

**PERANCANGAN ALAT FERMENTASI KAKAO OTOMATIS
BERBASIS MIKROKONTROLER ARDUINO UNO**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

**Diajukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik**



Disusun oleh:

ANGGARA TRUNA NEGARA

NIM. 0810633028

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2015

LEMBAR PERSETUJUAN
PERANCANGAN ALAT FERMENTASI KAKAO OTOMATIS BERBASIS
MIKROKONTROLER ARDUINO UNO

SKRIPSI
KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh :
ANGGARA TRUNA NEGARA
NIM. 0810633028

Telah diperiksa dan disetujui oleh :

Pembimbing I

Pembimbing II

Ir. Retnowati, MT
NIP. 19511224 198203 2 001

Rahmadwati, ST., MT., Ph.D
NIP. 19771102 200604 2 003

LEMBAR PENGESAHAN
PERANCANGAN ALAT FERMENTASI KAKAO OTOMATIS BERBASIS
MIKROKONTROLER ARDUINO UNO

SKRIPSI
JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh:
ANGGARA TRUNA NEGARA
NIM. 0810633028

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 13 Januari 2015

DOSEN PENGUJI :

Ir. Purwanto, MT.
NIP. 19540424 1986 1 001

Dr. Ir. Erni Yudaningsy, MT.
NIP. 19650913 199002 2 001

Goegoes Dwi Nusantoro, ST., MT.
NIP. 19711013 200604 1 001

Mengetahui,
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D.
NIP. 19741203 200012 1 001

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur kehadirat Allah SWT yang telah melimpahkan rahmad dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “PERANCANGAN ALAT FERMENTASI KAKAO OTOMATIS BERBASIS MIKROKONTROLER ARDUINO UNO”. Skripsi ini disusun sebagai salah satu syarat mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.

Pada kesempatan ini penulis mengucapkan terima kasih kepada Ibu Ir. Retnowati, MT dan Ibu Rahmadwati, ST., MT., Ph.D. sebagai dosen pembimbing skripsi yang telah memberikan bimbingan, saran dan motivasi sehingga skripsi ini dapat terselesaikan. Tidak lupa, ucapan terima kasih penulis ucapkan kepada:

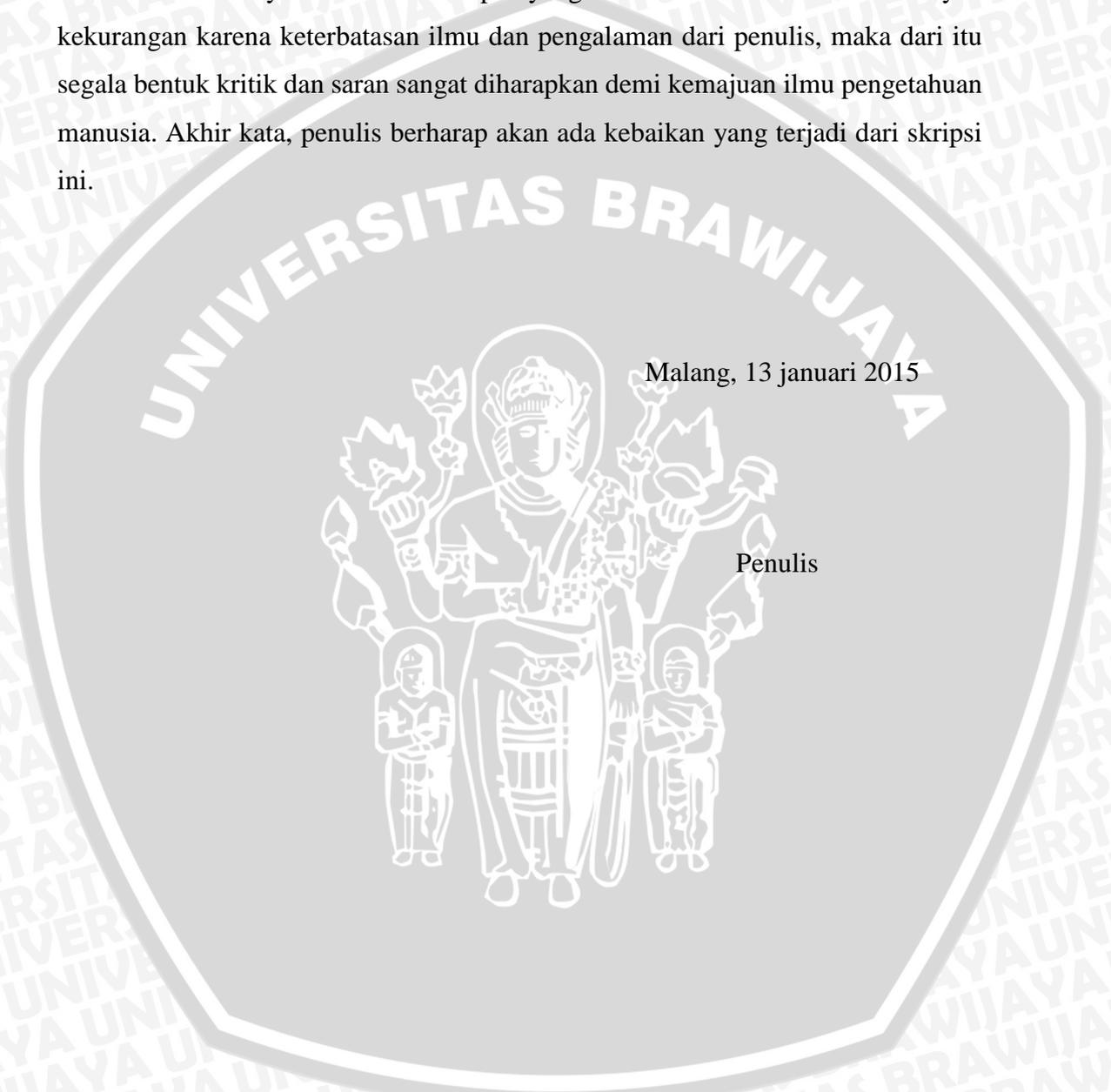
1. Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
2. Bapak Ir. Purwanto, MT selaku Ketua Kelompok Dosen dan Keahlian Teknik Kontrol.
3. Bapak Adharul Muttaqin, ST., MT selaku dosen pembimbing akademik.
4. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung telah membantu dalam penyelesaian skripsi ini.
5. Bapak Firul sebagai penanggung jawab di Pusat Penelitian Kopi dan Kakao Indonesia yang telah memberi izin untuk melakukan penelitian.
6. Mas Hendy sebagai peneliti di Pusat Penelitian Kopi dan Kakao Indonesia yang telah memberi saran dan informasi mengenai fermentasi kakao.
7. Mas Eko sebagai bagian pemasaran di Pusat Penelitian Kopi dan Kakao Indonesia yang telah memberi kemudahan dalam pengadaan buah kakao.
8. Keluarga. Ayah, Ibu, dan adek Moko yang telah memberi dukungan serta doa selama ini.
9. Teman-teman intra dan ekstra kampus yang telah banyak memberikan warna melalui gesekan-gesekan yang ada, terima kasih pengalamannya.
10. Mas Agung Adiyatma selaku orang yang telah banyak sekali membantu memudahkan dalam pengerjaan skripsi ini.

11. Teman kost lama KK21 dan kost baru yang banyak sekali memberi bantuan baik secara materi atau non materi selama proses pengerjaan skripsi.
12. Fandy, Frandicahya, Faiz, Maulana, Okta, Ihsan, Agus, Suhendra, Cahya, Ferdy, Fajar, Yudistira dll yang telah banyak membantu.

Penulis menyadari bahwa skripsi yang telah tersusun ini masih banyak kekurangan karena keterbatasan ilmu dan pengalaman dari penulis, maka dari itu segala bentuk kritik dan saran sangat diharapkan demi kemajuan ilmu pengetahuan manusia. Akhir kata, penulis berharap akan ada kebaikan yang terjadi dari skripsi ini.

Malang, 13 januari 2015

Penulis



DAFTAR ISI

KATA PENGANTAR..... i

DAFTAR ISI..... iii

DAFTAR TABEL vi

DAFTAR GAMBAR vii

ABSTRAK ix

BAB I. PENDAHULUAN

1.1 Latar Belakang 1

1.2 Rumusan Masalah..... 2

1.3 Batasan Masalah..... 2

1.4 Tujuan 3

1.5 Sistematika Penulisan 3

BAB II. TINJAUAN PUSTAKA

2.1 Teori Dasar Fermentasi Kakao..... 4

 2.1.1 Syarat Fermentasi Kakao 4

 2.1.2 Faktor-faktor yang mempengaruhi Fermentasi Kakao 5

2.2 Mikrokontroler Arduino Uno..... 5

 2.2.1 Catu Daya..... 7

 2.2.2 Memori..... 7

 2.2.3 *Input dan Output* 8

 2.2.4 Komunikasi..... 8

 2.2.5 *Programming* 9

 2.2.6 Perangkat Lunak (Arduino IDE)..... 9

 2.2.7 *Otomatis Software Reset* 9

2.3 Sensor Suhu LM-35..... 9

2.4 LCD..... 11

2.5 Motor DC 11

2.6 Sensor Rotary Encoder 12

2.7 PWM (*Pulse Width Modulation*)..... 14

2.8 Kontroler PID (Proporsional Integral Diferensial)..... 15

 2.8.1 Kontroler Proporsional..... 15



2.8.2 Kontroler Integral.....	16
2.8.3 Kontroler Diferensial	17
2.8.4 Kontroler Proporsional Integral Diferensial (PID)	17
2.8.5 Metode Tuning PID	19
2.9 Power MOSFET Motor Driver	22
BAB III. METODOLOGI PENELITIAN	
3.1 Studi Literatur	23
3.2 Perancangan Alat	24
3.2.1 Perancangan Perangkat Keras	24
3.2.2 Perancangan Perangkat Lunak	24
3.3 Pengujian Alat	24
3.4 Pengambilan Kesimpulan.....	25
BAB IV. PERANCANGAN DAN PEMBUATAN ALAT	
4.1 Tinjauan Umum.....	26
4.1.1 Blok Diagram Alat	27
4.1.2 Prinsip Kerja Alat	27
4.2 Perancangan Perangkat Keras	28
4.2.1 Perancangan Mekanik Kotak Fermentasi Kakao	28
4.2.2 Perancangan Catu Daya Sistem.....	28
4.2.3 Perancangan <i>Driver</i> Motor	29
4.2.4 Rangkaian LCD 16x2.....	30
4.2.5 Rangkaian Mikrokontroler Arduino Uno.....	31
4.2.6 Rangkaian Sensor Suhu LM35	31
4.2.7 Rangkaian <i>Push Button</i>	32
4.2.8 Rangkaian Komparator Sensor <i>Rotary Encoder</i>	32
4.3 Perancangan Perangkat Lunak.....	33
BAB V. PENGUJIAN DAN ANALISIS	
5.1 Pengujian <i>Driver</i>	35
5.2 Pengujian Motor DC	38
5.3 Pengujian Suhu.....	43
5.4 Pengujian Keseluruhan.....	45
BAB VI. KESIMPULAN DAN SARAN	

6.1. Kesimpulan 50

6.2. Saran 50

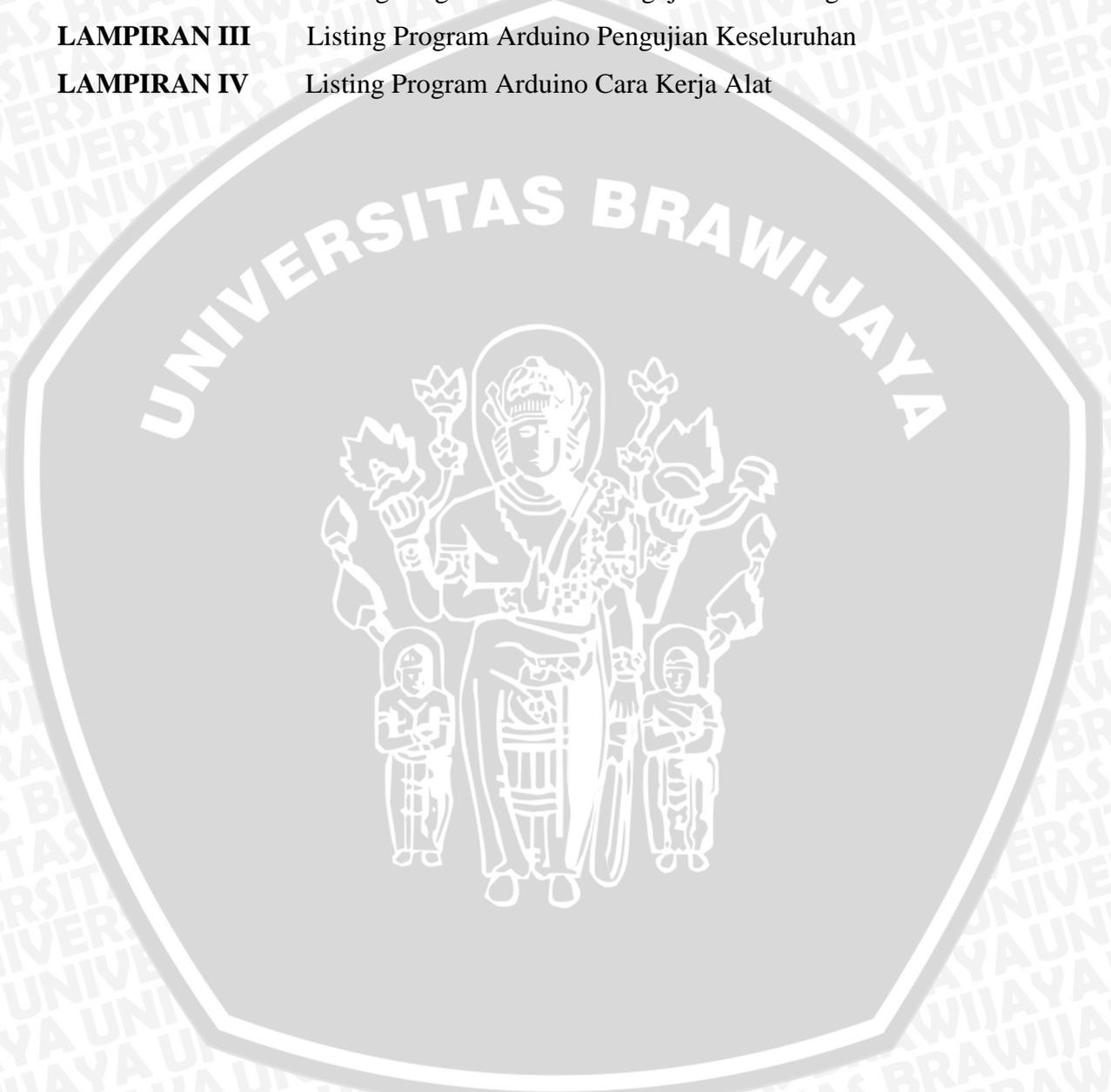
DAFTAR PUSTAKA 51

LAMPIRAN I Foto Alat

LAMPIRAN II Listing Program Arduino Pengujian RPM dengan PWM

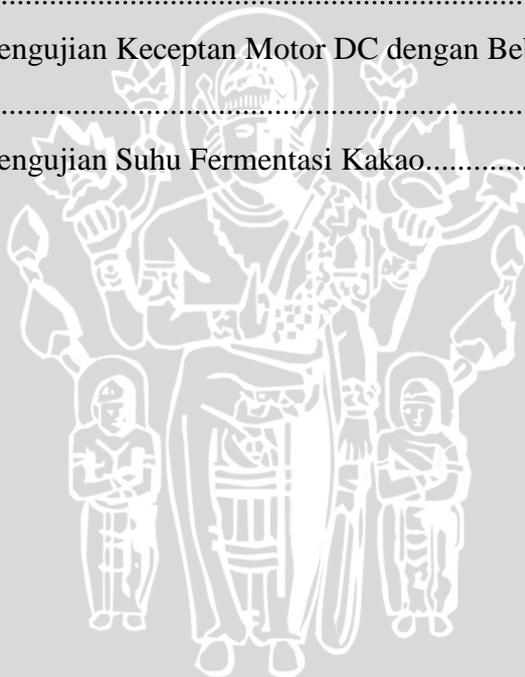
LAMPIRAN III Listing Program Arduino Pengujian Keseluruhan

LAMPIRAN IV Listing Program Arduino Cara Kerja Alat



DAFTAR TABEL

Tabel 2.1	Deskripsi Arduino Uno	6
Tabel 2.2	Aturan Penalaran Ziegler-Nichols Berdasarkan Respons Unit Step Dari Plan	20
Tabel 2.3	Aturan Dasar Ziegler-Nichols Berdasarkan <i>Critical GainKcr</i> dan <i>Critical PeriodPcr</i>	21
Tabel 5.1	Hasil Pengujian Driver Tanpa Moto..	36
Tabel 5.2	Hasil Pengujian Driver Menggunakan Motor	37
Tabel 5.3	Hasil Pengujian Kecepatan Motor DC Tanpa Beban Biji Kakao ..	39
Tabel 5.4	Hasil Pengujian Kecepatan Motor DC dengan Beban Biji Kakao 10 kg	40
Tabel 5.5	Hasil Pengujian Kecepatan Motor DC dengan Beban Biji Kakao 15 kg	41
Tabel 5.6	Hasil Pengujian Suhu Fermentasi Kakao.....	44



DAFTAR GAMBAR

Gambar 2.1	Board Arduino Uno	6
Gambar 2.2	Skema LM	10
Gambar 2.3	LM35 dengan <i>decouple</i> resistor dan LM35 dengan RC damper.....	10
Gambar 2.4	Bentuk LCD 16*2.....	11
Gambar 2.5	Skematik Motor DC Sederhana.....	12
Gambar 2.6	Skematik Rangkaian Driver Sederhana.....	12
Gambar 2.7	Rotari encoder absolut.....	13
Gambar 2.8	Rotari Encoder Relatif.....	13
Gambar 2.9	Rangkaian Tipikal Penghasil Pulsa pada <i>Rotary Encoder</i>	14
Gambar 2.10	Sinyal PWM secara umum	14
Gambar 2.11	Diagram Blok Kontroler Proporsional	15
Gambar 2.12	Diagram Blok Kontroler Integral	16
Gambar 2.13	Diagram Blok Kontroler Diferensial	17
Gambar 2.14	Diagram Blok Kontroler PID	18
Gambar 2.15	Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan	18
Gambar 2.16	Kurva Respon Unit Step yang Menunjukkan 25% <i>Maximum Overshoot</i>	19
Gambar 2.17	Respon Plan Terhadap Masukan Berupa <i>Unit Step</i>	19
Gambar 2.18	Respon Plan Berbentuk S	20
Gambar 2.19	Sistem Loop Tertutup dengan Kontroler Proporsional.....	21
Gambar 2.20	Osilasi Berkesinambungan dengan Periode <i>Pcr</i>	21
Gambar 2.21	<i>Power Mosfet Motor Driver</i>	22
Gambar 4.1	Diagram Blok Sistem	27
Gambar 4.2	Perancangan Mekanik Kotak Fermentasi Kakao	28
Gambar 4.3	Rangkaian Catu Daya 12 Volt.....	29
Gambar 4.5	Rangkaian Catu Daya 32 Volt.....	29
Gambar 4.6	Rangkaian <i>Driver</i> Motor DC.....	30
Gambar 4.7	Rangkaian Perencanaan LCD 16x2.....	30
Gambar 4.8	Rangkaian Perencanaan Mikrokontroler Arduino Uno	31

Gambar 4.9	Rangkaian Perencanaan Sensor Suhu LM35.....	32
Gambar 4.10	Rangkaian Perencanaan <i>Push Button</i>	32
Gambar 4.11	Rangkaian Perencanaan Komparator Sensor <i>Rotary Encoder</i> ..	33
Gambar 4.12	<i>Flowchart</i> Alat Fermentasi Kakao	33
Gambar 4.13	Program Arduino IDE 1.5.8	34
Gambar 5.1	Grafik Hubungan PWM dengan Tegangan Keluaran Driver Tanpa Motor.....	36
Gambar 5.2	Grafik Hubungan PWM dengan Tegangan Keluaran Driver Menggunakan Motor.....	37
Gambar 5.3	Grafik Hubungan PWM dengan Kecepatan Motor Tanpa Beban Biji Kakao	40
Gambar 5.4	Grafik Hubungan PWM dengan Kecepatan Motor Menggunakan Beban Biji Kakao 10 kg.....	41
Gambar 5.5	Grafik Hubungan PWM dengan Kecepatan Motor Menggunakan Beban Biji Kakao 15 kg.....	42
Gambar 5.6	Tampilan LCD saat 4 jam.....	43
Gambar 5.7	Tampilan LCD saat 48 jam.....	43
Gambar 5.8	Tampilan LCD saat 120 jam.....	43
Gambar 5.9	Grafik Hubungan Suhu Fermentasi terhadap Waktu.....	45
Gambar 5.10	Grafik Kecepatan Motor Terhadap Waktu dengan $K_p=1$	46
Gambar 5.11	Hasil Pengujian Kecepatan Motor DC dengan PID	47
Gambar 5.12	Gambar Biji Kakao Sebelum Fermentasi	48
Gambar 5.13	Gambar Biji Kakao Sebelum Fermentasi pada Alat.....	48
Gambar 5.14	Gambar Biji Kakao Sesudah Fermentasi.....	49
Gambar 5.15	Gambar Biji Kakao Sesudah Fermentasi pada Alat	49

ABSTRAK

Perancangan alat fermentasi kakao otomatis diharapkan dapat mempermudah pekerjaan petani kakao pada saat proses pembalikan dan pengadukan fermentasi kakao berdasarkan suhu yang tepat. Pengendalian kecepatan motor DC yang disesuaikan dengan suhu fermentasi kakao yang tepat pada proses pembalikan atau pengadukan dapat menyempurnakan alat fermentasi kakao konvensional yang masih diputar secara manual.

