surya Wibowo.,ST.,MT[3]

[1,3]Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

[1]vitriyani@students.telkomuniversity.ac.id, [2]porman@telkomuniversity.ac.id, [3]agungsw@telkomuniversity.ac.id

Abstrak

Hingga saat ini pada industri rumah makan menggunakan tenaga manusia untuk mengantar dan membawa makanan. Namun sering kali manusia melakukan kesalahan seperti menjatuhkan atau menumpahkan makanan dan minuman. Oleh karena itu troli makanan dapat menjadi alternatif sebagai alat pengantar makanan. Tetapi terkadang penggunaan troli masih menghadapi kendala seperti tumpahnya makanan dan minuman ketika melewati jalan yang tidak rata. Hal itu dikarenakan wadah makanan atau minuman pada troli tidak stabil.

Maka dari itu penelitian ini bertujuan untuk merancang sistem *smart trolley* dengan menggunakan sensor IMU (*Inertial Measurement Unit*) dan dikendalikan dengan metode PID sebagai algoritmanya *Smart Trolley* ini akan dikendalikan oleh Arduino uno sebagai mikrokontroler dan motor servo sebagai pergerak wadah. Beberapa perangkat yang digunakan untuk merancang *smart trolley* yaitu sensor IMU, Arduino Uno dan motor servo.

Dengan adanya penelitian ini diharapkan dapat menjadi alternatif yang lebih baik dalam penyelesaian masalah tersebut. Karena dari hasil penelitian yang didapatkan memiliki nilai *overshoot* yang kecil, yaitu sebesar 6% pada sudut *Roll* dan 5% pada sudut *Pitch*.

Kata Kunci : *Smart Trolley*, Sensor IMU, Arduino Uno, Motor Servo, Kendali PID

Abstract

Until now in the restaurant industry uses human labor to deliver and bring food. But often people make mistakes such as dropping or spilling food and drink. Therefore, food trolleys can be an alternative as a food delivery tool. But sometimes the use of trolleys still face obstacles such as the spill of food and drink when passing uneven roads. That's because the food or drink container on the trolley is not stable.

Therefore this research aims to design smart trolley system by using IMU sensor (Inertial Measurement Unit) and controlled by PID method as its algorithm Smart Trolley will be controlled by Arduino Uno as a microcontroller and servo motor as container movement. Some devices used to design smart trolley that is IMU sensor, Arduino Uno, and servo motor.

With this research is expected to be a better alternative in solving the problem. Because the results obtained have a small overshoot value, which is 6% at Roll angle and 5% at Pitch angle.

Keywords: Smart Trolley, IMU Sensor, Arduino Uno, Servo Motor, PID

Pendahuluan

Troli merupakan alat pembawa barang secara manual dimana makanan dan minuman yang dibawa troli sering terjatuh atau tumpah ketika medan yang dilalui miring atau tidak rata. Sehingga dirancang troli yang memiliki wadah yang tetap datar walaupun melalui medan miring atau tidak rata. Menjaga posisi wadah yang selalu datar pada troli dibutuhkan sensor kestabilan, motor servo dan mikrokontroler dengan metode kendali PI.

Perancangan *smart trolley* menggunakan sensor IMU (*Inertial Meausurement Unit*). Karakteristik yang menjadi kelebihan sensor IMU adalah memanfaatkan sistem pengukuran seperti gyroskop dan akselerometer untuk memperkirakan sudut-sudut sikap dari troli seperti *pitch* dan *roll*. Sehingga perancangan sistem *smart trolley* mendapatkan posisi yang *settle*.

1. Sistem Kendali

Sistem kendali adalah kumpulan alat untuk mengendalikan, memerintah dan mengatur keadaan dari suatu sistem [1]. Parameter yang mempengaruhi kerja sistem kendali, yaitu: pengukuran, membandingkan, perhitungan, dan perbaikan. Terdapat dua jenis sistem kendali yaitu kendali *open loop* dan kendali *closed loop*. Perbedaan antara kedua Kendali tersebut yaitu kendali *open loop* tidak mempunyai blok umpan balik (*feedback*) sedangkan kendali *closed loop* mempunyai blok umpan balik (*feedback*). Pada sistem kendali *closed loop* sinyal *error* dapat di ketahui dari selisih antara *feedback* dan sinyal *input*, dimana pengendali akan mengurangi *error* dan akan memberikan *output* sistem sesuai yang diinginkan.

