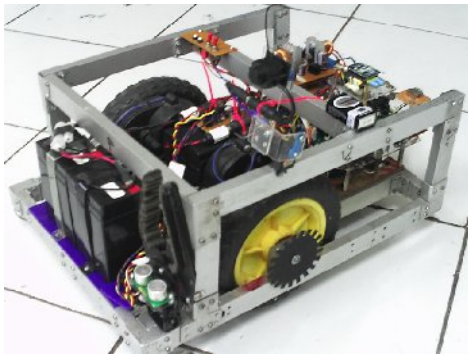
BAB IV
PENJELASAN DAN ANALISIS

Di dalam bab ini akan membahas mengenai implementasi dari hasil teori yang telah dipaparkan pada bab sebelumnya serta penjelasan mengenai pengujian, pengukuran dan analisis pada sistem yang telah dibuat. Pengujian sistem dibagi menjadi 2 bagian yaitu pengujian perangkat keras *(hardware)* dan perangkat lunak *(software)* yaitu algoritma gerak dasar dan analisa *respon* kendali PI pada kecepatan motor DC.

4.1 Penjelasan Mobile Robot

Implementasi dari dasar teori dan perancangan yang telah dipaparkan pada bab sebelumnya. Berhasil dibangun sebuah *mobile robot* yang mempunyai dimensi dengan panjang 50 cm, lebar = 40 cm dan tinggi = 25 cm. menggunakan dua roda utama sebagai penggerak dan *clater* sebagai penopang *body*, batterai *accu* dan batterai AA sebagai sumber catu daya. Pada bagian depan dipasangkan sebuah *gripper* untuk mengambil atau membawa objek dan sebuah kamera *mini wireless* yang dipasang pada bagian atas *frame*/*body* robot. Berikut adalah tampilan dari *mobile robot*:



Gambar 4.1 *Tampilan Mobile Robot*

Kemampuan yang dimiliki *mobile robot* ini diantaranya adalah bergerak lurus, belok kanan 45^0, belok kanan 90^0, belok kiri 45^0, belok kiri 90^0, dengan mengimplementasikan kendali PI sebagai algoritma dasar gerak.

4.2 Pengujian Perangkat Keras *(Hardware)*

4.2.1 Pengujian Sensor Optocoupler

Pengujian dilakukan dengan cara mengukur tegangan pada rangkaian sensor *optocoupler* dalam kondisi *high* dan *low* menggunakan *multimeter* dengan sumber catu daya sebesar ± 5V DC. Tujuan dari pengujian ini adalah sama untuk memastikan kondisi *high* atau *low*. Pada pengujian *optocoupler* sudah bisa terlihat begitu *optocoupler* terhalang warna hitam maka kondisinya akan *high* dan saat terhalang warna putih maka kondisinya akan *low*. Hasil pengujian dapat dilihat pada tabel 4.1 di bawah ini.

Tabel 4.1 *Hasil Pengujian Output Tegangan Optocoupler*

Kondisi	Tegangan	Keterangan
High	4,85 V	Diukur pada saat sensor terkena warna hitam
Low	0,128 V	Diukur pada saat sensor terkena warna putih

4.2.1.1 Pengujian Linieritas Sensor Optocoupler

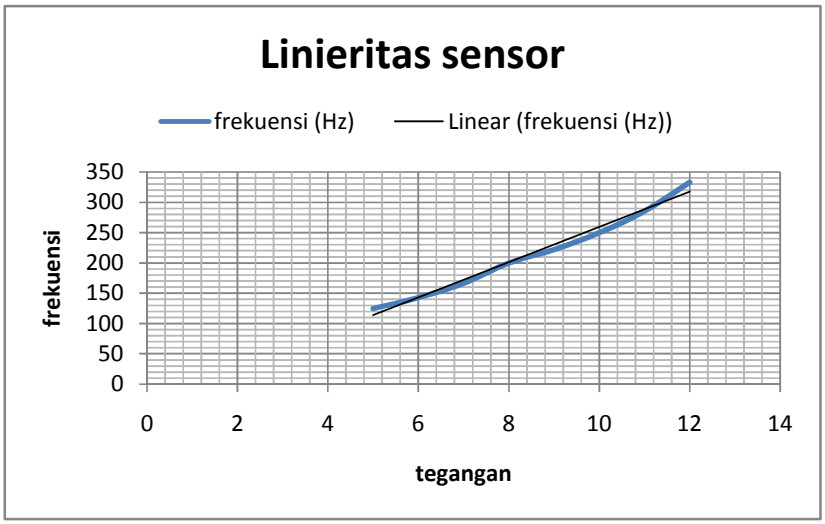
Pengujian linieritas sensor *optocoupler* dilakukan dengan cara menggerakan motor dengan memberikan nilai tegangan yang terus menaik sampai batas maksimum. Pengukuran *output* sensor *optocoupler* dilakukan dengan menggunakan osiloskop.

Tabel 4.2 merupakan hasil pengukuran frekuensi *optocoupler* menggunakan osiloskop.

Tabel 4.2 *Hasil Pengujian Output Frekuensi Optocoupler Terhadap Tegangan*

V motor (volt)	frekuensi (Hz)
5	125
6	142
7	166.
8	200
9	222
10	250
11	285
12	333

Dari tabel 4.2 diperoleh frekuensi 125 Hz untuk tegangan motor sebesar 5 volt dan 333 Hz untuk tegangan motor 12 volt.



