



**RANCANG BANGUN ALAT PENGERING TEMBAKAU OTOMATIS  
DENGAN METODE PENGOLAHAN CITRA DIGITAL  
MENGUNAKAN KONTROL *HYBRID FUZZY-PID***

**Skripsi**

Oleh

**Fathor Rohman**

**NIM 131910201105**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2017**



**RANCANG BANGUN ALAT PENGERING TEMBAKAU OTOMATIS  
DENGAN METODE PENGOLAHAN CITRA DIGITAL  
MENGUNAKAN KONTROL *HYBRID FUZZY-PID***

**Skripsi**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Elektro (S1)  
dan mencapai gelar Sarjana Teknik

Oleh

**Fathor Rohman**

**NIM 131910201105**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2017**

## PERSEMBAHAN

Alhamdulillah, puji syukur kehadiran Allah SWT atas kasih setia-Nya yang telah melimpahkan segala rahmat yang tak ternilai, sehingga saya bisa menyelesaikan penelitian ini. Akhirnya, saya persembahkan skripsi ini kepada:

1. Allah SWT yang Maha Pengasih lagi Maha Penyayang.
2. Nabi besar Muhammad SAW, yang menjadi suri tauladan bagi seluruh umat.
3. Ke-dua orang saya yaitu Bapak Muhammad dan Ibu Murtini serta kakak tercinta Raudatuljannah dan Munawaroh.
4. Teman-teman INTEL 2013 dan ELKA13 yang telah menjadikan kehidupan dimasa kuliah terasa sangat indah.
5. Ex-pengurus Robotika dan anggota UKM Robotika. Devry, Dika (Kambing), Riski (Bima), Rokhim, Rizky, Wandu (Bocil), Mas Wawan dan kawan-kawan terimakasih atas kebersamaan, kepedulian serta ilmu yang telah diberikan.
6. Seluruh asisten Laboratorium Teknik Elektro Universitas Jember. Terima kasih telah menjadi lawan diskusi yang menyenangkan.
7. Guru-guruku sejak TK hingga Perguruan Tinggi yang terhormat, terima kasih telah memberikan ilmu dan mendidik dengan penuh kesabaran.
8. Serta seluruh rekan-rekan yang penulis kenal dan rekan-rekan yang membaca skripsi ini. Terima kasih untuk segalanya.

**MOTTO**

“Allah akan meninggikan orang-orang yang beriman diantara kamu dan yang berilmu pengetahuan beberapa derajat”