Digunakan Kontroler PID untuk mengurangi kesalahan, sehingga putaran motor dapat sesuai dengan kecepatan yang diinginkan. Pada skripsi ini digunakan metode kedua dari teori Ziegler-Nichols. Dalam pembuatannya digunakan Arduino Uno, sensor suhu LM35, sensor *rotary encoder*, motor DC.

Dari hasil pengujian kontroler PID pada alat dengan menggunakan metode teori Ziegler-Nichols. Didapatkan parameter PID dengan nilai $K_p=0,6$, $K_i=0,324$, dan $K_d=0,2775$ yang menunjukkan bahwa respons sistem untuk pengendalian kecepatan putaran pada alat fermentasi kakao memiliki *error steady state* sebesar 4,3% serta mengalami *overshoot* kurang dari 70 ms.

BAB I PENDAHULUAN

1.1 Latar Belakang

Kakao merupakan salah satu komoditi andalan dalam sektor perkebunan, tercatat bahwa Indonesia merupakan salah satu produsen biji kakao terbesar di dunia setelah Pantai Gading dan Ghana. Walaupun Indonesia secara volume dan kuantitas termasuk dalam produsen utama, tetapi dari segi kualitas, biji kakao Indonesia masih rendah sehingga dalam pemasaran sering mengalami permasalahan seperti adanya *Automatic Detention* yang diterapkan salah satu negara konsumen.

Penanganan pascapanen yang kurang sempurna dengan kata lain penanganan pascapanen tanpa fermentasi yang menyebabkan kualitas biji kakao di Indonesia masih rendah. Fermentasi merupakan suatu proses produksi suatu produk dengan mikroba sebagai organisme pemroses. Salah satu tahapan penting dalam penanganan pascapanen kakao adalah proses fermentasi. Fermentasi biji Kakao bertujuan untuk memudahkan pelepasan zat lendir dari permukaan kulit biji dan membentuk cita rasa khas cokelat serta mengurangi rasa pahit dan sepat yang ada dalam biji kakao sehingga menghasilkan biji dengan mutu dan aroma yang baik, serta warna coklat cerah dan bersih.

Dalam fermentasi kakao dibutuhkan beberapa faktor yang diperhatikan agar menghasilkan biji kakao terbaik yaitu suhu dan waktu yang tepat untuk pengadukan dan pemindahan. Alat fermentasi kakao otomatis ini bekerja berdasarkan prinsip dasar alat fermentasi kakao konvensional sebelumnya contohnya pada alat fermentasi kakao konvensional dilakukan dengan menggunakan 2 kotak kayu yang diberi lubang-lubang kecil agar oksigen masuk. Untuk kotak skala kecil diperlukan berat maksimal 40 kg dan dalam proses fermentasi suhu dijaga sekitar 45-49⁰ C lalu dilakukan pemindahan dan pengadukan biji di kotak berikutnya setelah 48 jam (2 hari) atau saat suhu kakao 50⁰ C , proses fermentasi ini berlangsung selama 4-5 hari dengan 1 kali pemindahan dan pengadukan. Sedangkan alat fermentasi kakao otomatis yang akan dibuat ini mengatur kecepatan motor yang disesuaikan dengan suhu pada proses pembalikan fermentasi kakao yang difungsikan sebagai

pengaduk seperti halnya alat fermentasi konvensional sehingga menghasilkan suhu yang merata.

Diharapkan alat ini dapat mempermudah dalam menentukan proses pengadukan dan pembalikan tepat waktu dan merata sesuai dengan suhu yang tepat. Pada penelitian ini alat fermentasi kakao otomatis akan menggunakan mikrokontroler Arduino Uno sebagai kontroler otomatisnya dikarenakan memiliki fitur dan jumlah I/O yang sesuai dengan kebutuhan sistem.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas, maka didapatkan perumusan masalah sebagai berikut:

1. Bagaimana merancang dan membuat mikrokontroler Arduino Uno pada alat fermentasi kakao otomatis?
2. Bagaimana merancang dan membuat sistem pengaturan kecepatan motor DC pada alat fermentasi kakao otomatis dengan menggunakan kontroler PID?

1.3 Batasan Masalah

Karena begitu luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah dalam skripsi ini adalah :

1. Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM (*Pulse Width Modulation*) dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol *reset*.
2. Sensor suhu menggunakan tipe LM35 yang mempunyai range pengukuran suhu 25°- 50° C dengan resolusi 0,1°C.
3. Penentuan posisi dan jumlah sensor suhu disesuaikan dengan pola aliran udara.
4. Menggunakan sensor *rotary encoder*.
5. Kapasitas maksimal alat fermentasi kakao otomatis adalah 40 kg.

6. Posisi motor DC digunakan sebagai pengaduk atau pembalik diletakkan pada kotak kecil yang tersambung dengan kotak fermentais kakao.

1.4 Tujuan

Tujuan yang ingin dicapai dari skripsi ini adalah merancang dan membuat model percobaan (model eksperimental) alat fermentasi kakao otomatis berbasis mikrokontroler arduino uno sehingga dapat mengatur kecepatan motor DC dengan kontroler PID yang disesuaikan dengan suhu fermentasi kakao pada proses pembalikan atau pengadukan.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan serta sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi Penelitian

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Membahas perancangan sistem pengendalian proses alat fermentasi kakao dengan mikrokontroler arduino uno. Setelah itu, bagaimana menerapkannya dalam sistem secara keseluruhan.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem terhadap alat yang telah direalisasikan.

BAB VI Kesimpulan dan Saran

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar Fermentasi Kakao

Proses fermentasi berlangsung secara alamiah selama beberapa hari. Tahapan ini sangat penting dilalui untuk mempersiapkan biji kakao basah menjadi biji kakao kering bermutu tinggi dan layak dikonsumsi. Fermentasi biji kakao akan menumbuhkan citarasa, aroma dan warna, karena selama fermentasi terjadi perubahan fisik, kimiawi dan biologi di dalam biji kakao. Di dalam biji kakao akan terjadi penguraian senyawa polifenol, protein dan gula oleh enzim. Penguraian senyawa-senyawa tersebut akan menghasilkan talon aroma, perbaikan rasa dan perubahan warna.

2.1.1 Syarat Fermentasi Kakao

Beberapa syarat dalam fermentasi kakao adalah :

a. Lokasi

Lokasi Unit Fermentasi Biji Kakao (UFBK) harus memenuhi persyaratan sebagai berikut:

1. Bebas pencemaran dan tidak di daerah yang drainasenya buruk (tidak ada genangan air dan saluran tersumbat).
2. Dekat dengan sentra produksi sehingga menghemat biaya transportasi.
3. Dekat dengan sumber air.
4. Tersedia lokasi untuk pembuangan limbah sehingga tidak mengganggu lingkungan.
5. Sebaiknya tidak dekat dengan perumahan.

b. Bangunan

1. Bangunan harus dibuat berdasarkan perencanaan yang memenuhi persyaratan sesuai spesifikasi teknis dan kesehatan pada Pedoman Teknis.
2. Bangunan mudah dibersihkan, mudah sanitasi dan mudah dilakukan perawatan.

2.1.2 Faktor – Faktor yang Mempengaruhi Fermentasi Kakao

a. Berat kakao

Berat biji kakao yang akan difermentasi maksimal 40 kg apabila menggunakan ukuran kotak 40 x 40 x 50 cm. Hal ini terkait dengan kemampuan untuk menghasilkan panas yang cukup sehingga proses fermentasi dapat berjalan dengan baik.

b. Pembalikan

Setelah 48 jam atau + 2 hari proses fermentasi dilakukan pengadukan atau pembalikan.

c. Lama fermentasi

Untuk lama fermentasi optimal adalah 4-5 hari (4 hari bila udara lembab dan 5 hari bila udara terang). Proses fermentasi yang terlalu singkat (kurang dari 3 hari) menghasilkan biji yang berwarna ungu agak keabu-abuan (*Slaty*) dan bertekstur pejal. Sedangkan proses fermentasi yang terlalu lama (lebih dari 5 hari) menghasilkan biji rapuh dan berbau kurang sedap atau berjamur.

d. Sarana

Sarana fermentasi yang ideal adalah dengan menggunakan kotak dari kayu yang diberi lubang-lubang. Untuk skala kecil (40 kg biji kakao basah) diperlukan kotak dengan ukuran panjang dan lebar masing-masing 40 cm dan tinggi 50 cm. Untuk skala besar (700kg biji kakao basah) diperlukan kotak dengan ukuran lebar 100-120 cm, panjang 150-165 cm dan tingginya 50 cm.

e. Tinggi minimal

Tinggi tumpukan biji kakao minimal 40 cm agar dapat tercapai suhu fermentasi 45-49°C.

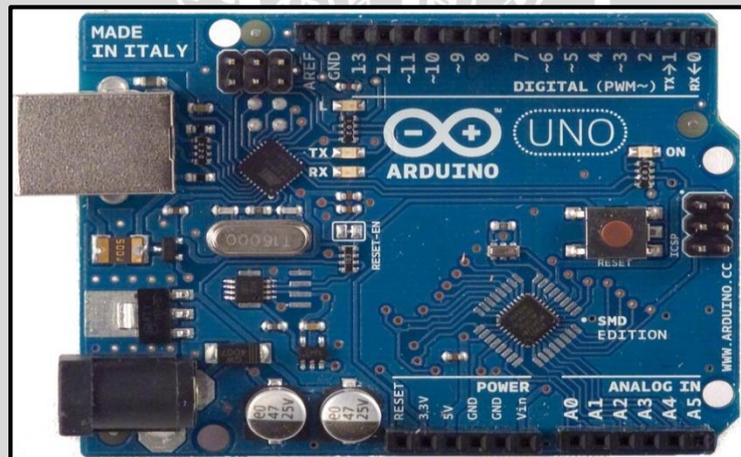
2.2 Mikrokontroler Arduino Uno

Mikrokontroler Arduino Uno adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital *input / output* pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 input *analog*, 16 MHz *osilator* kristal, koneksi USB, jack listrik tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan

menggunakan kabel USB atau sumber tegangan bisa didapat dari adaptor AC ke DC atau menggunakan baterai untuk menjalankannya. *Board* Arduino Uno memiliki fitur-fitur baru sebagai berikut :

- Pin out 1.0: ditambah SDA dan SCL pin yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin RESET, IO REF yang memungkinkan *shield-shield* untuk menyesuaikan tegangan yang disediakan dari board. Untuk ke depannya, *shield* akan dijadikan kompatibel atau cocok dengan *board* yang menggunakan AVR yang beroperasi dengan tegangan 5 volt dan dengan Arduino Due yang beroperasi dengan tegangan 3,3 volt. Yang ke dua merupakan sebuah pin yang tak terhubung, yang disediakan untuk tujuan kedepannya.
- Sirkit RESET yang lebih kuat
- Atmega 16U2 menggantikan 8U2

Board arduino uno dapat dilihat dalam Gambar 2.1.



Gambar 2.1 *Board* Arduino Uno

Sumber : *Adruino.cc*

Deskripsi Arduino Uno secara umum dapat dilihat dalam Tabel 2.1.

Tabel 2.1 Deskripsi Arduino Uno

<i>Microcontroller</i>	Atmega328
<i>Operasi Voltage</i>	5V
<i>Input Voltage</i>	7-12 V (Rekomendasi)
<i>Input Voltage</i>	6-20 V (<i>limits</i>)
I/O	14 pin (6 pin untuk PWM)

Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

2.2.1 Catu Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Eksternal (non-USB) tegangan dapat diperoleh dari sebuah adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan sebuah *center-positive plug* yang panjangnya 2.1mm ke *power jack* dari *board*. Kabel *lead* dari baterai dapat dimasukkan ke dalam *header pin Ground* (Gnd) dan Vin dari konektor POWER.

Board Arduino Uno dapat beroperasi pada sebuah suplai eksternal dari 6 sampai 20 volt. Jika disuplai kurang dari 7 volt, contohnya ketika pin 5V mendapat suplai kurang dari 5 volt dapat berakibat *board* arduino menjadi tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 - 12 volt.

10 Pin catu daya adalah sebagai berikut:

- VIN. Tegangan input ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator *on board*, atau diberikan oleh USB.
- 3,3 volt pasokan yang dihasilkan oleh regulator *on-board*. Menarik arus maksimum adalah 50 mA.
- GND.

2.2.2 Memori

ATmega328 memiliki 32 KB dengan 0,5 KB digunakan untuk *loading file*. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM.

2.2.3 Input & Output

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Fungsi-fungsi tersebut beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal dari 20-50 K_Ω. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat *attach Interrupt ()* fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit *output* PWM dengan *analog Write ()* fungsi.
- SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu *off*.

Arduino Uno memiliki 6 *input analog*, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara *default* sistem mengukur dari tanah sampai 5 volt.

- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk *input analog*. Digunakan dengan *analog Reference ()*.
- Reset.

Lihat juga pemetaan antara pin Arduino dan ATmega328 *port*. Pemetaan untuk ATmega8, 168 dan 328 adalah identik.

2.2.4 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan

UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file.Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi *interface* pada sistem.

2.2.5 Programming

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. ATmega328 pada Arduino Uno memiliki *bootloader* yang memungkinkan anda untuk meng-*upload* program baru tanpa menggunakan programmer *hardware* eksternal. Cara berkomunikasi menggunakan protokol dari bahasa C. Sistem ini dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat *firmware* baru. Atau anda dapat menggunakan *header* ISP dengan programmer eksternal.

2.2.6 Perangkat Lunak (Arduino IDE)

Lingkungan *open-source* Arduino memudahkan untuk menulis kode dan meng-*upload* ke *board* Arduino. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, avr-gcc, dan perangkat lunak sumber terbuka lainnya.

2.2.7 Otomatis Software Reset

Tombol *reset* Arduino Uno dirancang untuk menjalankan program yang tersimpan didalam mikrokontroler dari awal. Tombol *reset* terhubung ke ATmega328 melalui kapasitor 100nf. Setelah tombol *reset* ditekan cukup lama untuk me-*reset chip*, *software IDE* Arduino dapat juga berfungsi untuk meng-*upload* program dengan hanya menekan tombol *upload* di *software IDE* Arduino.

2.3 Sensor Suhu LM-35

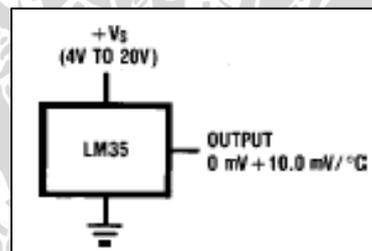
LM35 merupakan sensor suhu terintegrasi yang mempunyai tegangan keluaran yang linier. LM35 tidak membutuhkan kalibrasi eksternal yang

menyediakan akurasi $\pm 1/4^{\circ}\text{C}$ pada temperatur ruangan dan $\pm 3/4^{\circ}\text{C}$ pada kisaran -55 to $+150^{\circ}\text{C}$. LM35 dimaksudkan untuk beroperasi pada -55°C hingga $+150^{\circ}\text{C}$. Sensor LM35 umumnya akan naik sebesar 10mV setiap kenaikan 1°C (300mV pada 30°C). LM35 mempunyai impedansi keluaran yang rendah, keluaran yang linier dan kalibrasi yang tepat sehingga mudah untuk di hubungkan dengan rangkaian lain.

Adapun fitur yang ada pada LM35 adalah sebagai berikut :

- Kalibrasi dalam derajat celcius
- Faktor skala linier adalah $10\text{ mV}/^{\circ}\text{C}$
- Jangkauan suhu -55°C hingga $+150^{\circ}\text{C}$
- Tegangan operasi dari 4 V sampai 30 V
- Ketidakiirran hanya $\pm 1/4^{\circ}\text{C}$
- Impedansi keluaran kecil sebesar $0,1\ \Omega$ untuk arus beban 1mA

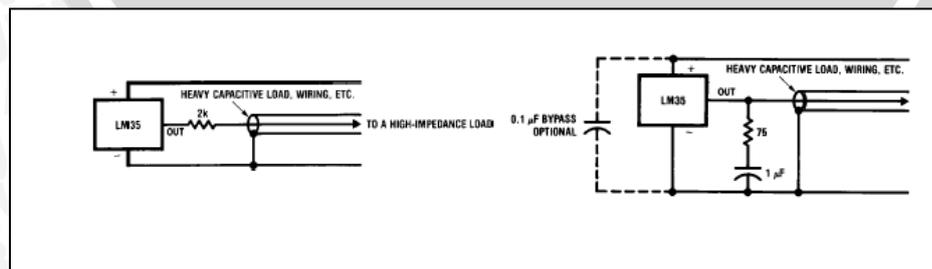
Dalam Gambar 2.2. ditunjukkan skema sensor suhu LM35.



Gambar 2.2 Skema LM35

Sumber: Datasheet LM35

LM35 mempunyai batas kemampuan untuk men-*drive* beban kapasitif. LM35 dapat men-*drive* 50 pF tanpa penanganan khusus. Jika beban bertambah maka untuk mengantisipasinya dengan mengisolasi atau men-*decouple* beban dengan resistor atau dengan menambah kapasitansi yang disusun seri dengan resistor antara *output* dan *ground* seperti ditunjukkan dalam Gambar 2.3.



