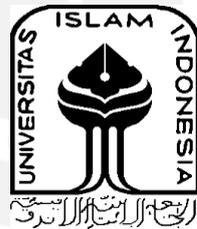


**Respon Atas Variasi Gangguan Setpoint pada Sistem Kontrol
PID Penyeimbang Bola dengan Penambahan Filter pada
Pembacaan Sensor Inframerah**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh:

Nama : Dzata Hadi Nugrah

No. Mahasiswa : 16525081

NIRM 2016080674

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

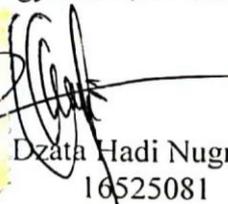
2021

PERNYATAAN KEASLIAN

Demi Allah, semua yang saya tulis pada Tugas Akhir ini adalah hasil karya saya sendiri kecuali kutipan atau ringkasan yang saya ambil sebagai referensi dan telah saya cantumkan sumber-sumbernya. Apabila dikemudian hari, pengakuan saya terbukti tidak benar, maka saya bersedia mengikuti hukuman ataupun sanksi yang diberikan, sesuai sanksi hukum yang berlaku di Universitas Islam Indonesia.

Yogyakarta, 17 Juni 2021




Dzata Hadi Nugrah
16525081

الجامعة الإسلامية
الاستدراالاتو

LEMBAR PENGESAHAN DOSEN PEMBIMBING
Respon Atas Variasi Gangguan Setpoint pada Sistem Kontrol
PID Penyeimbang Bola dengan Penambahan Filter pada
Pembacaan Sensor Inframerah

TUGAS AKHIR

Disusun Oleh:

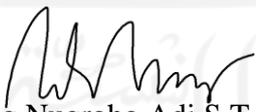
Nama : Dzata Hadi Nugrah

No. Mahasiswa : 16525081

NIRM 2016080674

Yogyakarta, 17 Juni 2021

Pembimbing I,


Agung Nugroho Adi S.T., M.T.

Pembimbing II,


Purtojo S.T., M.Sc.

LEMBAR PENGESAHAN DOSEN PENGUJI

Respon Atas Variasi Gangguan Setpoint pada Sistem Kontrol PID Penyeimbang Bola dengan Penambahan Filter pada Pembacaan Sensor Inframerah

TUGAS AKHIR

Disusun Oleh:

Nama : Dzata Hadi Nugrah

No. Mahasiswa : 16525081

NIRM : 2016080674

Tim Penguji

Agung Nugroho Adi S.T., M.T.

Ketua



Tanggal : 29 Juni 2021

Muhammad Ridlwan S.T., M.T.

Anggota I



Tanggal : 29 Juni 2021

Faisal Arif Nurgesang S.T., M.Sc.

Anggota II



Tanggal : 29 Juni 2021

Mengetahui

Ketua Jurusan Teknik Mesin




Dr. Eng. Risdiyono, ST, M.Eng.

HALAMAN PERSEMBAHAN

Tugas akhir ini disampaikan kepada kedua orang tua yang telah membiayai kuliah, memberikan kasih sayang serta memberikan pengetahuan yang sangat banyak untuk menjadi seseorang yang lebih baik dan masih banyak lagi hal yang sangat berharga yang telah diberikan yang akan sangat banyak jika gambarkan menggunakan kata-kata

Kepada kedua saudara yang sangat membantu dalam berhasilnya menyelesaikan kuliah.

Kepada dosen pembimbing 1 bapak Agung Nugroho Adi S.T., M.T. yang telah membantu dalam menyelesaikan tugas akhir, terima kasih juga untuk semua saran yang diberikan.

Kepada dosen pembimbing 2 bapak Purtojo S.T., M.Sc. yang telah membantu dalam pengerjaan tugas akhir, terima kasih juga untuk semua saran yang telah diberikan

Kepada teman-teman yang telah membantu menyemangati, memberi info, memberi canda dan tawa, meskipun kadang ada yang sering marah tidak jelas.

HALAMAN MOTTO

Hanya Kepada-Mu (Allah) Kami Menyembah Dan Hanya Kepada-Mu (Allah)
Kami Memohon Pertolongan
(Q.S Al-Fatihah, 5)

Jika Kalian Tidak Melakukannya Dan Tidak Akan Dapat Melakukannya, Maka
Takutlah Kepada Api Neraka Yang Materialnya Manusia Dan Batu. Api Neraka
Itu Disediakan Bagi Orang-Orang Kafir
(Q.S Al-Baqarah, 24)

Jangan Pernah Membatasi Diri Dalam Belajar, Karena Setiap Ilmu Yang Kita
Pelajari Akan Membantu Kita Di Masa Depan

KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Allah SWT, atas limpahan karunia dan rahmat yang telah diberikan kepada penulis sehingga dapat menyelesaikan tugas akhir sebagai salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi Mesin Fakultas Teknologi Industri Universitas Islam Indonesia.

Sholawat serta salam selalu diberikan kepada junjungan Nabi besar kita Muhammad SAW yang telah menyelamatkan umat dari Zaman kebodohan ke Zaman yang penuh karunia dari Allah SWT. Kemudian selama pelaksanaan tugas akhir, penyusun ingin berterima kasih kepada :

1. Orang tua tercinta serta kedua saudara. Yang selalu mendoakan dan memberi dukungan sehingga dapat terselesaikannya tugas akhir.
2. Bapak Agung Nugroho Adi S.T., M.T. yang selalu memberi masukan dan arahan sehingga penulis lebih mudah dalam menyelesaikan tugas akhir .
3. Bapak Purtojo S.T., M.Sc. yang selalu memberikan saran yang selalu memudahkan penulis dalam pengambilan keputusan.
4. Dan Jurusan Teknik Mesin Universitas Islam Indonesia, yang telah membantu dalam proses pembuatan alat dan administrasi.

penulis sepenuhnya menyadari bawah dalam mengerjakan tugas akhir ini banyak kekeliruan dan kekurangan, untuk itu penulis meminta maaf atas semua kekurangan yang ada di tugas akhir ini, semoga tugas akhir ini dapat bermanfaat bagi semua pihak yang membaca di masa yang akan datang, Amin.

Penulis

Yogyakarta, 17 Juni 2021

ABSTRAK

Penelitian tentang ball balancing sudah cukup banyak dari beberapa jurnal yang dibaca penelitian yang dilakukan menghadapi beberapa kendala yaitu sensor yang memiliki resolusi yang rendah sehingga mempersulit dalam mencapai keadaan steady state ada sistem. Kesalahan pada pembacaan sensor ultrasonic *noise* pada pembacaan sensor. Balancing ball adalah sebuah plant dimana pada plant tersebut terdapat bola yang dapat bergerak dan posisi dari perpindahan bola tersebut dapat diatur sesuai dengan set point yang diinginkan. Konsep Kontrol PID adalah mengurangi error yang terjadi pada sistem untuk mendapatkan hasil yang sesuai keinginan atau mendapat keadaan *steady state*. Sistem Kontrol PID yang akan digunakan pada alat ini menggunakan Arduino UNO sebagai mikrokontroler, *Sharp gp2y0a21yk0f* sebagai sensor yang akan membaca posisi bola, aktuator menggunakan servo *Hitec HS-255mg*, sensor yang memiliki resolusi berdasarkan jarak, dimana semakin jauh resolusi akan semakin kecil, sehingga pembacaan akan semakin tidak presisi, berdasarkan ini penulis akan membuat pembacaan jadi lebih baik dengan menambahkan filter pada pembacaan sensor. Setelah didapat pembacaan yang lebih baik, penelitian ini akan men-tuning nilai K_p , K_i dan K_d setelah itu pengujian akan dilakukan pada nilai K_p , K_i dan K_d dengan memberikan gangguan pada *setpoint*, dimana *setpoint* akan diubah ketika bola dalam keadaan *steady state*.

Kata kunci: Kontrol PID, *Filter*, *Steady State*, *Hanning Filter*, *Ball Balancing*

ABSTRACT

Research on ball balancing has been quite a lot from several journals that have been read. Research conducted faces several obstacles, namely sensors that have low resolutions, making it difficult to achieve steady state in the system. Errors in ultrasonic noise sensor readings on sensor readings. Balancing ball is a plant where in the plant there is a ball that can move and the position of the displacement of the ball can be adjusted according to the desired set point. The concept of PID control is to reduce errors that occur in the system to get the desired results or get a steady state. The PID control system that will be used in this tool uses Arduino UNO as a microcontroller, Sharp gp2y0a21yk0f as a sensor that will read the position of the ball, actuator uses a Hitec HS-255mg servo, a sensor that has a resolution based on distance, where the farther the resolution will be smaller, so the reading is will be increasingly imprecise, based on this the author will make the readings better by adding filters to the sensor readings. After getting a better reading, this study will adjust the values of K_p , K_i and K_d after which the test will be carried out on the values of K_p , K_i and K_d by giving disturbances to the setpoint, where the setpoint will be changed when the ball is in a steady state.

Keywords: Control PID, Hanning filter, Steady state, Filter, Ball Balancing

DAFTAR ISI

HALAMAN JUDUL	1
PERNYATAAN KEASLIAN	2
LEMBAR PENGESAHAN DOSEN PEMBIMBING	3
LEMBAR PENGESAHAN DOSEN PENGUJI	4
HALAMAN PERSEMBAHAN	5
HALAMAN MOTTO	6
KATA PENGANTAR	7
ABSTRAK	8
ABSTRACT	9
DAFTAR ISI	10
DAFTAR GAMBAR	13
Bab 1 Pendahuluan	15
1.1 Latar Belakang	15
1.2 Rumusan Masalah	16
1.3 Batasan Masalah	16
1.4 Tujuan Penelitian atau Perancangan	17
1.5 Manfaat Penelitian atau Perancangan	17
1.5.1 Manfaat Penelitian bagi Peneliti	17
1.5.2 Manfaat Penelitian bagi Universitas	17
1.6 Sistematika Penulisan	17
1.6.1 BAB I – Pendahuluan	17
1.6.2 BAB II – Tinjauan Pustaka	18
1.6.3 BAB III – Metode Penelitian	18
1.6.4 BAB IV – Hasil dan Pembahasan	18
1.6.5 BAB V – Penutup	18
Bab 2 Tinjauan Pustaka	19
2.1 Kajian Pustaka	19
2.2 Dasar Teori	21
2.2.1 Sistem kendali <i>open loop</i> dan <i>closed loop</i>	21
2.2.2 Kontrol <i>PID</i>	22
2.2.3 Kontrol Proporsional	22

2.2.4	Kontrol Integral	22
2.2.5	Kontrol Derivatif	23
2.2.6	Hanning Filter Data	23
Bab 3	Metode Penelitian	24
3.1	Alur Penelitian	24
3.2	Peralatan dan Bahan	24
3.2.1	Perangkat keras	24
3.2.2	Perangkat lunak	25
3.2.3	Bahan	25
3.3	Perancangan sistem alat	25
3.3.1	Perancangan Rangkaian Perangkat Keras	26
3.3.2	Penjelasan program PID	27
3.3.3	Perancangan Program PID kontrol	32
3.3.4	Perancangan desain 3D	33
3.3.3.1	Desain Penampang V	34
3.3.3.2	Skematik Desain yang Disesuaikan dengan Motor Servo	36
3.3.3.3	Perhitungan Beban yang Diterima Motor Servo	37
Bab 4	Hasil dan Pembahasan	40
4.1	Hasil Pengujian	40
4.1.1	Kalibrasi <i>Sharp Gp2y0a21yk0f</i>	40
4.1.2	<i>Tuning</i> Nilai PID menggunakan filter	41
4.1.2.1	<i>Tuning</i> Nilai K_p (Konstanta Proporsional)	41
4.1.2.2	<i>Tuning</i> Nilai K_d (Konstanta Derivatif)	42
4.1.2.3	<i>Tuning</i> Nilai K_i (Konstanta Integral)	43
4.1.3	Pengujian Nilai Konstanta PID dan Filter	44
4.1.3.1	Pengujian Nilai K_p (Konstanta Proporsional)	44
4.1.3.2	Pengujian Nilai K_d (Konstanta Derivatif)	46
4.1.3.3	Pengujian Pengaruh Jumlah Sampel Pada filter	47
4.1.3.4	Pengujian Nilai K_i (Konstanta Integral)	49
4.1.3.5	<i>Running Test</i> Kontrol PID dengan Filter	51
4.2	Analisis Pembahasan	53
Bab 5	Penutup	55

5.1	Kesimpulan	55
5.2	Saran atau Penelitian Selanjutnya	55
	DAFTAR PUSTAKA	56
	Lampiran Program	57
	Lampiran <i>Datasheet</i> Sharp GP2Y0A21YK0F	62



DAFTAR GAMBAR

Gambar 2-1 Diagram Sistem Kendali <i>Open Loop</i>	21
Gambar 2-2 Diagram Sistem Kendali <i>Close Loop</i>	21
Gambar 3-1 Diagram Alur Perancangan	24
Gambar 3-2 Rangkaian elektrik dari sistem	26
Gambar 3-3 <i>Library</i>	27
Gambar 3-4 <i>Input</i> dan <i>Output</i>	27
Gambar 3-5 <i>Variable</i>	28
Gambar 3-6 Konstanta PID dan Filter	28
Gambar 3-7 <i>Void setup</i>	29
Gambar 3-8 Program <i>Tuning</i> PID dengan Potensiometer	29
Gambar 3-9 Program filter	30
Gambar 3-10 Program PID	31
Gambar 3-11 Fungsi menghasilkan nilai adc menjadi centimeter dengan nilai kalibrasi sensor	32
Gambar 3-12 <i>Flowchart</i> Program PID	33
Gambar 3-13 Desain 3D alat menggunakan inventor 2017	34
Gambar 3-14 Penampang V	34
Gambar 3-15 Skema Dari Gaya Dan Pengaruh Sudut Penampang Terhadap r ...	35
Gambar 3-16 Desain Tampak Samping.....	36
Gambar 3-17 Skematik berdasarkan gambar 3-16	36
Gambar 3-18 Dimensi dari Skematik Alat dalam satuan milimeter.....	37
Gambar 3-19 Model Pembebanan dari alat	37
Gambar 3-20 Beban merata menjadi terpusat	38
Gambar 4-1 Grafik output tegangan dengan pembacaan jarak.	40
Gambar 4-2 Grafik Excel dari kalibrasi.....	41
Gambar 4-3 Grafik osilasi menggunakan $K_p = 1$ dan sampel data 2-5.....	42
Gambar 4-4 Grafik <i>tuning</i> K_d dengan $K_p = 1$	43
Gambar 4-5 <i>Tuning</i> nilai K_i dengan $K_p = 1$, $K_d = 550$	44
Gambar 4-6 <i>Tuning</i> $K_p = 1$ dengan $K_d = 550$	45
Gambar 4-7 <i>Tuning</i> $K_p = 1,5$ dengan $K_d = 550$	45
Gambar 4-8 <i>Tuning</i> $K_p = 2$ dengan $K_d = 550$	46

Gambar 4-9 Grafik Pengujian Jumlah Data Sampel Genap	47
Gambar 4-10 Grafik Pengujian Jumlah Data Sampel Ganjil	48
Gambar 4-11 perbandingan sampel 5,6 dan 7 dengan menambah $K_i = 0.01$	49
Gambar 4-12 Pengujian K_i 0.007,0.008,0.009 dan 0.01.....	50
Gambar 4-13 Pengujian K_i 0.01,0.011,0.012 dan 0.013.....	50
Gambar 4-14 <i>Running test</i> Kontrol PID	51
Gambar 4-15 3 <i>Running test</i> terbaik.....	52
Gambar 4-16 <i>Running test ke-5</i>	53
Gambar 4-17 Perbesaran dari gambar 4-25	53



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penelitian tentang *ball balancing* sudah cukup banyak dari beberapa jurnal yang dibaca penelitian yang dilakukan menghadapi beberapa kendala yaitu sensor yang memiliki resolusi yang rendah sehingga mempersulit dalam mencapai keadaan steady state ada sistem. Kesalahan pada pembacaan sensor ultrasonic pada umumnya disebabkan oleh tingkat sensitivitas yang kurang baik (Ivan Candra Utama, 2016).

Penggunaan filter pada pembacaan sensor bertujuan untuk mengurangi *noise* dari pembacaan sensor. Hanning filter adalah metode filter data yang menggunakan data sebelumnya sebagai sampel yang digunakan dalam filter (Iskandar and Saputra, n.d.). Jika di buat persamaan data terbaru akan menjadi data lama pertama, dan data lama pertama akan menjadi data lama kedua, sehingga tidak membuat memperlama waktu dari pengambilan sampel data. Hanning filter cocok untuk sampel data yang *continue* (Iskandar and Saputra, n.d.).

Sistem kendali adalah salah hal yang sangat penting dalam sebuah industri, terutama di industri otomasi yang dimana sistem otomatis harus diberikan sistem kendali untuk menjaga kualitas dari sistem (Yunardi, 2017). Sistem kendali yang digunakan salah satunya adalah sistem kendali PID.

Sistem kendali *open loop* adalah sistem kendali yang tidak memiliki umpan balik dari *output* atau dengan kata lain *output* dari sistem tidak mempengaruhi sistem kendali. Sedangkan sistem kendali *closed loop* adalah sistem kendali yang memiliki umpan balik dari *output* yang dihasilkan, dengan demikian *output* dari sistem ini adakan memiliki pengaruh terhadap sistem kendali (Supardi, 2012)

Di dunia sekarang sudah cukup banyak mobil listrik yang memiliki teknologi *self driving*. *Self driving* merupakan salah satu jenis kontrol mobil otomatis yang dapat memudahkan manusia dalam berkendara tanpa campur tangan pengemudi (Taufiqurrahman and Riyadi, 2016). Dimana teknologi *self driving* juga menerapkan sistem PID *Control*.

Sistem kendali P, P+I dan P+D adalah beberapa tipe dari sistem kendali PID, penggunaan dari ketiga sistem kendali ini tidak sama dikarenakan memiliki karakteristik yang berbeda, oleh karena itu peruntukan dari sistem kendali akan berbeda (Raharjo Sulistiyo Utomo, 2007).

Oleh karena itu pada penelitian ini akan menggunakan sistem kendali P+I+D dan berfokus pada pengurangan *noise* pada sensor sehingga akan mempermudah sistem mencapai keadaan *steady state*, kemudian sistem akan diberikan gangguan pada *setpoint* untuk melihat bagaimana sistem mempertahankan keadaan *steady state* saat diberi gangguan.