2. Dasar Teori Perancangan

2.1 Kendali PID

Sistem kontrol PID (*Proportional-Integral-Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut (Feed Back).

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*), dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan yang diinginkan. Algortima PID *independent* dapat direpresentasikan pada persamaan (2.1).

$$U(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} \quad (2.1)$$

Dimana:

U(t) = *output* kendali
e(t) = selisih antara *set point* dengan *output*
K_p = *gain* proporsional
K_i = *gain* integral
K_D = *gain derivative*

Sedangkan dalam bentuk PID *dependent* kita hanya perlu mengganti $K_i = K_p/T_i$ dan $KD = K_p.T_D$. Bisa dilihat bahwa bentuk *dependent* dari sebuah PID nilai dari K_i dan K_D bergantung dengan nilai dari K_p.

2.1.1. Kendali Proportional

Kendali Proportional ini dikenal sebagai penguatan / gain. Jika nilai Kp ini terus bertambah makan penguatan system juga akan bertambah sehingga dapat digunakan untuk memperbesar kecepatan respond an mengurangi *error steady state*. Penggunaan kendali proposional ini sering tidak memuaskan karena penambahan nilai Kp akan membuat system lenih sensitive dan cenderung mengakibatkan ketidakstabilan. Persamaan kendali proportional dapat direpresentasikan pada persamaan (2.2).

$$u(t) = Kp\ e(t) \quad (2.2)$$

2.1.2. Kendali Integratif

Fungsi utama dari kendali Integral adalah memastikan bahwa output akan sesuai dengan input pada keadaan steady state. Dengan kendali Integral, sedikit error positif akan selalu meningkatkjan sinyal control, dan sedikit error negative akan menurunkan sinyal control. Persamaan Ki dapat dipresentasikan pada persamaan (2.3).

$$u(t) = Ki \int_0^t \big(e(t)d(t)\big) \quad (2.3)$$

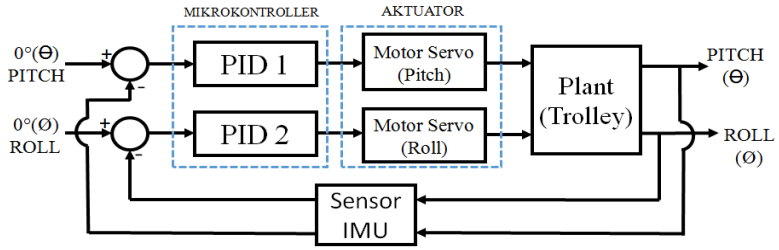
2.1.3. Kendali Derivatif

sifat dari kendali D ini dalam konteks "kecepatan" atau *rate* dari *error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi *error* yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan *error* sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler *Derivative* tidak dapat dipakai sendiri.

$$u(t) = Kd \frac{de(t)}{dt} \quad (4)$$

2.2 Blok Diagram

Sistem troli dirancang untuk menyeimbangkan wadah troli apabila terjadi perubahan sudut atau rotasi sehingga sistem akan otomatis bekerja untuk menyeimbangkan wadah troli.