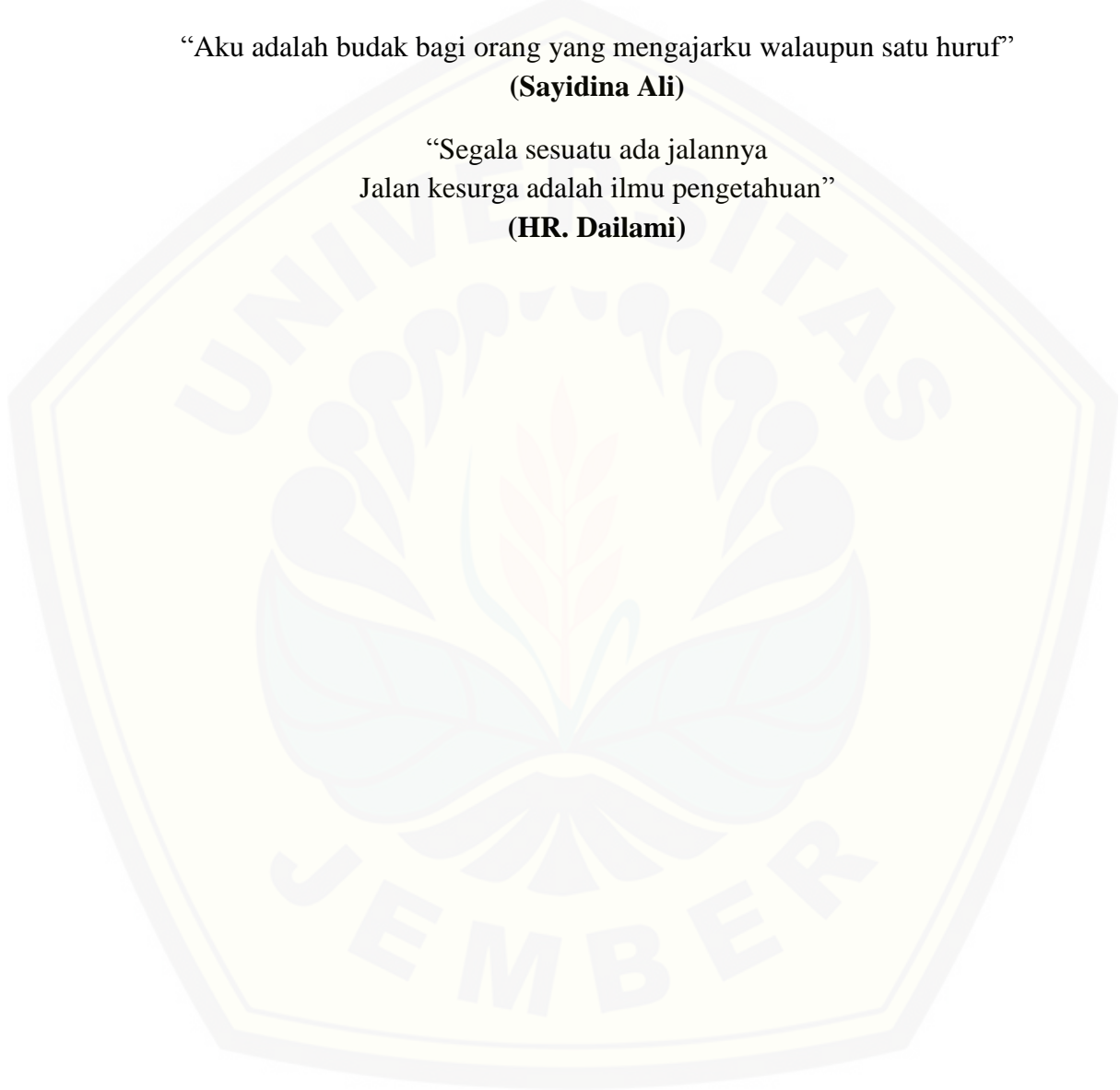
**(QS : Al-Mujaadilah : 11)**

“Aku adalah budak bagi orang yang mengajarku walaupun satu huruf”

**(Sayidina Ali)**

“Segala sesuatu ada jalannya  
Jalan kesurga adalah ilmu pengetahuan”

**(HR. Dailami)**



**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Fathor Rohman

NIM : 131910201105

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ” Rancang Bangun Alat Pengering Tembakau Otomatis Dengan Metode Pengolahan Citra Digital Menggunakan Kontrol *Hybrid Fuzzy-PID*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, September 2017

Yang menyatakan,

Fathor Rohman

NIM 131910201105

**SKRIPSI**

**RANCANG BANGUN ALAT PENERING TEMBAKAU OTOMATIS  
DENGAN METODE PENGOLAHAN CITRA DIGITAL  
MENGUNAKAN KONTROL *HYBRID FUZZY-PID***

Oleh

Fathor Rohman

NIM 131910201105

Pembimbing :

Dosen Pembimbing Utama : Khairul Anam, S.T., M.T., Ph.D

Dosen Pembimbing Anggota : Wahyu Muldayani, S.T., M.T.

**PENGESAHAN**

Skripsi berjudul “**Rancang Bangun Alat Pengering Tembakau Otomatis dengan Metode Pengolahan Citra Digital Menggunakan Kontrol *Hybrid Fuzzy-PID***” karya Fathor Rohman telah diuji dan disahkan pada :