Gambar 2.3 LM35 dengan *decouple* resistor dan LM35 dengan RC damper

Sumber : Datasheet LM35

2.4 LCD

Liquid Crystal Display (LCD) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator ataupun layar komputer. Tipe LCD dot matrik dengan jumlah karakter 16*2. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. Bentuk LCD 16*2 dapat dilihat dalam Gambar 2.4.

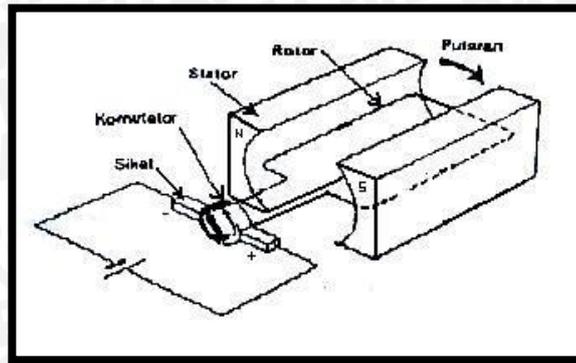


Gambar 2.4 Bentuk LCD 16*2

Sumber : engineersgarage.com

2.5 Motor DC

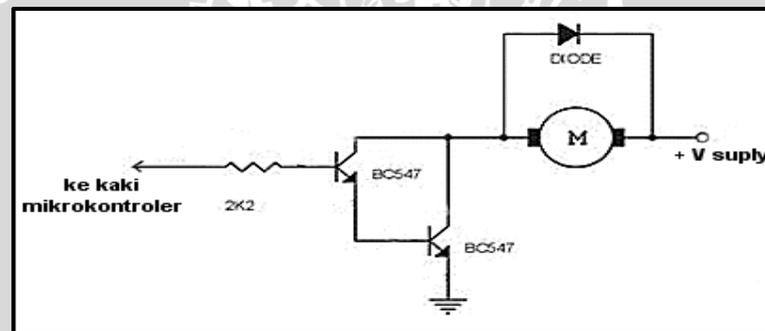
Motor listrik merupakan perangkat *electromagnetic* yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk memutar *impeller* pompa, *fan*, *blower* menggerakkan kompresor dan mengangkat bahan. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik *phase* tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Skematik motor DC sederhana dapat dilihat dalam Gambar 2.5.



Gambar 2.5 Skematik Motor DC Sederhana

Sumber : Kismet F [1994:98]

Satu tegangan DC dari baterai menuju lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar 2.5 disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet. Skematik rangkaian *driver* dapat dilihat dalam Gambar 2.6.



Gambar 2.6 Skematik Rangkaian *Driver* Sederhana

Sumber : elektronika-dasar.web.id

2.6 Sensor Rotary Encoder

Rotary encoder atau yang dikenal dengan *shaft encoder* adalah perangkat elektronika yang digunakan untuk mengkonversi sudut dari perputaran poros atau roda ke dalam kode digital. Komponen ini biasa digunakan dalam bidang robotika, perangkat computer dan perangkat elektronik lainnya. *Rotary encoder* dibedakan menjadi dua jenis yakni *rotary encoder* absolut dan relatif.

Rotari encoder absolut merupakan jenis sensor yang mampu menghasilkan kode digital yang unik untuk masing-masing sudut poros. Pada prinsipnya sensor ini tersusun atas plat baja dan kontak. Plat baja dipotong dan disusun dengan pola tertentu kemudian ditempelkan pada suatu poros. Plat baja dan kontak ini secara

prinsip kerja menyerupai saklar dalam kondisi ON apabila keduanya saling bersentuhan dan OFF apabila terpisah. Plat baja dan kontak diatur sedemikian rupa sehingga menghasilkan kondisi yang berbeda untuk tiap-tiap sudut poros. Bentuk fisik dari rotari encoder absolut ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Rotari Encoder Absolut

Sumber : Sigit, 2007

Rotari encoder relative tidak dapat mengukur posisi sudut poros melainkan hanya mengukur perubahan sudut poros terhadap posisi sudut sebelumnya. Rotary ini digunakan ketika metode rotary absolut tidak dapat digunakan. Secara prinsip system ini terdiri dari piringan yang dipasang pada poros dan sensor optic. System ini menggunakan metode saklar optic, missal photodiode, untuk menghasilkan pulsa listrik yang digunakan sebagai masukan bagi rangkaian control elektronika. Gambar 2.8 di bawah ini menunjukkan bentuk fisik *rotary encoder* relatif.

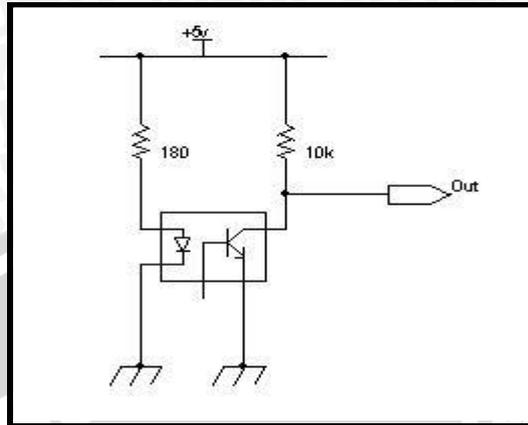


Gambar 2.8 Rotari Encoder Relatif

Sumber: Sigit, 2007

Salah satu kekurangan system relatif adalah tidak dapat menentukan arah putaran poros. Agar dapat mengetahui arah putaran poros, pada system harus ditambahkan dua buah sensor optic yang dipasang pada sudut berbeda. Pembacaan arah putaran dan jumlah putaran dilakukan dengan membaca dan membandingkan sinyal keluaran dari kedua buah sensor. Tipe *rotary encoder* ini dikenal dengan

quadrature encoder. Rangkaian tipikal penghasil pulsa pada *rotary encoder* ditunjukkan dalam Gambar 2.9.

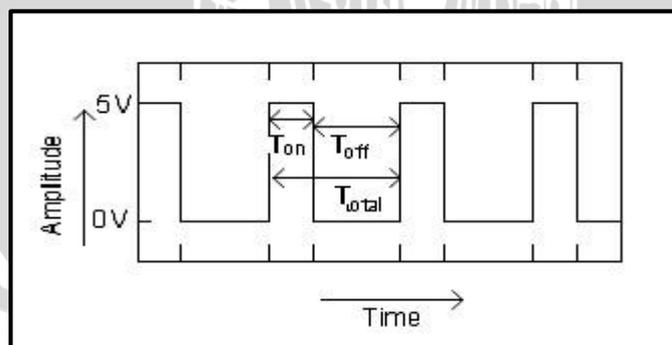


Gambar 2.9 Rangkaian Tipikal Penghasil Pulsa pada *Rotary Encoder*

Sumber: Sekilas *rotary encoder* (wordpress :2012)

2.7 PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Pada AVR ATmega16 *duty cycle* ditentukan oleh *Output Compare Match A* (OC1A). Sinyal PWM (*pulse width modulation*) secara umum dapat dilihat dalam Gambar 2.10.



Gambar 2.10 Sinyal PWM Secara Umum

Sumber: www.electronics-scheme.com

$$Duty\ Cycle = \frac{T_{on}}{T} \times 100\% \dots\dots\dots(1-1)$$

$$V_{DC} = Duty\ Cycle \times V_{cc} \dots\dots\dots(1-2)$$



Sedangkan frekuensinya dapat ditentukan dengan rumus sebagai berikut:

$$fOCn = \frac{fclk}{N \cdot 256} \dots\dots\dots(1-3)$$

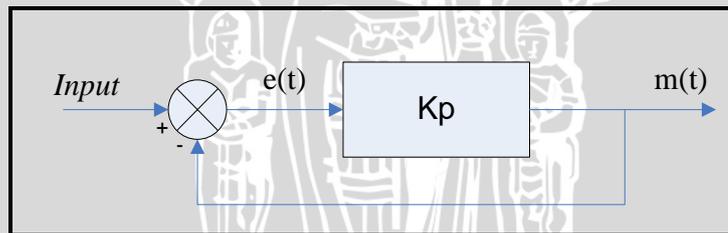
Timer/counter yang digunakan pada PWM ini yaitu *Timer/Counter0* (8 bit) dengan metode Fast PWM dan *Prescaler factor(N)* yaitu 256.

2.8 Kontroler PID (Proporsional Integral Diferensial)

2.8.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding atau proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Dapat dilihat dalam Gambar 2.11 menunjukkan diagram blok yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



Gambar 2.11 Diagram Blok Kontroler Proporsional

Sumber: Ogata, 1995: 157

Pada pengendali proporsional hubungan antara keluaran kontroler *m(t)* dan sinyal kesalahan *e(t)* adalah

$$m(t) = Kp e(t) \dots\dots\dots(2-1)$$

Sumber: Ogata, 1995: 157

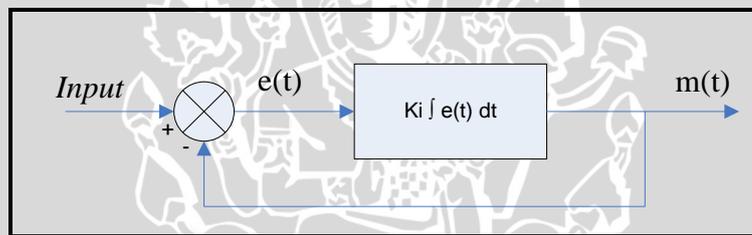
dengan *Kp* adalah penguatan proporsional. Keluaran *m(t)* hanya tergantung pada *Kp* dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan *Kp* akan menaikkan penguatan sistem sehingga dapat

digunakan untuk memperbesar kecepatan respons dan mengurangi kesalahan keadaan mantap.

2.8.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah *plant* tidak memiliki unsur integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantap nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Dapat dilihat dalam Gambar 2.12 yang menunjukkan diagram blok kontroler integral.



Gambar 2.12 Diagram Blok Kontroler Integral

Sumber: Ogata, 1995: 158

Nilai keluaran kontroler $m(t)$ sebanding dengan integral sinyal kesalahan $e(t)$, sehingga :

$$\frac{dm(t)}{dt} = Ki \cdot e(t) \dots\dots\dots(2-2)$$

$$m(t) = Ki \int_0^t e(t) dt \dots\dots\dots (2-3)$$

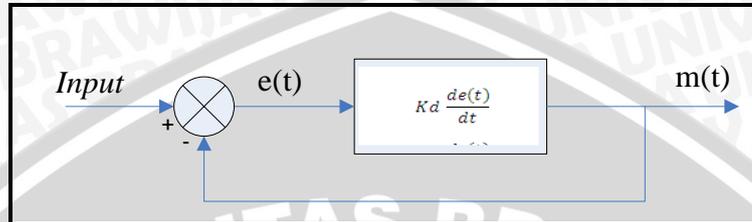
Sumber: Ogata, 1995: 157

dengan K_i adalah konstanta integral. Jika sinyal kesalahan $e(t)=0$, maka laju perubahan sinyal kendali integral $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi kontrol integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respons.



2.8.3 Kontroler Diferensial

Kontroler diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Dapat dilihat dalam Gambar 2.13 berikut yang menunjukkan diagram blok pada kontroler diferensial.



Gambar 2.13 Diagram Blok Kontroler Diferensial

Sumber: Ogata, 1995: 177

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$.

Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \dots \dots \dots (2-4)$$

Sumber: Ogata, 1995: 179

Kontroler diferensial akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respons awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler diferensial tidak bisa digunakan tanpa ada kontroler lain.

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu:

2.8.4 Kontroler Proporsional Integral Diferensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral diferensial (PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah



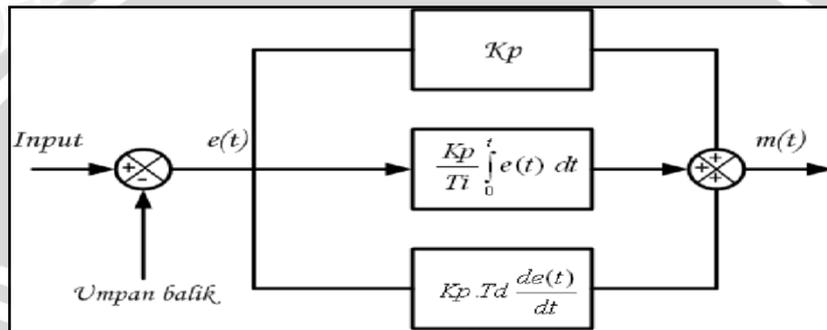
sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.14.

Aksi kontrolnya dinyatakan sebagai:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \dots \dots \dots (2-5)$$

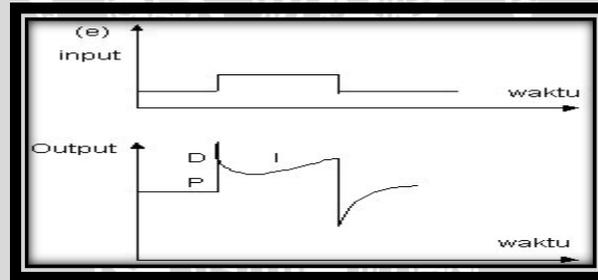
Sumber: Ogata, 1995: 183

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respons, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.



Gambar 2.14 Diagram Blok Kontroler PID

Sumber: Philips, 1996: 220



Gambar 2.15 Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan Kontroler PID

Sumber: Gunterus, 1994:8-11

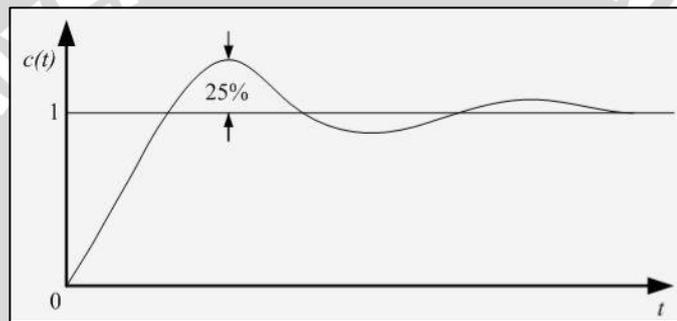
Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan diferensial. Dapat dilihat dalam Gambar 2.15 diatas yang menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P,I dan D. Penyetelan konstanta K_p , T_i dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pada respons sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.8.5 Metode Tuning PID

Metode Ziegler-Nichols.

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan. (Ogata, K., 1997)

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, seperti ditunjukkan dalam Gambar 2.16.

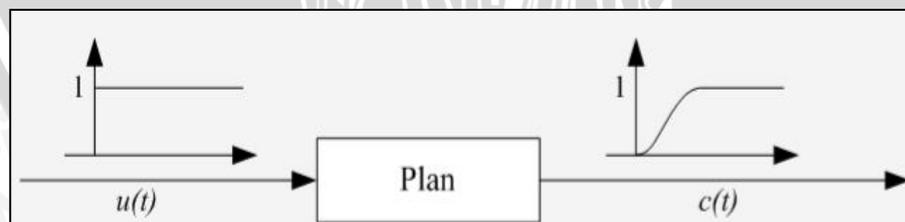


Gambar 2.16 Kurva Respon *Unit Step* yang Menunjukkan 25% *Maximum Overshoot*

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

a) Metode Pertama

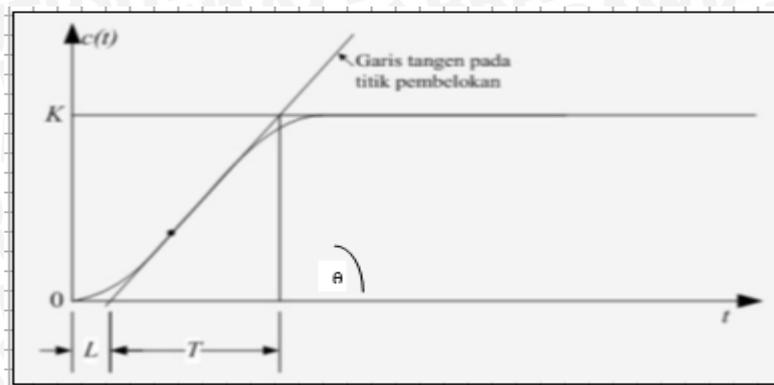
Metode pertama atau sering disebut metode kurva reaksi, respon dari *plan* dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.17



Gambar 2.17 Respons Plan Terhadap Masukan Berupa *Unit Step*

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Jika dalam *plan* tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respons unit *step* berbentuk seperti huruf S, seperti dalam Gambar 2.18. Jika respons tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



Gambar 2.18 Respons Plan berbentuk S

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.18 Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots(2-6)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.2 (Ogata, K., 1997)

Tabel 2.2 Aturan Penalaan Ziegler-Nichols Berdasarkan Respons Unit Step Dari Plan

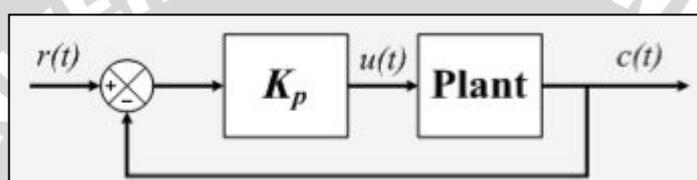
Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5 L$

Sumber: Ogata, K. 1997

b) Metode Kedua

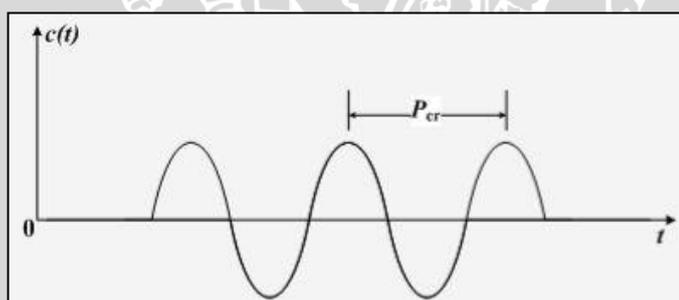
Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan

kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.19 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam gambar 2.20 Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p, T_i, T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.3. (Ogata, K., 1997)



Gambar 2.19 Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997



Gambar 2.20 Osilasi Berkesinambungan dengan periode P_{cr}

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

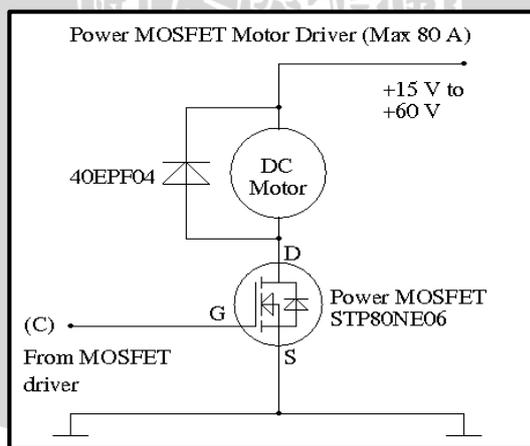
Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr}

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1,2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.9 Power MOSFET Motor Driver

Driver motor transistor mosfet ini membutuhkan rangkaian driver khusus. Alasan untuk ini adalah bahwa perpindahan tegangan gerbang transistor mosfet membutuhkan arus transien yang tinggi (2A) karena beban kapasitif relatif tinggi. Transistor mosfet dan kekuatan dioda harus memiliki *heatsink* yang relatif besar untuk motor arus tinggi. Ini adalah ide yang baik untuk menggunakan komponen listrik (yaitu transistor mosfet dan daya diode) pada kartu yang terpisah dengan *heatsink*. Komponen listrik harus terhubung ke pasokan listrik atau akumulator dan motor dengan kabel cukup tebal dan kabel lainnya. Kabel tembaga itu menggunakan 1,5 mm (1,77 mm²) timah berlapis antara kekuatan komponen dan konektor kabel. Ini cukup untuk motor yang menggunakan arus sampai 20 A rata-rata. Kabel juga harus memiliki luas minimal 2 mm². Untuk motor arus tinggi kabel yang sesuai harus lebih tebal. Seperti yang ditunjukkan dalam peringatan itu sangat penting pada pengaturan listrik pendek, terutama ketika terhubung ke akumulator lebih besar. Selalu menguji dengan *power supply* yang terbatas saat ini dan motor kecil pertama untuk melihat bahwa segala sesuatu bekerja. Hal ini juga dianjurkan untuk memiliki sekering 10 atau 15 A (lambat) terhubung ke *power supply* atau akumulator. Gambar *power MOSFET motor driver* ditunjukkan dalam Gambar 2.21 berikut:



Gambar 2.21 Power MOSFET Motor Driver

Sumber : spaennare.se

BAB III

METODOLOGI PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif yang diwujudkan dalam bentuk prototipe, yaitu perencanaan dan perealisasiannya agar dapat menampilkan unit kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat secara umum adalah sebagai berikut:

- Studi literatur
- Perancangan alat
- Realisasi pembuatan alat
- Pengujian alat dan analisa data
- Pengambilan kesimpulan

3.1 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem pengendalian Alat fermentasi kakao yang menggunakan mikrokontroler Arduino Uno, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu mikrokontroler Arduino Uno.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah :

- a. Studi tentang teori dasar proses fermentasi kakao
 - Faktor-faktor yang mempengaruhi fermentasi kakao.
 - Suhu fermentasi kakao.
 - Proses pengadukan dan pembalikan sesuai suhu fermentasi.
- b. Studi tentang rangkaian sensor
 - Karakteristik spesifikasi sensor suhu LM35 dan *rotary encoder*.
 - Penerapan sensor suhu dan *rotary encoder* sesuai sistem.
 - Karakteristik komponen elektronik yang digunakan diantaranya meliputi LCD 16*2, Mikrokontroler Arduino Uno, *Power Mosfet Motor driver*, transistor sebagai saklar dan motor DC.

c. Mikrokontroler ATmega328

- Karakteristik umum mikrokontroler Arduino Uno yang berbasis mikrokontroler ATmega328.
- Cara memprogram mikrokontroler Atmega328 sebagai pusat pengontrol kecepatan motor saat proses pembalikan dan pengadukan fermentasi kakao.