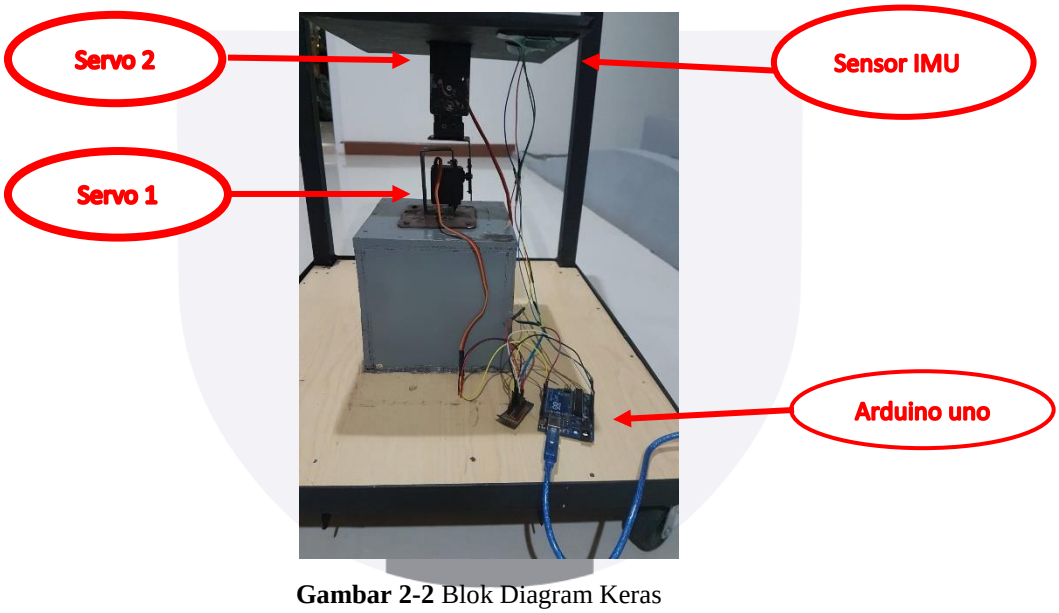


Gambar 2.1 Diagram Blok

Gambar 2-1 adalah blok diagram sistem kerja troli. Perancangan dibutuhkan sudut sebagai masukan dari sistem. Arduino Uno berperan sebagai mikrokontroler utama sistem, motor servo berperan sebagai aktuator dan troli berperan sebagai *plant*. *Feedback* sistem adalah sensor IMU dan keluaran yang dihasilkan adalah sudut. Perubahan sudut yang terjadi akan dibaca oleh sensor IMU kemudian akan diproses ke dalam Arduino Uno dan dikeluarkan dengan sinyal yang sesuai untuk menggerakkan aktuator (motor servo). Namun apabila sudut belum memenuhi keluaran yang diinginkan maka sistem akan terus bekerja.

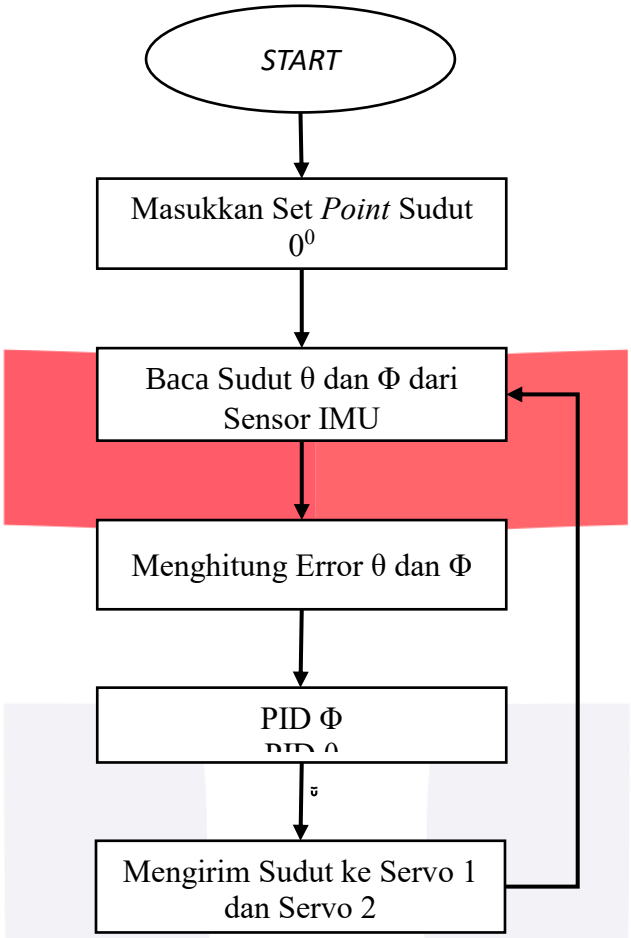
2.3 Desain Perangkat Keras

blok hardware pada tugas akhir ini dapat digambarkan sebagai berikut :



Gambar 2-2 Blok Diagram Keras

2.4 Diagram Alir



Gambar 2-3 Diagram Alir

Gambar 2-3 merupakan diagram alir program utama sistem. Penentuan *set point* dilakukan untuk mengatur sudut, *set point* pada sistem adalah 0^0 inisialisasi dilakukan oleh mikrokontroler. Setelah dilakukan penginisialisasi, mikrokontroler membaca informasi yang dikirim oleh sensor IMU dan menghitung *error* kemudian mikrokontroler mengirimkan sinyal untuk menggerakkan motor servo agar wadah troli menjadi seimbang. Keseimbangan yang didapatkan kemudian dibandingkan oleh sensor IMU dengan *set point*. Jika nilai sudah sesuai, maka mikrokontroller akan mengirim sinyal kepada aktuator sesuai dengan yang *set point* 0^0.