Hari : Selasa

Tanggal : 12 September 2017

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Anggota I,

Khairul Anam, S.T., M.T., Ph.D  
NIP 197804052005011002

Wahyu Muldayani, S.T., M.T.  
NRP 760016799

Anggota II,

Anggota III,

Widya Cahyadi S.T., M.T.  
NIP 198511102014041001

Ike Fibriani S.T., M.T.  
NIP 198002072015042001

Mengesahkan  
Dekan,

Dr. Ir. Entin Hidayah, M.U.M.  
NIP 196612151995032001

## RINGKASAN

**Rancang Bangun Alat Pengering Tembakau Otomatis dengan Metode Pengolahan Citra Digital Menggunakan Kontrol *Hybrid Fuzzy-PID***; Fathor Rohman; 131910201105; 2017; 79 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Daun tembakau merupakan bahan baku utama dalam pembuatan rokok. Terdapat bermacam-macam pengolahan pasca panen daun tembakau untuk dijadikan bahan baku rokok. Salah satunya yaitu dengan proses pengeringan *flue-cured*. Proses pengeringan *flue-cured* merupakan proses pengeringan daun tembakau menggunakan aliran udara panas dalam oven (*curing-barn*). Dalam proses pengeringan tembakau menggunakan *flue-cured*, terdapat beberapa fase pengeaturan suhu yang disesuaikan dengan tingkat kekeringan daun tembakau.

Proses pengeringan dengan *flue-cured* umumnya masih menggunakan bahan baku kompor gas atau batu bara untuk mengatur suhu dalam oven. Proses pengeringan dengan bahan baku kompor gas atau batu bara relatif sulit untuk mengatur kestabilan suhu dalam oven, serta membutuhkan pengawasan yang ketat dalam mengatur suhu oven sesuai perubahan warna daun tembakau agar menghasilkan daun yang di inginkan.

Dengan kemajuan teknologi, proses pengaturan suhu dan pengawasan yang ekstra tidak perlu dilakukan oleh peteni apabila menggunakan alat pengering tembakau otomatis. Untuk mendeteksi tingkat kekeringan menggunakan pengolahan citra digital dan untuk mengatur kestabilan suhu oven menggunakan *element heater* yang dikontrol menggunakan metode *hybrid fuzzy-PID*.

Dalam pengujian kestabilan suhu menggunakan kontrol PID hanya menggunakan satu *setpoint* yaitu 40°C. Pengujian kontrol suhu dilakukan beberapa tahap dengan cara *trial and error* diantaranya kontrol P, kontrol PI kontrol PD dan Kontrol PID. Selain dengan cara *trial and error* juga dilakukan pengujian PID *Ziegler-Nichols* tipe 2 yang kemudian dibandingkan dengan kontrol PID *trial and*



*error*. Hasil pengujian kontrol PID secara *trial and error* lebih baik dibandingkan dengan kontrol PID *Ziegler-Nichols* tipe 2.

Setelah melakukan pengujian kontrol PID dilakukan pengujian kontrol *Fuzzy-PID*. Kontrol *fuzzy* berfungsi untuk *men-tuning* otomatis konstantan  $K_p$ ,  $K_i$  dan  $K_d$  pada kontrol PID ketika kontrol mengalami perubahan *setpoint* diantaranya 32°C, 38°C, 40°C, 42°C, 49°C dan 60°C. Berubah *setpoint* didasarkan pada tingkat kekeingan daun tembakau.

Alat pengering tembakau ini terintegrasi dengan sebuah *wabcam* dengan *interface* USB yang dapat mendeteksi warna yang berkerja secara *real time*. Pemanfaat *webcame* ini dapat dikembangkan sehingga dapat digunakan sebagai sensor untuk memonitoring perubahan sebuah benda melalui beberapa tahap pengolahan citra.

Pada proses pengeringan tembakau akan menghasilkan perubahan warna dari daun tembakau. Untuk mendeteksi perubahan warna daun tembakau dilakukan menggunakan metode pengolahan citra digital. Pada pengolahan citra digital berfungsi mendeteksi warna tertentu dari daun tembakau. Model pengolahan ini merupakan pengolahan warna dari RGB, *grayscale* sampai menjadi citra biner. Pada proses akhir yang bertujuan untuk memberikan informasi untuk menghitung jumlah pixel putih pada citra biner yang merepresentasikan perubahan warna tertentu yang ingin dideteksi. Karena dalam proses pengeringan tembakau terjadi perubahan warna maka jumlah pixel warna putih akan berubah, perubahan jumlah pixel ini akan dikirim pada mikrokontroller yang nantinya diolah untuk dijadikan perubahan *setpoint* untuk kontrol *hybrid fuzzy-PID*.