1.2 Rumusan Masalah

Perumusan masalah yang akan dibahas pada proposal ini adalah :

1. Bagaimana merancang sistem kendali penyeimbang bola menggunakan metode PID?
2. Bagaimana mengurangi *noise* dari pembacaan sensor ?
3. Bagaimana reaksi sistem saat diberi gangguan perubahan *set point* yang konsisten?

1.3 Batasan Masalah

Pembatasan masalah pada proposal kali ini bertujuan untuk membatasi ruang lingkup dari pembahasan agar lebih jelas dan tidak menyebar ke hal yang tidak diinginkan. Berikut adalah Batasan – Batasan masalah pada proposal sebagai berikut :

1. Hasil dari penelitian akan fokus pada pengurangan *steady state error*
2. Penambahan filter untuk mengurangi *noise* pada pembacaan sensor.
3. Penelitian akan fokus pada sistem kendali PID.
4. Filter yang digunakan menggunakan Hanning filter
5. Sensor yang di gunakan adalah Sharp IR gp2y0a21yk0f
6. Motor servo yang digunakan ada Hitec HS-225mg.
7. Hasil dari alat yang dibuat berupa *prototype*.
8. Bola yang digunakan adalah bola Billiard ukuran 2 inchi dengan berat 160 gr.

1.4 Tujuan Penelitian atau Perancangan

1. Menghasilkan sistem yang dapat mencapai keadaan *steady state* dengan sistem kendali PID
2. Mengurangi *noise* dari pembacaan sensor
3. Sistem dapat mempertahankan keadaan *steady state* saat diberi gangguan

1.5 Manfaat Penelitian atau Perancangan

Manfaat penelitian ini dibagi menjadi beberapa kepentingan yaitu :

1.5.1 Manfaat Penelitian bagi Peneliti

- a. Meningkatkan kemampuan peneliti dalam mengaplikasikan teori yang telah didapatkan selama kuliah.
- b. Memberikan pemahaman lebih pada peneliti mengenai sistem *PID control*.
- c. Peneliti mengetahui permasalahan yang terjadi selama pembuatan alat.

1.5.2 Manfaat Penelitian bagi Universitas

- a. Hasil dari penelitian dapat digunakan untuk menunjang pembelajaran di perkuliahan.
- b. Memudahkan dosen dalam menyampaikan materi yang berkaitan tentang sistem *PID Control*.
- c. Dapat digunakan sebagai salah satu alat praktikum pada laboratorium universitas yang berkaitan.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini sebagai berikut .

1.6.1 BAB I – Pendahuluan

Bab ini akan menjelaskan latar belakang dari masalah yang akan diangkat dan di bahas pada proposal ini.

1.6.2 BAB II – Tinjauan Pustaka

Bab ini berisi tentang kajian pustaka dan dasar teori yang akan digunakan dalam penelitian dan perancangan.

1.6.3 BAB III – Metode Penelitian

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian dan metode penelitian yang digunakan.

1.6.4 BAB IV – Hasil dan Pembahasan

Bab ini menjelaskan hasil dari penelitian serta pembahasan dari hasil tersebut.

1.6.5 BAB V – Penutup

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian dan metode penelitian yang digunakan.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Berikut merupakan kajian pustaka dari beberapa penelitian yang kemudian digunakan sebagai acuan dalam penulisan kali ini.

Pada jurnal Ivan Candra Utama dkk (2016), penelitian yang dilakukan menggunakan Arduino Uno untuk mikrokontroler kemudian sensor ultrasonik HC-SR04 untuk sensor jarak dan motor servo yang tidak diketahui mereknya, pada penelitian yang dilakukan oleh Ivan Candra Utama lebih memfokuskan pada penggunaan *LabVIEW*. Pada penelitian ini memiliki kekurangan dimana sensor ultrasonik yang memiliki sensitivitas kurang baik sehingga pembacaan sering terjadi kesalahan, kemudian peneliti memberikan saran agak memperbaiki mekanik dari penelitian tersebut.

Pada jurnal Orlanda Merti Wijaya dkk (2018) pada jurnal ini menggunakan Arduino UNO sebagai mikrokontroler, sensor ultrasonic dan motor servo SG90. Jurnal ini membahas tentang perancangan kontrol PID dengan metode Ziegler-Nichols *tuning and error* untuk mendapatkan sebuah kontroler PID yang sesuai dengan nilai *setpoint*.

Pada jurnal Muhammad Taufiqurrahman dkk (2016) membahas tentang perancangan *self-driving* dengan metode PD, dimana menggunakan beberapa komponen mulai dari Driver EMS 30A, UBEC 5V-3A, Mikrokontroler ATmega16, sensor ultrasonic, 2 baterai Lipo 12 cell 2200mAh, Mini Computer Raspberry Pi 2, UBEC 5V-5A. Pada jurnal ini melakukan beberapa pengujian diantaranya, pengujian perangkat keras, pengujian *color filtering*, pengujian eliminasi, pengujian kecepatan komunikasi data, pengujian kinematika dan *free run*, pengujian kontrol kestabilan PD. Pengujian yang dilakukan pada jurnal ini untuk mengetahui kondisi dari *prototype autonomous* yang dirancang. Pengujian juga dilakukan untuk menjelaskan kestabilan dari kemampuan sistem *self-driving* dalam mengikuti lintasan yang diberikan.

Pada jurnal Riky Tri Yunardi dkk (2017) menganalisa kinerja dari sensor inframerah dan ultrasonik. Sensor IR yang digunakan ada Sharp GY2Y02YK0F sedangkan sensor ultrasonic ada modul SRF04. Pada jurnal ini dilakukan dengan rentang jarak 15 cm – 150 cm, kemudian hasil dari penelitian pada jurnal ini menjelaskan bahwa sensor Inframerah memiliki presisi tinggi dibandingkan dengan sensor Ultrasonik untuk jarak kurang dari 15 cm sedangkan untuk jarak jauh hingga 150 cm sama-sama memiliki tingkat akurasi yang tinggi dengan persentase 96,88 – 98,16 %.

Pada jurnal Surya Setiawan dkk (2007) yang membahas tentang penerapan inverse kinematic untuk pergerakan kaki robot biped. Pada jurnal ini robot biped dibuat menggunakan servo Tower Pro MG995, sedangkan untuk mikrokontrolernya menggunakan Arduino UNO. Di Dalam jurnal ini ada membahas tentang perhitungan untuk kinematika pada *link* dan *joint*.

Pada jurnal Iskandar and Saputra dkk (n.d.) membahas tentang perbandingan dari average filter dan Hanning filter yang diaplikasikan pada pembacaan alat ukur berat, dari jurnal ini didapati bahwa penggunaan metode average filter mendekati dari hasil beban yang sebenarnya.

Pada jurnal Raharjo Sulistiyo Utomo (2007) membahas tentang perbandingan sistem kendali P, P+I dan P+D, disimpulkan bahwa ketiga sistem kendali diatas memiliki karakteristik yang berbeda. Oleh karena itu peneliti memberi saran untuk menambah variasi sistem kendali yang digunakan.

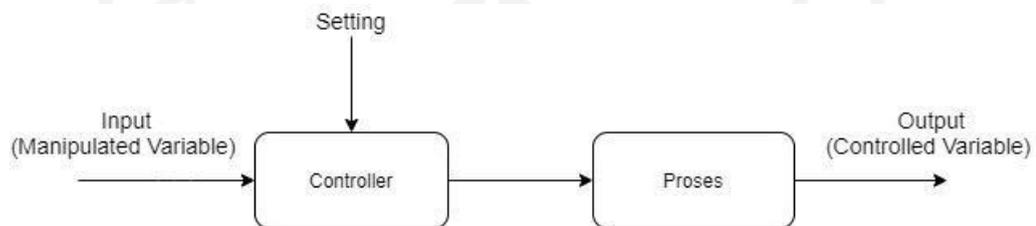
Dari semua jurnal di atas didapatkan beberapa ide untuk dikembangkan sebagai penelitian selanjutnya, dimana pada penelitian selanjutnya akan menggunakan sistem kendali P+I+D, dikarenakan pada penelitian sebelumnya hanya menggunakan kendali P, P+I, dan P+D pada *ball and beam*, kemudian akan menambahkan Hanning filter pada pembacaan sensor untuk mengurangi *noise* dikarenakan pada penelitian Ivan Candra Utama dkk (2016) terjadi *noise* pada pembacaan sensor, dan kemudian akan diberi variasi gangguan pada *set point* untuk melihat respon dari sistem.

2.2 Dasar Teori

Seperti yang telah disebutkan sebelumnya, pengontrol *PID* akan digunakan sebagai program pada sistem yang akan dibuat. Penjelasan detail *PID* akan dijelaskan setelah ini.

2.2.1 Sistem kendali *open loop* dan *closed loop*

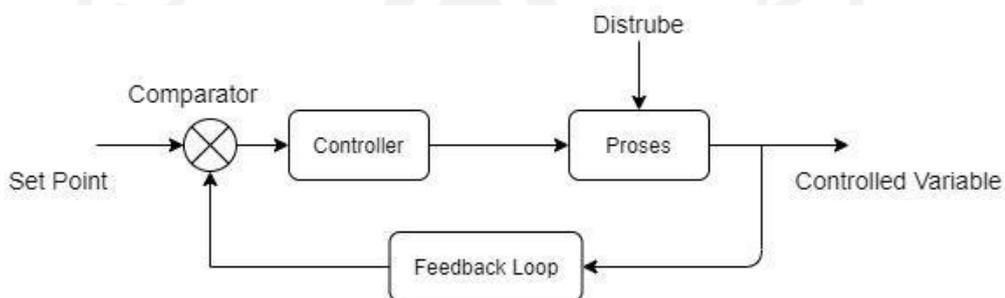
Sistem kendali *open loop* adalah sistem kendali yang tidak memiliki umpan balik dari *output* atau dengan kata lain *output* dari sistem tidak mempengaruhi sistem kendali.



Gambar 2-1 Diagram Sistem Kendali *Open Loop*

Diadaptasi dari : (Raharjo Sulistiyo Utomo, 2007)

Sistem kendali *close loop* adalah sistem kendali yang memiliki umpan balik dari *output* yang dihasilkan, dengan demikian *output* dari sistem ini adakan memiliki pengaruh terhadap sistem kendali



Gambar 2-2 Diagram Sistem Kendali *Close Loop*

Diadaptasi dari : (Raharjo Sulistiyo Utomo, 2007)

2.2.2 Kontrol PID

Proporsional Integral Derivatif, ada sistem kontrol yang berfungsi untuk memberikan koreksi secara otomatis pada mekanisme loop yang memberi umpan balik. Sistem akan secara terus menerus mengontrol sistem agar tetap pada daerah *setpoint*. Jarak antara *set point* dengan bola disebut dengan error $e(t)$. perumusan dari program *PID* dapat dilihat pada persamaan dibawah.

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad 2.1$$

2.2.3 Kontrol Proporsional

Tujuan dari kontrol proporsional adalah untuk mengurangi error pada *steady state* dengan cara memberikan sinyal kendali yang nilainya proporsional terhadap nilai sinyal error dari sistem. Jika nilai proporsional terlalu tinggi akan mengakibatkan osilasi yang besar. Persamaan dari kontrol proporsional sebagai berikut (Wijaya, 2018).

$$u_P(t) = K_P e(t) \quad 2.2$$

2.2.4 Kontrol Integral

Kontrol integral memiliki kontrol yang berdasarkan waktu error, jadi pada sistem ini sistem dikoreksi dengan jumlah kesalahan yang sebelumnya, dengan kata lain nilai dari kesalahan akan semakin kecil seiring waktu. Persamaan dari kontrol integral sebagai berikut (Wijaya, 2018).

$$u_I(t) = K_I \int_0^t e(\tau) d\tau \quad 2.3.2.3$$

2.2.5 Kontrol Derivatif

Kontrol derivative memiliki fungsi untuk mengurangi *overshoot*, dan mempercepat waktu sistem untuk stabil (Itm-Ex, n.d.). Hal ini dikarenakan perubahan yang dihasilkan oleh kontrol ini akan dikalikan dengan nilai error. Untuk persamaan dari kontrol derivative sebagai berikut (Wijaya, 2018).

$$u_D(t) = K_D \frac{de(t)}{dt} \quad 2.4$$

2.2.6 Hanning Filter Data

Pada filter data penulis menggunakan metode Hanning filter dimana data sample data menggunakan data sebelumnya. Jika di buat persamaan data terbaru akan menjadi data lama pertama, dan data lama pertama akan menjadi data lama kedua, sehingga tidak membuat memperlama waktu dari pengambilan sampel data. Hanning filter sangat cocok untuk sampel data yang *continue* (Iskandar and Saputra, n.d.). Berikut persamaan dari Hanning filter sebagai berikut:

$$Han_{average} = (Han_1 + Han_2 + Han_3 + Han_4 \dots + Han_n) / n \quad 2.4$$

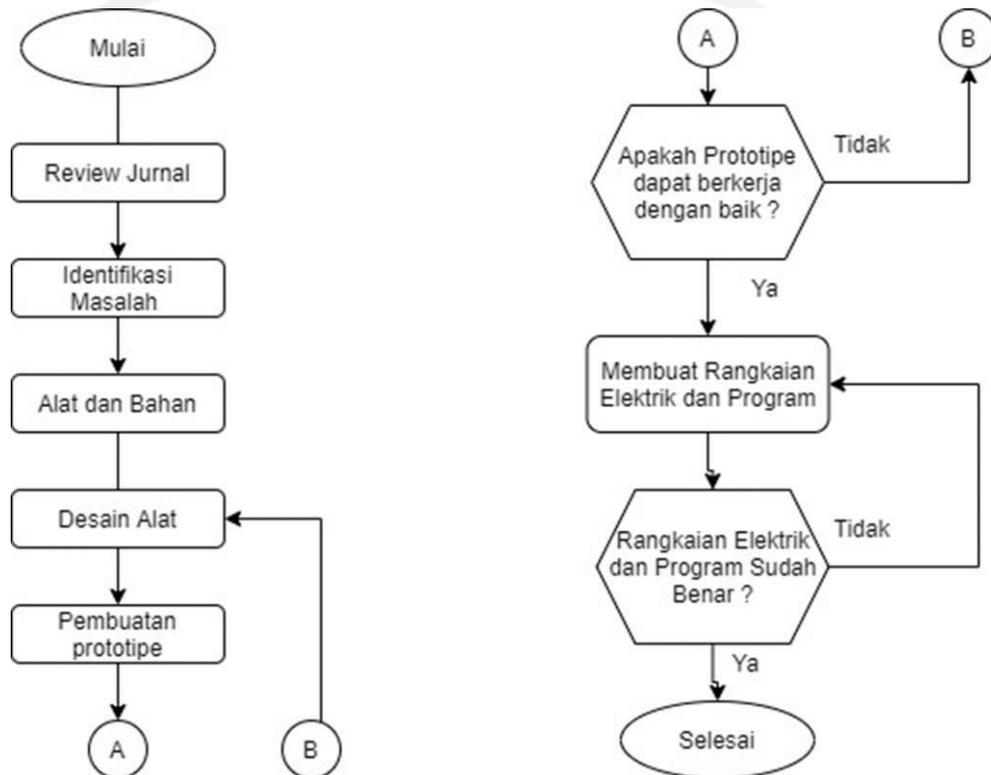
Keterangan:

- Han_1 = sampel data terbaru dari pembacaan sensor
- Han_2 = sampel data sebelumnya yang pertama
- Han_3 = sampel data sebelumnya yang kedua
- Han_4 = sampel data sebelumnya yang ketiga
- Han_n = sampel data sebanyak n
- n = jumlah sampel data

BAB 3 METODE PENELITIAN

3.1 Alur Penelitian

Perancangan alat PID kontrol untuk bola dan sinar. Di bawah ini adalah diagram alur dari penelitian.



Gambar 3-1 Diagram Alur Perancangan

3.2 Peralatan dan Bahan

Ada beberapa komponen yang dibutuhkan untuk membuat produk PID kontrol untuk ball balancing, yaitu:

3.2.1 Perangkat keras

- Laptop/ PC untuk membuat program dan mendesain prototype.
- Multimeter

- Solder
- Cat spray warna hitam

3.2.2 Perangkat lunak

- Arduino IDE untuk membuat program PID
- Solidworks 2013 untuk mendesain produk
- Microsoft Excel 2019 untuk perhitungan kalibrasi sensor IR

3.2.3 Bahan

- Arduino UNO
- Sharp IR GP2Y0A21YK0F untuk sensor jarak
- Servo Hitec HS-225mg
- Power supply 12V 5A
- Stepdown dc
- Breadboard
- Bola billiard ukuran 2 inchi
- Kable jumper male dan female
- Timah
- Kapasitor 20 mikrofarad

3.3 Perancangan sistem alat

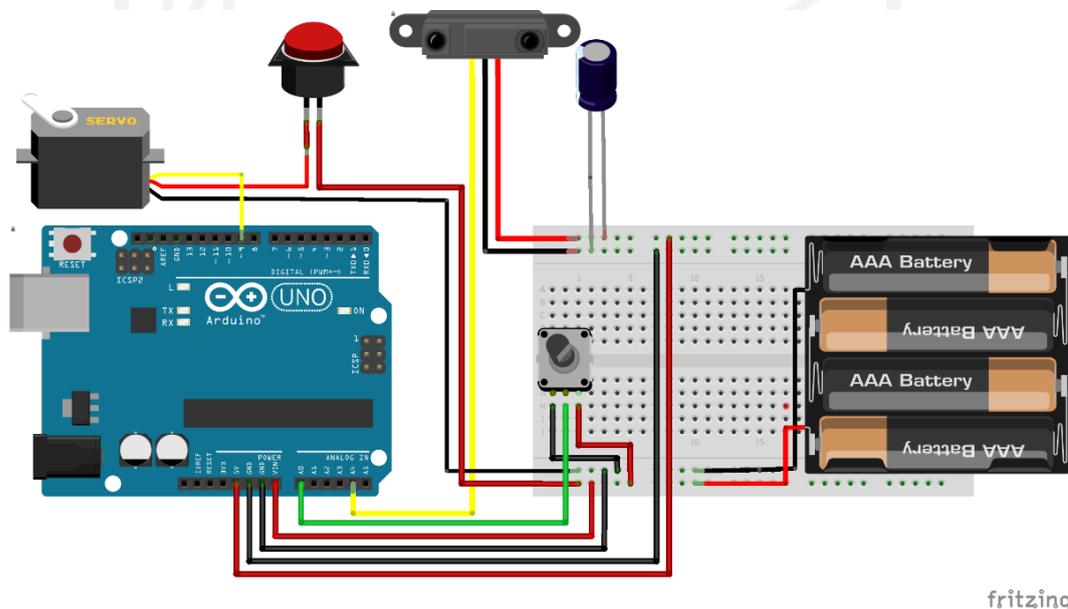
Alat yang dibuat memiliki fungsi untuk menstabilkan bola sesuai dengan set point yang telah ditentukan, alat ini merupakan salah satu aplikasi dari PID kontrol. Dimana dalam alat ini akan menampilkan hubungan antara nilai proporsional, integral dan derivatif. Program PID akan dibuat berdasarkan turunan matematika dari rumus PID kontrol. Nilai dari PID akan dimasukkan secara manual disesuaikan dengan hasil atau perubahan yang dihasilkan.