3. Hasil Analisa dan Pengujian

3.1. Pengujian Gerakan Motor Servo

Pengujian gerakan motor servo bertujuan untuk mengetahui berapa selisih nilai sudut yang terhitung oleh servo. Pengujian dilakukan dengan cara mengubah sudut pada servo dan membandingkan dengan sudut yang diberikan pada program Arduino Uno. Ditunjukkan seperti tabel 3-1 merupakan data perbandingan pembacaan sudut pada servo dan program Arduino Uno.

Tabel 3-1 Gerakan Motor Servo

No	Sudut pada Busur	Sudut pada Program	Selisih
1	0^0	0^0	0^0
2	20^0	25^0	0^0
3	40^0	45^0	0^0
4	50^0	50^0	0^0
5	100^0	100^0	0^0
6	150^0	150^0	0^0
7	180^0	180^0	0^0
Rata-Rata Error			0^0

Pengujian tabel 3-1 disimpulkan bahwa pergerakkan motor servo didapatkan sesuai dengan sudut yang diinginkan.

3.2. Pengujian pada Sumbu *Roll* dan *Pitch*

Pengujian gerakan terhadapa sumbu *roll* dan *pitch* dilakukan untuk melihat perbandingan sistem apabila nilai Kp dan Ki di ubah. Pengujian dilakukan dengan melihat respon sistem terhadap *error* yang terjadi. Hasil yang didapat adalah berupa data sudut dari sumbu *roll* dan sumbu *pitch*. Data yang didapat akan dibuat grafik dengan format waktu terhadap sudut. Grafik yang ditampilkan menunjukkan osilasi sudut yang terjadi ketika pengujian dilakukan terhadap berapa lama waktu yang dibutuhkan untuk mencapai keadaan *settle*. Sehingga sudut yang ditampilkan pada grafik merupakan sudut dalam bentuk derajat dan waktu yang ditunjukan dalam grafik merupakan waktu dalam hitungan mili detik.

3.2.1. Pengujian Pada Sumbu *Roll*

Pengujian ini dilakukan dengan Simulink matlab dan hardware.

Tabel 3-2 Hasil Percobaan Pada Sumbu *Roll*

Jenis Kontroler	Settling time	Sudut	Peak data simulasi	Peak data pengujian	%OS	Keterangan
P	0.5 detik	30^0	24^0	40^0	40%	Sistem masih mengalami osilasi yang besar diawal
PI	0.5 detik	20^0	23^0	21^0	10%	osilasi sistem semakin membaik namun *settling time* melambat
PI	1.5 detik	15^0	17^0	18^0	6%	*Overshoot* dan osilasi membaik namun *settling time* melambat

Tabel 3-2 adalah pengujian sumbu *roll* dimana sistem yang bekerja dengan menggunakan kontroler PI akan bekerja lebih baik, namun masih diperlukan penyesuaian *settling time* agar sistem bekerja lebih optimal. Sehingga sistem yang bekerja dengan *settling time* lebih lambat mampu mengurangi osilasi yang terjadi diawal sistem dan memperbaiki %OS nya.

3.2.2 Pengujian pada Sumbu Pitch

Tabel 3-3 Hasil Pengujian Sumbu *Pitch*

Jenis Kontroler	Settling time	Sudut	Peak data simulasi	Peak data pengujian	%OS	Keterangan
P	1.5 Detik	30^0	17^0	40^0	57.5%	Sistem masih mengalami osilasi yang besar diawal
PI	2 Detik	20^0	24^0	30^0	20%	osilasi sistem semakin membaik namun *settling time* melambat
PI	2.5 Detik	35^0	38^0	40^0	5%	osilasi sistem semakin membaik namun *settling time* melambat

Tabel 3-3 merupakan sistem yang bekerja dengan menggunakan kontroler PI bisa bekerja lebih baik, namun masih diperlukan penyesuaian *settling time* agar sistem dapat bekerja lebih optimal. Sehingga sistem yang bekerja dengan *settling time* lebih lambat mampu mengurangi osilasi yang terjadi diawal sistem dan memperbaiki %OS nya.