## PRAKATA

Puji syukur kehadiran Allah SWT yang maha kuasa atas segalanya, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Alat Pengering Tembakau Otomatis dengan Metode Pengolahan Citra Digital Menggunakan Kontrol *Hybrid Fuzzy-PID*”. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak yang turut memberikan bantuan berupa motivasi, inspirasi, bimbingan, doa, fasilitas dan dukungan lainnya yang membantu memperlancar pengerjaan skripsi ini.

Untuk itu penulis mengucapkan terimakasih kepada.

1. Allah SWT yang telah melimpahkan rezeki, rahmat, hidayah dan karunia serta kasih sayang-Nya sehingga penulis mampu menyelesaikan skripsi ini.
2. Nabi besar Muhammad SAW, yang telah menjadi suri tauladan bagi seluruh umat.
3. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku Dekan Fakultas Teknik Universitas Jember
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember;
5. Bapak Khairul Anam, S.T., M.T., Ph.D, Bapak Wahyu Muldayani, S.T., M.T. dan Bapak Agung Prawira Negara S.T., M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini;
6. Bapak Widya Cahyadi S.T., M.T. , Ibu Ike Fibriani S.T., M.T. dan Bapak Bambang Supeno S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
7. Keluarga besar INTEL’UJ 2013, terimakasih telah memberikan arti kekeluargaan yang luar biasa.
8. Keluarga besar Civitas Akademia Jurusan Teknik Elektro Universitas Jember. Serta semua pihak yang tidak dapat disebutkan satu per satu, yang telah mendukung dalam penyelesaian skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 4 Agustus 2017

Penulis



DAFTAR ISI

HALAMAN SAMPUL .....	i
HALAMAN JUDUL .....	ii
PERSEMBAHAN .....	iii
MOTTO .....	iv
PERNYATAN.....	v
HALAMAN PEMBIMBING .....	vi
PENGESAHAN .....	vii
RINGKASAN .....	viii
PRAKATA .....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
DAFTAR LAMPIRAN .....	xix
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.4 Tujuan.....	3
1.5 Batsan Maslah .....	4
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Tembakau .....	5
2.2 <i>Heating Element</i> .....	10
2.3 Sensor DHT22 .....	11
2.4 <i>Fun/Kipas Aksial</i> .....	14
2.5 Kamera .....	15
2.6 Pengolahan Citra .....	16
2.7 Arduino Mega 2560 .....	18
2.8 Sistem Kendali PID ( <i>Proportional Integral Derivative</i> ) .....	19
2.8.1 Cara Mentuning Manual Parameter PID .....	20
2.8.2 Metoda Ziegler-Nichols .....	20

<b>2.9 Kontrol Fuzzy Logic .....</b>	<b>23</b>
2.9.1 Teori Himpunan Logika Fuzzy .....	23
2.9.2 Fungsi Kanggotaan .....	23
2.9.3 Oprerator Dasar Zadeh Untuk Operasi Himpunan Fuzzy .	24
2.9.4 Fuzzyfikasi .....	24
2.9.5 Penentuan <i>Rule Base</i> .....	24
2.9.6 Defuzzyfikasi .....	24
<b>2.10 Hybrid Fuzzy-PID .....</b>	<b>25</b>
<b>BAB 3. METODOLOGI PENELITIAN .....</b>	<b>26</b>
<b>3.1 Waktu dan Tempat .....</b>	<b>26</b>
3.1.1 Tempat .....	26
3.1.2 Waktu .....	26
<b>3.2 Tahap Penelitian .....</b>	<b>27</b>
<b>3.3 Perancangan Perangkat Keras .....</b>	<b>29</b>
3.3.1 Alat dan Bahan .....	29
3.3.2 Desain Alat .....	30
3.3.3 Rancangan Bentuk dan Cara Kerja Alat .....	32
3.3.4 Desain Elektrik .....	33
<b>3.4 Perancangan Perangkat Lunak .....</b>	<b>37</b>
3.4.1 Sistem Pengolahan Citra .....	37
3.4.2 Sistem Kontrl PID .....	45
3.4.3 Sistem Fuzzy Logic .....	46
<b>3.5 Flowchart .....</b>	<b>55</b>
<b>3.6 Metode Pengujian dan Pengumpulan Data .....</b>	<b>56</b>
<b>3.7 Metode Analisa Data .....</b>	<b>56</b>
<b>BAB 4. HASIL DAN PEMBAHASAN.....</b>	<b>58</b>
<b>4.1 Pengujian Sensor Suhu .....</b>	<b>59</b>
<b>4.2 Pengujian Sensor Kamera.....</b>	<b>63</b>
<b>4.3 Pengujian Kontrol PID dengan <i>Tuning Manual</i>.....</b>	<b>66</b>
4.3.1 Pengujian Kontrol P .....	67
4.3.2 Pengujian Kontrol PI .....	68