Gambar 4.2 *Grafik Linieritas Frekuensi Optocoupler Terhadap Tegangan*

Dari tabel 4.2 dan grafik pada gambar 4.1 terlihat bahwa hubungan antara tegangan pada motor dengan frekuensi adalah linier.

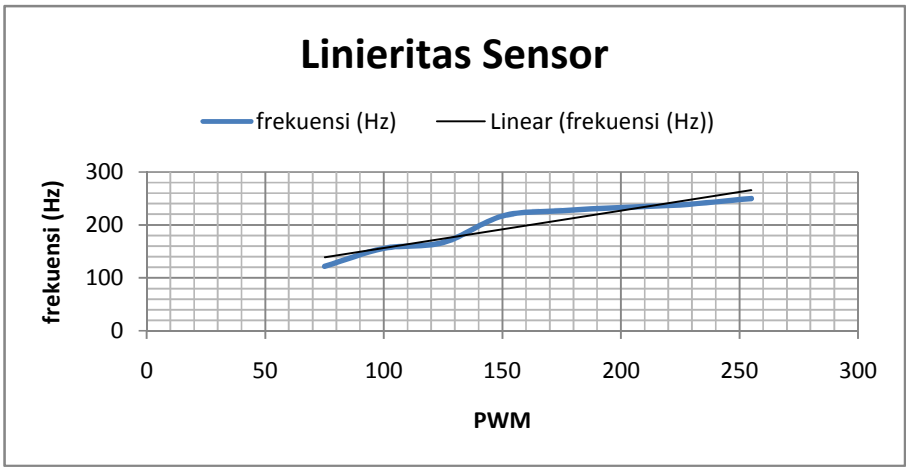
Sebagai perbandingan pengukuran linieritas dilakukan pengujian dengan meberikan nilai PWM atau lama pulsa aktif pada *driver* motor yang telah di hubungkan dengan dengan motor DC.

Berikut adalah hasil pengukuran frekuensi sensor *optocoupler* menggunakan osiloskop.

Tabel 4.3 *Hasil Pengujian Output Frekuensi Optocoupler Terhadap PWM*

PWM	Frekuensi (Hz)
75	122
100	156
125	167
150	217
175	227
200	233
225	238
255	250

Dari tabel 4.3 diperoleh frekuensi 122 Hz untuk PWM sebesar 75 dan frekuensi 250 Hz untuk PWM sebesar 255.



Gambar 4.3 *Grafik Linieritas Frekuensi Optocoupler Terhadap PWM*

Dari tabel 4.3 dan grafik pada gambar 4.2 terlihat bahwa hubungan antara PWM pada *driver* motor dengan frekuensi adalah linier.

4.2.2 Pengujian Catu Daya

Pengukuran catu daya dilakukan dengan menggunakan *multimeter* digital yang dihubungkan dengan kaki *input* untuk mengukur *input* tegangan dan kaki *output* untuk mengukur *output* tegangan, IC *regulator* yang diukur adalah LM2940 dan LM2575. Berikut adalah tabel hasil pengukuran.

Tabel 4.4 *Hasil Pengukuran Catu Daya*

IC	Input (Volt)	Output (Volt)
LM 2940	9,5	4,9
LM 2575	11,59	5,1

Dari hasil pengujian catu daya diperoleh nilai *output* tegangan IC LM2940 sebesar 4,9 volt dengan tegangan *input* sebesar 9,5 volt, sedangkan IC LM 2575 diperoleh nilai *output* teganagn sebesar 5,1 volt dengan *input* tegangan sebesar 11,59 volt.

4.3 Pengujian Perangkat Lunak *(Software)*

4.3.1 Gerak Dasar

Gerak dasar merupakan gerak yang digunakan pada *mobile robot* untuk melakukan gerakkan sesuai dengan instruksi yang diberikan melalui komuikasi serial pada PC.

4.3.1.1 Maju 100 cm

Hasil pengujian perbandingan pembacaan sensor *optocoupler* dan jarak pada gerak maju 100 cm pada *mobile robot* dapat dilihat pada tabel 4.5.

Tabel 4.5 *Hasil Pengukuran Optocoupler dan Jarak Untuk Gerak Maju 100 cm*

Percobaan ke-	Optocoupler (klip)	Jarak (cm)	$(\text{Jarak (cm)})^2$
1	2000	100	10000
2	2000	100	10000
3	2000	105	11025
4	2000	95	9025
5	2000	95	9025
6	2000	100	10000
7	2000	105	11025
8	2000	100	10000
9	2000	95	9025
10	2000	100	10000
Jumlah		995	99125

Pengujian dilakukan dengan memberikan jumlah *count* klip pada *shaft/encoder.*

Rata-rata jarak pada tabel 4.5 adalah:

Untuk perhitungan rata-rata jarak dari sepuluh kali percobaan:

$$\text{Rata} - \text{rata} = \frac{100 + 100 + 105 + 95 + 95 + 100 + 105 + 100 + 95 + 100}{10} = \frac{995}{10}$$

$$\text{Rata} - \text{rata} = 99{,}5$$

Standar deviasi rata-rata :

$$= \frac{1}{} \sqrt{\frac{\sum \quad -(\sum \quad)}{-1}}$$

$$= \frac{1}{10} \sqrt{\frac{10(99125) - (995)}{9}} = 1{,}17$$

Sehingga hasil pengukurannya :

$$(\cancel{99}{,}5 \pm 1{,}17)$$

4.3.1.2 Belok Kanan 45^0

Hasil pengujian perbandingan pembacaan sensor *optocoupler* dan sudut pada gerak belok kanan 45^0 pada *mobile robot* dapat dilihat pada tabel 4.6.