3.2 Perancangan Alat

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pada alat fermentasi kakao tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

1. penentuan spesifikasi alat.
2. pembuatan diagram blok sistem keseluruhan.
3. perancangan mekanik.
4. perancangan perangkat elektrik yang terdiri dari mikrokontroler dan rangkaian sensor.
5. Perancangan perangkat lunak menggunakan IDE arduino 1.5.8 beta.

3.2.1 Perancangan Perangkat Keras

- a. Pembuatan mekanik alat, *power supply*, *push button* dan LCD.
- b. Pembuatan minimum sistem dan driver motor.
- c. Pembuatan rangkaian sensor suhu dan rotary encoder sesuai sistem.

3.2.2 Perancangan Perangkat Lunak

Setelah mengetahui perangkat keras yang dirancang, maka dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Parameter yang diperoleh dari hasil perhitungan kemudian diterapkan kedalam mikrokontroler Aduino Uno dengan menggunakan bahasa C++.

3.3 Pengujian Alat

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem

telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut :

- Menguji pada tiap-tiap blok rangkaian.
- Kalibrasi pada sensor yang digunakan.
- Menggabungkan beberapa blok rangkaian menjadi sebuah sistem.
- Mengadakan pengujian sistem secara keseluruhan.
- Mengevaluasi hasil pengujian sistem secara keseluruhan.

3.4 Pengambilan Kesimpulan.

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

4.1 Tinjauan Umum

Bab ini menjelaskan mengenai spesifikasi alat, perancangan perangkat keras dari alat fermentasi kakao, yang meliputi diagram blok sistem, cara kerja sistem, *flowchart* atau diagram alir kerja sistem, dan gambar rancangan alat. Selain perancangan perangkat keras akan dijelaskan juga mengenai perancangan perangkat lunak.

Perancangan pengontrolan kecepatan motor yang disesuaikan suhu fermentasi dan beban dalam alat fermentasi kakao ini dirancang dengan spesifikasi sebagai berikut :

Spesifikasi *electric* terdiri dari :

1. Mengkondisikan motor DC saat suhu pembalikan dan pengadukan pada suhu 50°C.
2. Mengkondisikan motor DC dengan kecepatan putaran
3. Mengkondisikan motor DC dengan nilai PID
4. Menggunakan Mikrokontroler Arduino Uno sebagai pengontrol utama.
5. Menggunakan LCD (16 kolom x 2 baris) sebagai penampil informasi.
6. Menggunakan sensor suhu LM35.
7. Menggunakan rangkaian *driver* sebagai penggerak dari motor DC, yang berfungsi sebagai pengaduk atau pembalik motor.
8. Menggunakan sensor *rotary encoder* sebagai pengatur kecepatan motor.

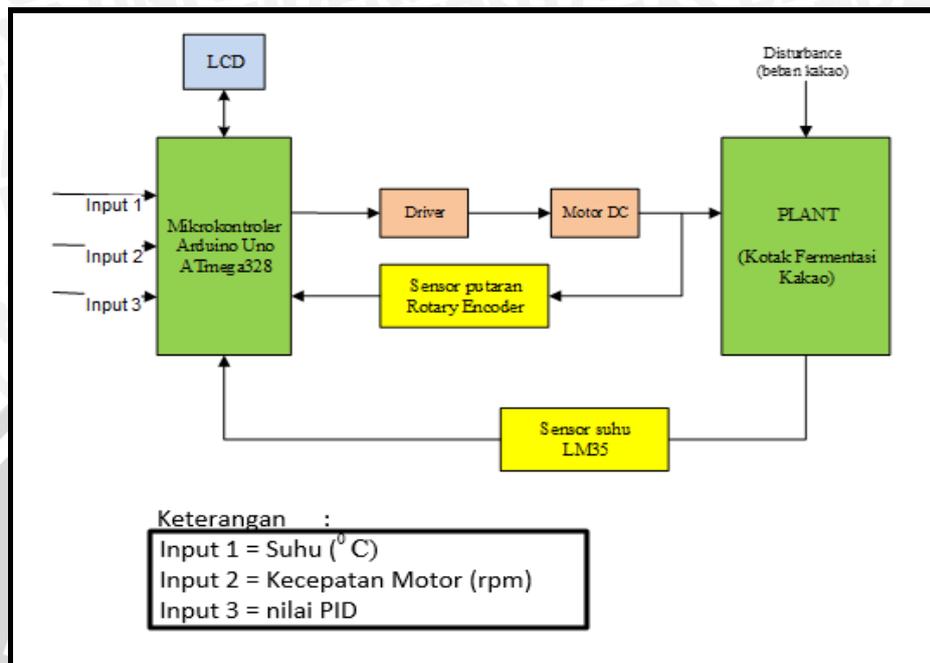
Spesifikasi *non electric* terdiri dari :

Kotak kayu yang memiliki lebar 38cm dan tinggi 45cm yang dibuat dengan bahan- bahan berikut :

1. Triplek
2. Partikel *wood*
3. Kayu
4. Paku
5. Mur dan baut
6. *Bearing* dan engsel

4.1.1 Blok Diagram Alat

Perancangan diagram blok sistem ditunjukkan dalam Gambar 4.1 :



Gambar 4.1 Diagram Blok Sistem

Sumber : Perancangan

4.1.2 Prinsip Kerja Alat

Untuk lebih memudahkan dalam perencanaan, maka perlu dijabarkan deskripsi kerja sistem secara keseluruhan. Mula-mula suhu pada alat fermentasi kakao disetting pada saat 50° C. Kemudian di-*setting* juga kecepatan putaran motor dan masukan nilai PID yang sudah ditentukan. Sinyal digital akan masuk ke dalam mikrokontroler dan langsung diproses oleh mikrokontroler. Setelah itu, pada kotak kayu fermentasi akan dimasukkan kakao dengan beban 40 kg dan proses fermentasi ini membutuhkan waktu 4-5 hari.

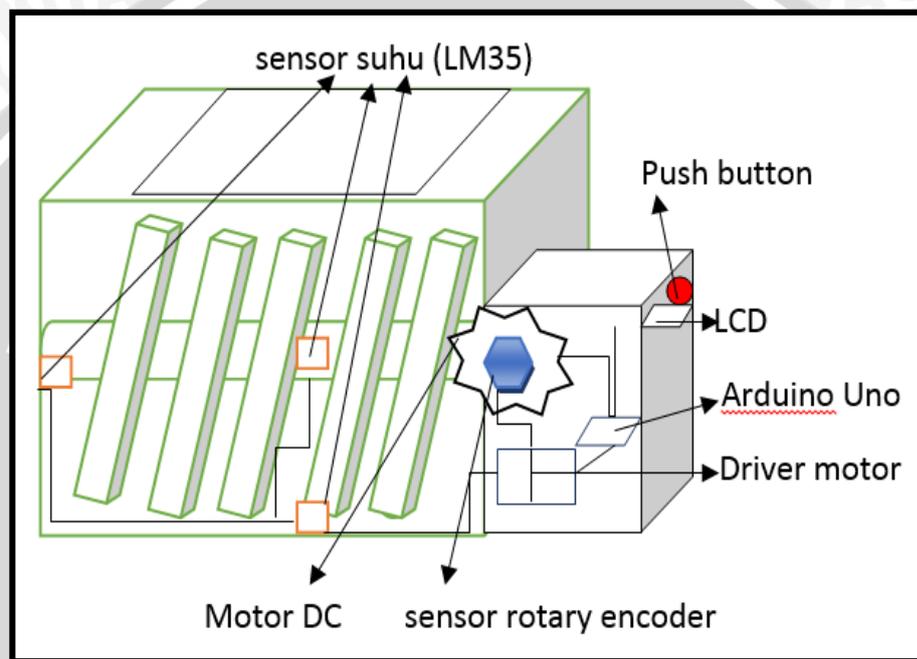
Mikrokontroler akan menggerakkan motor DC ketika suhu mencapai 50° C pada ketiga sensor suhu LM35 yang terpasang di bagian tengah, samping dan bawah dari dalam kotak fermentasi. Selain itu mikrokontroler akan mengatur kecepatan motor agar stabil dengan nilai kecepatan tertentu dan PID yang sudah dimasukkan. Proses pembalikan atau pengadukan akan berlangsung selama 2 menit yang sudah di-*setting* di mikrokontroler karena jika diaduk terlalu lama maka tekstur biji akan rusak. Prinsip kerja alat ini pada intinya terjadi ketika proses

pembalikan atau pengadukan selanjutnya tinggal menunggu proses fermentasi selesai.

4.2 Perancangan Perangkat Keras

4.2.1 Perancangan Mekanik Kotak Fermentasi Kakao

Perancangan mekanik dilakukan untuk merancang pemodelan kotak fermentasi kakao. Secara umum rancangan mekanik pemodelan kotak fermentasi kakao ditunjukkan dalam Gambar 4.2.



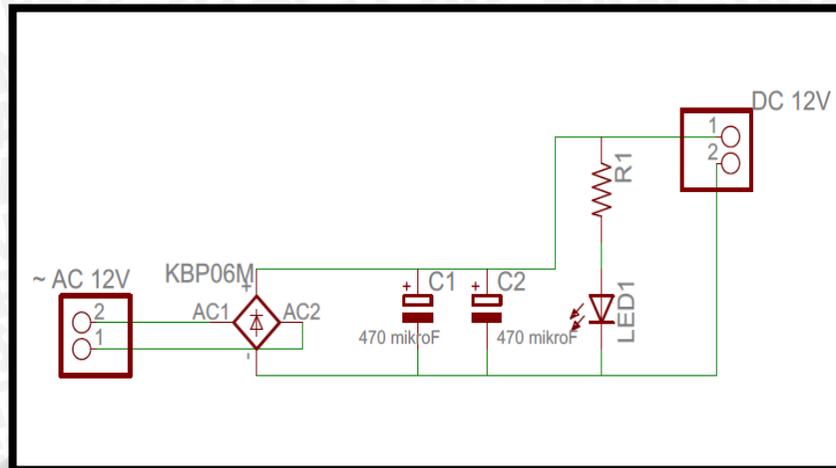
Gambar 4.2 Perancangan Mekanik Kotak Fermentasi Kakao

Sumber : Perancangan

4.2.2 Perancangan Catu Daya Sistem

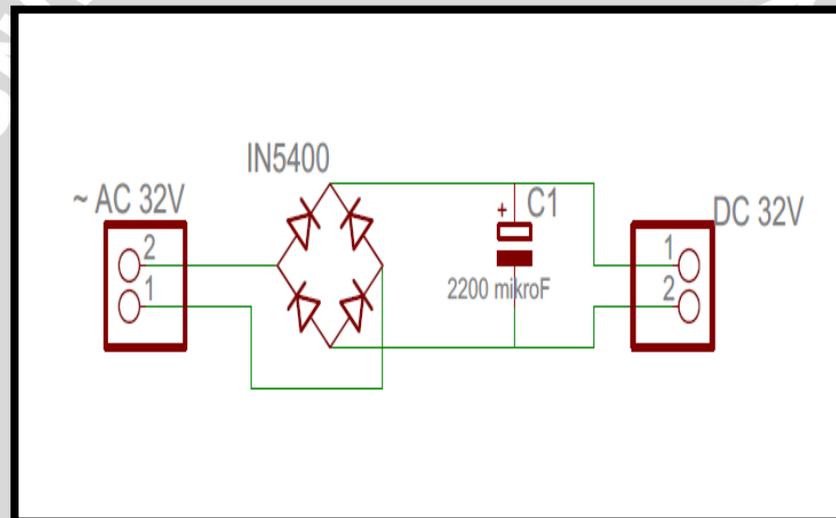
Pada pembuatan alat dalam skripsi ini membutuhkan 2 jenis besaran tegangan yaitu 12 volt untuk mengaktifkan rangkaian mikrokontroler arduino uno dan kipas pendingin pada IC *driver* motor, sedangkan sebagai penggerak motor DC dan *driver* motor menggunakan catu daya sebesar 32 volt.

Dalam pembuatan catu daya menggunakan jenis travo CT 3 ampere untuk catu daya 32 volt dan 500 mA untuk 12 volt. Gambar catu daya 12 volt dan 32 volt ditunjukkan dalam Gambar 4.3 dan 4.4.



Gambar 4.3 Rangkaian Catu daya 12 volt

Sumber : Perancangan

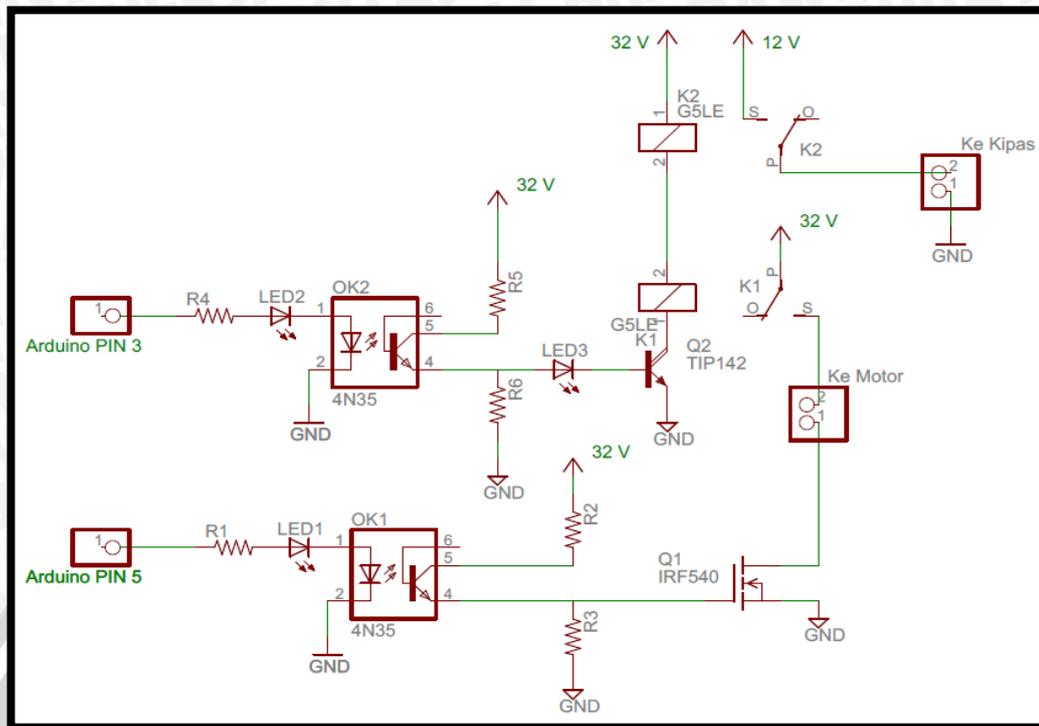


Gambar 4.4 Rangkaian Catu daya 32 volt

Sumber : Perancangan

4.2.2 Perancangan Driver Motor

Untuk *Driver* motor menggunakan *interface optocoupler* yang berfungsi untuk melindungi rangkaian dari lonjakan tegangan pada beban agar tidak mengganggu bagian pengolah data. Pada rangkaian ini terdapat transistor TIP142 yang terhubung ke arduino pin 3 dan *power mosfet* IRF540 yang terhubung ke arduino pin 5 serta terdapat kipas pendingin untuk menghindari *overheat* yang terhubung dengan catu daya 12 volt. Gambar rangkaian *driver* motor ditunjukkan dalam Gambar 4.5.

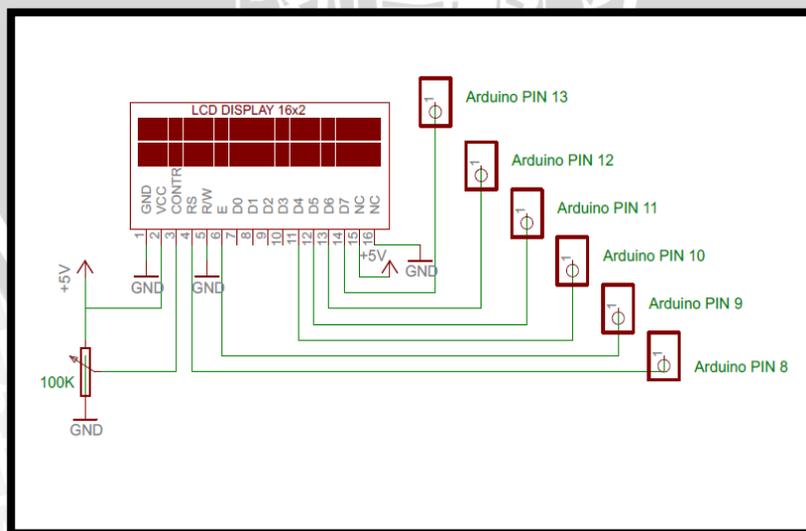


Gambar 4.5 Rangkaian Driver Motor DC

Sumber : Perancangan

4.2.4 Rangkaian LCD 16x2

Pada rangkaian LCD 16x2 berfungsi untuk menampilkan data suhu pada T1,T2 dan T3, nilai PWM serta kecepatan putaran motor dengan satuan rpm. Gambar rangkaian LCD 16x2 ditunjukkan dalam Gambar 4.6.