Alat ini diharapkan bisa menjadi salah satu gambaran dalam pengaplikasian PID kontrol yang lebih optimal, dikarenakan penelitian kali ini akan memfokuskan pada pengoptimalan dari sistem PID kontrol.

Sistem yang dimiliki dari alat ini adalah sensor Sharp IR menjadi input dari posisi bola kemudian sistem kontrol PID akan menjalankan program PID sehingga output dari program akan dilaksanakan oleh servo Hitec-mg225

3.3.1 Perancangan Rangkaian Perangkat Keras

Perancangan perangkat keras akan dimulai dari power supply 5V di paralel ke servo dan ke Arduino UNO, kemudian dari Arduino UNO akan dikeluarkan input 5V sensor Sharp IR, sebelum dihubungkan ke sensor Sharp IR akan di dihubungkan terlebih dahulu ke kapasitor 25k. Untuk lebih jelas akan digambarkan melalui gambar 3.1 sebagai berikut.



Gambar 3-2 Rangkaian elektrik dari sistem
Didesain menggunakan : Fritzing

Semua rangkaian kabel yang berwarna merah dan hitam adalah positif dan negatif dari komponen. Berikut keterangan dan alasan memilih komponen di atas rangkaian:

1. Arduino UNO, mendapatkan *supply* daya dari baterai 6V dari VIN dan GND. Penggunaan Arduino UNO pada penelitian dikarenakan komponen sering digunakan serta telah tersedia, sehingga akan mengurangi biasa perancangan.
2. Servo Hitec-225mg, kabel kuning servo masuk ke pin PWM 9 di Arduino. Alasan menggunakan servo Hitec-225mg dikarenakan servo memiliki gear

metal yang kuat dan kecepatan dari motor yang cukup baik untuk penelitian ini.

3. Sharp IR, kabel kuning masuk ke pin analog 4. Penggunaan sensor Sharp IR dikarenakan sensor cahaya memiliki respon lebih cepat dibandingkan sensor suara, berdasarkan penelitian sebelumnya Sharp IR memiliki resolusi yang cukup baik dibanding sensor yang lain serta dengan harga yang relatif murah.
4. Power supply digambarkan menggunakan baterai pada gambar diatas, dimana power supply akan masuk ke pin VIN dan GND dari Arduino UNO. Sistem sangat membutuhkan power *supply* sebagai daya tambahan mengingat sistem menggunakan servo, sensor IR dan Potensiometer.
5. Kapasitor, akan dipasang sebelum masuk sensor Sharp IR. Penggunaan kapasitor pada sensor IR dikarenakan saran yang ada di dalam *datasheet* dari sensor IR tersebut.
6. Potensiometer, kabel hijau yang dihubungkan ke pin analog 0. Potensiometer akan digunakan untuk mempermudah *tuning* nilai PID saat penelitian.
7. Saklar, digunakan pada positif dari servo untuk mematikan servo, jika terjadi sesuatu yang darurat.

3.3.2 Penjelasan program PID



```
#include <Servo.h>
```

Gambar 3-3 Library

Program diawali dengan *library* yang digunakan. *Library* yang digunakan yaitu `#include <Servo.h>`, *library* ini digunakan untuk servo Hitec HS-225mg.

```
////////////////////////////////Input/output////////////////////////////////
int Analog_in = A4;
const int Pot_in = A0 ;
float Pot ;
long quad ;
Servo myservo; //
////////////////////////////////////////////////////////////////
```

Gambar 3-4 Input dan Output

Kemudian dilanjutkan dengan menentukan *input* dan *output* yang digunakan. Input yang digunakan untuk sensor Sharp IR yaitu `int Analog_in = A4` pada pin analog A4, kemudian `const int Pot_in = A0` adalah input Potensiometer sedang untuk *output* digunakan `Servo myservo` untuk penamaan servo.

```

////////////////////////////////////Variabel////////////////////////////////////
int Read = 0;
float distance = 0.0;
float time; //Variabels waktu
double distance_previous_error, distance_error;
int period =25; // waktu period akan mempengaruhi response Kd
////////////////////////////////////

```

Gambar 3-5 Variable

Pada gambar diatas adalah variabel yang digunakan pada program ini yaitu `int Read = 0`, variabel ini adalah untuk mengatur nilai input yang masuk tidak negatif. Kemudian `float distance = 0.0` program ini digunakan untuk menentukan nilai yang dihasilkan berupa desimal, `float time` yaitu variabel untuk waktu, `float distance_previous_error, distance_error` variabel ini digunakan untuk jarak yang dibaca dari *input* sensor dan `int period = 25` yaitu variable waktu untuk pembacaan pada *input* (waktu *refresh* pembacaan 25 ms).

```

////////////////////////////////////PID constants////////////////////////////////////
double kp=1; //ZN = 0.8
double ki=0.1;
double kd=550; // ZN = 1.25
double distance_setpoint = 20;
double distance_setpoint_prev = 0;
double P, I, D, U;
double P_prev = 0, I_prev = 0, D_prev = 0, U_prev = 0, distance_error_prev1 = 0;
double distance_error_prev2 = 0, distance_error_prev3 = 0, distance_error_prev4 = 0;
double distance_error_prev5 = 0, distance_error_prev6 = 0, distance_error_prev7 = 0;
double distance_error_prev8 = 0, distance_error_prev9 = 0;
////////////////////////////////////

```

Gambar 3-6 Konstanta PID dan Filter

Program diatas adalah konstanta yang digunakan pada program PID diantaranya `double kp = 1` adalah nilai proporsional dari PID, `double ki = 0.01` adalah nilai integral dari PID, `double kd = 550` adalah nilai derivatif dari PID, `int distance_setpoint = 20` adalah nilai tengah dari PID atau nilai *steady state* yang

diinginkan dan *double P, I, D, U* adalah penamaan rumus yang digunakan pada program PID. Kemudian untuk seterusnya adalah parameter yang digunakan untuk filter.

```
void setup() {  
  //analogReference(EXTERNAL);  
  Serial.begin(9600);  
  myservo.attach(9); // PIN servo 9  
  myservo.write(90);  
  pinMode(Analog_in, INPUT);  
  pinMode(Pot_in, INPUT);  
  time = millis();  
}
```

Gambar 3-7 *Void setup*

Gambar diatas adalah *void setup* diantaranya, *Serial.begin (9600)* adalah kecepatan pengiriman dan penerimaan data pada mikrokontroler, *myservo.attach (9)* adalah servo menggunakan pin digital 9, *pinMode (Analog_in, INPUT)* adalah program untuk mengatur sensor Sharp sebagai input, *pinMode (Pot_in, INPUT)* adalah setup untuk Potensiometer sebagai input dan *time=millis()* adalah *variable time* yang sama dengan *milis*, *milis* adalah waktu yang ada pada mikrokontroler.

```
Pot = analogRead (Pot_in);  
// distance_setpoint = map(Pot,0,1023,8,30);  
// kp= map(Pot,0,1023,0,102.3);  
// ki= map(Pot,0,1023,0,102.3);  
////////////////////////////////////
```

Gambar 3-8 Program *Tuning* PID dengan Potensiometer

Program diatas akan digunakan saat *tuning* dari kontrol PID, hal ini dilakukan untuk mempermudah dalam melakukan *tuning*.

```

if (millis() > time+period) // setelah 25 ms
{
    time = millis();
    distance = get_dist(100);
    distance_error = distance_setpoint - distance;
// distance_error = (distance_error +distance_error_prev1)/2 ;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2)/3;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + di
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + di
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + di
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + dist
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + di
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2 + di
// distance_error_prev1 = distance_error;
// distance_error_prev2 = distance_error_prev1;
// distance_error_prev3 = distance_error_prev2;
// distance_error_prev4 = distance_error_prev3;
// distance_error_prev5 = distance_error_prev4;
// distance_error_prev6 = distance_error_prev5;
// distance_error_prev7 = distance_error_prev6;
// distance_error_prev8 = distance_error_prev7;
// distance_error_prev9 = distance_error_prev8;

```

Gambar 3-9 Program filter

`if (milis() > time+period)` dimaksudkan untuk memanggil program, dimana program agar berjalan setelah 25 ms, `time = milis()` untuk mengembalikan parameter `time` ke waktu `milis`, `distance = get_dist(100)` memanggil parameter hasil dari adc menjadi `distance`. `distance_error = distance-setpoint - distance`. Program setelahnya adalah program filter yang terdiri dari 2 sampel data sampai dengan 10 sampel data. Program secara lengkap ada di lampiran.

```

P = kp * distance_error;
float dist_diference = distance_error - distance_previous_error;
D = kd / period *(distance_error - distance_previous_error);
if(-3 < distance_error && distance_error < 3)
{
I = I + (ki * distance_error);
}
else
{
I = 0;
}
U = P + I + D;
U = map(U, -150, 150, 0, 150);

if (U < 20) {U = 20;}
if (U > 160) {U = 160;}
myservo.write(U+20);
quad = (distance_error*distance_error);
distance_previous_error = distance_error;
}
}

```

Gambar 3-10 Program PID

$P = kp * distance_error$ adalah program untuk menentukan nilai proporsional, $int\ dist_diference = distance_error - distance_previous_error$ yaitu *variable* untuk nilai derivatif. $D = kd * ((distance_error - distance_previous_error) / period)$ adalah rumus atau program untuk mencari nilai derivatif, kemudian program dari $I = I + (ki * distance_error)$; sampai dengan $I = 0$ adalah untuk menjelaskan bahwa nilai integral hanya akan berjalan saat *distance_error* kurang 3 dan lebih dari -3.

Kemudian $U = P + I + D$ adalah gabungan dari hasil setiap kontrol P,I dan D sedangkan $U = map(U, -150, 150, 0, 150)$ dan $myservo.write(PID_total+20)$ adalah mengkonversi nilai PID menjadi *output* sudut pada servo, penambahan +20 pada *myservo.write* dikarenakan sudut servo pada alat dimulai dari sudut 20 dan $distance_previous_error = distance_error$ untuk mengembalikan nilai *distance_error*.

```

float get_dist(int n) // harus float
{
  long sum=0; // fungsinya untuk membuat hasil dari sum 0 bukan bilangan negatif
  for(int i=0;i<n;i++) // membatasi bawah konversi adc berdasarkan percentase 0- 1
  {
    sum= sum + analogRead (Analog_in); // fungsinya untuk membuat hasil dari sum 0
  }
  float adc=sum/n;
  float distance_cm = 17555 * pow(adc, -1.202); // pow (x,y) == pow(base,exponent)
}

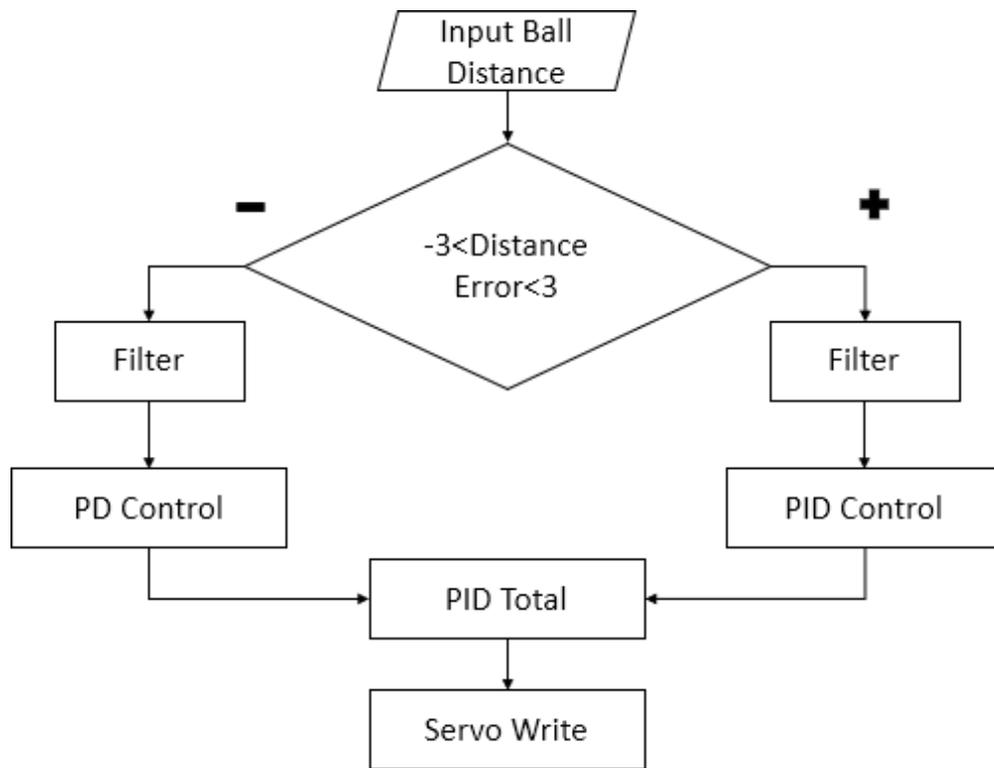
```

Gambar 3-11 Fungsi menghasilkan nilai adc menjadi centimeter dengan nilai kalibrasi sensor

Program diatas merupakan fungsi yang akan dipanggil ketika awal program PID dimulai. *int get_dist(int n)* adalah fungsi yang akan dipanggil, *long sum= 0*, *for(int i=0;i<n;i++)* dan *sum=sum+analogRead(Analog_in)* adalah untuk memastikan nilai sum tidak negatif jika tidak ada nilai analog maka anak terbaca 0. *Float adc=sum/n* adalah untuk mengkonversi nilai analog menjadi digital, *float distance_cm = 17569.7 * pow(adc, -1.2062)* adalah merubah nilai sensor menjadi centimeter, *Serial.println (distance_cm)* untuk menampilkan di serial monitor pembacaan *distance_cm* dan *return (distance_cm)* untuk mengembalikan nilai awal.

3.3.3 Perancangan Program PID kontrol

Untuk perancangan perangkat lunak akan menggunakan Arduino IDE, program akan dibuat berdasarkan flowchart yang ada di gambar 3-12. Dimana sensor Sharp IR akan memberikan input nilai berupa posisi dengan satuan centimeter, kemudian nilai error dari setpoint terhadap posisi saat ini akan dikelola oleh program PID, berdasarkan nilai KP, KI dan KD yang telah ditentukan sehingga akan menghasilkan output yang akan diteruskan oleh servo Hitec-mg 225. Jika nilai set point dipenuhi maka program akan berhenti.



Gambar 3-12 *Flowchart* Program PID

3.3.4 Perancangan desain 3D

Desain 3D dilakukan menggunakan Autodesk Inventor 2017, desain akan diproduksi dengan 3 macam material, diantaranya.

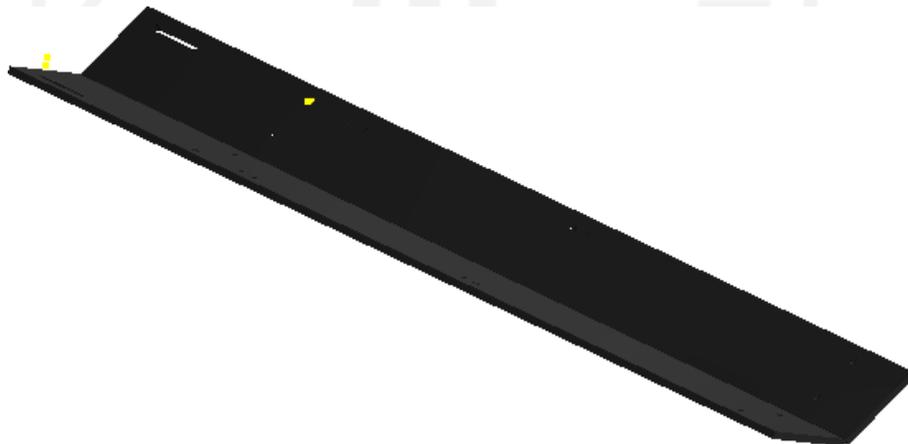
1. Filament PLA
2. Acrylic
3. Multiplek



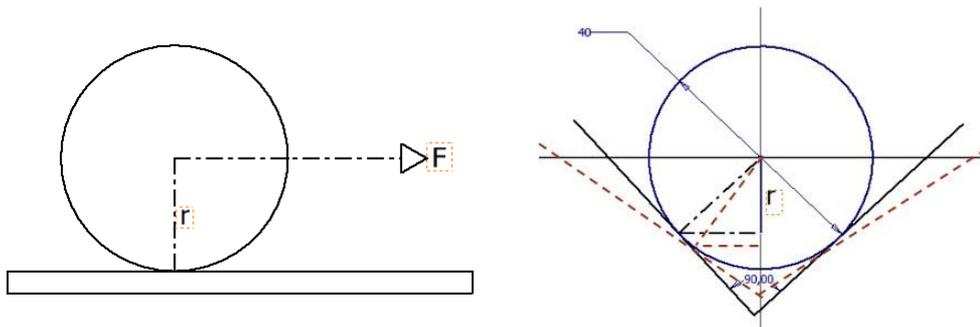
Gambar 3-13 Desain 3D alat menggunakan inventor 2017

3.3.3.1 Desain Penampang V

Berdasarkan penelitian sebelumnya penampang V yang digunakan tidak dihitung sudutnya, akan tetapi asumsi dari desain adalah siku-siku atau 90 derajat. Sudut penampang V sangat penting untuk dihitung dikarenakan akan berpengaruh pada pergerakan bola.