3.4. Pengujian Sistem dengan Menggunakan Beban

Tabel 3-4 Hasil Pengujian dengan Beban pada Sumbu *Roll*

Beban	**Sudut**	**Respon Waktu**
100gram	20^0	1.01 Sekon
200gram	20^0	1.16 Sekon
300gram	20^0	1.24 Sekon
400gram	20^0	1.44 Sekon
500gram	20^0	1.65 Sekon

Tabel 3-4 merupakan pengujian dengan beban dimana salah satu contohnya pengujian beban 100gram dengan sudut gangguan 20^0 maka respon waktu yang dibutuhkan adalah 1.01 detik selanjutnya pengujian dilakukan sebanyak 5 kali dengan menggunakan beban yang berbeda bahwa semakin berat beban yang ditampung maka semakin lama waktu respon motor servo bekerja.

Tabel 3-5 Hasil Pengujian dengan Beban pada Sumbu *Pitch*

Beban	Sudut	Respon Waktu
100gram	20^0	1.52 Sekon
200gram	20^0	1.72 Sekon
300gram	20^0	1.95 Sekon
400gram	20^0	2 Sekon
500gram	20^0	2.02 Sekon

Tabel 3-5 merupakan pengujian dengan beban dimana salah satu contohnya pengujian beban 100gram dengan sudut gangguan 20^0 maka respon waktu yang dibutuhkan adalah 1.52 detik selanjutnya pengujian dilakukan sebanyak 5 kali dengan menggunakan beban yang berbeda bahwa semakin berat beban yang ditampung maka semakin lama waktu respon motor servo bekerja.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis sistem yang dibuat penguji dapat menyimpulkan bahwa :

1. Dari hasil pengujian ini sensor IMU (*Inertial Measurement Unit*) dapat membaca sudut sikap (*pitch* dan *roll*).
2. Dari hasil pengujian pada sumbu *roll* didapatkan nilai parameter Kp = 0.412 dan Ki= 5.395 yang bekerja dengan ts = 1.5 detik agar sistem bisa bekerja dengan optimal dengan kondisi overshoot sebesar 6% atau 4% kurang dari *%overshoot* yang dirancang.
3. Dari hasil pengujian pada sumbu *pitch* didapatkan nilai parameter Kp=0.349 dan Ki= 3.092 yang bekerja dengan ts=2.5 detik agar sistem bisa bekerja dengan optimal dengan kondisi *overshoot* sebesar 5% atau 5% lebih dari *%overshoot* yang diancang.
4. Dari hasil pengujian didapat bahwa semakin berat beban yang ditampung maka semakin lama waktu respon motor servo bekerja.

5. Daftar Pustaka

[1] Shofiudin Aji, “SISTEM KONTROL OTOMATIS PADA AC SPLIT”, Universitas Negeri Semarang

[2] Muhammad Alfiansyah, Rudy Dikairono dan Pujiono, “Rancang Bangun Inertial Measurement Unit Untuk Unmanned Aerial Vehicles “Quadrotor””, JURNAL TEKNIK ITS Vol. 1, No. 1 (Sept. 2012).

[3] Ahmad Yani, “PEMBELAJARAN PERANCANGAN SISTEM KONTROL PID DENGAN MENGGUNAKAN SOFTWARE MATLAB”, STT HARAPAN MEDAN

[4] Ikhsan, Hendra Kurniawan, “IMPLEMENTASI SISTEM KENDALI CAHAYA DAN SIRKULASI UDARA RUANGAN DENGAN MEMANFAATKAN PC DAN MIKROKONTROLER ATMEGA8”, Vol. 3 No. 1 April 2015

[5] Frangki R. Misah, Sherwin R.U.A. Sompie, ST., MT, M. Dwisnanto Putro, ST., M.Eng, “Pengendalian Lengan Robot Pemindah Objek Dengan Smartphone Android”, vol. 4 no. 5 (2015)