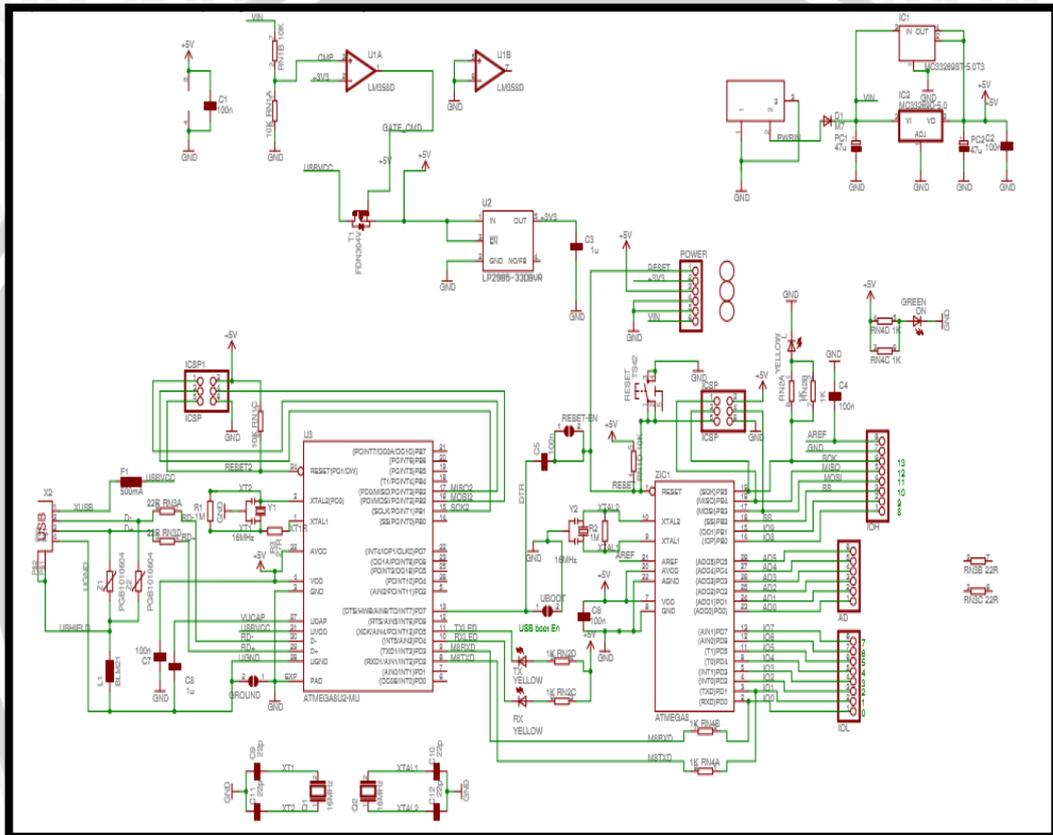


Gambar 4.6 Rangkaian Perencanaan LCD 16x2

Sumber : Perancangan

4.2.5 Rangkaian Mikrokontroler Arduino Uno

Pada perancangan ini, menggunakan mikrokontroler Arduino Uno sebagai pengatur motor agar berputar pada saat suhu yang sudah ditentukan dan mengatur kecepatan motor. Rangkaian Arduino Uno ini terhubung dengan LCD 16x2 pada PIN 13,12,11,10,9, dan 8, ketiga sensor suhu, *push button*, *driver motor* dan *rotary encoder*. Gambar rangkaian mikrokontroler Arduino Uno ditunjukkan dalam Gambar 4.7.



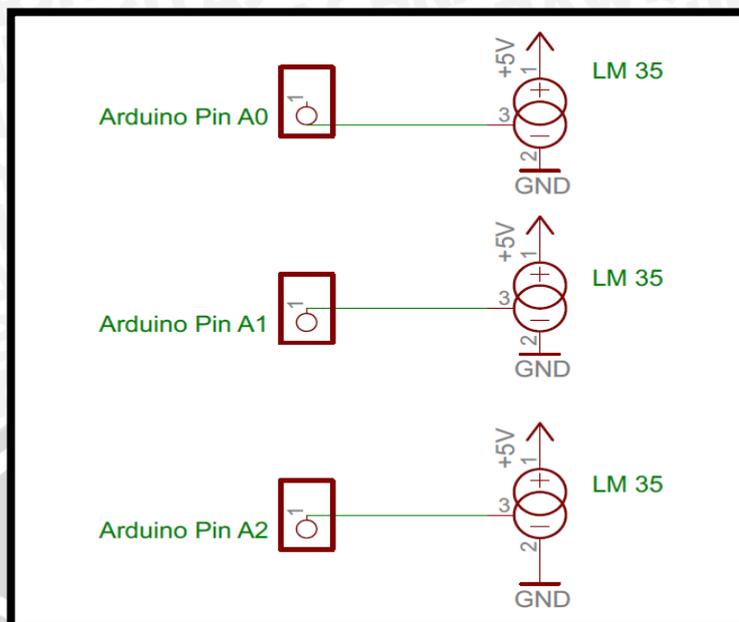
Gambar 4.7 Rangkaian Perencanaan Mikrokontroler Arduino UNO

Sumber : *Adruino.cc*

4.2.6 Rangkaian Sensor Suhu LM35

Rangkaian sensor suhu menggunakan jenis sensor suhu LM35 yang terpasang 3 buah didalam kotak fermentasi. Sensor suhu 1 atau T1 terletak disamping kiri kotak ,sensor suhu 2 atau T2 terletak ditengah kotak dan sensor suhu 3 atau T3 terletak dibawah kotak. Tujuan peletakkannya agar dapat memonitor suhu disetiap tempat berbeda sehingga memudahkan dalam menentukan suhu yang tepat pada proses pengadukan atau pembalikan. Gambar rangkaian sensor suhu LM35 ditunjukkan dalam Gambar 4.8.



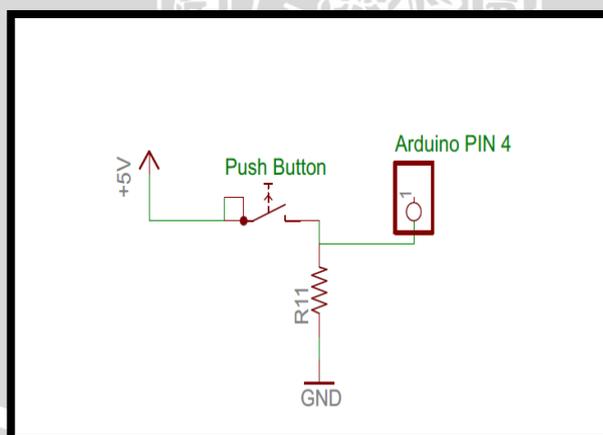


Gambar 4.8 Rangkaian Perencanaan Sensor Suhu LM35

Sumber : Perancangan

4.2.7 Rangkaian Push Button

Pembuatan rangkaian *push button* yang terpasang pada arduino PIN 4 bertujuan untuk tombol pemutar motor. Pada saat suhu sudah tercapai dan motor tidak berputar maka peran tombol *push button* sangat dibutuhkan. Gambar rangkaian *push button* ditunjukkan dalam Gambar 4.9.



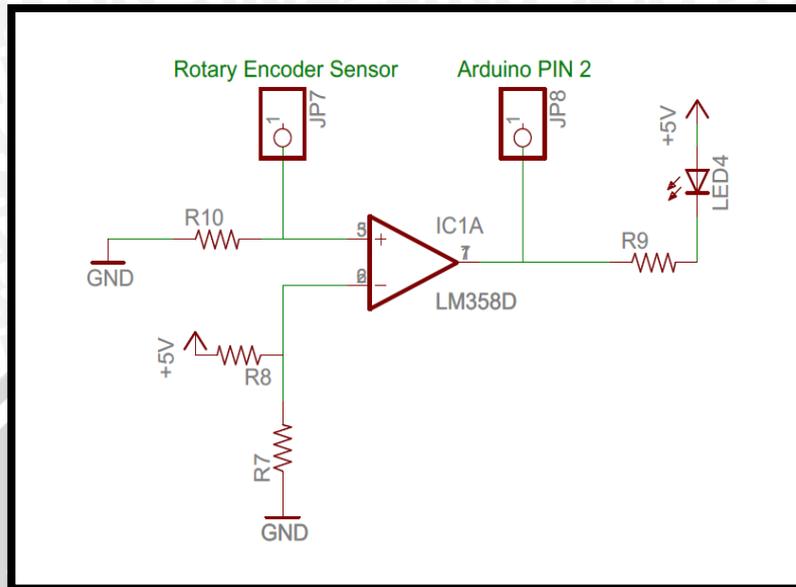
Gambar 4.9 Rangkaian Perencanaan Push Button

Sumber : Perancangan

4.2.8 Rangkaian Komparator Sensor Rotary Encoder

Pada rangkaian komparator sensor *rotary encoder* terdapat op-amp IC1A sebagai penguat dan sensor *rotary encoder* yang terdapat dalam perancangan ini

menggunakan 7 lubang piringan. Gambar rangkaian komparator sensor *rotary encoder* ditunjukkan dalam Gambar 4.10.

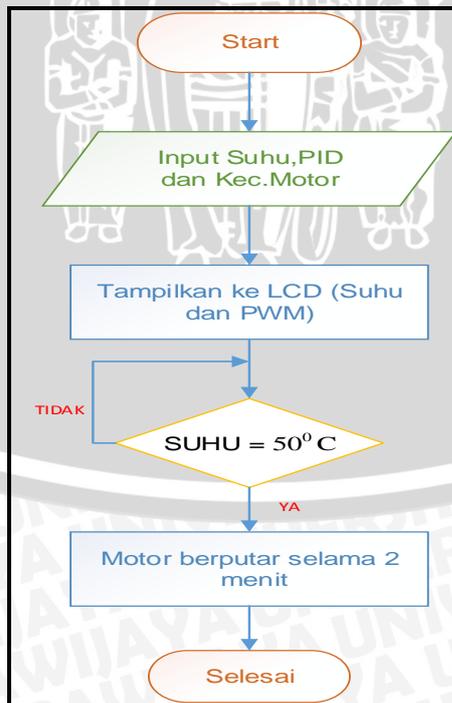


Gambar 4.10 Rangkaian Perencanaan Komparator Sensor *Rotary Encoder*

Sumber : Perancangan

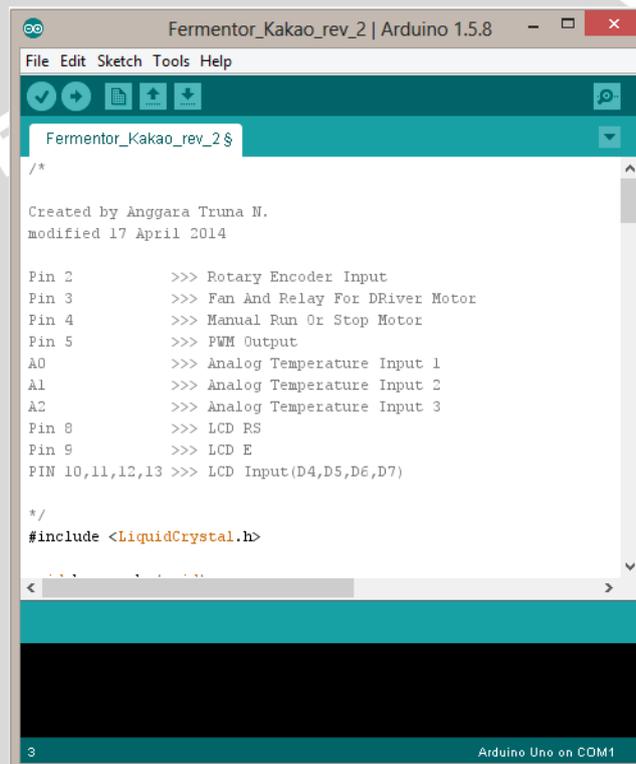
4.3 Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan dalam Gambar 4.11 berikut :



Gambar 4.11 *Flowchart* alat fermentasi kakao

Mikrokontroler arduino uno memiliki IDE atau *Integrated development environment*. IDE adalah sebuah *software* PC yang berfungsi untuk menulis *syntax-syntax* agar arduino tersebut bisa berfungsi seperti apa yang kita inginkan. Pada alat ini program yang digunakan adalah arduino IDE 1.5.8 beta untuk memberikan perintah ke dalam arduino uno. Pengoperasian program ini menggunakan bahasa C++ kemudian di masukan ke dalam arduino uno sebagai *board controller*. Penampakan program arduino IDE 1.5.8 pada fermentor kakao ditunjukkan dalam Gambar 4.12.



```
File Edit Sketch Tools Help
Fermentor_Kakao_rev_2 $
/*
Created by Anggara Truna N.
modified 17 April 2014

Pin 2      >>> Rotary Encoder Input
Pin 3      >>> Fan And Relay For DRIVER Motor
Pin 4      >>> Manual Run Or Stop Motor
Pin 5      >>> PWM Output
A0         >>> Analog Temperature Input 1
A1         >>> Analog Temperature Input 2
A2         >>> Analog Temperature Input 3
Pin 8      >>> LCD RS
Pin 9      >>> LCD E
PIN 10,11,12,13 >>> LCD Input (D4,D5,D6,D7)

*/
#include <LiquidCrystal.h>

3 Arduino Uno on COM1
```

Gambar 4.12 Program Arduino IDE 1.5.8

BAB V

PENGUJIAN DAN ANALISIS

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu:

1. Pengujian *driver*
2. Pengujian motor DC
3. Pengujian suhu
4. Pengujian keseluruhan sistem

5.1 Pengujian *Driver*

- a. Tujuan
 - Mengetahui hubungan antara PWM dengan tegangan keluaran *driver* tanpa motor dan dengan motor.
- b. Peralatan
 1. Alat fermentasi kakao
 2. *Notebook* dengan aplikasi arduino IDE
 3. Multimeter
- c. Langkah Pengujian *Driver* Tanpa Motor
 1. Menghidupkan alat tanpa menekan *push button* agar motor tidak bergerak.
 2. Mengubah – ubah nilai PWM melalui program arduino IDE.
 3. Mengukur tegangan keluaran pada keluaran *driver* menggunakan multimeter.
- d. Langkah Pengujian *Driver* Menggunakan Motor
 1. Menghidupkan alat dan menekan *push button* agar motor bergerak.
 2. Mengubah – ubah nilai PWM melalui program arduino IDE.
 3. Mengukur tegangan keluaran pada keluaran *driver* menggunakan multimeter.

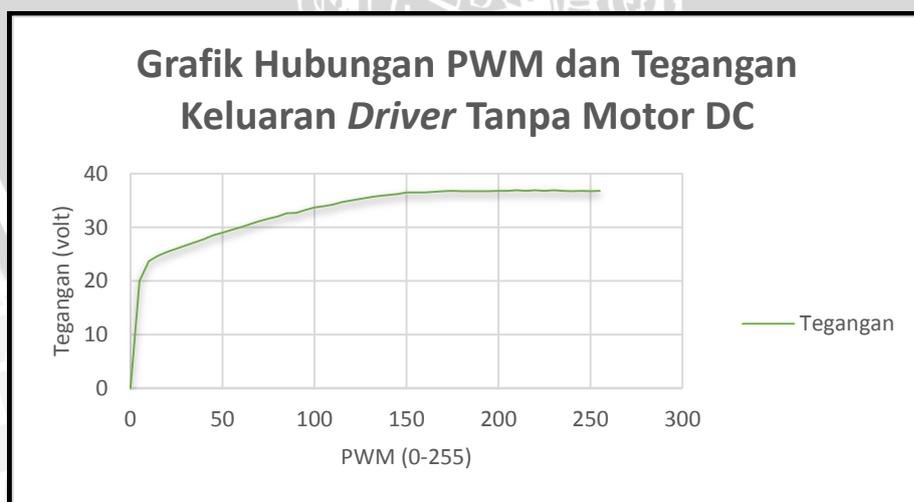
e. Hasil Pengujian *Driver* Tanpa Motor

Tabel hasil pengujian *driver* tanpa motor ditunjukkan dalam Tabel 5.1.

Tabel 5.1 Hasil Pengujian *Driver* Tanpa Motor

No.	PWM	Tegangan (volt)	No.	PWM	Tegangan (volt)	No.	PWM	Tegangan (volt)
1	0	0	19	90	32.7	37	180	36.7
2	5	20	20	95	33.2	38	185	36.7
3	10	23.7	21	100	33.7	39	190	36.7
4	15	24.7	22	105	33.9	40	195	36.7
5	20	25.4	23	110	34.2	41	200	36.8
6	25	26	24	115	34.7	42	205	36.8
7	30	26.6	25	120	35	43	210	36.9
8	35	27.2	26	125	35.3	44	215	36.8
9	40	27.8	27	130	35.6	45	220	36.9
10	45	28.5	28	135	35.8	46	225	36.8
11	50	29	29	140	36	47	230	36.9
12	55	29.5	30	145	36.2	48	235	36.8
13	60	30	31	150	36.5	49	240	36.7
14	65	30.6	32	155	36.5	50	245	36.8
15	70	31.1	33	160	36.5	51	250	36.7
16	75	31.6	34	165	36.6	52	255	36.8
17	80	32	35	170	36.7			
18	85	32.6	36	175	36.8			

Grafik hasil pengujian *driver* tanpa motor ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Grafik Hubungan PWM dengan Tegangan Keluaran *Driver* Tanpa Motor

Dalam Tabel 5.1 dan Gambar 5.1 diketahui bahwa hubungan nilai PWM dengan tegangan berbanding lurus yaitu semakin besar nilai PWM maka semakin besar juga nilai besaran tegangan yang dihasilkan.

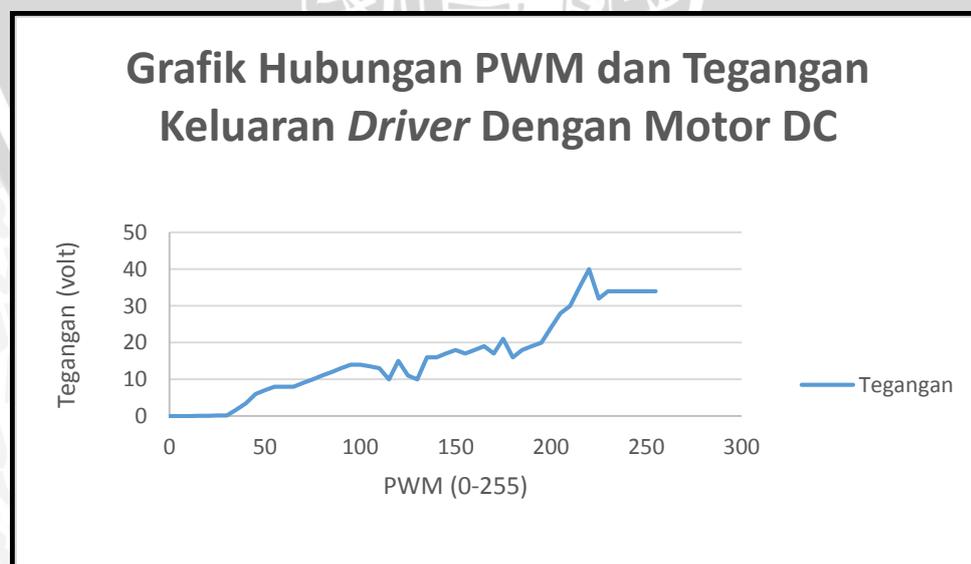
f. Hasil Pengujian *Driver* Menggunakan Motor

Tabel hasil pengujian *driver* menggunakan motor ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian *Driver* Menggunakan Motor

No.	PWM	Tegangan (volt)	No.	PWM	Tegangan (volt)	No.	PWM	Tegangan (volt)
1	0	0	19	90	13	37	180	16
2	5	0	20	95	14	38	185	18
3	10	0	21	100	14	39	190	19
4	15	0.1	22	105	13.6	40	195	20
5	20	0.1	23	110	13	41	200	24
6	25	0.2	24	115	10	42	205	28
7	30	0.2	25	120	15	43	210	30
8	35	1.7	26	125	11	44	215	35
9	40	3.5	27	130	10	45	220	40
10	45	6	28	135	16	46	225	32
11	50	7	29	140	16	47	230	34
12	55	8	30	145	17	48	235	34
13	60	8	31	150	18	49	240	34
14	65	8	32	155	17	50	245	34
15	70	9	33	160	18	51	250	34
16	75	10	34	165	19	52	255	34
17	80	11	35	170	17			
18	85	12	36	175	21			

Grafik hasil pengujian *driver* menggunakan motor ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Grafik Hubungan PWM dengan Tegangan Keluaran *Driver* Menggunakan Motor

Dalam Tabel 5.2 dan Gambar 5.2 diketahui bahwa nilai besaran tegangan keluaran menurun setelah menggunakan motor dan penurunan tersebut dapat dilihat pada awal – awal nilai PWM 0 sampai nilai PWM 210 yang sudah mencapai nilai besaran tegangan keluaran 30 volt tetapi terdapat satu kali lonjakan tegangan pada nilai PWM 220. Pada pengujian tanpa menggunakan motor dan menggunakan motor tidak terjadi kebocoran tegangan yang ditunjukkan pada nilai PWM 0 dengan nilai besaran tegangan keluarannya juga 0 volt hal ini disebabkan karena dalam rangkaian ini digunakan *interface optocoupler* sehingga tidak terjadi kebocoran tegangan dari minimum sistem.