4.3.3 Pengujian Kontrol PD .....	69
4.3.4 Pengujian Kontrol PID .....	70
<b>4.4 Pengujian Kontrol PID Ziegler-Nichols tipe 2 (closed loop)...</b>	<b>71</b>
<b>4.5 Pengujian Hybrid Fuzzy-PID.....</b>	<b>74</b>
<b>4.6 Sistem Pengolahan Citra Digital .....</b>	<b>75</b>
4.6.1 Pengujian Warna Hijau dengan Satu <i>Setpoint</i> Suhu.....	76
4.6.2 Pengujian Warna Hijau dengan Dua <i>Setpoint</i> Suhu .....	77
4.6.3 Pengujian Warna Hijau dengan Empat <i>Setpoint</i> Suhu.....	78
4.6.4 Pengujian Warna Kuning dengan Lima <i>Setpoint</i> Suhu .....	80
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>	<b>83</b>
5.1 Kesimpulan .....	83
5.2 Saran .....	83
<b>DAFTAR PUSTAKA .....</b>	<b>84</b>
<b>LAMPIRAN .....</b>	<b>86</b>

**DAFTAR GAMBAR**

Gambar 2.1 Perkebunan Tembakau di Indonesia .....	6
Gambar 2.2 Bentuk-Bentuk <i>Tubular Heater</i> .....	10
Gambar 2.3 Sensor DHT22 .....	11
Gambar 2.4 Diagram blok DHT22 .....	13
Gambar 2.5 Kipas Aksial .....	14
Gambar 2.6 A4Tech PK-710G .....	15
Gambar 2.7 Nilai Warna RGB dalam Hexadesimal .....	16
Gambar 2.8 Komposisi Warna RGB .....	16
Gambar 2.9 <i>Board</i> Arduino Mega 2560 .....	18
Gambar 2.10 Blok Diagram Pengendali PID .....	19
Gambar 2.11 Sistem Diberi <i>Input Step</i> .....	20
Gambar 2.12 Proses Desain Penentuan Parameter L dan T .....	21
Gambar 2.13 Sistem <i>closed loop</i> dengan Menggunakan Kp Saja .....	21
Gambar 2.14 Proses Mendesain Menentukan Parameter Pcr .....	22
Gambar 2.15 <i>Defuzzyfikasi</i> . .....	25
Gambar 2.16 <i>Hybrid Fuzzy-PID</i> .....	25
Gambar 3.1 <i>Flowchart</i> Penelitian .....	27
Gambar 3.2 Desain Oven Tembakau .....	30
Gambar 3.3 Diagram Blok Perangkat Keras .....	31
Gambar 3.4 Oven Tembakau .....	31
Gambar 3.5 Diagram alur alat pengering tembakau otomatis .....	32
Gambar 3.6 Rangkain DHT22 dengan Arduino .....	34
Gambar 3.7 Rangkain <i>Dimmer</i> .....	34
Gambar 3.8 Rangkain <i>Relay Dimmer</i> .....	35
Gambar 3.9 Kipas angin motor DC .....	36
Gambar 3.10 Rangkain Driver Motor .....	36
Gambar 3.11 <i>Flowchart</i> Pengolahan Citra Digital .....	37
Gambar 3.12 Warna RGB daun tembakau.....	38