Gambar 3-14 Penampang V



Gambar 3-15 Skema Dari Gaya Dan Pengaruh Sudut Penampang Terhadap r

Bola yang bergelinding akan berotasi dan translasi, dimana pada untuk gerak rotasi memiliki persamaan sebagai berikut.

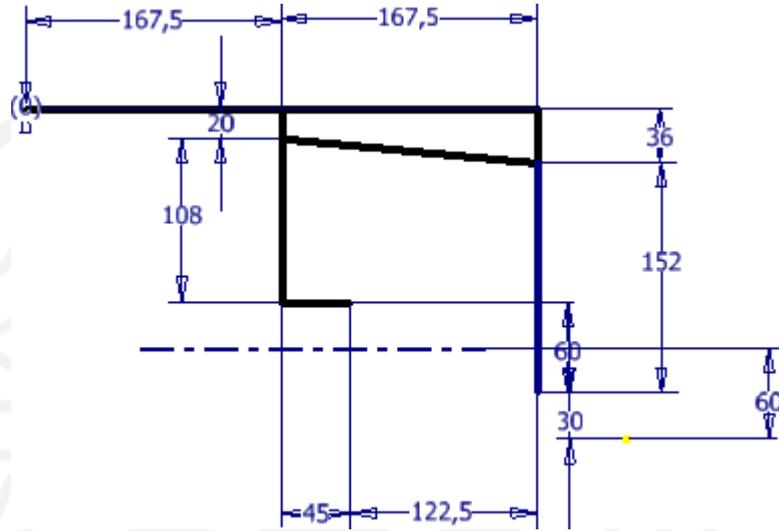
$$\tau = rF \sin \theta \quad 3.1$$

Keterangan :

- F = Gaya dari bola
- $\sin \theta$ = Arah dari gaya
- r = Jarak gaya ke rotasi
- τ = Torsi

Berdasarkan gambar 3-15 dan rumus torsi, sudut penampang memiliki hubungan terhadap jarak pusat bola ke penampang, dimana semakin besar r maka akan semakin besar torsi. Semakin lebar penampang semakin besar r, dari penelitian sebelumnya yang menggunakan penampang V yang siku-siku, di penelitian ini akan menggunakan sudut yang lebih dai siku-siku (sudut lebih tumpul) sehingga nilai r akan bertambah.

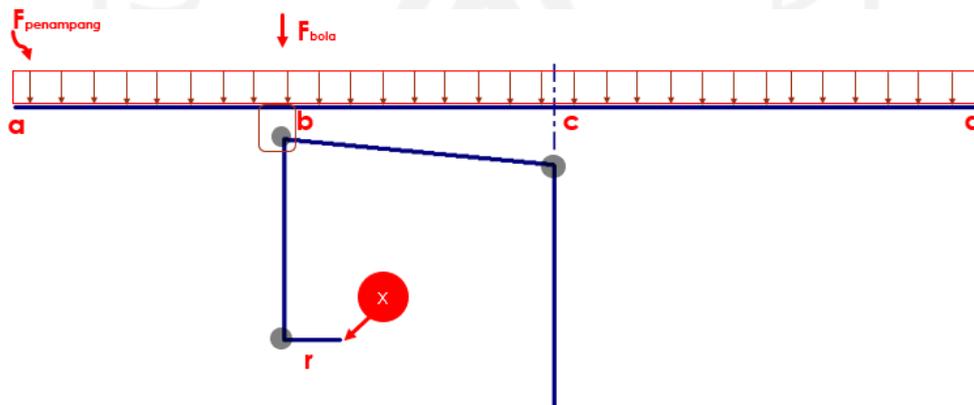
- L_α = Perpindahan sudut α
- L_β = Perpindahan sudut β
- α = sudut perpindahan L_1
- β = sudut perpindahan L_3



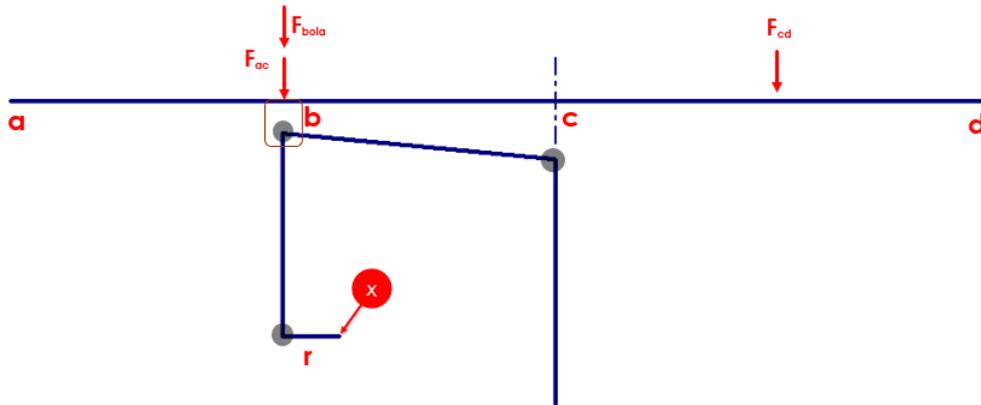
Gambar 3-18 Dimensi dari Skematik Alat dalam satuan milimeter

3.3.3.3 Perhitungan Beban yang Diterima Motor Servo

Berdasarkan gambar 3-17 panjang lengan yang menghubungkan servo ke tiang D adalah 45 mm berikut persamaan gaya yang di terima Servo.



Gambar 3-19 Model Pembebanan dari alat



Gambar 3-20 Beban merata menjadi terpusat

Berat dari penampang ditambah dengan berat dari bola, dimana asumsi bahwa penampang adalah beban merata dan bola adalah beban terpusat.

Diketahui :

M_{bola}	$= 0.16 \text{ Kg}$	$g = 9.81 \text{ m/s}^2$
$M_{penampang}$	$= 0.41 \text{ Kg}$	$X = \text{Letak servo}$
L_{ab}	$= 0.15 \text{ m}$	$r = 0.045 \text{ m}$
L_{ac}	$= 0.3 \text{ m}$	$L_{ad} = 0.5 \text{ m}$
L_{cd}	$= 0.2 \text{ m}$	

Hal perlu di ketahui dari informasi diatas dan berdasarkan gambar 3-18 adalah besar torsi yang dibutuhkan servo untuk mengangkat beban. Berikut penjelasannya.

- $F_{penampang} = M_{penampang} \times g / L_{ad}$
 $= 0.41 \text{ kg} \times 9.81 \text{ m/s}^2 / 0.5 \text{ m}$
 $= 8.04 \text{ N/m}$
- $F_{bola} = M_{bola} \times g$
 $= 0.16 \text{ kg} \times 9.81 \text{ m/s}^2$
 $= 1.57 \text{ N}$
- $F_{ac} = F_{penampang} \times L_{ac}$
 $= 8.04 \text{ N/m} \times 0.3 \text{ m}$
 $= 2.41 \text{ N}$
- $F_{cd} = F_{penampang} \times L_{cd}$
 $= 8.04 \text{ N/m} \times 0.2 \text{ m}$
 $= 1.61 \text{ N}$

- $\sum F_y = F_b$
 $F_{bola} + F_{ac} - F_{cd} = F_b$
 $1.57 \text{ N} + 2.41 \text{ N} - 1.61 \text{ N} = F_b$
 $2.37 \text{ N} = F_b$
- Torsi yang diterima servo = $r \times F_b$
 $= 0.045 \text{ m} \times 2.37 \text{ N}$
 $= 0.107 \text{ N.m}$

Berdasarkan data sheet motor servo dapat menahan hingga = 0.48 N.m yang dimana empat kali lebih besar dari bebannya, sehingga torsi yang diberikan motor masih bisa menahan beban dari bola dan strukturnya.



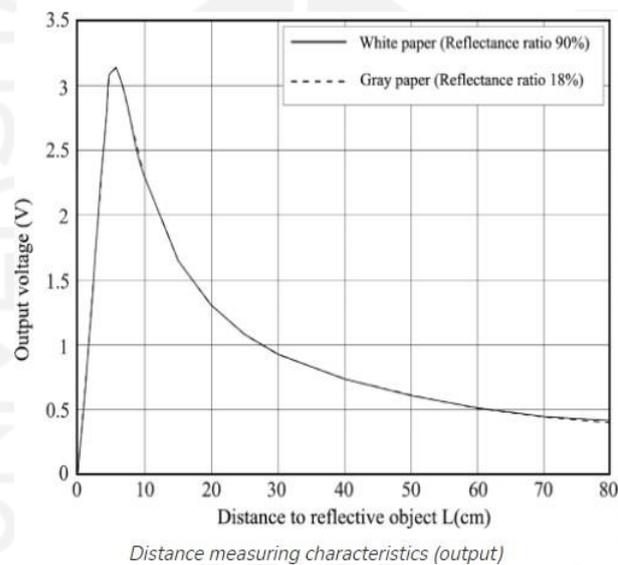
BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian

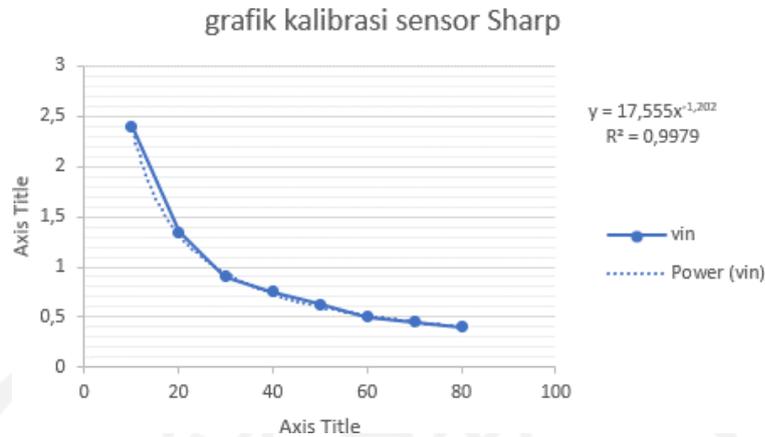
4.1.1 Kalibrasi *Sharp Gp2y0a21yk0f*

Pembacaan yang di hasilkan dari sensor *Sharp gp2y0a21yk0f* berupa nilai analog, sehingga dibutuhkannya konversi ke satuan jarak. Kalibrasi adalah bertujuan menyesuaikan grafik kalibrasi nilai persamaa grafik untuk mendapatkan persamaan matematika grafik yang kemudian dirubah menjadi program.



Gambar 4-1 Grafik output tegangan dengan pembacaan jarak.

Sumber gambar dari : *datasheet Sharp gp2y0a21yk0f analog distance*



Gambar 4-2 Grafik Excel dari kalibrasi

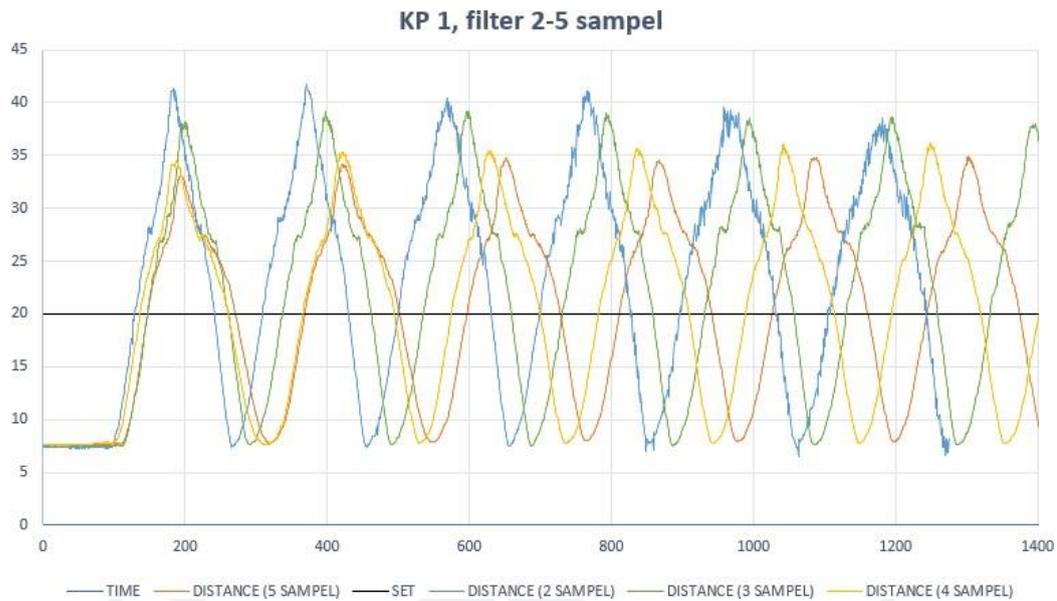
Kalibrasi dari sensor diambil dari *datasheet* yang tersedia, kemudian akan dirumuskan menggunakan *Microsoft excel* untuk mencari persamaan matematikanya dari persamaan matematika akan dirubah menjadi program yang di masukan ke Arduino IDE. Dapat di lihat persamaa matematikanya $y = 17,555x^{-1,202}$, untuk programnya dapat dilihat di gambar 3-11.

4.1.2 Tuning Nilai PID menggunakan filter

Perlu di ketahui penambahan filter pada program untuk mengurangi *noise* dari pembacaan sensor.

4.1.2.1 Tuning Nilai K_p (Konstanta Proporsional)

Perlu diketahui penambahan filter pada program untuk mengurangi *noise* dari pembacaan sensor. Untuk mencari nilai K_p pada program PID, kemudian mencoba mencari menggunakan prinsip yang ada metode *Ziegler Nichols*, dimana untuk mencari nilai K_p yang kecil untuk mendapatkan sistem bisa berosilasi, dari sini didapat nilai K_p = 1 agar sistem bisa berosilasi menyentuh batas atas dan bawah dari sistem. Jika K_p dibawah 1 maka bola berosilasi tapi tidak maksimal dan jika lebih besar kan lebih sulit dikendalikan.

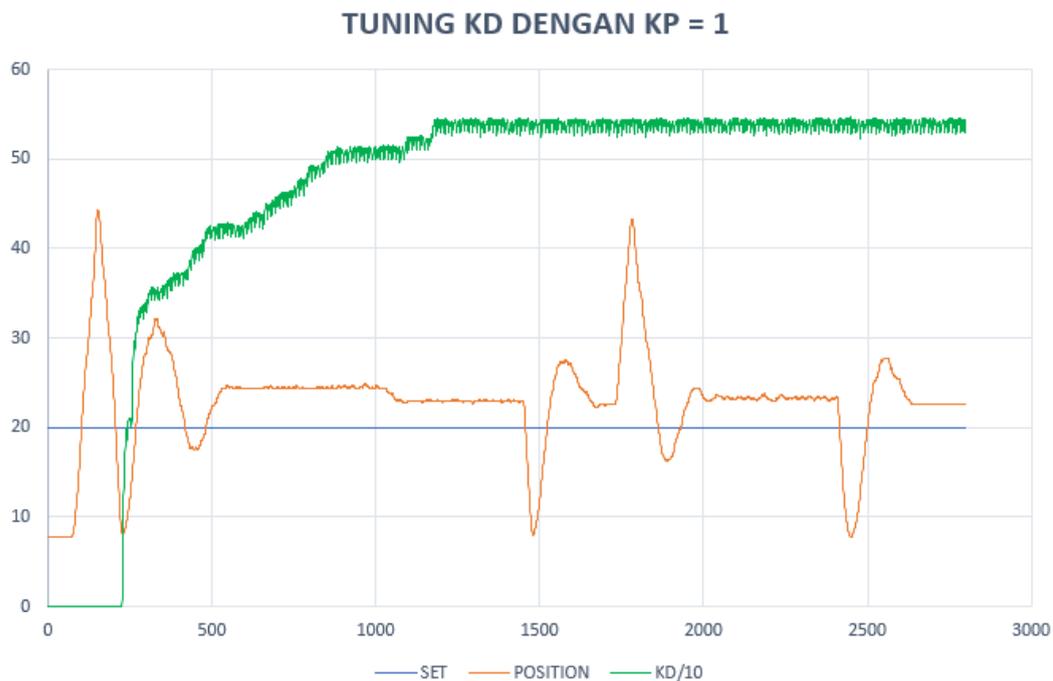


Gambar 4-3 Grafik osilasi menggunakan $K_p = 1$ dan sampel data 2-5

Dari grafik diatas diketahui bahwa set point = 20 dan sampel-sampel yang ditampilkan memperlihatkan bahwa semakin banyak sampel dari filter semakin stabil pembacaan dari sensor. Pada bagian ini hanya akan fokus pada *tuning* nilai PID, sehingga jumlah sampel data yang digunakan adalah 3 (garis yang berwarna hijau). Penambahan filter untuk mempermudah *tuning* nilai PID.

4.1.2.2 *Tuning* Nilai K_d (Konstanta Derivatif)

Setelah mendapatkan K_p , dilanjutkan untuk mencari nilai K_d dengan metode iterasi akan tetapi pada penelitian ini penambahan nilai K_d akan menggunakan Potensiometer, dimana perbandingan nilai analog dengan Potensiometer adalah 1:1. Berikut adalah grafik *tuning* K_d .