5.2 Pengujian Motor DC

a. Tujuan

- Mengetahui hubungan antara PWM dengan kecepatan motor DC tanpa menggunakan beban biji kakao dan dengan beban biji kakao.
- Mengetahui kekuatan motor DC terhadap beban biji kakao.

b. Peralatan

1. Alat fermentasi kakao
2. *Notebook* dengan aplikasi arduino IDE
3. Biji kakao

c. Langkah Pengujian Kecepatan Motor DC Tanpa Beban Biji Kakao

1. Menghidupkan alat fermentasi dengan LCD yang menampilkan nilai PWM dan kecepatan putaran motor DC.
2. Menjalankan program yang telah dibuat pada arduino IDE yang berfungsi membaca sinyal PWM.
3. Menghitung nilai rata-rata kecepatan motor DC yang ditampilkan pada setiap nilai PWM.

d. Langkah Pengujian Kecepatan Motor DC Dengan Beban Biji Kakao 10 kg

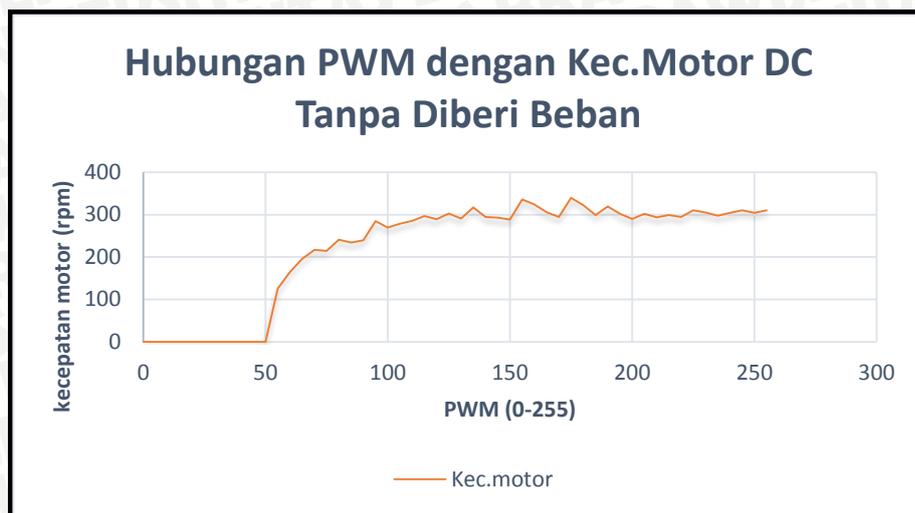
1. Menghidupkan alat fermentasi dengan LCD yang menampilkan nilai PWM dan kecepatan putaran motor DC.
2. Memasukkan biji kakao 10 kg ke kotak fermentasi
3. Menjalankan program yang telah dibuat pada program arduino IDE yang berfungsi membaca sinyal PWM.

4. Menghitung nilai rata-rata kecepatan motor DC yang ditampilkan pada setiap nilai PWM.
 - e. Langkah Pengujian Kecepatan Motor DC Dengan Beban Biji Kakao 15 kg
 1. Menghidupkan alat fermentasi dengan LCD yang menampilkan nilai PWM dan kecepatan putaran motor DC.
 2. Memasukkan biji kakao 15 kg ke kotak fermentasi
 3. Menjalankan program yang telah dibuat pada program arduino IDE yang berfungsi membaca sinyal PWM.
 4. Menghitung nilai rata-rata kecepatan motor DC yang ditampilkan pada setiap nilai PWM
 - f. Hasil Pengujian Kecepatan Motor DC Tanpa Beban Biji Kakao
- Tabel hasil pengujian kecepatan motor DC tanpa beban biji kakao ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian Kecepatan Motor DC Tanpa Beban Biji Kakao

No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)
1	0	0	19	90	239.63	37	180	322.09
2	5	0	20	95	284.84	38	185	298.73
3	10	0	21	100	269.75	39	190	319.14
4	15	0	22	105	278.69	40	195	302.17
5	20	0	23	110	285.04	41	200	290.1
6	25	0	24	115	296.77	42	205	301.79
7	30	0	25	120	289.32	43	210	293.32
8	35	0	26	125	302.95	44	215	298.9
9	40	0	27	130	290.55	45	220	294.48
10	45	0	28	135	316.95	46	225	310.44
11	50	0	29	140	294.74	47	230	304.7
12	55	125.62	30	145	293.3	48	235	297.75
13	60	164.66	31	150	288.66	49	240	304.24
14	65	196.1	32	155	335.64	50	245	309.89
15	70	217.09	33	160	323.67	51	250	304.5
16	75	214.23	34	165	305.9	52	255	310.09
17	80	241.08	35	170	294.47			
18	85	234.2	36	175	339.99			

Grafik hasil pengujian kecepatan motor DC tanpa beban biji kakao ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Grafik Hubungan PWM dengan Kecepatan Motor Tanpa Beban Biji Kakao

Dalam Tabel 5.3 dan Gambar 5.3 diketahui bahwa nilai kecepatan motor mulai muncul setelah nilai PWM 50 pada pengujian tanpa beban biji kakao.

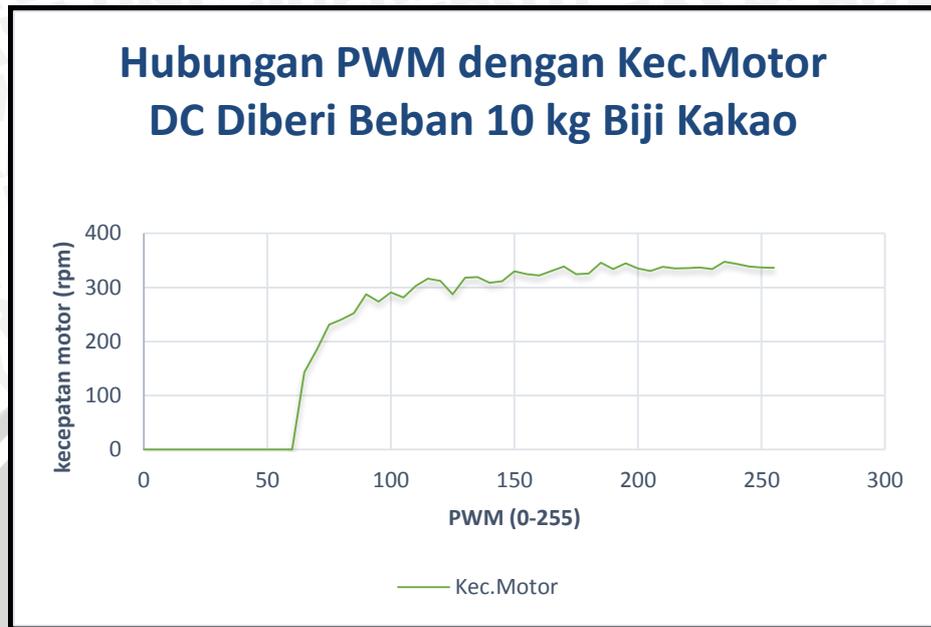
g. Hasil Pengujian Kecepatan Motor DC dengan Beban Biji Kakao 10 kg

Tabel hasil pengujian kecepatan motor DC dengan beban biji kakao 10 kg ditunjukkan dalam Tabel 5.4.

Tabel 5.4 Hasil Pengujian Kecepatan Motor DC dengan Beban Biji Kakao 10 kg

No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)
1	0	0	19	90	287.2	37	180	325.77
2	5	0	20	95	273.49	38	185	346.05
3	10	0	21	100	290.89	39	190	334.14
4	15	0	22	105	281.42	40	195	344.42
5	20	0	23	110	302.81	41	200	335.35
6	25	0	24	115	316.08	42	205	330.33
7	30	0	25	120	311.9	43	210	338.39
8	35	0	26	125	287.31	44	215	335.21
9	40	0	27	130	317.98	45	220	335.71
10	45	0	28	135	319.26	46	225	337.2
11	50	0	29	140	308.82	47	230	334.05
12	55	0	30	145	311.37	48	235	347.51
13	60	0	31	150	329.92	49	240	343.25
14	65	143.31	32	155	324.36	50	245	338.95
15	70	185.45	33	160	322.08	51	250	337.26
16	75	231.49	34	165	330.77	52	255	336.54
17	80	240.4	35	170	338.6			
18	85	252.25	36	175	324.48			

Grafik hasil pengujian kecepatan motor DC dengan beban biji kakao 10 kg ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Grafik Hubungan PWM dengan Kecepatan Motor DC Menggunakan Beban Biji Kakao 10 kg.

Dalam Tabel 5.4 dan Gambar 5.4 diketahui bahwa nilai kecepatan motor mulai muncul setelah nilai PWM 60 dan mengalami peningkatan kecepatan motor pada pengujian dengan beban biji kakao 10 kg saat PWM 150 - 255.

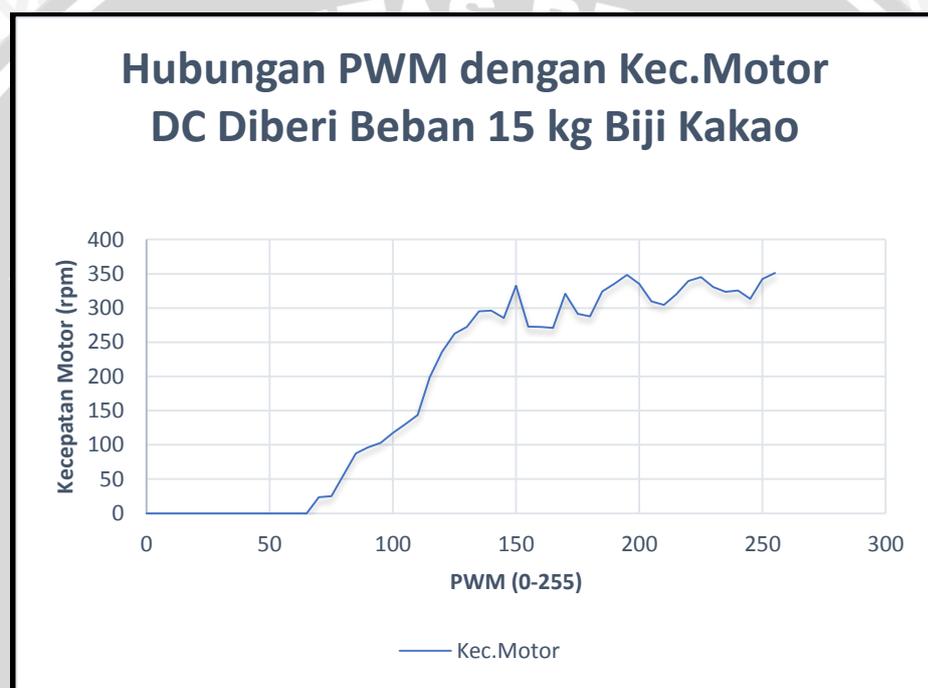
- h. Hasil Pengujian Kecepatan Motor DC dengan Beban Biji Kakao 15 kg
Tabel hasil pengujian kecepatan motor DC dengan beban biji kakao 15 kg ditunjukkan dalam Tabel 5.5.

Tabel 5.5 Hasil Pengujian Kecepatan Motor DC Tanpa Beban Biji Kakao

No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)
1	0	0	19	90	96.2	37	180	287.98
2	5	0	20	95	103.1	38	185	324.04
3	10	0	21	100	117.5	39	190	336.01
4	15	0	22	105	130	40	195	348.64
5	20	0	23	110	143.57	41	200	335.5
6	25	0	24	115	199.1	42	205	309.84
7	30	0	25	120	235.84	43	210	304.37
8	35	0	26	125	262.45	44	215	319.85
9	40	0	27	130	272.38	45	220	339.73
10	45	0	28	135	295.38	46	225	344.97
11	50	0	29	140	296.14	47	230	330.58

No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)	No.	PWM	Kec.Motor (rpm)
12	55	0	30	145	285.54	48	235	323.62
13	60	0	31	150	332.33	49	240	325.78
14	65	0	32	155	272.82	50	245	313.63
15	70	23.6	33	160	272.4	51	250	342.26
16	75	25	34	165	270.93	52	255	351.23
17	80	56.4	35	170	320.81			
18	85	87.55	36	175	291.44			

Grafik hasil pengujian kecepatan motor DC dengan beban biji kakao 10 kg ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Grafik Hubungan PWM dengan Kecepatan Motor DC Menggunakan Beban Biji Kakao 15 kg.

Dalam Tabel 5.5 dan Gambar 5.5 diketahui bahwa nilai kecepatan motor mulai muncul setelah nilai PWM 65 dan mengalami peningkatan kecepatan motor pada pengujian dengan beban biji kakao 15 kg saat PWM 185 – 255.

- i. Hasil Pengujian dengan Beban Lebih dari 15 kg Biji Kakao
 - Setelah melakukan pengujian dengan beban biji kakao lebih dari 15 kg motor DC tidak dapat berputar sehingga pengujian pada alat fermentasi ini hanya dibatasi untuk pengukuran berat biji kakao sampai 15 kg.

5.3 Pengujian Suhu

a. Tujuan

- Mengetahui hubungan dan respon suhu pada proses fermentasi biji kakao terhadap waktu fermentasi biji kakao pada alat.

b. Langkah Pengujian Suhu Fermentasi Biji Kakao pada Alat

- Mencatat nilai suhu T1, T2 dan T3 yang ditampilkan pada LCD setiap 4 jam sekali selama 5 hari

c. Hasil Pengujian Suhu Fermentasi Biji Kakao

Gambar tampilan LCD pada saat 4 jam, 48 jam dan 120 hari ditunjukkan dalam Gambar 5.6, 5.7, dan 5.8

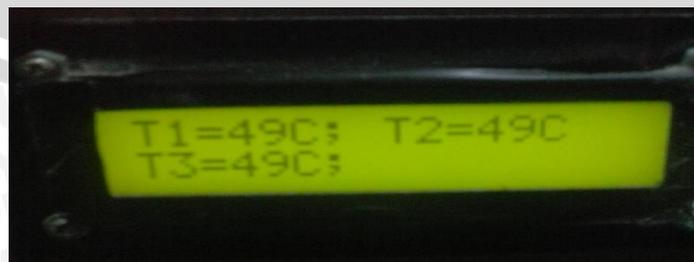


Gambar 5.6 Tampilan LCD saat 4 jam.



Gambar 5.7 Tampilan LCD saat 48 jam.

Pada saat LCD menampilkan suhu T1, T2, dan T3 sama dengan 50⁰ C maka pada saat itu motor DC akan berputar untuk melakukan proses pengadukan dan pembalikan.



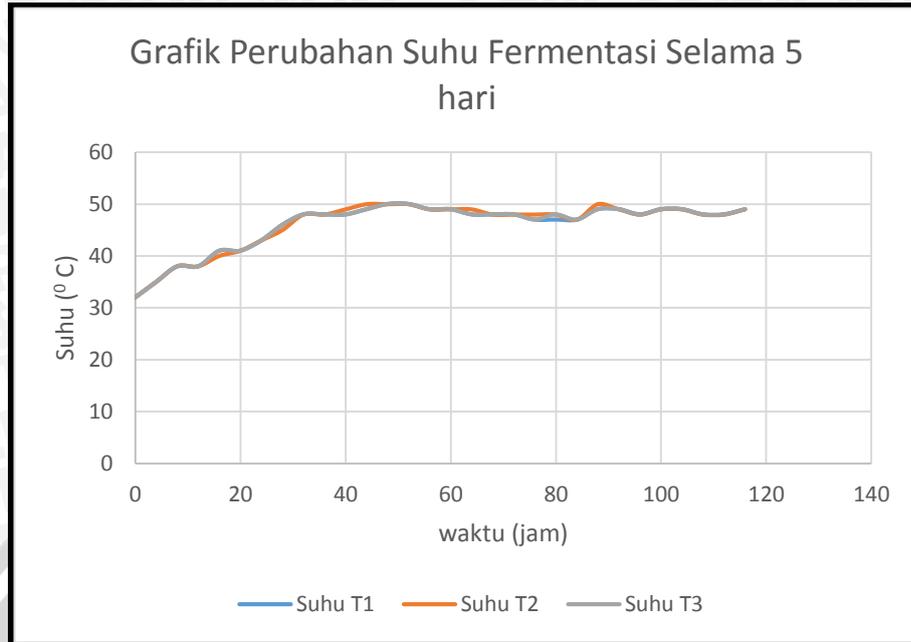
Gambar 5.8 Tampilan LCD saat 120 jam.

Tabel hasil pengujian suhu fermentasi biji kakao pada alat fermentasi kakao ditunjukkan dalam Tabel 5.6.

Tabel 5.6 Hasil Pengujian Suhu Fermentasi Kakao

No.	Jam	Suhu T1(⁰ C)	Suhu T2(⁰ C)	Suhu T3(⁰ C)
1	0	32	32	32
2	4	35	35	35
3	8	38	38	38
4	12	38	38	38
5	16	41	40	41
6	20	41	41	41
7	24	43	43	43
8	28	46	45	46
9	32	48	48	48
10	36	48	48	48
11	40	48	49	48
12	44	49	50	49
13	48	50	50	50
14	52	50	50	50
15	56	49	49	49
16	60	49	49	49
17	64	48	49	48
18	68	48	48	48
19	72	48	48	48
20	76	47	48	47
21	80	47	48	48
22	84	47	47	47
23	88	49	50	49
24	92	49	49	49
25	96	48	48	48
26	100	49	49	49
27	104	49	49	49
28	108	48	48	48
29	112	48	48	48
30	116	49	49	49
31	120	49	49	49

Grafik hasil pengujian suhu fermentasi biji kakao pada alat fermentasi biji kakao ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Grafik Hubungan Suhu Fermentasi terhadap Waktu

Dalam Tabel 5.6 dan Gambar 5.9 diketahui bahwa suhu fermentasi kakao terus meningkat setelah 4 jam hingga mencapai suhu 50°C pada 48 jam, setelah menetap 50°C pada 52 jam suhu akan kembali naik turun kisaran 49°C - 47°C dihitungan jam berikutnya .