Tabel 4.6 *Hasil Pengukuran Optocoupler Untuk Sudut Belok Kanan 45^0*

Percobaan ke-	Optocoupler (klip)	Sudut (^{0})	(Sudut (^{0}))^2
1	200	45	2025
2	200	45	2025
3	200	45	2025
4	200	45	2025
5	200	50	2500
6	200	45	2025
7	200	45	2025
8	200	50	2500
9	200	45	2025
10	200	40	1600
Jumlah		455	20775

Pengujian dilakukan dengan memberikan jumlah *count* klip pada *shaft/encoder.*

Rata-rata sudut pada tabel 4.6 adalah:

Untuk perhitungan rata-rata sudut dari sepuluh kali percobaan:

$$\text{Rata} - \text{rata} = \frac{45 + 45 + 45 + 45 + 50 + 45 + 45 + 50 + 45 + 40}{10} = \frac{455}{10} = 45{,}5°$$

Standar deviasi rata-rata :

$$= \frac{1}{} \sqrt{\frac{\sum \quad -(\sum \quad)}{-1}}$$

$$= \frac{1}{10} \sqrt{\frac{10(20775) - (455)}{9}} = 0{,}89°$$

Sehingga hasil pengukurannya :

$$(\cancel{45},5 \pm 0{,}89)°$$

4.3.1.3 Belok Kiri 45^0

Hasil pengujian perbandingan pembacaan sensor *optocoupler* dan sudut pada gerak belok kiri 45^0 pada *mobile robot* dapat dilihat pada tabel 4.7.

Tabel 4.7 *Hasil Pengukuran Optocoupler Untuk Sudut Belok Kiri 45^0*

Percobaan ke-	Optocoupler (klip)	Sudut (^{0})	(Sudut (^{0}))^2
1	200	45	2025
2	200	50	2500
3	200	45	2025
4	200	45	2025
5	200	40	1600
6	200	45	2025
7	200	45	2025
8	200	50	2500
9	200	45	2025
10	200	45	2025
Jumlah		455	20775

Pengujian dilakukan dengan memberikan jumlah *count* klip pada *shaft/encoder.*

Rata-rata sudut pada tabel 4.7 adalah:

Untuk perhitungan rata-rata sudut dari sepuluh kali percobaan

$$\text{Rata} - \text{rata} = \frac{45 + 50 + 45 + 45 + 40 + 45 + 45 + 50 + 45 + 45}{10} = \frac{455}{10} = 45{,}5°$$

Standar deviasi rata-rata :

$$= \frac{1}{} \sqrt{\frac{\sum \quad -(\sum \quad)}{-1}}$$

$$= \frac{1}{10} \sqrt{\frac{10(20775) - (455)}{9}} = 0{,}89°$$

Sehingga hasil pengukurannya :

$$(45{,}5 \pm 0{,}89)°$$

4.3.1.4 Belok Kanan 90^0

Hasil pengujian perbandingan pembacaan sensor *optocoupler* dan sudut pada gerak belok kanan 90^0 pada *mobile robot* dapat dilihat pada tabel 4.8.

Tabel 4.8 *Hasil Pengukuran Optocoupler Untuk Sudut Belok Kanan 90^0*

Percobaan ke-	Optocoupler (klip)	Sudut (^{0})	(Sudut (^{0}))^2
1	400	90	8100
2	400	90	8100
3	400	85	7225
4	400	90	8100
5	400	90	8100
6	400	85	7225
7	400	85	7225
8	400	90	8100
9	400	90	8100
10	400	90	8100
Jumlah		885	78375

Pengujian dilakukan dengan memberikan jumlah *count* klip pada *shaft/encoder.*

Rata-rata jarak pada tabel 4.8 adalah:

Untuk perhitungan rata-rata sudut dari sepuluh kali percobaan:

$$\text{Rata} - \text{rata} = \frac{90 + 90 + 85 + 90 + 90 + 85 + 85 + 90 + 90 + 90}{10} = \frac{885}{10} = 88{,}5°$$

Standar deviasi rata-rata :

$$= \frac{1}{} \sqrt{\frac{\sum \quad -(\sum \quad)}{-1}}$$

$$= \frac{1}{10} \sqrt{\frac{10(78375) - (885)}{9}} = 0{,}76°$$

Sehingga hasil pengukurannya :

$$(88{,}5 \pm 0{,}76)°$$

4.3.1.5 Belok Kiri 90^0

Hasil pengujian perbandingan pembacaan sensor *optocoupler* dan sudut pada gerak belok kiri 90^0 pada *mobile robot* dapat dilihat pada tabel 4.9.