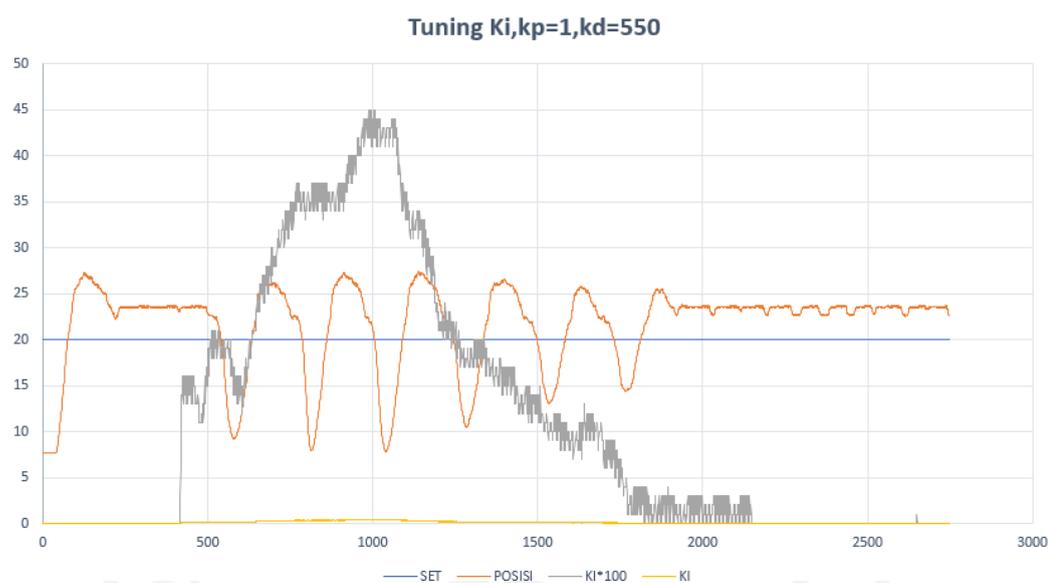


Gambar 4-4 Grafik *tuning* K_d dengan $K_p = 1$

Dari gambar 4-11 dapat dilihat bahwa sistem beresilasi kemudian diberi input kontrol derivatif, *overshoot* dari sistem mulai berkurang hingga pada kisaran nilai $K_d = 550$ sistem sudah mulai stabil dan *steady state error* berkurang, kemudian di kisaran data 1500 (sumbu x), diberi gangguan pada sistem berupa dorongan untuk melihat apakah sistem sudah berada dalam kondisi *steady state*, jika dilihat gangguan berupa dorongan ke atas dan kebawah masih bisa dikendalikan oleh sistem. Nilai K_d pada sistem telah dibagi 10 untuk memudahkan dalam melihat perubahan grafik dari sistem terhadap penambahan nilai K_d .

4.1.2.3 *Tuning* Nilai K_i (Konstanta Integral)

Dari gambar 4-11 dapat dilihat masih memiliki *steady state error* yang cukup besar di kisaran 3 cm dari *set point*, oleh karena itu diperlukannya penambahan kontrol Integral.



Gambar 4-5 Tuning nilai K_i dengan $K_p = 1$, $K_d = 550$

Pada gambar 4-12 metode penambahan nilai K_i sama dengan K_d , nilai yang K_i yang ditampilkan di grafik dikalikan 100 agar lebih mudah untuk dilihat pengaruh dari penambahan kontrol integral pada sistem. Jika dilihat sistem masih berada dalam keadaan *steady state* pada K_i di kisaran 0 sampai dengan 0,2 saat melebihi itu sistem akan berosilasi dikarenakan hal ini nilai K_i kembali di turunkan sehingga pada $K_i = 0.03$ sistem kembali *steady state*, yang dimana rentang 0 – 0.03 akan menjadi acuan untuk pengujian dalam mendapatkan nilai K_i yang lebih tepat.

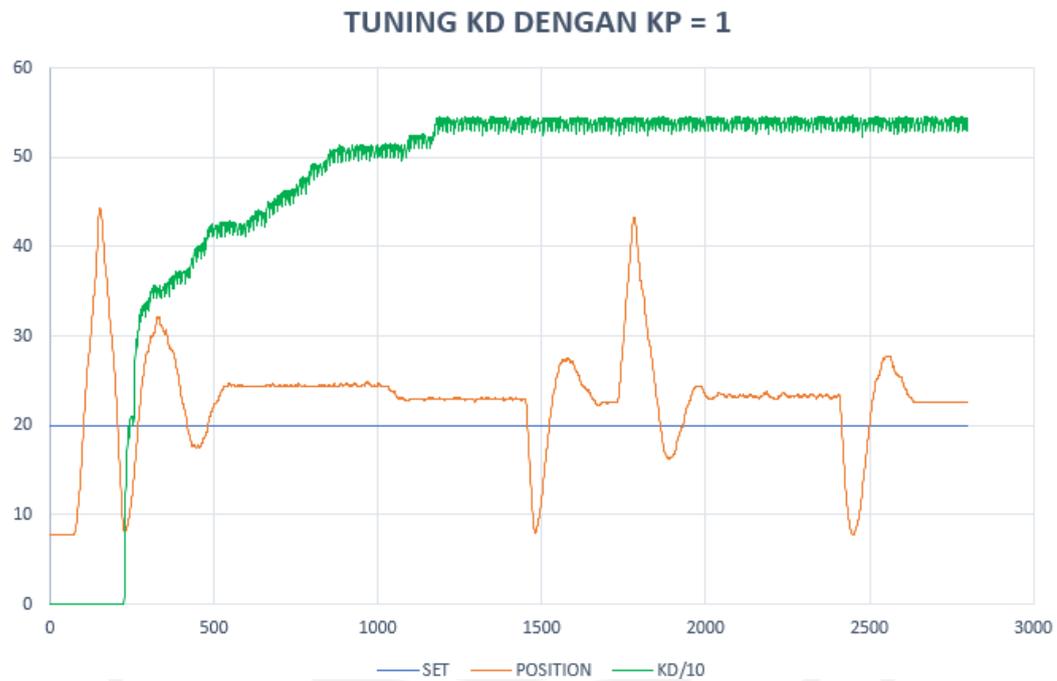
4.1.3 Pengujian Nilai Konstanta PID dan Filter

Pada proses *tuning* sistem sudah bisa mencapai keadaan *steady state* akan tetapi *steady state error* masih cukup besar. Pada bagian ini akan mencari rentang nilai yang lebih untuk mencapai keadaan *steady state* dan memiliki *steady state error* yang lebih kecil.

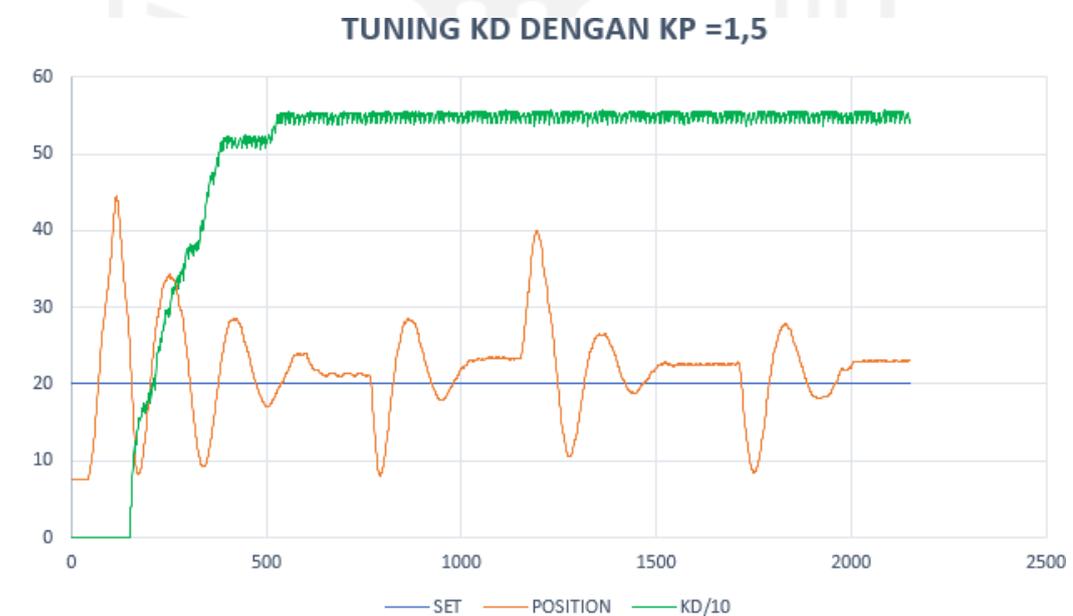
4.1.3.1 Pengujian Nilai K_p (Konstanta Proporsional)

Steady state error dan respon dari sistem memiliki hubungan dengan kontrol proporsional, jika K_p lebih besar maka respon akan lebih cepat dan *steady state error* akan lebih kecil sedangkan K_p lebih kecil menunjukkan hasil sebaliknya, akan tetapi K_p yang lebih besar bisa membuat sistem sulit dikendalikan (Setiawan,

2015). Oleh karena itu diperlukan pengujian untuk melihat hasil dari penambahan nilai K_p .

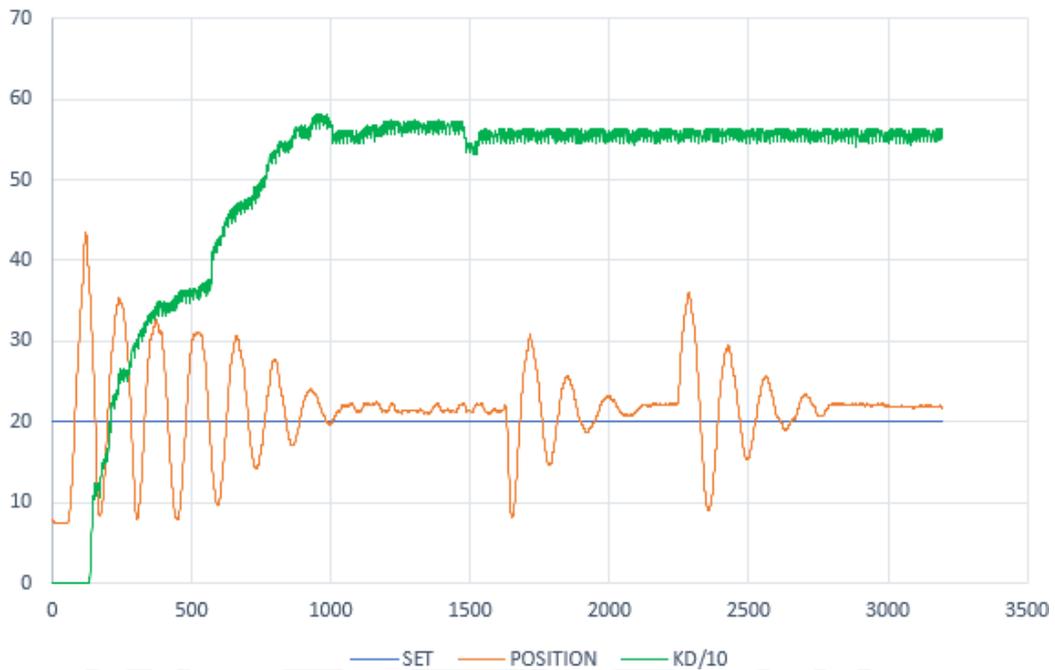


Gambar 4-6 *Tuning* $K_p = 1$ dengan $K_d = 550$



Gambar 4-7 *Tuning* $K_p = 1,5$ dengan $K_d = 550$

TUNING KD DENGAN KP 2



Gambar 4-8 Tuning $K_p = 2$ dengan $K_d = 550$

Dari gambar 4-13 sampai gambar 4-15 penambahan nilai K_p membuat sistem jadi lebih sulit dikendalikan, untuk *steady state error* dari sistem akan berkurang seiring dengan pertambahan nilai K_p .

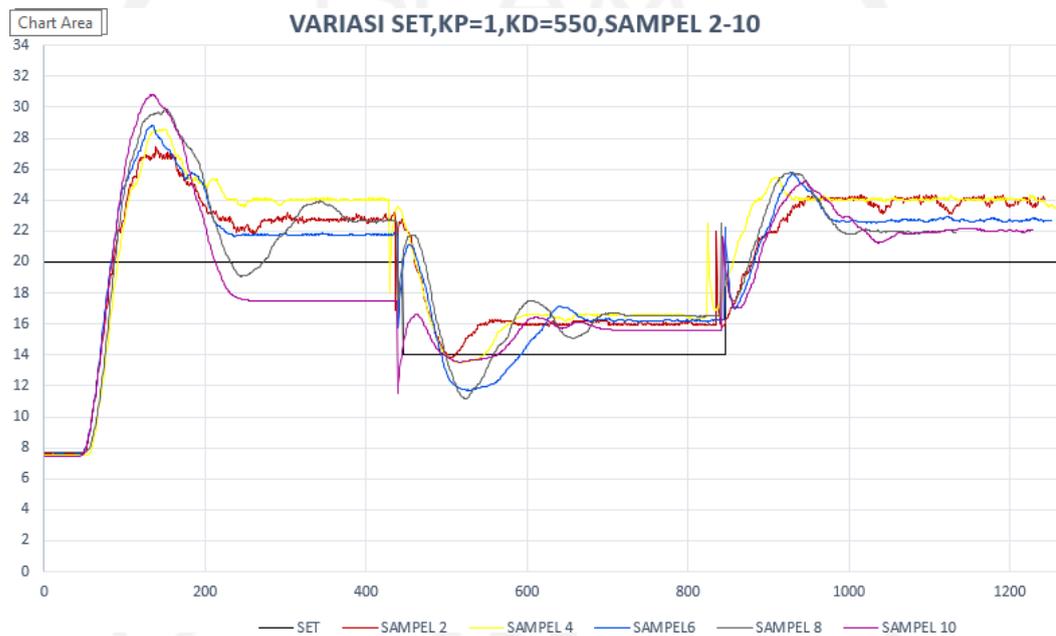
Berdasarkan data diatas pada penelitian ini akan menggunakan $K_p = 1$, dikarenakan osilasi dari sistem lebih sedikit sehingga sistem lebih mudah untuk dikendalikan. *Steady state error* yang masih cukup besar akan di kurangi menggunakan kontrol integral

4.1.3.2 Pengujian Nilai K_d (Konstanta Derivatif)

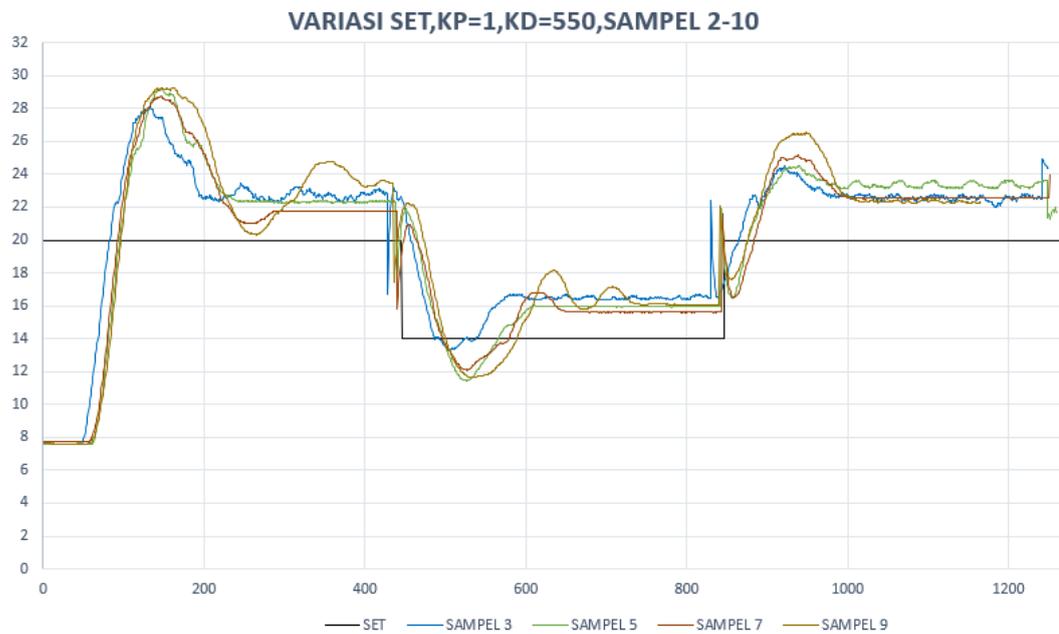
Secara umum kontrol derivatif berfungsi untuk meningkatkan stabilitas serta mempercepat respon output proses dari sistem, akan tetapi kontrol derivatif akan selalu menghasilkan *steady state error* (Setiawan, 2015). pada penelitian ini akan fokus untuk mengurangi *steady state error*, dan juga dapat dilihat dengan nilai $K_d = 550$ sudah membuat sistem stabil dan dapat dikendalikan.

4.1.3.3 Pengujian Pengaruh Jumlah Sampel Pada filter

Pada pengujian ini akan dilihat pengaruh dari jumlah sampel data dari filter terhadap *steady state error*. Pengujian akan menggunakan $K_p = 1$ dan $K_d = 550$, dan 9 sampel data yaitu 2 sampel data sampai 9 sampel data. Pengujian dilakukan dengan memberikan gangguan pada *set point* secara konsisten, saat sistem diaktifkan 10 detik pertama *set point* = 20, kemudian 10 detik kedua akan berada di *set point* = 14, dan 10 detik terakhir akan kembali ke *set point* = 20.



Gambar 4-9 Grafik Pengujian Jumlah Data Sampel Genap

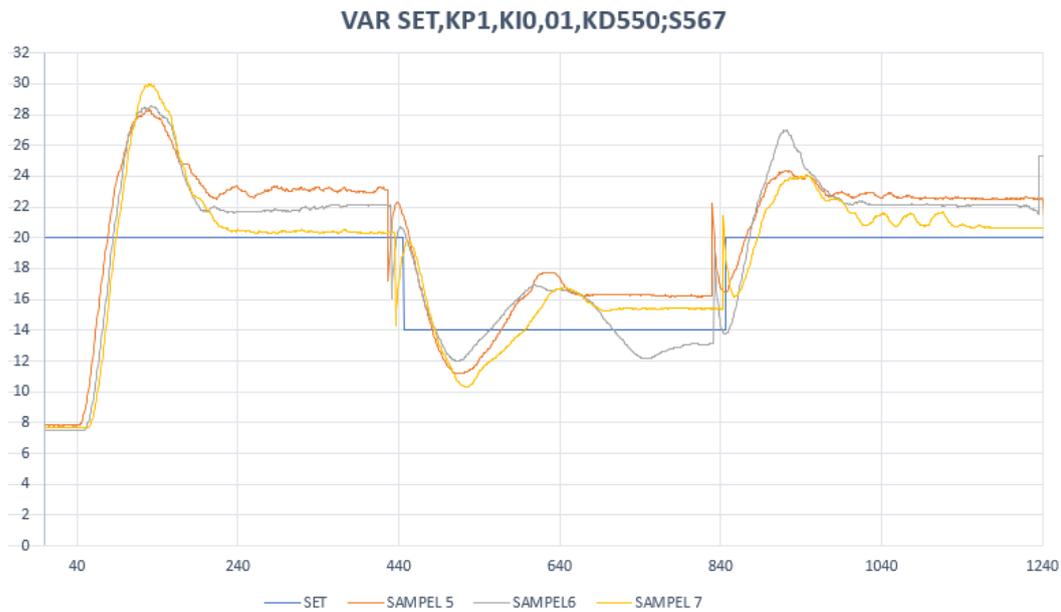


Gambar 4-10 Grafik Pengujian Jumlah Data Sampel Ganjil

Dari gambar 4-16 dan 4-17 dapat dilihat perbedaan hasil dari setiap jumlah sampel data, data sampel ganjil dan genap hanya bertujuan untuk memudahkan pengamatan. Dari 9 sampel dari grafik di atas akan dipilih yang terbaik, parameter pemilihannya adalah grafik memiliki *steady state error* yang kecil, dikarenakan fokus pada pengujian untuk mendapatkan hasil *steady state* yang kecil.