5.4 Pengujian Keseluruhan

a. Tujuan

Untuk pengujian ini bertujuan untuk mengetahui keberhasilan dari keseluruhan sistem yang dirancang dan perbandingan biji kakao sebelum dan sesudah fermentasi. Pada pengujian ini semua sistem digabungkan menjadi satu kesatuan dan menggunakan kontrol PID dengan metode kedua dari teori Ziegler-Nichols sebagai sistem kontrolnya. Pengujian dilakukan untuk mengetahui kontrol sistem alat fermentasi kakao.

b. Peralatan

1. *Notebook* dengan software arduino IDE
2. Alat fermentasi kakao

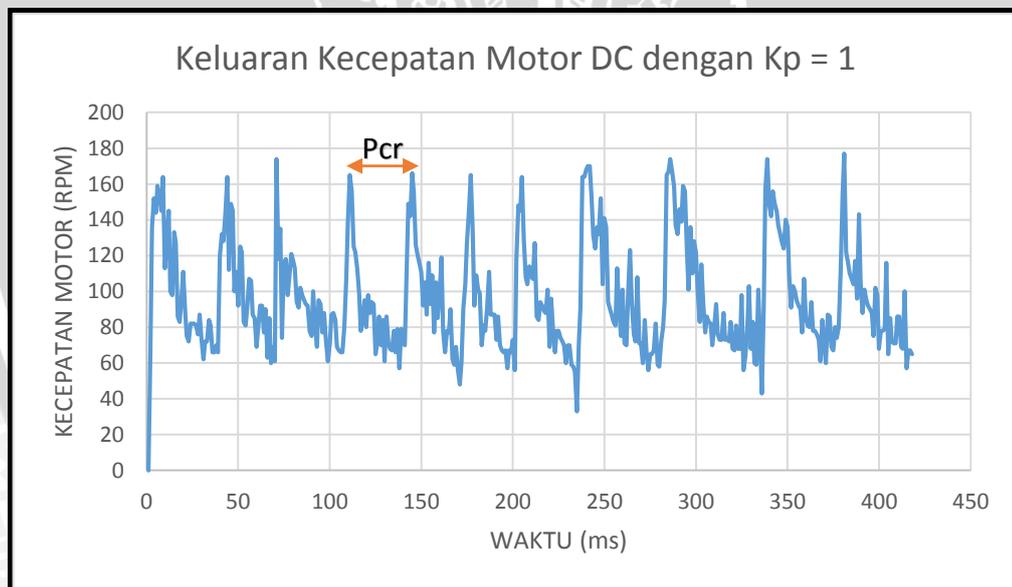
c. Langkah Pengujian I

Menentukan nilai parameter kontrol PID (K_p , K_i , dan K_d).

- Menghidupkan alat fermentasi kakao dengan menghubungkan ke sumber arus AC.
- Memberikan penguatan proporsional dari nol hingga alat dapat memberikan respons berupa osilasi berkesinambungan, dengan penguatan integral nol dan penguatan diferensial nol.
- Membuat grafik yang menyatakan hubungan antara kecepatan putaran motor dengan waktu.
- Menentukan nilai K_{cr} dan P_{cr} yaitu K_{cr} merupakan nilai penguatan kritis dan P_{cr} merupakan nilai periode (T).
- Menentukan parameter kontroler PID berdasarkan metode kedua Ziegler-Nichols (metode loop tertutup).

d. Hasil Pengujian I

Dengan menggunakan penguatan K_p sebesar 1 maka didapatkan respons seperti dalam Gambar 5.10



Gambar 5.10 Grafik Kecepatan Motor Terhadap Waktu dengan $K_p=1$

Respons sistem menampilkan data setiap 0,1 sekon, maka dari grafik diketahui bahwa nilai

$$\begin{aligned}
 P_{cr} &= \text{jumlah data} \times \text{waktu antar data} \\
 &= 37 \times 0,1 \\
 &= 3,7
 \end{aligned}$$

$$\text{dan } K_{cr} = 1$$

Dengan menggunakan metode kedua dari teori Ziegler-Nichols , maka akan didapatkan nilai sebagai berikut :

$$K_p = 0,60 \times K_{cr} = 0,60 \times 1 = 0,6$$

$$T_i = 0,5 \times P_{cr} = 0,5 \times 3,7 = 1,85$$

$$T_d = 0,125 \times P_{cr} = 0,125 \times 3,7 = 0,4625$$

Nilai – nilai penguatan K_i dan K_d dapat dicari dengan :

$$K_i = \frac{K_p}{T_i} = \frac{0,6}{1,85} = 0,324$$

$$K_d = K_p \times T_d = 0,6 \times 0,4625 = 0,2775$$

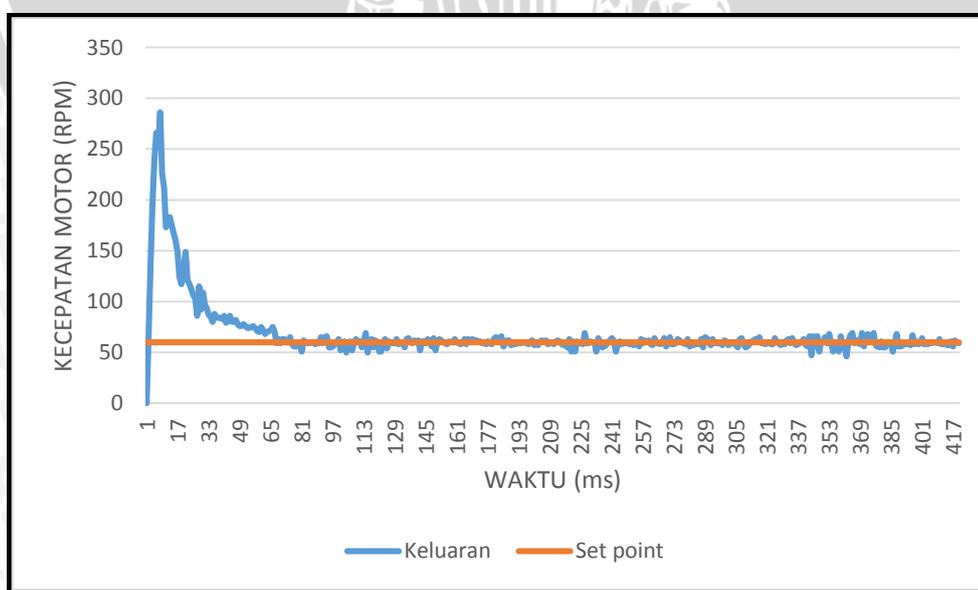
e. Langkah Pengujian II

Selanjutnya melakukan pengujian pada alat fermentasi kakao dengan memasukkan nilai parameter PID (K_p , K_i , dan K_d) yang sudah didapatkan.

- Memasukan nilai kecepatan motor 60 rpm dan nilai parameter PID (K_p , K_i , K_d) pada program yang telah dibuat pada arduino IDE.
- Menjalankan alat fermentasi kakao lalu mengamati dengan serial monitor pada program arduino IDE dan mencatat nilai kecepatan motor dan error pada data yang dikeluarkan program.

f. Hasil Pengujian II

Tabel hasil pengujian kecepatan motor DC dengan PID dengan nilai $K_p=0,6$, $K_i=0,324$ dan $K_d=0,2775$ pada alat fermentasi kakao ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Hasil Pengujian Kecepatan Motor DC dengan PID tanpa Beban

Dari Gambar 5.11, diketahui bahwa hasil respon memiliki % *error* sebagai berikut:

$$\begin{aligned}\% E_{ss} &= \frac{1}{N} \times \sum_{i=1}^N \left| \frac{Pv(i) - \text{setpoint}}{\text{setpoint}} \right| \times 100 \% \\ &= \frac{1}{353} \times 15,2 \times 100\% \\ &= 4,3\%.\end{aligned}$$

g. Hasil Perbandingan Biji Kakao Sebelum dan Sesudah Fermentasi.

Dari hasil pengujian alat didapatkan hasil perbandingan sesudah dan sebelum dilakukan fermentasi biji kakao menggunakan alat ini selama 5 hari yang ditunjukkan dalam Gambar 5.12, 5.13, 5.14 dan 5.15.



Gambar 5.12 Gambar Biji Kakao Sebelum Fermentasi



Gambar 5.13 Gambar Biji Kakao Sebelum Fermentasi pada Alat



Gambar 5.14 Gambar Biji Kakao Sesudah Fermentasi



Gambar 5.15 Gambar Biji Kakao Sesudah Fermentasi pada Alat

Pada fermentasi biji kakao dengan alat ini terjadi perubahan warna biji kakao menjadi kecoklatan dan tidak berjamur yang menandakan bahwa proses fermentasi berlangsung dengan baik. Setelah pengujian ini dapat disimpulkan kontribusi alat dan kelebihan alat fermentasi ini dengan fermentasi konvensional adalah dapat memonitoring suhu secara tepat sehingga proses pengadukan berdasarkan suhu bukan lagi waktu, mempersempit lahan karena hanya membutuhkan 1 kotak dan membantu petani agar tidak perlu lagi untuk melakukan pengadukan atau pembalikan dengan tangan.

BAB VI PENUTUP

6.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut.

1. Mikrokontroler Arduino Uno pada alat fermentasi kakao otomatis ini dapat :
 - Menghitung dan mengirim kecepatan motor DC sesuai dengan masukan dari sensor *rotary encoder*.
 - Menghitung dan mengirim nilai suhu sesuai dengan masukan dari ketiga sensor suhu LM35.
 - Mengontrol kecepatan motor DC pada beban kakao maksimal 15 kg yang disesuaikan dengan suhu pembalikan atau pengadukan yaitu 50 °C selama 2 menit.
2. Penggunaan kontroler PID pada alat fermentasi kakao otomatis melalui metode kedua dari teori Ziegler-Nichols menghasilkan parameter PID ($K_p=0,6$, $K_i=0,324$ dan $K_d=0,2775$) pada *set point* kecepatan motor DC 60 rpm dan memiliki *error stady state* sebesar 4,3% serta mengalami *overshoot* kurang dari 70 ms.

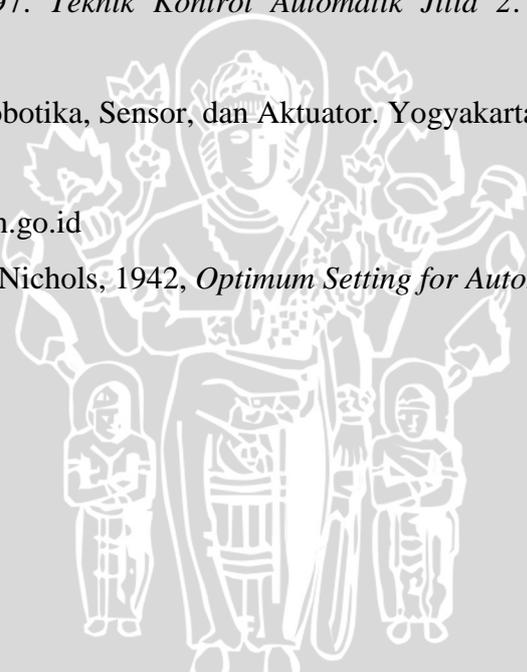
6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah:

1. Penggunaan motor yang lebih kuat selain motor DC yaitu menggunakan motor AC.
2. Pengujian sistem kontrol kecepatan motor menggunakan beban kakao untuk mendapatkan respon sistem kontrol alat terhadap beban kakao.
3. Penggunaan piringan sensor *rotary encoder* sebaiknya memiliki lebih banyak lubang sehingga data yang didapatkan lebih akurat.
4. Penentuan parameter PID menggunakan metode lain selain Ziegler-Nichols untuk mendapatkan respon yang lebih bagus.
5. Untuk hasil fermentasi dari alat ini sebaiknya di uji ke lembaga yang sudah berpengalaman menangani fermentasi kakao.

DAFTAR PUSTAKA

- Arduino.cc, Arduino uno
 Datasheet.
 Ditjenbun.pertanian.go.id
 Gunterus, Frans. 1994. *Falsafah Dasar: Sistem Pengendalian Proses*. Jakarta: PT. Elex Media Komputindo.
 Iccri.net
 Kismet, Fadillah.1999. Instalasi Motor Listrik. Bandung: PT Angkasa
 Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1*. Jakarta. Penerbit Erlangga.
 Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta. Penerbit Erlangga.
 Sigit, Riyanto. 2007 Robotika, Sensor, dan Aktuator. Yogyakarta : Graha Ilmu
 Spaennare.se
 Yogya.litbang.pertanian.go.id
 Ziegler, J. G. dan N.B. Nichols, 1942, *Optimum Setting for Automatic Kontrollers*, Tans.ASME.



LAMPIRAN I

FOTO ALAT



Gambar Penampakan Alat Fermentasi Kakao



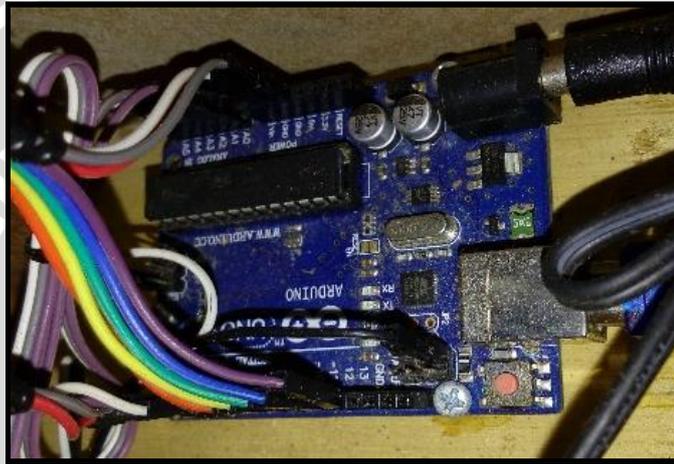
Gambar Catu Daya 32 volt dan 12 volt



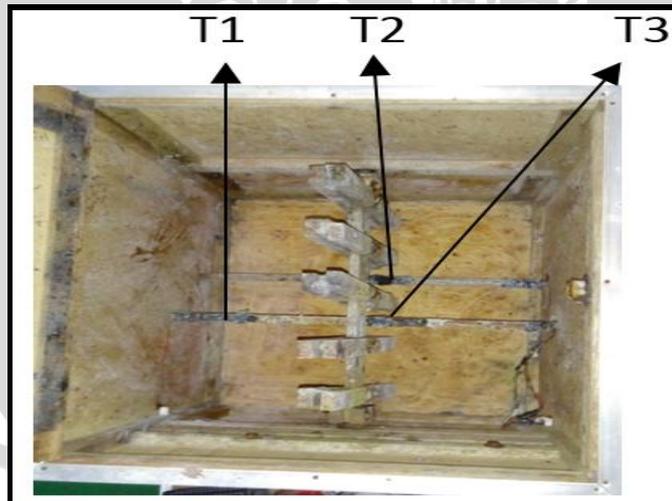
Gambar Driver Motor DC



Gambar LCD 16x2



Gambar Mikrokontroler Arduino Uno



Gambar Sensor Suhu LM35



Gambar Push Button



Gambar Sensor Rotary Encoder

LAMPIRAN II

LISTING PROGRAM ARDUINO PENGUJIAN RPM DENGAN PWM

```
Pin 2    >>> Rotary Encoder Input
Pin 3    >>> Fan And Relay For DRiver Motor
Pin 4    >>> Manual Run Or Stop Motor
Pin 5    >>> PWM Output
A0       >>> Analog Temperature Input 1
A1       >>> Analog Temperature Input 2
A2       >>> Analog Temperature Input 3
Pin 8    >>> LCD RS
Pin 9    >>> LCD E
PIN 10,11,12,13 >>> LCD Input(D4,D5,D6,D7)
*/
#include <LiquidCrystal.h>
void baca_suhu(void);
void tampilkan_suhu(void);
void motorCommand(void);
void baca_putaran(void);
int sensorPin1    = A0;
int sensorPin2    = A1;
int sensorPin3    = A2;
int fanAndRelay   = 3 ;
```

```
int motorStartCommand = 4 ;
int pwmOut           = 5 ;
LiquidCrystal lcd(8,9,10,11,12,13);
int temperature1 = 0;
int temperature2 = 0;
int temperature3 = 0;
int datasamples1[8];
int datasamples2[8];
int datasamples3[8];
int MaxT = 50;
int i,b;
long a;
int pwm;
int sp;
double actual,error,lastError,dError,errorSum,dt,kp,ki,kd,out;
unsigned long time,lastTime;
volatile float periode = 0;
volatile float lastPeriode = 0;
volatile int rpm;
volatile int e;
long previousMillis = 0;
long interval = 10000;
```



```
long previousMillis2 = 0;

long interval2 = 1000;

void setup() {

  pinMode(fanAndRelay, OUTPUT);

  pinMode(pwmOut, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(motorStartCommand, INPUT);

  sp = 60;

  kp = 0.09;//0.09;

  ki = 9;//0.0005;

  kd = 9;//0.05;

  digitalWrite(fanAndRelay, HIGH);

  attachInterrupt(0, encoder_interrupt, FALLING);

  Serial.begin(9600);

}

void loop(){

  unsigned long currentMillis = millis();

  if(currentMillis - previousMillis > interval) {

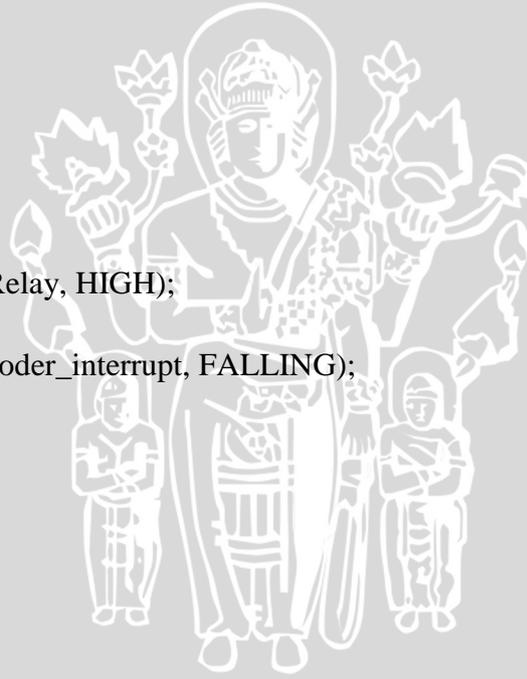
    previousMillis = currentMillis;

    pwm=(pwm+5)%256;

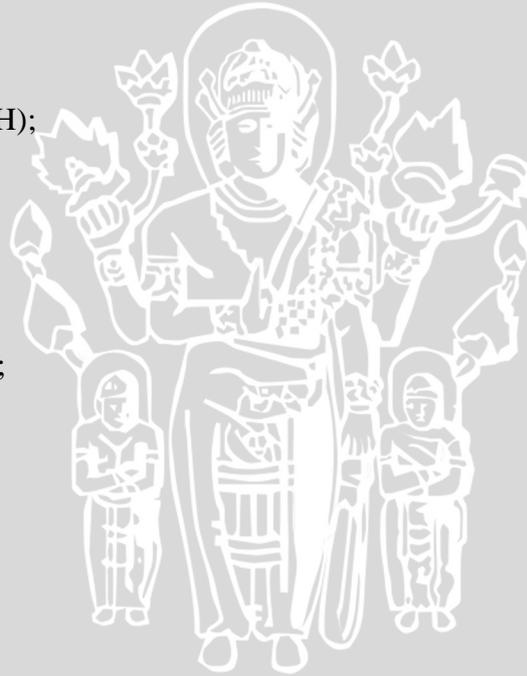
    analogWrite(pwmOut,pwm);

  }

}
```



```
if(currentMillis - previousMillis2 > interval2) {  
  previousMillis2 = currentMillis;  
  Serial.print("PWM= ");  
  Serial.print("\t");  
  Serial.print(pwm);  
  Serial.print("\t");  
  Serial.print(" RPM= ");  
  Serial.print("\t");  
  Serial.println(rpm);  
  digitalWrite(9, HIGH);  
  lcd.begin(16, 2);  
  lcd.setCursor(0, 0);  
  lcd.print("PWM=");  
  lcd.print(pwm);  
  lcd.print(" itr=");  
  lcd.print(e);  
  lcd.setCursor(0, 1);  
  lcd.print("RPM=");  
  lcd.print(rpm);  
  lcd.print(" T=");  
  lcd.print (periode);  
}
```



```
}  
void encoder_interrupt(){  
    if (e==7) {  
        e=0;  
        periode = (micros() - lastPeriode)/1000000;  
        rpm= 60/periode;  
    }  
    if (e==0) {  
        lastPeriode = micros();  
    }  
    e++;  
}
```