Tabel 4.9 *Hasil Pengukuran Optocoupler Untuk Sudut Belok Kiri 90^0*

Percobaan ke-	Optocoupler (klip)	Sudut (^{0})	(Sudut (^{0}))^2
1	400	90	8100
2	400	90	8100
3	400	85	7225
4	400	90	8100
5	400	90	8100
6	400	85	7225
7	400	90	8100
8	400	90	8100
9	400	90	8100
10	400	90	8100
Jumlah		890	79250

Pengujian dilakukan dengan memberikan jumlah *count* klip pada *shaft/encoder.*

Rata-rata jarak pada tabel 4.9 adalah:

Untuk perhitungan rata-rata sudut dari sepuluh kali percobaan

$$\text{Rata} - \text{rata} = \frac{90 + 90 + 85 + 90 + 90 + 85 + 90 + 90 + 90 + 90}{10} = \frac{890}{10} = 89°$$

Standar deviasi rata-rata :

$$= \frac{1}{} \sqrt{\frac{\sum \quad -(\sum \quad)}{-1}}$$

$$= \frac{1}{10} \sqrt{\frac{10(79250) - (890)}{9}} = 0{,}67°$$

Sehingga hasil pengukurannya :

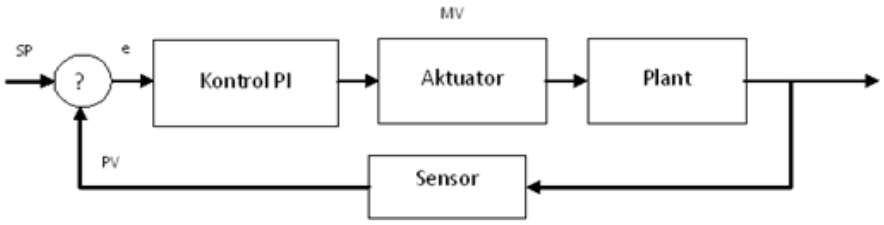
$$(89 \pm 0{,}67)°$$

4.3.2 Pengujian Kendali PI

Pengujian kendali PI dilakukan pada RPM 2000 – 3500 dengan nilai acuan masukan *(setpoint)* pada RPM 3000. Nilai RPM *(Rotasi per Minute)* diperoleh dari:

$$= \frac{60}{h} \times$$

Pemberian nilai parameter Kp dan Ki untuk memperoleh *respon* sistem yang optimal dilakukan dengan menggunakan metode manual *(hand tunning/trial error)*. Hasil *output* (proses variable = VP) ditampilkan pada layar monitor dengan memanfaatkan program *debug* yang ada pada *compiler*.



Gambar 4.4 *Diagram Blok Kendali PI*

Pengujian diawali dengan memberikan nilai *setpoint* (SP) pada sistem. Jika terjadi *error* (e) kendali PI akan mengembalikan nilai *output* ke *setpoint*-nya dengan merubah nilai variable manipulasi (MV). Informasi perubahan *error* di kirim kembali ke sistem sebagai variabel proses (PV).

Berikut di bawah ini merupakan penggalan instruksi untuk prosedur kontrol PI pada *mobile* robot:

```
Right_motor_dir1 = 1
Right_motor_dir2 = 0
Right_motor_enable = 1
Right_motor_pwm = Right_pwm

Pulsein Right_opto , Pind , 7 , 0
Rpm_kanan = 1000 / Right_opto

Error = Setpoint - Rpm_kanan
P = Kp * Error
Temp = Ki * Error
I = Temp + Ki_error_acc
Pi = P + I

If Pi < -150 Then
    Ki_error_acc = -150 – P
Elseif Pi > 150 Then
    Ki_error_acc = 150 – P
Else
    Ki_error_acc = I
End If

Pi = P + Ki_error_acc

Temp = Pi + 150
Temp1 = Temp * 180
Temp2 = Temp / 300
Right_motor_pwm = 75 + Temp2
```

Tunning kendali PI dilakukan dengan mengatur nilai Kp (konstanta *proporsional*) dan Ki (konstanta *integral*). Pengujian dilakukan dengan memberikan nilai Kp = 1 sampai 10 dan Ki = 1 sampai 8.

Pengujian ini bertujuan untuk mengetahui *respon time* terbaik dari masing-masing hasil *tunning* yang dilakukan.