Jika dilihat dari gambar 4-16 grafik terbaik terdapat pada jumlah sampel 6, sedangkan pada gambar 4-17 terdapat pada sampel 5 dan 7 karena kedua sampel ini memiliki *steady state error* yang kecil dan stabil.

Kemudian dari ketiga sampel terpilih akan diuji kembali dengan menambahkan kontrol integral, berdasarkan gambar 4-12 rentang nilai K_i dari 0-0,13. Disini yang memilih nilai K_i secara acak pada rentang tersebut yaitu 0.01.

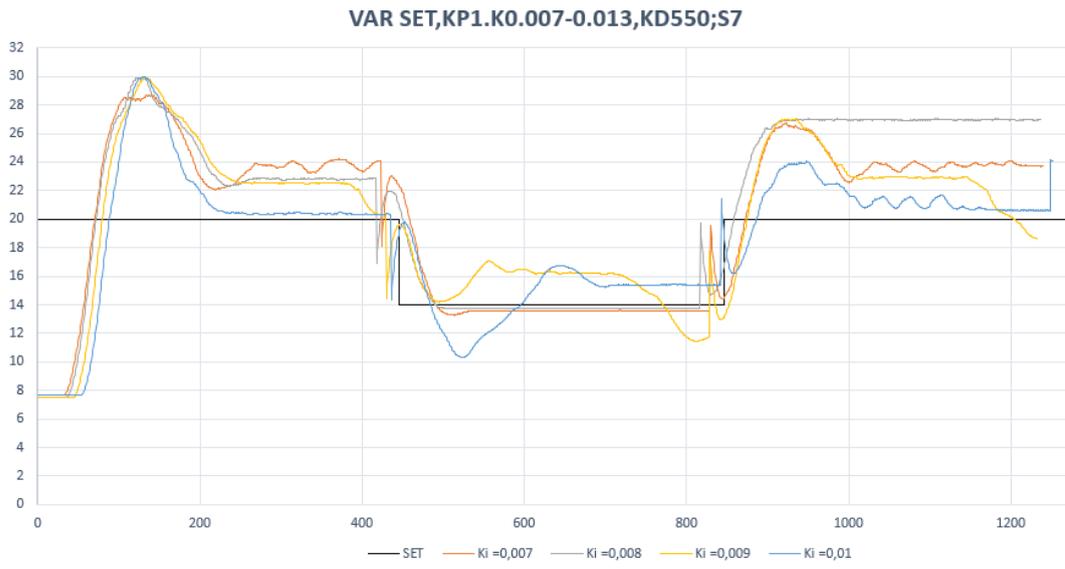


Gambar 4-11 perbandingan sampel 5,6 dan 7 dengan menambah $K_i = 0.01$

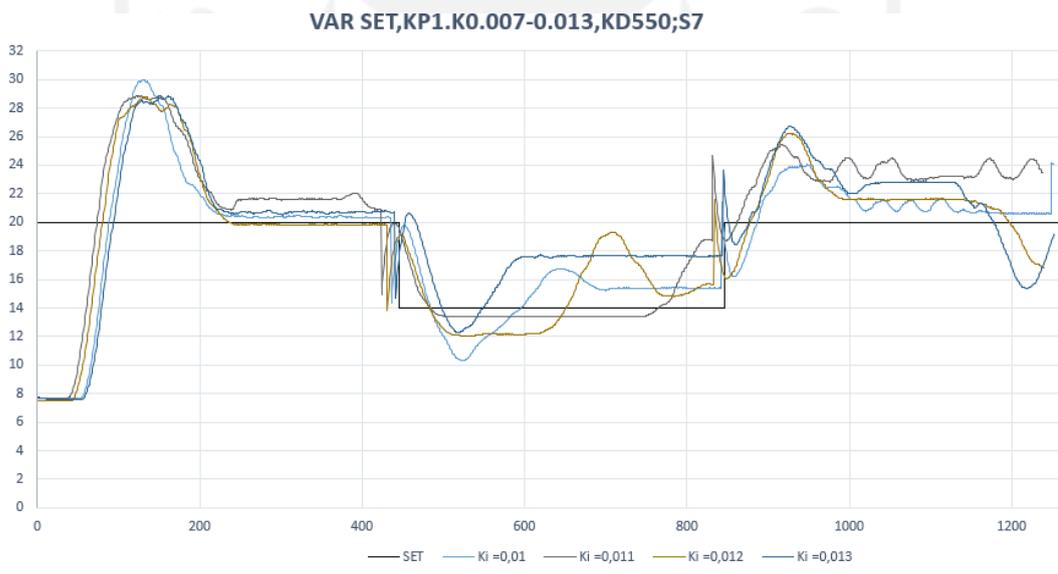
Dari gambar 4-18 dilihat bawah sampel 7 memiliki *steady state error* yang sangat kecil dan stabil, atas dasar ini pada penelitian ini akan menggunakan filter dengan 7 sampel data.

4.1.3.4 Pengujian Nilai K_i (Konstanta Integral)

Semakin besar nilai K_i akan membuat sistem lebih cepat beresilasi dan tentunya akan semakin kasar pergerakan dari sistem, sedangkan semakin kecil nilai K_i akan membuat sistem lebih sedikit beresilasi dan relatif lama, akan tetapi dengan adanya kontrol integral bisa dikatakan akan menghilangkan *steady state error* (Setiawan, 2015). Pengujian akan dilakukan dengan cara yang sama dengan memberikan gangguan *set point* secara konsisten dengan K_i 0.007-0.013.



Gambar 4-12 Pengujian K_i 0.007,0.008,0.009 dan 0.01



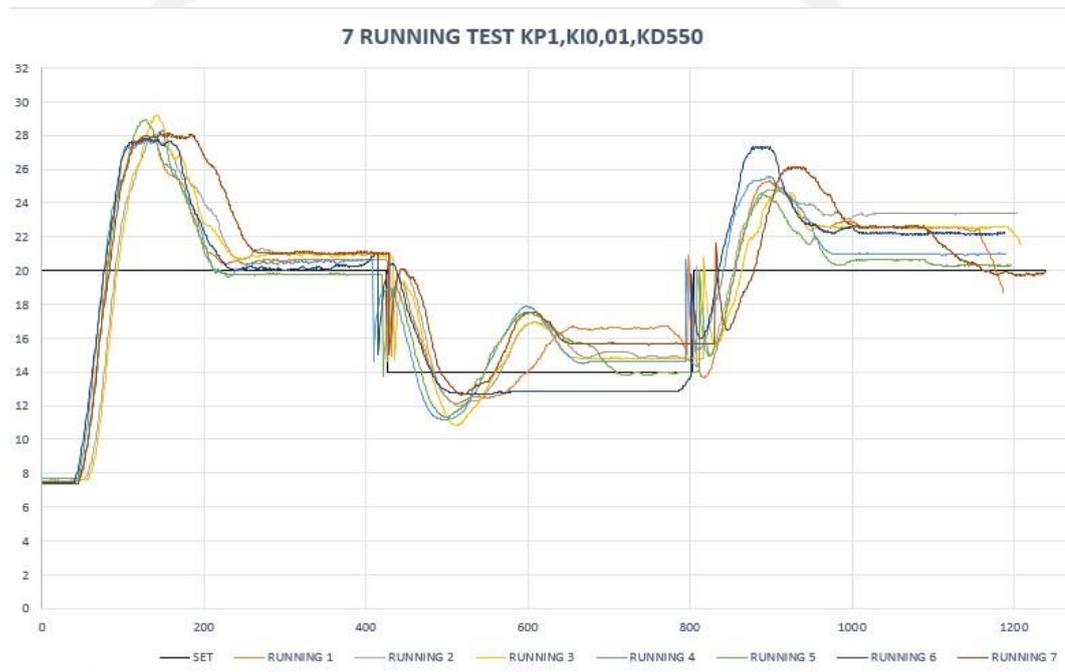
Gambar 4-13 Pengujian K_i 0.01,0.011,0.012 dan 0.013

Dari gambar 4-19 dan 4-20 adalah hasil pengujian untuk melihat *steady state error* yang paling kecil. Pada gambar 4-19 dapat dilihat bahwa K_i dengan nilai 0.01 memiliki *steady state error* yang paling kecil, sedang pada gambar 4-20 untuk 10 detik pertama $K_i = 0.012$ memiliki *steady state error* yang kecil sedangkan 10 detik kedua $K_i = 0.011$ memiliki *steady state error* yang paling kecil, secara keseluruhan $K_i = 0.01$ memiliki *steady state error* yang kecil di setiap 10

detik dan lebih stabil, oleh karena itu untuk pemilihan nilai K_i akan menggunakan $K_i = 0.01$.

4.1.3.5 *Running Test* Kontrol PID dengan Filter

Pengujian ini bertujuan untuk melihat konsistensi dari kontrol PID yang telah ditentukan dimana untuk $K_p = 1$, $K_i = 0.01$ dan $K_d = 550$, *running test* akan di ulang sebanyak 7 kali. Berikut hasilnya.



Gambar 4-14 *Running test* Kontrol PID

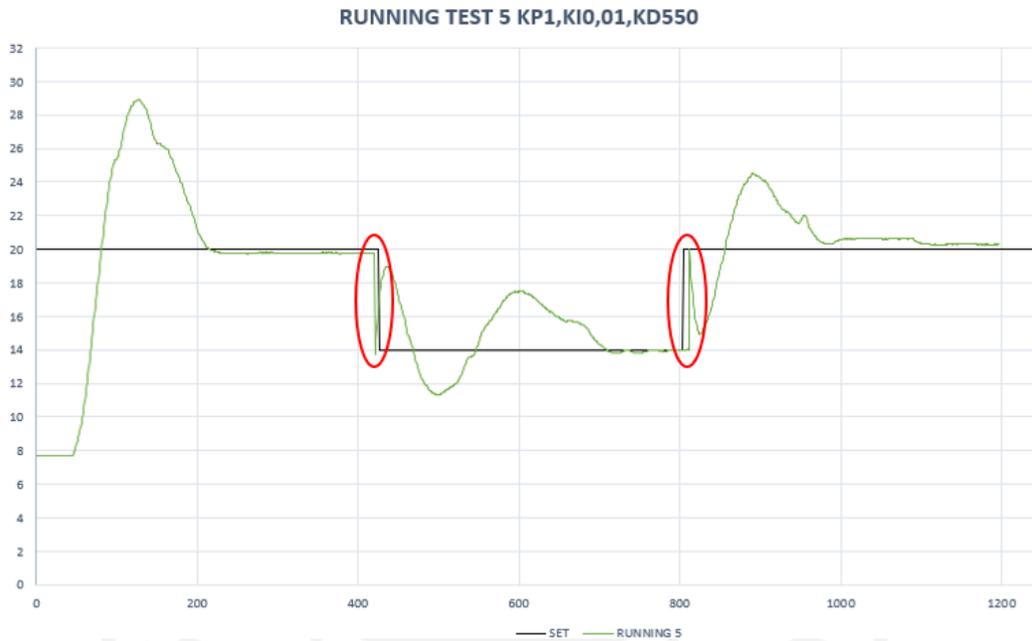
Jika dilihat dari hasil gambar 4-21 pada 10 detik pertama *steady state error* ada di kisaran 1 cm dari *set point*, untuk 10 detik kedua ada 3 tes yang memiliki *steady state error* masih di bawah 2cm, kemudian 10 detik terakhir *steady state error* masih cukup besar. 3 *Running test* terbaik yaitu.



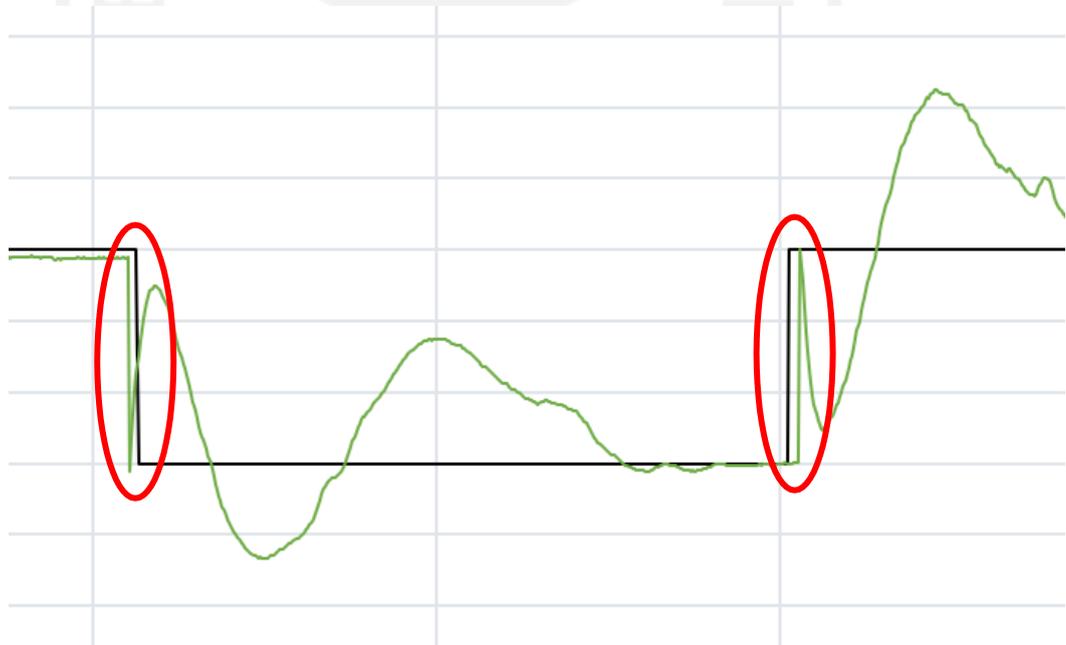
Gambar 4-15 3 *Running test* terbaik

Dari gambar 4-22 tiga percobaan terbaik ada pada percobaan ke 4, 5 dan 6, percobaan ke 5 memiliki *steady state error* dibawah 1 cm hampir 0. Pemilihan ini berdasarkan fokus penelitian yaitu untuk mengurangi *steady state error* dari sistem.

4.2 Analisis Pembahasan



Gambar 4-16 Running test ke-5



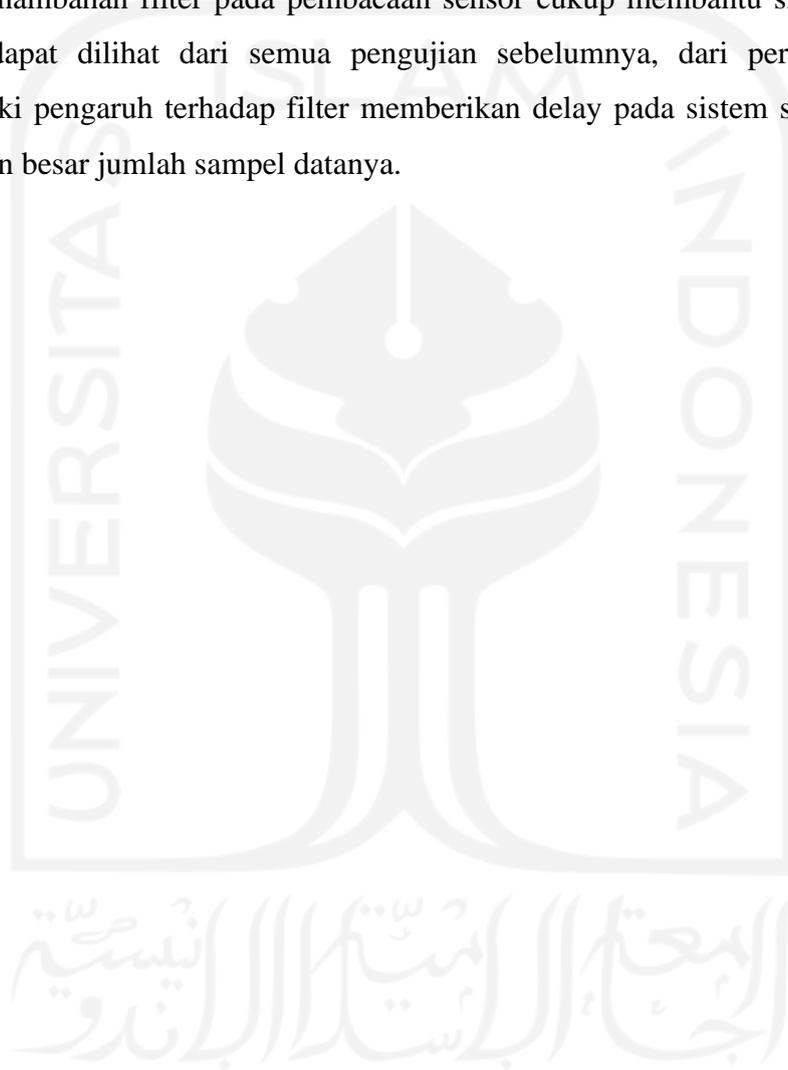
Gambar 4-17 Perbesaran dari gambar 4-25

Dari gambar 4-25 dan 4-26 dapat dilihat bawah grafik memiliki *noise* yang cukup besar, hal ini diduga disebabkan perubahan tegangan diakibatkan perubahan

set point yang diakibatkan input dari Potensiometer, asumsi ini setelah memperhatikan visual dari sistem yang tidak memiliki pergerakan berdasarkan grafik yang dimunculkan.

Pada proses tuning dari nilai K_i dan K_d diterapkan menggunakan input Potensiometer untuk memudahkan, akan tetapi karena penggunaan Potensiometer hanya satu jadi tidak bisa *men-tuning* secara bersamaan.

Penambahan filter pada pembacaan sensor cukup membantu sistem kendali yang dapat dilihat dari semua pengujian sebelumnya, dari percobaan filter memiliki pengaruh terhadap filter memberikan delay pada sistem sesuai dengan semakin besar jumlah sampel datanya.