LAMPIRAN III

LISTING PROGRAM ARDUINO PENGUJIAN KESELURUHAN

```
Pin 2    >>> Rotary Encoder Input
Pin 3    >>> Fan And Relay For DRiver Motor
Pin 4    >>> Manual Run Or Stop Motor
Pin 5    >>> PWM Output
A0       >>> Analog Temperature Input 1
A1       >>> Analog Temperature Input 2
A2       >>> Analog Temperature Input 3
Pin 8    >>> LCD RS
Pin 9    >>> LCD E
PIN 10,11,12,13 >>> LCD Input(D4,D5,D6,D7)
*/
#include <LiquidCrystal.h>
void baca_suhu(void);
void tampilkan_suhu(void);
void motorCommand(void);
void baca_putaran(void);
int sensorPin1    = A0;
int sensorPin2    = A1;
int sensorPin3    = A2;
int fanAndRelay   = 3 ;
```

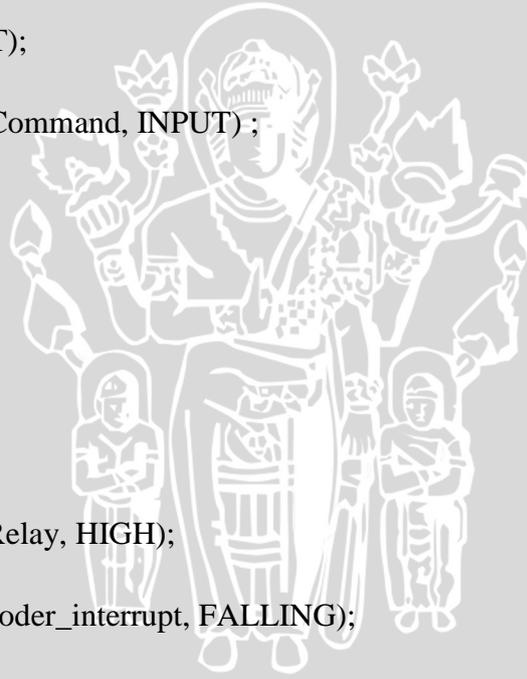
```
int motorStartCommand = 4 ;  
int pwmOut = 5 ;  
LiquidCrystal lcd(8,9,10,11,12,13);  
int temperature1 = 0;  
int temperature2 = 0;  
int temperature3 = 0;  
int datasamples1[8];  
int datasamples2[8];  
int datasamples3[8];  
int MaxT = 50;  
int i,b;  
long a;  
int pwm;  
int ratapwm;  
int sp;  
double actual,error,lastError,dError,errorSum,dt,kp,ki,kd,out;  
unsigned long time,lastTime;  
volatile float periode = 0;  
volatile float lastPeriode = 0;  
volatile int rpm;  
volatile int e;  
long previousMillis = 0;
```



```
long interval = 100;
long previousMillis2 = 0;
long interval2 = 100;
long previousMillis3 = 0;
long interval3 = 50000;
void setup() {
  pinMode(fanAndRelay, OUTPUT);
  pinMode(pwmOut, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(motorStartCommand, INPUT);

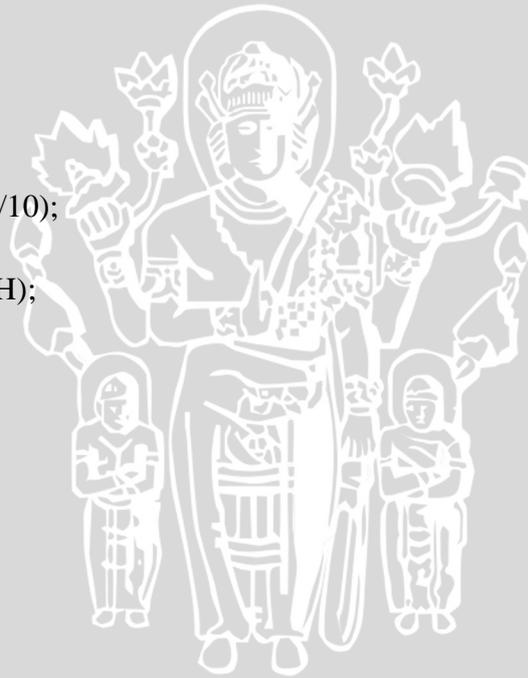
  sp = 60;
  kp = 0.6;
  ki = 0.324;
  kd = 0.2775;

  digitalWrite(fanAndRelay, HIGH);
  attachInterrupt(0, encoder_interrupt, FALLING);
  Serial.begin(9600);
}
void loop(){
  unsigned long currentMillis = millis();
  baca_suhu();
  tampilkan_suhu();
```



```
if(currentMillis - previousMillis3 > interval3) {  
    previousMillis3 = currentMillis;  
    kp=kp;  
    analogWrite(pwmOut,pwm);  
}  
baca_suhu();  
tampilkan_suhu();  
if(currentMillis - previousMillis > interval) {  
    previousMillis = currentMillis;  
    analogWrite(pwmOut,out);  
    hitung_PID();  
    sinyal_kontrol();  
    ratapwm=(ratapwm+pwm)/2;  
}  
if(currentMillis - previousMillis2 > interval2) {  
    previousMillis2 = currentMillis;  
    //Serial.print("PWM= " );  
    //Serial.print("\t");  
    Serial.print(kp);  
    Serial.print("\t");  
    //Serial.print(" RPM= ");  
    //Serial.print("\t ");
```

```
Serial.println(rpm/10);  
  
// Serial.print("\t");  
  
//Serial.print(" Err= ");  
  
//Serial.print("\t");  
  
// Serial.print("\t");  
  
//Serial.println(ratapwm);  
  
//Serial.print("\t");  
  
//Serial.print(" Out= ");  
  
//Serial.print("\t");  
  
Serial.print("\t");  
  
//Serial.println(error/10);  
  
digitalWrite(9, HIGH);  
  
lcd.begin(16, 2);  
  
lcd.setCursor(0, 0);  
  
lcd.print("Kp=");  
  
lcd.print(kp);  
  
lcd.print(" RPM=");  
  
lcd.print(rpm/10);  
  
lcd.setCursor(0, 1);  
  
lcd.print("ERR=");  
  
lcd.print(error/10);  
  
lcd.print(" ");
```



```
lcd.print (ratapwm);

rpm=0;
}
}

void encoder_interrupt(){
  if (e==7) {
    e=0;
    periode = (micros() - lastPeriode)/1000000;
    rpm= 60/periode;
  }

  if (e==0) {
    lastPeriode = micros();
  }
  e++;
}

void hitung_PID(void){
  actual=rpm;
  time=millis();
  dt=time-lastTime;
  error=sp*10-actual;
  errorSum = errorSum + error*dt;
  dError=error-lastError;
```



```
out= kp*error + ki*errorSum + kd*dError/dt;
lastTime=time;
lastError=error;
}

void sinyal_kontrol(void){
    if (out>=255){
        out=255;
    }
    if (out<=0){
        out=32;
    }
    pwm=out;
    //analogWrite(pwmOut,pwm);
}

void baca_suhu(void) {
    temperature1 = (5.0*analogRead(sensorPin1)*100.0)/1024.0;
    temperature2 = (5.0*analogRead(sensorPin2)*100.0)/1024.0;
    temperature3 = (5.0*analogRead(sensorPin2)*100.0)/1024.0;
}

void tampilkan_suhu(void) {
    digitalWrite(9, HIGH);
    lcd.begin(16, 2);
```



```
lcd.setCursor(0, 0);  
lcd.print("T1=");  
lcd.print(temperature3);  
lcd.print("C; T2=");  
lcd.print(temperature2);  
lcd.print("C");  
lcd.setCursor(0, 1);  
lcd.print("T3=");  
lcd.print(temperature3);  
lcd.print("C; ");  
}
```



LAMPIRAN IV

LISTING PROGRAM ARDUINO CARA KERJA ALAT

```
Pin 2    >>> Rotary Encoder Input
Pin 3    >>> Fan And Relay For DRiver Motor
Pin 4    >>> Manual Run Or Stop Motor
Pin 5    >>> PWM Output
A0       >>> Analog Temperatue Input 1
A1       >>> Analog Temperature Input 2
A2       >>> Analog Temperature Input 3
Pin 8    >>> LCD RS
Pin 9    >>> LCD E
PIN 10,11,12,13 >>> LCD Input(D4,D5,D6,D7)
*/
#include <LiquidCrystal.h>
void baca_suhu(void);
void tampilkan_suhu(void);
void motorCommand(void);
void baca_putaran(void);
int sensorPin1    = A0;
int sensorPin2    = A1;
int sensorPin3    = A2;
int fanAndRelay   = 3 ;
```

```
int motorStartCommand = 4 ;
int pwmOut = 5 ;
LiquidCrystal lcd(8,9,10,11,12,13);
int temperature1 = 0;
int temperature2 = 0;
int temperature3 = 0;
int datasamples1[8];
int datasamples2[8];
int datasamples3[8];
int MaxT = 50;
int i,b;
long a;
int pwm;
int sp;
int hold=0;
double actual,error,lastError,dError,errorSum,dt,kp,ki,kd,out;
unsigned long time,lastTime;
volatile float periode = 0;
volatile float lastPeriode = 0;
volatile int rpm;
long previousMillis = 0;
long interval = 180000; // 3 menit
```

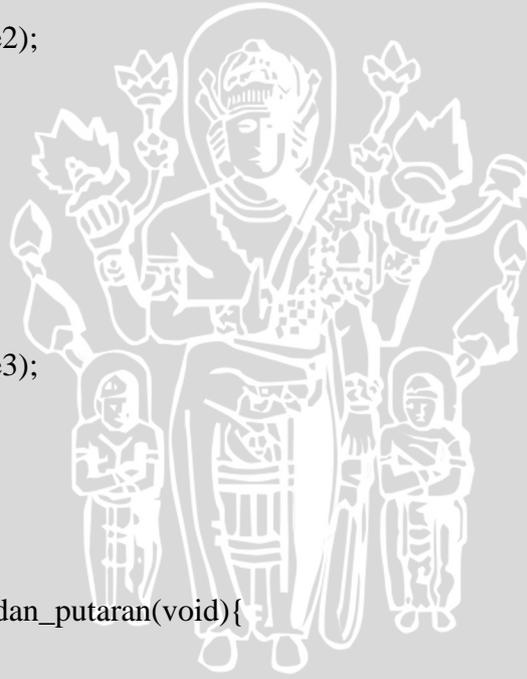


```
void setup() {  
  pinMode(fanAndRelay, OUTPUT);  
  pinMode(pwmOut, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(motorStartCommand, INPUT);  
  
  sp = 60;  
  kp = 0.6;  
  ki = 0.324;  
  kd = 0.2775;  
}  
  
void loop(){  
  baca_suhu();  
  motorCommand();  
}  
  
void encoder_interrupt(){  
  periode = (micros() - lastPeriode);  
  lastPeriode = micros();  
}  
  
void baca_suhu(void) {  
  temperature1 = (5.0*analogRead(sensorPin1)*100.0)/1024.0;  
  temperature2 = (5.0*analogRead(sensorPin2)*100.0)/1024.0;  
  temperature3 = (5.0*analogRead(sensorPin2)*100.0)/1024.0;
```

UNIVERSITAS BRAWIJAYA



```
}  
  
void tampilkan_suhu(void) {  
    digitalWrite(9, HIGH);  
  
    lcd.begin(16, 2);  
  
    lcd.setCursor(0, 0);  
  
    lcd.print("T1=");  
  
    lcd.print(temperature1);  
  
    lcd.print("C; T2=");  
  
    lcd.print(temperature2);  
  
    lcd.print("C");  
  
    lcd.setCursor(0, 1);  
  
    lcd.print("T3=");  
  
    lcd.print(temperature3);  
  
    lcd.print("C; ");  
}  
  
void tampilkan_suhu_dan_putaran(void){  
  
    a++;  
  
    if (a== 5000 && b==0){  
  
        digitalWrite(9, HIGH);  
  
        lcd.begin(16, 2);  
  
        lcd.setCursor(0, 0);  
  
        lcd.print("SP=");  
  
    }  
}
```



```
lcd.print(sp);
```

```
lcd.print(" PWM=");
```

```
lcd.print(pwm);
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Err=");
```

```
lcd.print(int (error));
```

```
lcd.print(" RPM=");
```

```
lcd.print(rpm);
```

```
a=0;
```

```
b=1;
```

```
}
```

```
if (a== 5000 && b==1){
```

```
tampilkan_suhu();
```

```
a=0;
```

```
b=0;
```

```
}
```

```
}
```

```
void motorCommand(void){
```

```
if (digitalRead(motorStartCommand)== HIGH){
```

```
digitalWrite(fanAndRelay, HIGH);
```

```
attachInterrupt(0, encoder_interrupt, FALLING);
```

```
baca_putaran();
```



```
tampilkan_suhu_dan_putaran();  
  
hitung_PID();  
  
sinyal_kontrol();  
  
}  
  
else {  
  
    if ((temperature1 < MaxT || temperature2 < MaxT || temperature3 < MaxT)  
        && hold==0 ) {  
  
        analogWrite(pwmOut,0);  
  
        digitalWrite(fanAndRelay, LOW);  
  
        detachInterrupt(0);  
  
        tampilkan_suhu();  
  
        delay(1000);  
  
    }  
  
}  
  
if ((temperature1 >= MaxT || temperature2 >= MaxT || temperature3 >= MaxT)  
    && hold==1 ) {  
  
    hold=1;  
  
    digitalWrite(fanAndRelay, HIGH);  
  
    attachInterrupt(0, encoder_interrupt, FALLING);  
  
    baca_putaran();  
  
    tampilkan_suhu_dan_putaran();  
  
    hitung_PID();  
  
    sinyal_kontrol();
```

```
unsigned long currentMillis = millis();

if(currentMillis - previousMillis > interval) {
  previousMillis = currentMillis;

  hold=0;
}
}

else {
  if (digitalRead(motorStartCommand)== LOW){

    analogWrite(pwmOut,0);

    digitalWrite(fanAndRelay, LOW);

    detachInterrupt(0);

    tampilkan_suhu();

    delay(1000);

  }
}

void baca_putaran (void){

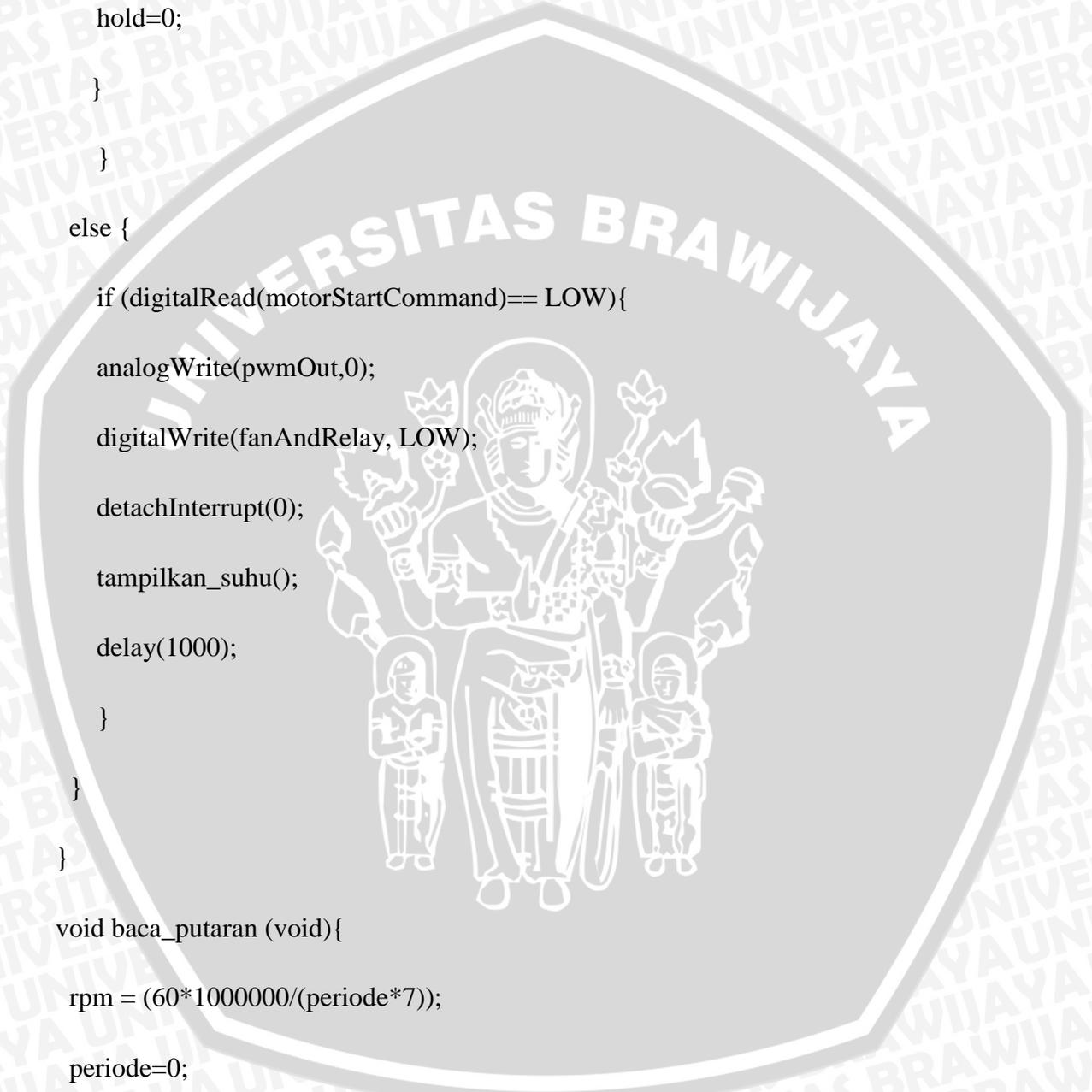
  rpm = (60*1000000/(periode*7));

  periode=0;

}

void hitung_PID(void){

  actual=rpm;
```



```
time=millis();  
dt=time-lastTime;  
error=sp-actual;  
errorSum = errorSum + error*dt;  
dError=error-lastError;  
out= kp*error + ki*errorSum + kd*dError/dt;  
lastTime=time;  
lastError=error;  
}  
void sinyal_kontrol(void){  
  if (out>=255){  
    out=255;  
  }  
  if (out<=0){  
    out=35;  
  }  
  pwm=out;  
  analogWrite(pwmOut,pwm);  
}
```