Gambar 3.13 a) Histogram Red, b) Histogram Green dan c) Hitogram Blue..	38
Gambar 3.14 Nilai Mean dan Standart deviasi warna.....	40
Gambar 3.15 Perubahan citra RGB ke <i>Grayscale</i> .....	41
Gambar 3.16 Pengambilan <i>sample</i> pixel warna hijau dengan nilai R=143, G=184 dan B=152 pada koordinat $I_{rgb}(147,113)$ .....	42
Gambar 3.17 a) Citra <i>grayscale</i> dan b) Citra biner .....	44
Gambar 3.18 <i>Flowchart</i> Program Kontrol PID .....	45
Gambar 3.19 <i>Membership function input error</i> .....	49
Gambar 3.20 <i>Membership function input delta error</i> .....	50
Gambar 3.21 <i>Membership function output Kp</i> .....	51
Gambar 3.22 <i>Membership function output Ki</i> .....	51
Gambar 3.23 <i>Membership function output Kd</i> .....	51
Gambar 3.24 <i>Flowchat</i> Kendali <i>Fuzzy-PID</i> Suhu .....	55
Gambar 3.25 Diagram Blok Fase Pengeringan Sistem.....	57
Gambar 4.1 Grafik Kelineiaran Hasil Pengujian Sensor Suhu .....	60
Gambar 4.1 Grafik Respon Keluaran Kontrol P .....	67
Gambar 4.2 Grafik Respon Keluaran Kontrol PI .....	68
Gambar 4.3 Grafik Respon Keluaran Kontrol PD .....	69
Gambar 4.4 Grafik Respon Keluaran Kontrol PID .....	70
Gambar 4.5 Proses menentukan parameter Pcr .....	72
Gambar 4.6 Grafik respon kontrol PID <i>Ziegler-Nichols</i> tipe 2 .....	72
Gambar 4.7Grafik Respon Kendali <i>Hybrid</i> .....	74