BAB 5

PENUTUP

5.1 Kesimpulan

- a) Sistem dapat mencapai Keadaan *steady state* menggunakan kontrol PID, dimana nilai $K_p = 1$, $K_i = 0.01$ dan $K_d = 550$.
- b) Penambahan filter pada program akan mengurangi *noise* dari pembacaan sensor, selain itu jumlah data filter harus disesuaikan dengan sistem agar sistem tidak memiliki respon yang lambat.
- c) Sistem dapat mempertahankan keadaan *steady state* saat diberi gangguan secara konsisten. Gangguan yang diberikan berupa perubahan nilai *set point*.

5.2 Saran atau Penelitian Selanjutnya

- a) Dapat di kembangkan dengan menambah input kendali nilai PID menggunakan perangkat keras, sehingga memudahkan dalam *tuning* dari kontrol PID. Hal ini juga mempermudah proses penelitian.
- b) Mencari sensor yang lebih stabil untuk mendapatkan hasil yang lebih maksimal.
- c) Mencari metode filter yang lain sehingga dapat mengurangi delay dari sistem

DAFTAR PUSTAKA

- Iskandar, T., Saputra, H.M., n.d. Perbandingan Average Filter dengan Hanning Filter pada Pengolahan Sinyal Load Cell 6.
- Ivan Candra Utama, 2016. PID Control For Ping-Pong Balance With Ultrasonic Sensor And Servo Motor Based On Labview. Telekontran J. Ilm. Telekomun. Kendali Dan Elektron. Terap. 4, 68–77. <https://doi.org/10.34010/telekontran.v4i2.1891>
- Raharjo Sulistiyo Utomo, 2007. Perancangan dan Pembuatan kendali PID.
- Setiawan, I., 2015. KONTROL PID UNTUK PROSES INDUSTRI 210.
- Supardi, N.I., 2012. PENGGUNAAN PLC DALAM PENGONTROLAN TEMPERATUR, SIMULASI PADA PROTOTYPE RUANGAN 2, 8.
- Taufiqurrahman, M., Riyadi, M.A., 2016. PERANCANGAN SELF DRIVING DENGAN METODE KONTROL PD PADA SISTEM TRACKING AUTONOMOUS CAR 7.
- Wijaya, E.O.M., 2018. PERANCANGAN SISTEM KESEIMBANGAN BALL AND BEAM DENGAN MENGGUNAKAN PENGENDALI PID BERBASIS ARDUINO UNO. J. Tek. Elektro 7.
- Yunardi, R.T., 2017. Analisa Kinerja Sensor Inframerah dan Ultrasonik untuk Sistem Pengukuran Jarak pada Mobile Robot Inspection. Setrum Sist. Kendali-Tenaga-Elektron.-Telekomun.-Komput. 6, 33. <https://doi.org/10.36055/setrum.v6i1.1583>
- Data sheet Sharp gp2y0a21yk0f diakses dari <https://www.pololu.com/file/0J85/gp2y0a21yk0f.pdf> pada 15/01/2021

Lampiran Program

```
#include <Servo.h>

////////////////////////////////Input/output////////////////////////////////

int Analog_in = A4;
const int Pot_in = A0 ;
float Pot ;
long quad ;
Servo myservo; //
////////////////////////////////

////////////////////////////////Variabel////////////////////////////////

int Read = 0;
float distance = 0.0;
float time; //Variabels waktu
double distance_previous_error, distance_error;
int period =25; // waktu period akan mempengaruhi response Kd
////////////////////////////////

////////////////////////////////PID constants////////////////////////////////

double kp=1; //ZN = 0.8
double ki=0.1;
double kd=550; // ZN = 1.25
double distance_setpoint = 20;
double distance_setpoint_prev = 0;
double P, I , D , U ;
double P_prev = 0, I_prev = 0, D_prev = 0, U_prev = 0,distance_error_prev1 = 0 ;
double distance_error_prev2 = 0,distance_error_prev3 = 0,distance_error_prev4 =
0;
double distance_error_prev5 = 0,distance_error_prev6 = 0,distance_error_prev7 =
0;
```

```

double distance_error_prev8 = 0,distance_error_prev9 = 0 ;
////////////////////////////////////

void setup() {
  //analogReference(EXTERNAL);
  Serial.begin(9600);
  myservo.attach(9); // PIN servo 9
  myservo.write(90);
  pinMode(Analog_in,INPUT);
  pinMode(Pot_in,INPUT);
  time = millis();
}
// pada filter pid di pertinmbangkan menggunakan filter p,i dan d. karena akan
berbeda hasil jika filter tidak dipukul rata
void loop() {

  Pot = analogRead (Pot_in);
  // distance_setpoint = map(Pot,0,1023,8,30);
  // kp= map(Pot,0,1023,0,102.3);
  // ki= map(Pot,0,1023,0,102.3);
  //////////////////////////////////////

  if (millis() > time+period) // setelah 25 ms
  {
    time = millis();
    distance = get_dist(100);
    distance_error = distance_setpoint - distance;
  // distance_error = (distance_error +distance_error_prev1)/2 ;
  // distance_error = (distance_error +distance_error_prev1 +
distance_error_prev2)/3;
  // distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+ distance_error_prev3)/4;

```

```

// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+ distance_error_prev3 + distance_error_prev4)/5 ;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+ distance_error_prev3 + distance_error_prev4+distance_error_prev5)/6;
    distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+
        distance_error_prev3
        +
distance_error_prev4+distance_error_prev5+distance_error_prev6)/7;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+
        distance_error_prev3
        +
distance_error_prev4+distance_error_prev5+distance_error_prev6
+distance_error_prev7)/8;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+
        distance_error_prev3
        +
distance_error_prev4+distance_error_prev5+distance_error_prev6
+distance_error_prev7+ distance_error_prev8)/9;
// distance_error = (distance_error +distance_error_prev1 + distance_error_prev2
+
        distance_error_prev3
        +
distance_error_prev4+distance_error_prev5+distance_error_prev6
+distance_error_prev7+ distance_error_prev8+distance_error_prev9)/10;
    distance_error_prev1 = distance_error;
    distance_error_prev2 = distance_error_prev1;
    distance_error_prev3 = distance_error_prev2;
    distance_error_prev4 = distance_error_prev3;
    distance_error_prev5 = distance_error_prev4;
    distance_error_prev6 = distance_error_prev5;
//    distance_error_prev7 = distance_error_prev6;
//    distance_error_prev8 = distance_error_prev7;
//    distance_error_prev9 = distance_error_prev8;
    P = kp * distance_error;
float dist_diference = distance_error - distance_previous_error;
    D = kd / period *(distance_error - distance_previous_error);
    if(-3 < distance_error && distance_error < 3)

```

```

{
I = I + (ki * distance_error);
}
else
{
I = 0;
}
U = P + I + D;
U = map(U, -150, 150, 0, 150);

if (U < 20) {U = 20;}
if (U > 160) {U = 160;}
myservo.write(U+20);
quad = (distance_error*distance_error);
distance_previous_error = distance_error;
}
}
// fungsi untuk mengkalibrasi sensor

float get_dist(int n) // harus float
{
long sum=0; // fungsinya untuk membuat hasil dari sum 0 bukan bilangan negatif
for(int i=0;i<n;i++) // membatasi bawah konversi adc berdasarkan percentase 0-
100
{
sum= sum + analogRead (Analog_in); // fungsinya untuk membuat hasil dari
sum 0 bukan bilangan negatif
}

float adc=sum/n;
float distance_cm = 17569.7 * pow(adc, -1.2062); // pow (x,y) ==
pow(base,exponent)

```

```
//Serial.print(" Set point = ");
Serial.print(distance_setpoint);
Serial.print(";");
//Serial.print(" // ball at = ");
Serial.print(distance_setpoint+(-distance_error));
//Serial.print(" // Jarak = ");
//Serial.print(distance_cm);
//Serial.print(" // Kp = ");
//Serial.print(kp);
//Serial.print(" // Ki = ");
//Serial.print(ki);
//Serial.print(" // Kd = ");
//Serial.print(";");
//Serial.print(";");
//Serial.print(kd);
Serial.println(";");
return(distance_cm); //mengembalikan nilai distance_cm ke awal
}
```

Lampiran *Datasheet* Sharp GP2Y0A21YK0F

SHARP

Distance Measuring Sensor Unit
Measuring distance: 10 to 80 cm Analog
output type



DESCRIPTION

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

Features

Distance measuring range : 10 to 80 cm

Analog output type

Package size : 29.5×13×13.5 mm

Consumption current : Typ. 30 mA

Supply voltage : 4.5 to 5.5 V

Agency approvals/Compliance

1. Compliant with RoHS directive
(2002/95/EC)

Touch-less switch

(Sanitary equipment, Control of illumination, etc.)

Robot cleaner

Sensor for energy saving

(ATM, Copier, Vending machine)

Amusement equipment (Robot, Arcade game machine)

Applications



Notice The content of data sheet is subject to change without prior notice.

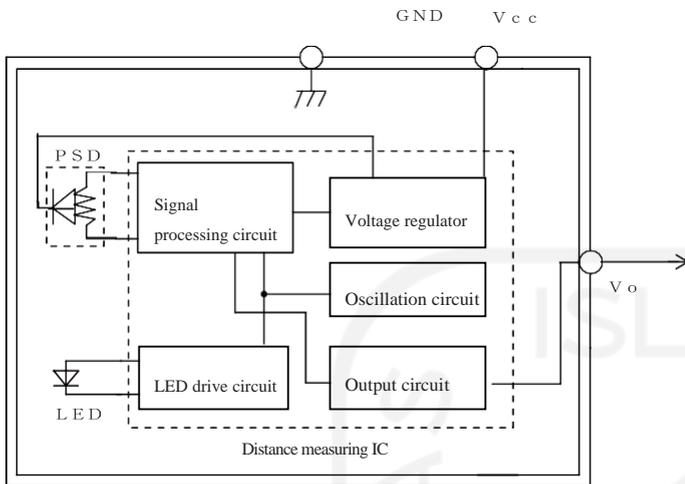
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

1 Sheet No.:E4-A00201EN

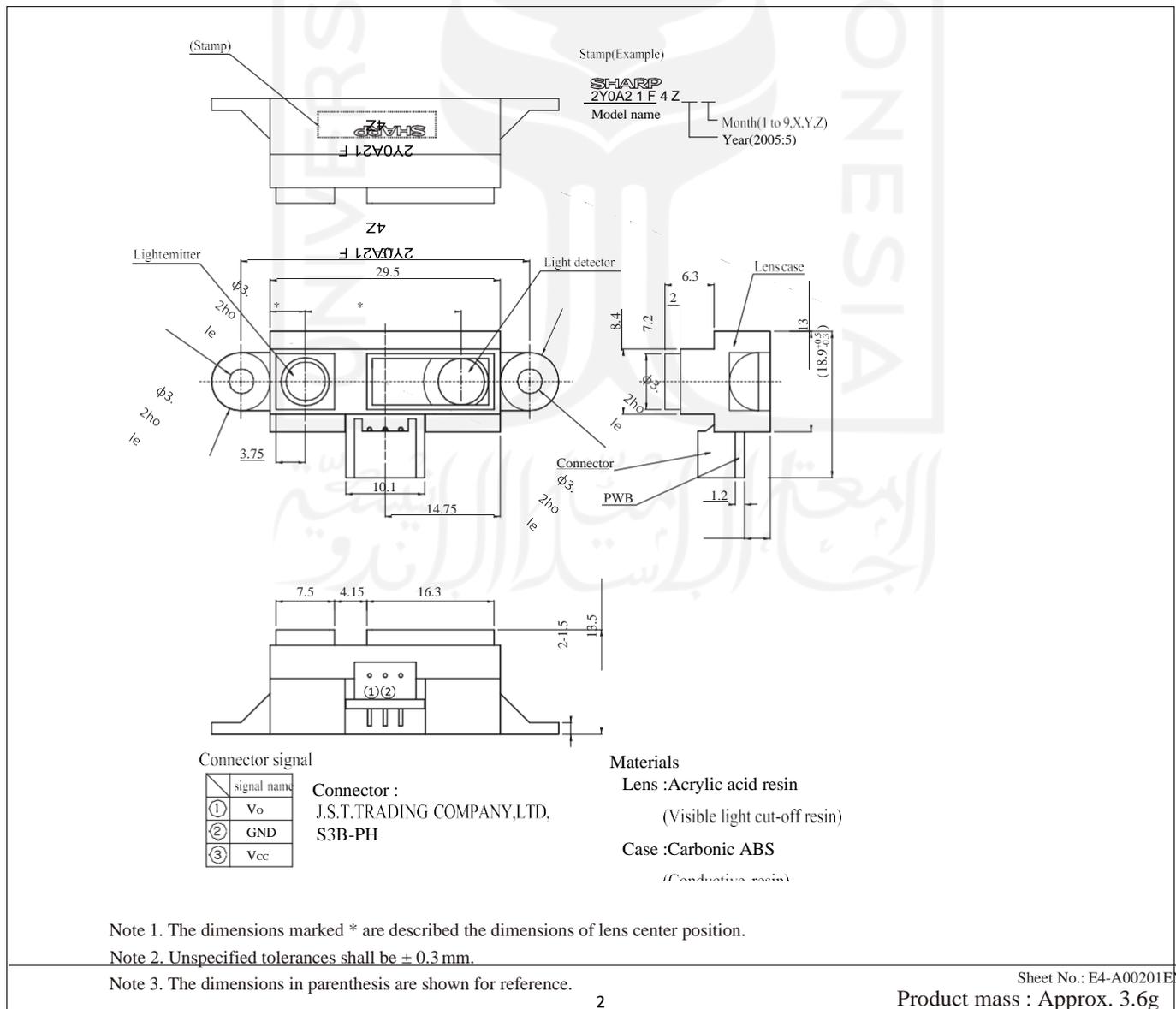
Date Dec.01.2006

©SHARP Corporation

Block diagram



Outline Dimensions (Unit : mm)



Absolute Maximum Ratings (T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	-0.3 to +7	V
Output terminal voltage	V _O	-0.3 to V _{CC} +0.3	V
Operating temperature	T _{opr}	-10 to +60	°C
Storage temperature	T _{stg}	-40 to +70	°C

Electro-optical Characteristics (T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I _{CC}	L=80cm (Note 1)	—	30	40	mA
Distance measuring	ΔL	(Note 1)	10	—	80	cm
Output voltage	V _O	L=80cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV _O	Output voltage difference between L=10cm and L=80cm (Note 1)	1.65	1.9	2.15	V

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27 • white face, reflectance; 90%)

Recommended operating conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	4.5 to 5.5	V