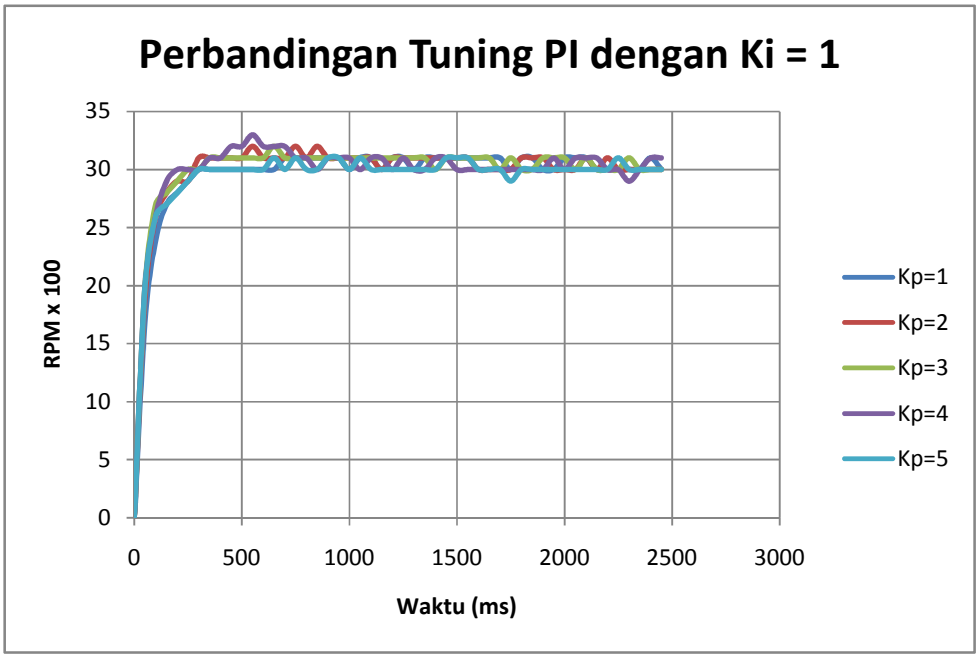
Percobaan ke-1

Dari hasil percobaan dan pengukuran yang ke-1 dengan memberikan nilai Ki = 1 dan Kp diubah dari 1 sampai 10 didapat hasil seperti tabel 4.10 berikut:

Tabel 4.10 *Perbandingan Tunning Parameter Kp dan Ki = 1 Terhadap Waktu*

Waktu (ms)	RPM x 100 (setpoint 3000 RPM)									
	Ki=1									
	Kp=1	Kp=2	Kp=3	Kp=4	Kp=5	Kp=6	Kp=7	Kp=8	Kp=9	Kp=10
0	0	0	0	0	0	0	0	0	0	0
50	17	19	20	20	20	21	8	7	7	7
100	24	26	27	26	26	29	13	12	12	13
150	27	28	28	29	27	32	18	17	17	18
200	28	29	29	30	28	33	21	21	20	21
250	29	29	30	30	29	32	24	24	23	23
300	30	31	30	30	30	33	26	25	23	23
350	30	31	31	31	30	32	27	25	25	24
400	30	31	31	31	30	32	27	26	25	25
450	30	31	31	32	30	32	28	27	25	25
500	30	31	31	32	30	31	27	27	26	25
550	30	32	31	33	30	30	28	27	26	25
600	30	31	31	32	30	31	28	27	27	25
650	30	31	32	32	31	31	28	27	27	26
700	31	31	31	32	30	31	28	27	27	27
750	31	32	31	31	31	30	29	27	27	27
800	31	31	31	31	30	30	29	27	27	26
850	31	32	31	30	30	30	28	27	28	27
900	31	31	31	31	31	30	28	28	27	27
950	31	31	31	31	31	29	30	29	28	27
1000	30	31	31	31	30	30	29	28	28	27
1050	31	31	31	30	31	30	29	29	28	27
1100	31	31	31	31	30	31	30	29	28	28
1150	30	30	31	31	30	31	31	29	28	27
1200	31	31	31	30	30	30	30	29	28	27
1250	31	31	31	31	30	30	31	28	29	28
1300	30	31	31	30	30	30	29	29	29	28
1350	31	31	31	30	30	31	28	29	28	28
1400	31	31	30	31	30	29	28	30	29	28
1450	31	31	31	31	31	30	29	30	28	29
1500	31	31	31	30	31	30	26	27	29	30

Dari tabel 4.10 dapat dijelaskan nilai RPM menaik dari nol sampai dengan *setpoint* yang diinginkan yaitu sebesar 3000 RPM terhadap waktu. Data RPM pada tabel di atas mempunyai skala 1 : 100. Dengan algorima kendali PI, nilai kecepatan motor (RPM) akan berada pada *setpoint* atau disekitar *setpoint* ketika sistem telah stabil. Untuk nilai konstanta Kp = 1 dan Ki = 1 pertama kali mencapai *setpoint* pada waktu 300 ms, sedangkan untuk nilai Kp = 10 dan Ki = 1 pertama kali mencapai *setpoint* pada waktu 1850 ms atau 1,85 detik. Perbedaan tersebut diakibatkan hasil kali konstanta Kp dan Ki terhadap *error*-nya. Untuk melihat perbandingannya dapat dilitat pada gambar 4.5.



Gambar 4.5 *Grafik Perbandingan Tunning PI dengan Ki = 1*

Gambar 4.5 merupakan grafik perbandingan nilai RPM terhadap waktu untuk nilai Parameter Ki = 1 dan Kp dengan nilai 1 sampai 5. Kecepatan motor bergerak dari nol menuju *setpoint* sebesar 3000 RPM. Osilasi gelombang untuk nilai konstanta Kp = 5 lebih kecil dibandingkan dengan osilasi gelombang nilai konstanta Ki = 1. *Tunning* nilai konstanta Kp dan konstanta Ki akan mempengaruhi bentuk osilasi dan *respon time*-nya.