**DAFTAR TABEL**

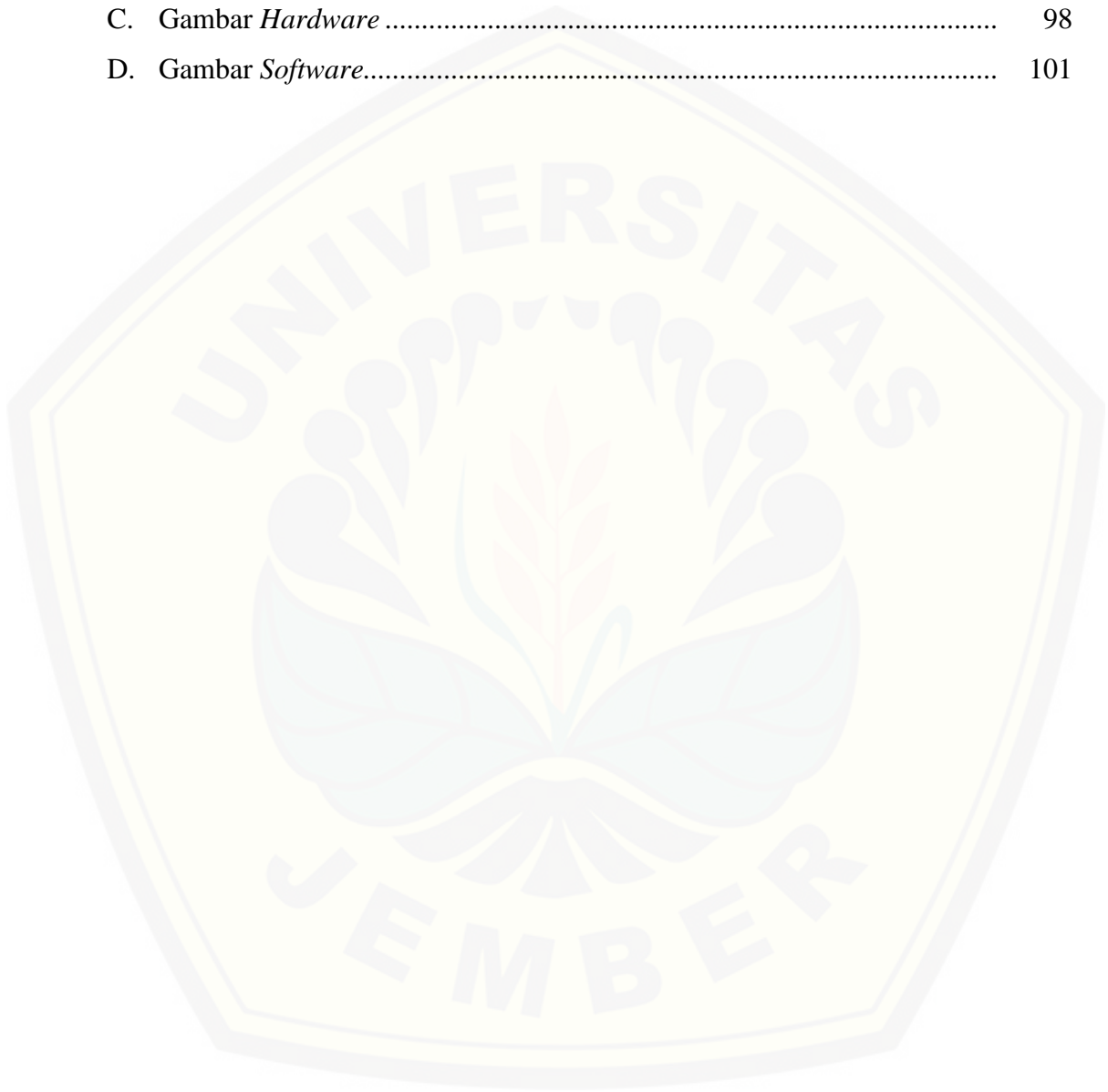
Tabel 2.1 Proses <i>Flue-Curing</i> .....	9
Tabel 2.2 Spesifikasi sensor DHT22 .....	13
Tabel 2.3 Spesifikasi kamera Havit HV-V622 .....	15
Tabel 2.4 Parameter PID .....	20
Tabel 2.5 Parameter PID untuk ZN tipe 1 .....	21
Tabel 2.6 Parameter PID Untuk ZN Tipe 2 .....	22
Tabel 3.1 <i>Timieline</i> Penelitian .....	26
Tabel 3.2 Daftar Peralatan yang Digunakan .....	29
Tabel 3.3 Indeks warna RGB daun tembakau.....	39
Tabel 3.4 <i>Rule output</i> Kp .....	53
Tabel 3.5 <i>Rule output</i> Ki .....	53
Tabel 3.6 <i>Rule output</i> Kd .....	53
Tabel 4.1 Perbandingan Sensor Suhu DHT22 dengan Digital Thermometer	60
Tabel 4.2 Perbandingan Hasil Kalibrasi Sensor Suhu Dht22 Dengan Digital Thermometer .....	61
Tabel 4.3 Nilai Aktual Terkoreksi Suhu Menggunakan Digital Termometer	62
Tabel 4.4 Nilai Aktual Terkoreksi Suhu Menggunakan Sensor DHT22 .....	62
Tabel 4.5 Perbandingan jumlah pixel saat mengalami perubahan tegangan sumber dengan tegangan awal 12,2volt dan jumlah pixel $\pm$ 47000 pixel ...	63
Tabel 4.6 Pengujian Citra Warna Kuning .....	64
Tabel 4.7 Pengujian Citra Warna Hijau .....	65
Tabel 4.8 Tanggapan Suhu dengan Kontrol P .....	67
Tabel 4.9 Tanggapan Suhu dengan Kontrol PI .....	68
Tabel 4.10 Tanggapan Suhu dengan Kontrol PD .....	69
Tabel 4.11 Tanggapan Suhu dengan Kontrol PID .....	70
Tabel 4.12 Perbandingan kontrol PID <i>tuning</i> manual dan <i>Ziegler-Nichols</i> ...	73
Tabel 4.13 Citra digital oven tembakau 1 tahap pengeringan .....	76
Tabel 4.14 Citra digital oven tembakau 2 tahap pengeringan .....	77
Tabel 4.15 Jumlah pixel warna kuning sebagai <i>setpoint</i> suhu .....	78

Tabel 4.16 Citra digital oven tembakau 4 tahap pengeringan dengan warna kuning sebagai <i>setpoint</i> .....	79
Tabel 4.17 Jumlah pixel warna hijau sebagai <i>setpoint</i> suhu .....	80
Tabel 4.18 Citra digital oven tembakau dengan 5 tahap pengeringan dengan warna kuning sebagai <i>setpoint</i> suhu .....	81



**DAFTAR LAMPIRAN**

A. <i>Listing</i> Program Arduino Mega 2560 .....	86
B. <i>Listing</i> Program VB .....	95
C. Gambar <i>Hardware</i> .....	98
D. Gambar <i>Software</i> .....	101



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Tembakau adalah tanaman yang digunakan sebagai bahan baku utama pembuatan rokok. Sehingga pemerintah telah menetapkan bahwa industri rokok adalah sebagai salah satu industri prioritas nasional (Anonim, 2010a). Di Indonesia merupakan penghasil tembakau yang cukup baik. Berbagai macam tembakau komersial diantaranya dihasilkan di daerah-daerah tertentu. Waktu tanam, lokasi penanaman, kultivar, dan pengolahan pascapanen sangat menentukan kualitas dari tembakau.