Fig. 1 Timing chart

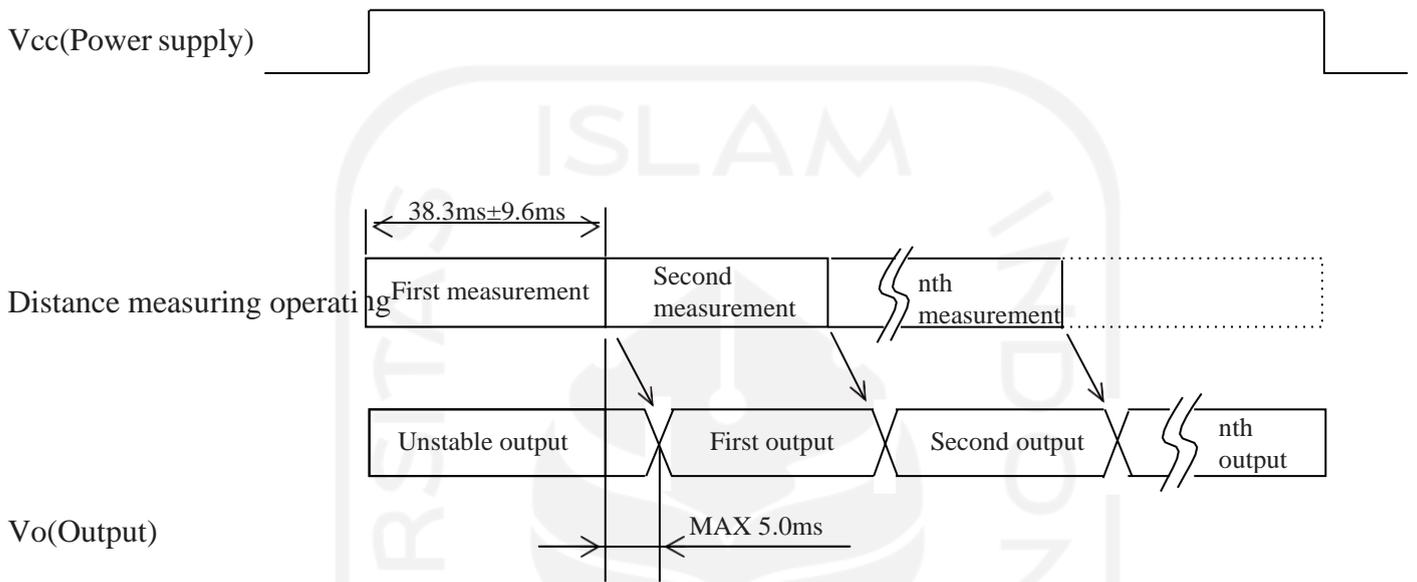
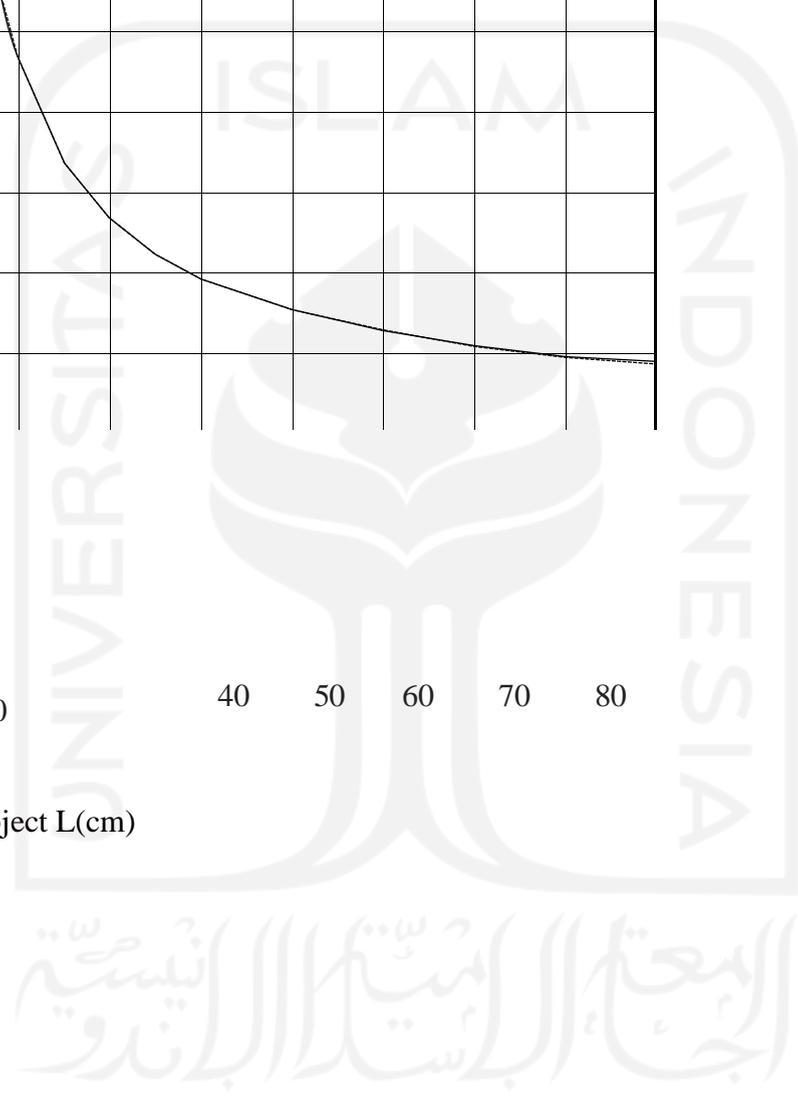
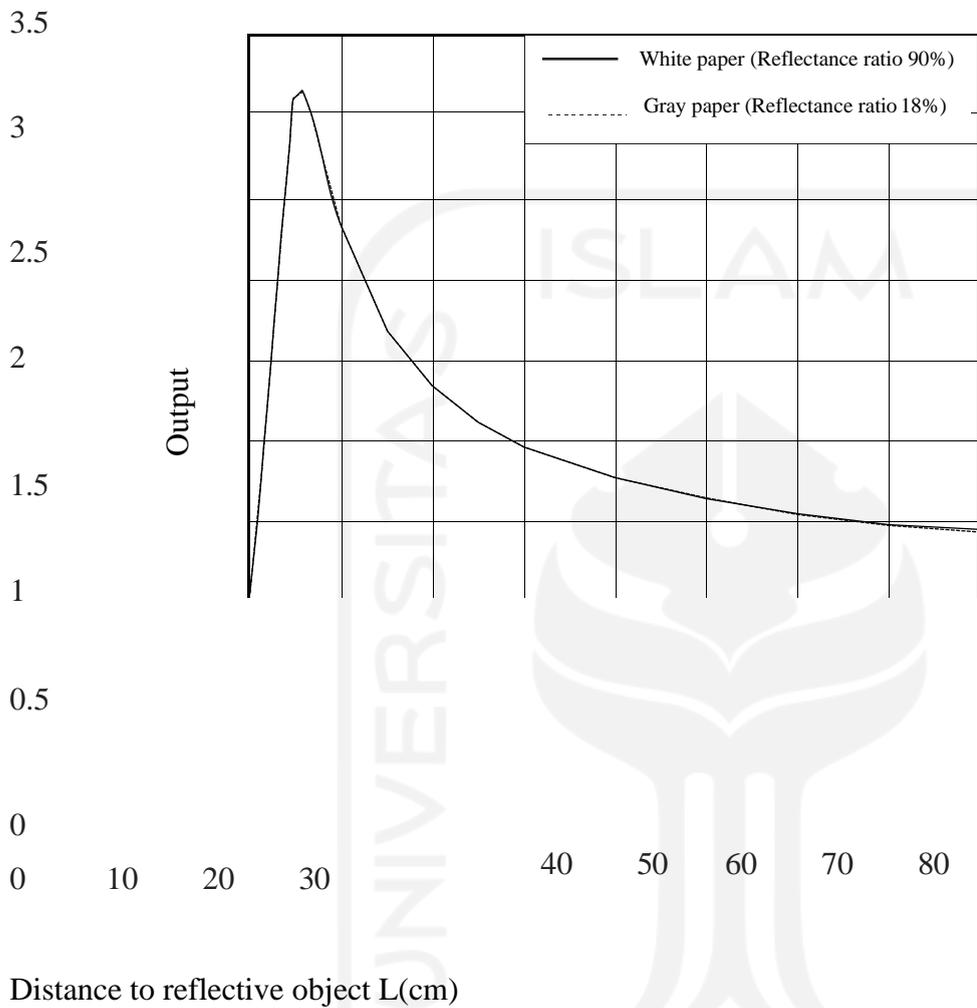
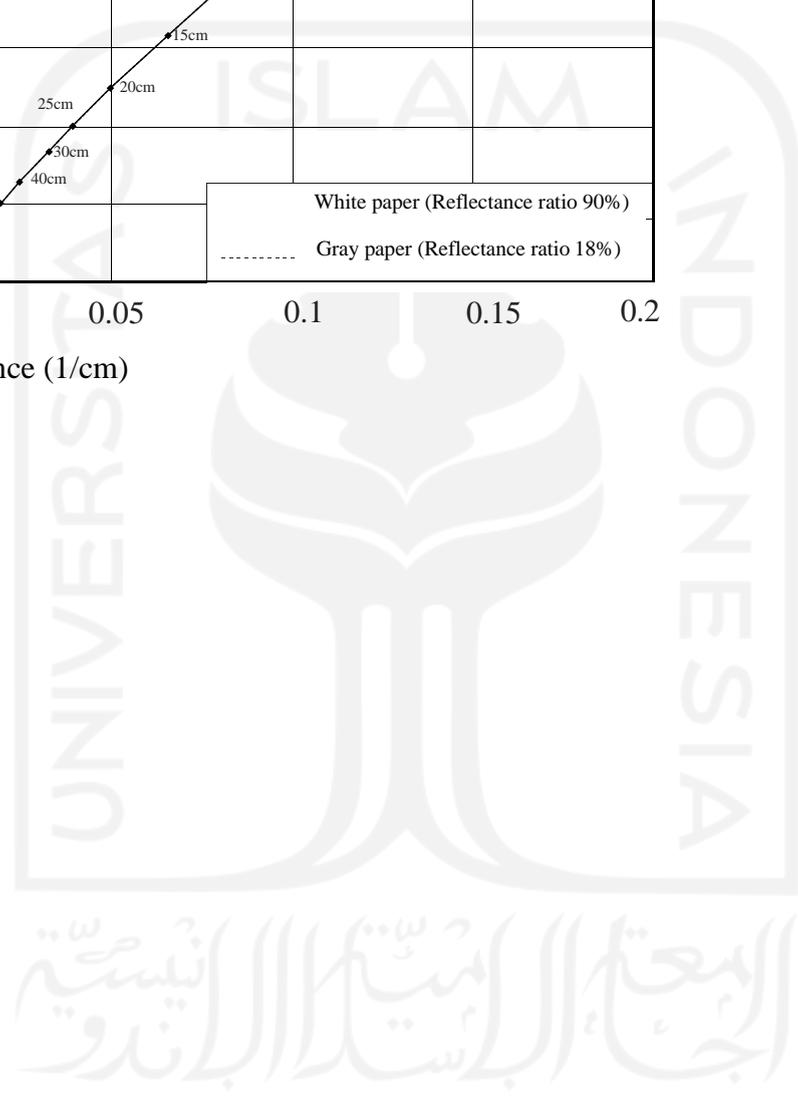
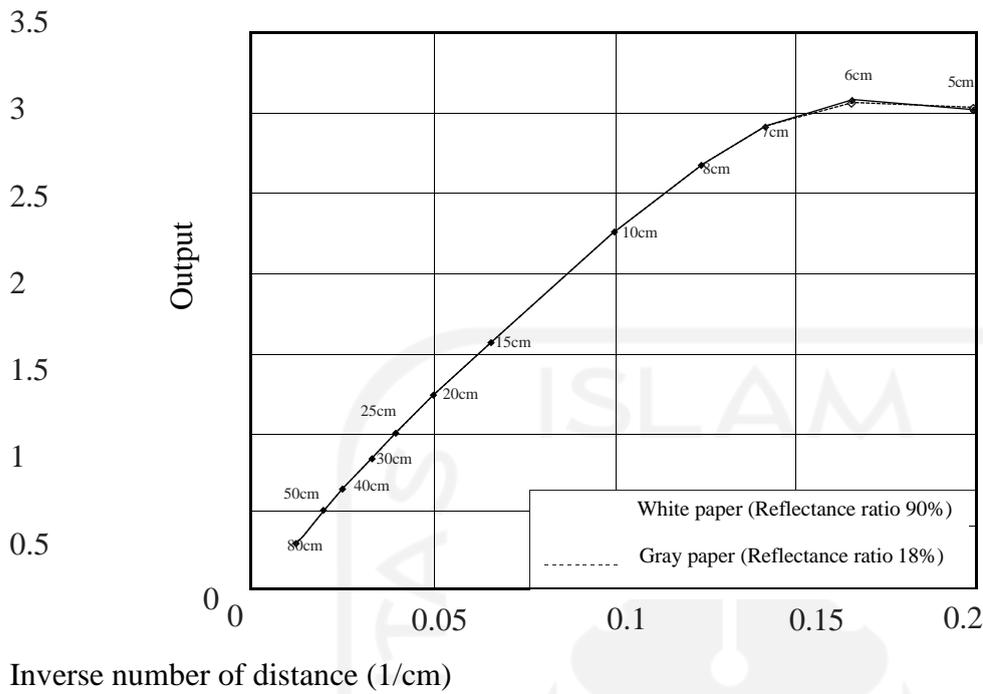


Fig. 2 Example of distance measuring characteristics(output)





Notes

Advice for the optics

The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.

Please don't do washing. Washing may deteriorate the characteristics of optical system and so on.

Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

Advice for the characteristics

In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 870 \pm 70\text{nm}$), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.

In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.

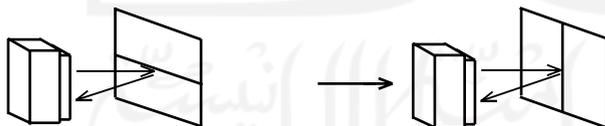
When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.

Distance to a mirror reflector can not be sometimes measured exactly.

In case of changing the mounting angle of this product, it may measure the distance exactly.

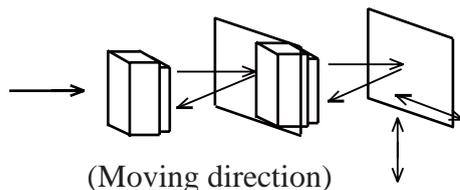
In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.

(Incorrect) (Correct)



In order to decrease deviation of measuring distance by moving direction of the reflective object, it shall be recommended to set the sensor that the moving direction of the object and the line between emitter center and detector center are vertical.

(Incorrect) (Correct)



(Moving direction)

SHARP

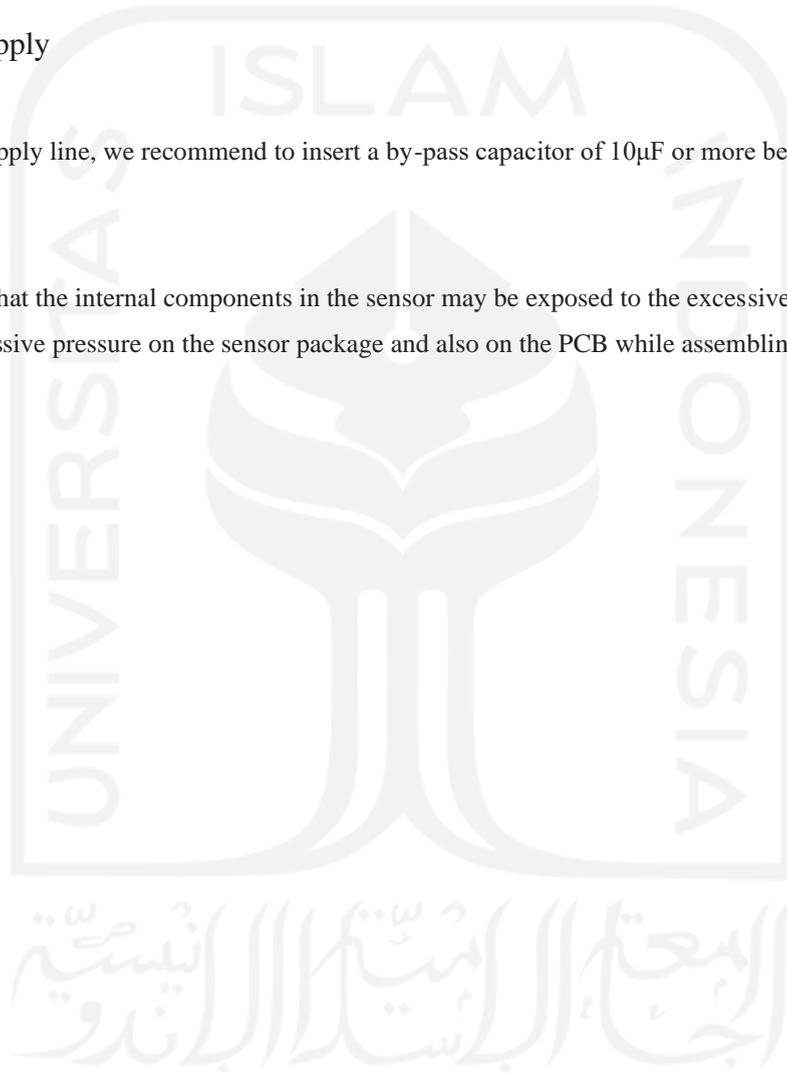
GP2Y0A21YK0F

Advice for the power supply

In order to stabilize power supply line, we recommend to insert a by-pass capacitor of 10 μ F or more between Vcc and GND near this product.

Notes on handling

There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.



Presence of ODC etc.

This product shall not contain the following materials.

And they are not used in the production process for this product.

Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

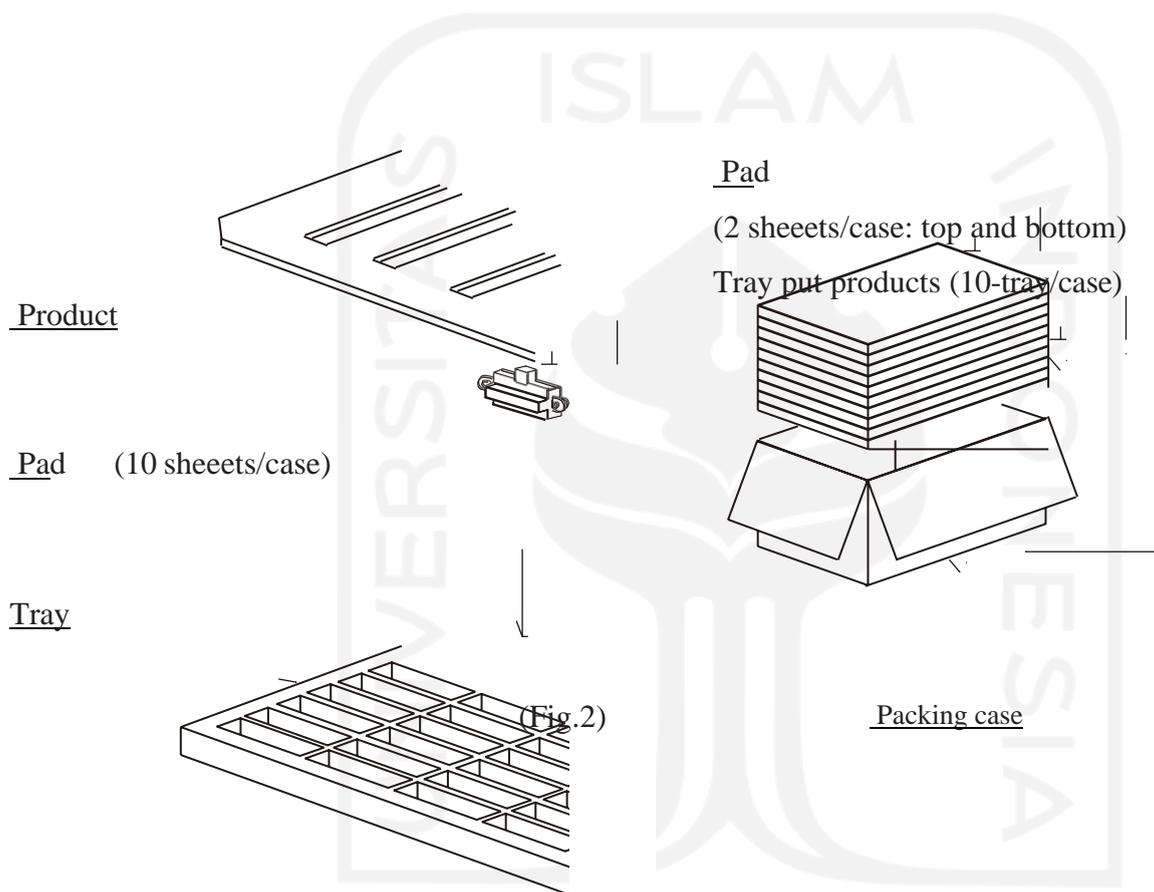
This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).

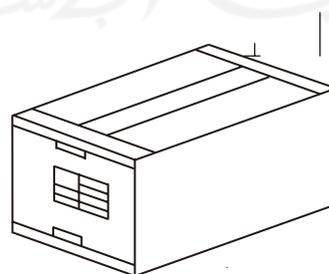


Package specification

Package composition



Craft tape
(Fig.1)



(Fig.3)

Model No. Quantity Date

Packaging method

SHARP

GP2Y0A21YK0F

Put products of 100pcs. in tray. packing method is showed in the above fig.(Fig.1)

Put them(10-tray) in the packing box. Put pads on their top and bottom. And put pads on each trays(Total 10 sheets) (Fig.2).

Seal the packing box with craft tape.

Print the model No.,quantity,inspection date (1000 pcs./a packing box)(Fig.3).



Important Notices



The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

الجامعة الإسلامية
الاستدراكية