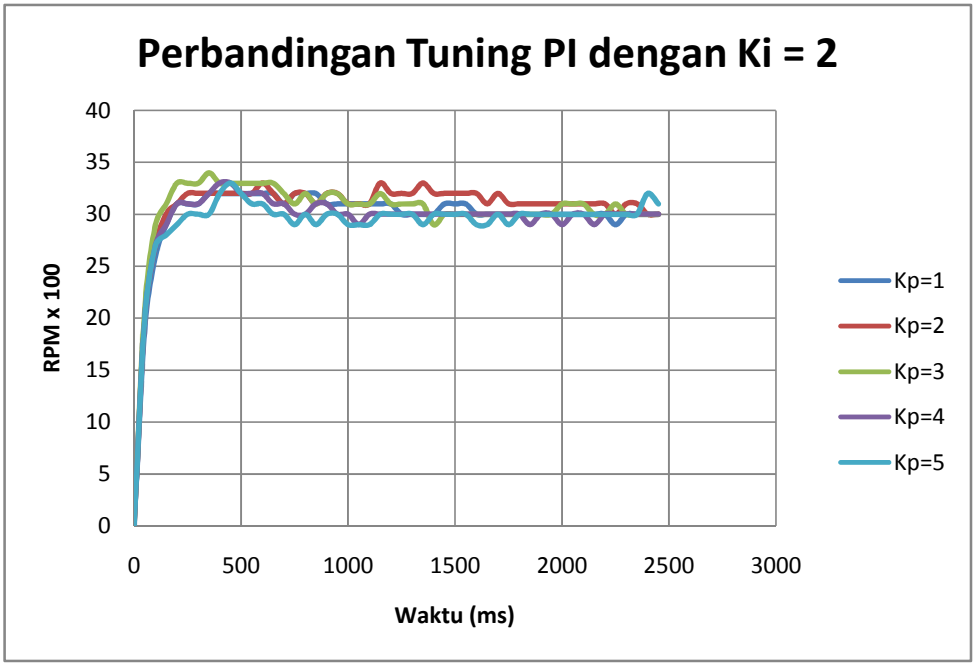
Percobaan ke-2

Dari hasil percobaan dan pengukuran dengan memberikan nilai Ki = 2 dan Ki diubah dari 1 sampai 10 didapat hasil seperti pada tabel 4.11 berikut:

Tabel 4.11 *Perbandingan Tunning Parameter Kp dan Ki =2 Tererhadap Waktu*

Waktu (ms)	RPM x 100 (setpoint 3000 RPM)									
	Ki=2									
	Kp=1	Kp=2	Kp=3	Kp=4	Kp=5	Kp=6	Kp=7	Kp=8	Kp=9	Kp=10
0	0	0	0	0	0	0	0	0	0	0
50	19	20	21	20	20	8	7	7	7	8
100	26	27	29	27	27	13	12	12	12	13
150	29	30	31	29	28	18	17	17	16	19
200	31	31	33	31	29	21	21	21	21	22
250	31	32	33	31	30	25	25	25	24	24
300	31	32	33	31	30	28	27	27	26	25
350	32	32	34	32	30	30	28	28	27	27
400	32	32	33	33	32	32	30	29	28	27
450	32	32	33	33	33	31	31	30	30	27
500	32	32	33	32	32	32	31	30	29	28
550	32	32	33	32	31	32	32	31	30	29
600	32	33	33	32	31	31	31	30	30	29
650	32	32	33	31	30	31	30	31	30	29
700	31	31	32	31	30	32	32	31	30	29
750	32	32	31	30	29	32	31	31	31	29
800	32	32	32	30	30	31	32	30	31	27
850	32	31	31	31	29	32	32	31	30	27
900	31	32	32	31	30	32	32	31	29	28
950	31	32	32	30	30	31	31	31	30	28
1000	31	31	31	30	29	32	31	31	31	28
1050	31	31	31	29	29	31	31	31	31	29
1100	31	31	31	30	29	31	27	30	29	28
1150	31	33	32	30	30	30	32	31	27	30
1200	31	32	31	30	30	31	31	31	29	29
1250	30	32	31	30	30	31	32	30	28	30
1300	30	32	31	30	30	31	29	30	28	31
1350	30	33	31	30	29	32	31	30	29	30
1400	30	32	29	30	30	30	31	30	27	30
1450	31	32	30	30	30	31	30	31	29	31
1500	31	32	30	30	30	31	31	29	27	30

Dari tabel 4.11 dapat dijelaskan nilai RPM menaik dari nol sampai dengan *setpoint* yang diinginkan yaitu sebesar 3000 RPM terhadap waktu. Data RPM pada tabel di atas mempunyai skala 1 : 100. Dengan algorima kendali PI, nilai kecepatan motor (RPM) akan berada pada *setpoint* atau disekitar *setpoint* ketika sistem telah stabil. Untuk nilai konstanta Kp = 1 dan Ki = 1 pertama kali mencapai *setpoint* pada waktu 1250 ms atau 1,25 detik, sedangkan untuk nilai konstanta Kp = 5 dan Ki = 1 pertama kali mencapai *setpoint* pada waktu 250 ms. Perbedaan tersebut diakibatkan hasil kali konstanta Kp dan Ki terhadap *error*-nya. Untuk melihat perbandingannya dapat dilitat pada gambar 4.6.



Gambar 4.6 *Grafik Perbandingan Tunning PI dengan Ki = 2*

Gambar 4.6 merupakan grafik perbandingan nilai RPM terhadap waktu untuk nilai Parameter Ki = 1 dan Kp dengan nilai 1 sampai 5. Kecepatan motor bergerak dari nol menuju *setpoint* sebesar 3000 RPM. Osilasi gelombang untuk nilai konstanta Kp = 5 lebih kecil dibandingkan dengan osilasi gelombang nilai konstanta Ki = 3. *Tunning* nilai konstanta Kp dan konstanta Ki akan mempengaruhi bentuk osilasi dan respon time-nya.

Pengujian sama dilakukan pada parameter Kp dan Ki yang lain, sampai dengan nilai Kp = 10 dan Ki = 8. Untuk tabel dan grafik pengujian dapat dilihat pada lampiran.

4.3.2.1 Respon Time

Setelah melakukan *tunning* parameter konstanta Kp dan Ki, kita dapat mengetahui seberapa baik hasil *output* yang diperoleh setelah melakukan *tunning* pada parameter kendali PI. kita dapat mengetahui apakah *tunning* yang dilakukan itu baik atau tidak baik dilihat pada *respon time* yang diperoleh setelah melakukan *tunning* parameter Kp dan Ki pada kendali PI

Respon time merupakan waktu pertama kali yang dibutuhkan untuk mencapai keadaan stabil. Diawali dengan menentukan batas *respon time* terbaik sesuai dengan yang kita kehendaki.

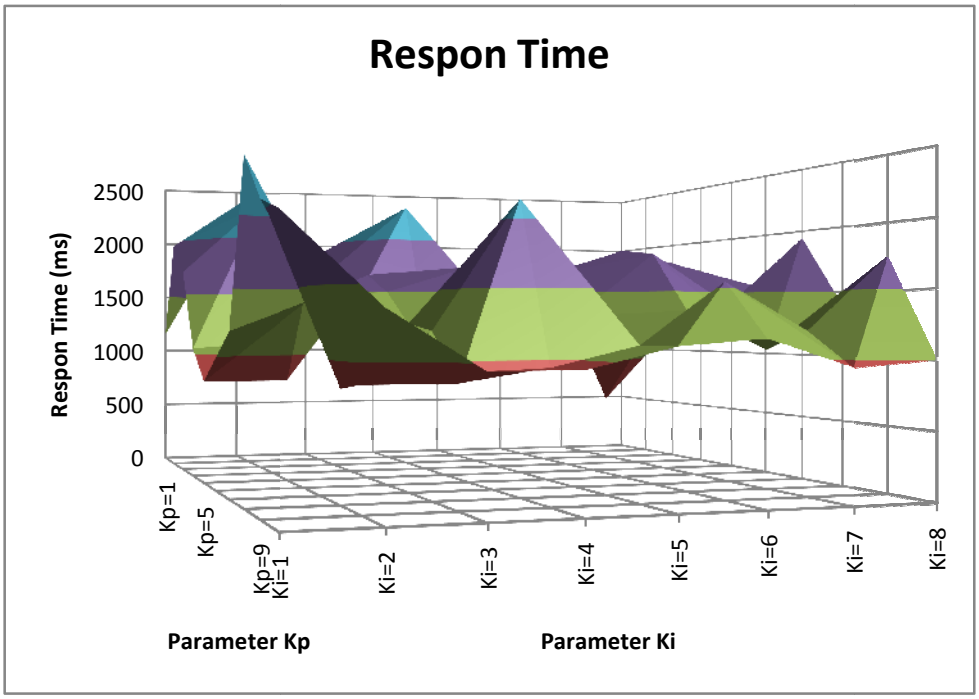
Setelah mendapatkan perbandingan hasil kombinasi *tunning* parameter Kp dan Ki kita dapat mengetahui *respon time* pada masing-masing *tunning*. *Respon time* dari masing-masing *tunning* parameter Kp dna Ki dapat dilihat pada tabel 4.12.

Tabel 4.12 *Perbandingan Respon Time (milisecond)*

Kp \ Ki	1	2	3	4	5	6	7	8
1	1150	2300	1100	1400	1350	1200	1100	1750
2	1950	2400	1900	1350	1350	1200	1550	1500
3	1700	2300	1750	2300	1450	1150	1250	1050
4	1050	1100	1350	1250	1350	1200	1300	1700
5	800	800	1650	900	850	1650	1800	1550
6	1000	1400	1500	1700	1650	1850	1100	1550
7	1200	1450	1250	1300	1300	1800	1400	1950
8	2400	800	1200	2150	700	1350	1050	1400
9	2050	850	850	950	1100	1550	1200	1750
10	1950	1350	950	950	1100	1150	950	1000

Tabel 4.12 merupakan tabel nilai *respon time* dari masing-masing hasil *tunning* konstanta Kp dan Ki. Nilai repon time terbaik berada pada Kp = 8 dan Ki = 5, yaitu sebesar 700 *milisecond.* Dan catatan waktu *respon time* terlama yaitu pada Kp = 1 dan Ki = 8 juga untuk nilai Kp = 2 dan Ki = 2

Untuk mengetahu perbandingan *respon time* secara grafik dapat dilihat pada gambar 4.7.



Gambar 4.7 *Grafik Respon Time Kendali PI*

Dari gambar 4.7 dapat dijelaskan bahwa *repon time* terbaik dari hasil *tunning* parameter Kp dan Ki adalah *respon time* untuk nilai parameter Kp = 8 dan Ki = 5 dangan nilai respon time sebesar 700 ms.

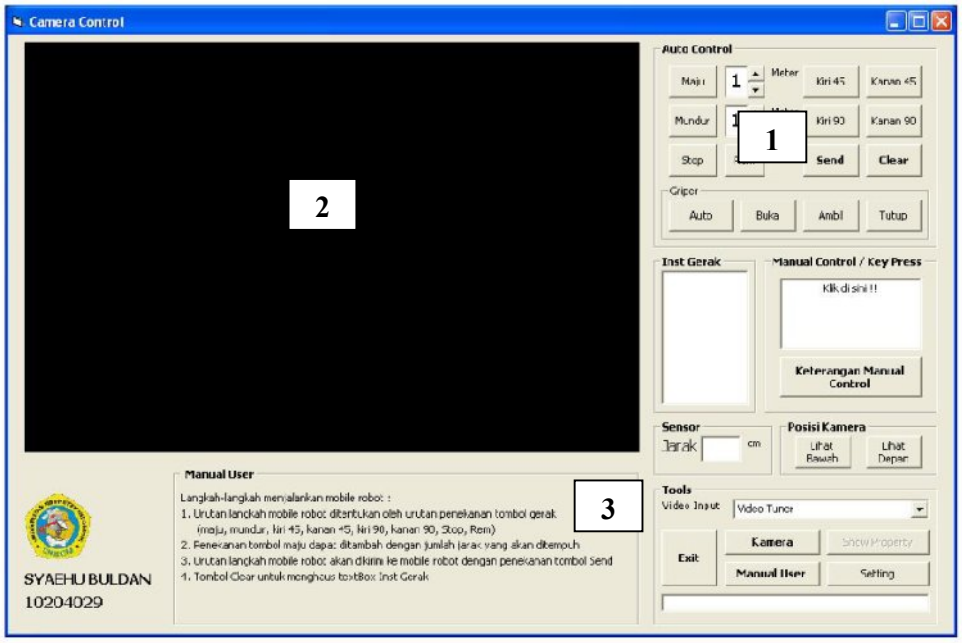
Pada kontrol PI secara praktis akan sangat mempengaruhi dinamika keluaran variable *output* proses (PV) yang sedang dikontrol. Jika kedua parameter kontrol (Kp dan Ki) ini dipilih secara tepat, maka respon yang didapat akan optimal (sesuai harapan). Pembesaran atau pengecilan nilai Kp dan Ki dari nilai optimalnya dapat menyebabkan respon menjadi terlalu lambat atau terlalu

berosilasi. Nilai parameter PI yang optimal pada dasarnya sangat tergantung dari model dan nilai parameter proses yang dikontrol.

4.3.3 Penjelasan Interface Visual Basic

Berikut ini akan dijelaskan mengenai implementasi *software* Visual Basic sebagai *interface* kendali *mobile robot*. *Software* ini akan ditampilkan pada layar monitor PC dan digunakan sebagai *input* masukan untuk *mobile robot*.

Menu utama meliputi objek-objek seperti *form*, *button* (tombol), *textbox* (kotak isian) *label* dan *activeX*. implementasi menu utama dapat dilihat pada gambar 4.8.



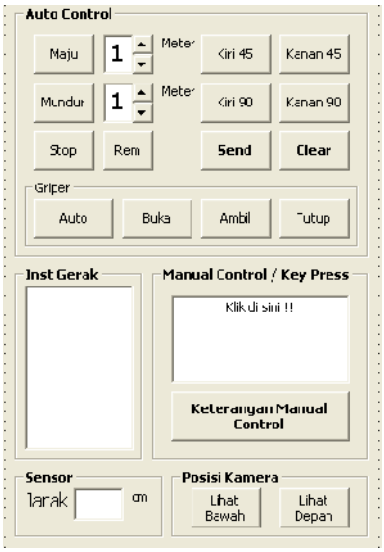
Gambar 4.8 *Tampilan Menu Utama*

Gambar 4.8 merupakan tampilan *window* untuk kendali *mobile robot* yang di dalamnya terdapat 3 bagian utama yaitu:

4.3.3.1 Input Panel Kendali Mobile Robot

Pada bagian ini merupakan bagian tombol *input* gerak diantaranya tombol "Maju" untuk melakukan gerak maju, Tombol "Mundur" untuk gerak mundur,

"Kiri 45" untuk belok kiri 45^0, "Kiri 90" untuk belok kiri 90^0dan tombol "Send" untuk menggirim karakter ke *mobile robot*.



Gambar 4.9 *Input Panel*

Instruksi yang akan diberikan kepada *mobile robot* pertama kali di*input*kan lalu dikirimkan dengan menekan tombol "Send".

4.3.3.2 Activex Video Capture

Pada bagian ini akan menampilkan gambar lingkungan yang ada di depan *mobile robot*. Gambar 4.10 meunjukan lingkungan di depan *mobile robot*.



Gambar 4.10 *Gambar Lingkungan di Depan Robot*

4.3.3.3 Setting Panel

Bagian ini untuk digunakan untuk mengatur pencahayaan pada bagian *activex video capture.*



Gambar 4.11 *Setting Panel*

Pada bagian ini juga terdapat pemilihan *video input*, tombol "Preview" untuk mulai menampilkan gambar dan tombol "Exit" untuk keluar dari program.