Berdasarkan cara pengolahan pascapanen, dikenal tembakau kering-angin (*air-cured*), kering-asap (*fire-cured*), kering-panas (*flue-cured*), dan kering-jemur (*sun-cured*). Salah satu contoh tembakau menggunakan cara pengolahan *flue-cured* yaitu tembakau virginia. Pengeringan tembakau virginia menggunakan aliran udara panas dalam proses pengovenan (*curing-barn*) dan menghasilkan kerosok yang berwarna lemon atau *orange* (Peedin 1999).

Pengeringan dapat dilakukan dengan penjemuran yang memanfaatkan sinar matahari atau dengan cara buatan. Pengeringan buatan mempunyai kelebihan diantaranya mengatasi pengaruh cuaca dengan kelembaban nisbi yang tinggi sepanjang tahun dan juga untuk meningkatkan mutu hasil pengeringan.

Prosesnya pengeringan tembakau dibagi menjadi dua yaitu pengeringan secara alami dan secara mekanis atau buatan. Darmadjati et al. (1992) menyatakan bahwa proses pengeringan dengan sumber panas sinar matahari mempunyai potensi untuk dikembangkan. Karena energi sinar matahari berfluktuasi terhadap musim sehingga tingkat adopsinya masih terhambat. Dengan penggunaan *system in store* yaitu dengan menggunakan penukar panas seperti yang diungkapkan oleh Srzednicki et al. (2001) dapat menutupi kelemahan pengeringan dengan energi surya. Sistem tersebut lebih menguntungkan karena menghasilkan kualitas yang bagus namun dengan biaya yang agak lebih mahal.

Penelitian yang pernah dilakukan mengenai pengeringan tembakau yaitu dilakukan oleh Tirtosastro et al. (2003). Penelitian ini bertujuan untuk merekayasa dan menguji dengan kombinasi kolektor surya bagian atap oven dan kompor LPG untuk pengovenan tembakau virginia. Penelitian ini didasarkan oleh kebiasaan cara pengovenan yang berlaku dan hasil penelitian dievaluasi berdasar aspek teknis dan aspek ekonomi. Lokasi percobaan di emplasemen P.T. Sadhana Arif Nusa, Lombok Timur, Nusa Tenggara Barat. Penggunaan bahan bakar LPG dan kolektor surya sebagai sumber energi dapat memenuhi target suhu yang diinginkan (30-70 °C) pada fase-fase pengovenan.

Sementara yang pernah dilakukan oleh Ichsani et al. (2001) mengenai proses pengeringan tembakau. Hasil penelitiannya menyatakan bahwa dengan menggunakan proses pengeringan sistem energi panas gabungan antara energi surya dan penukar panas lebih menguntungkan bila dibandingkan dengan pengeringan dengan sinar matahari saja dan pengeringan dengan penukar kalor saja. Letak keunggulannya adalah tidak tergantung pada waktu dan pada biaya yang relatif murah.

Penelitian ini menjadi penting artinya terkait dengan kebutuhan riil petani tembakau. Cara pengolahan tembakau dengan menggunakan metode aliran udara panas di dalam oven saat ini masih menggunakan cara konvensional yaitu menggunakan kompor. Pada oven konvensional pengaturan suhu dilakukan secara manual. Pengawasan yang ketat dilakukan agar hasil pengovenan tidak mengalami kerusakan.

Kelemahan menggunakan cara konvensional yaitu proses pengovenan masih menggunakan bahan bakar minyak tanah, LPG atau kayu bakar. Apalagi akhir-akhir ini terjadi pembatasan minyak tanah bersubsidi dan sulit memperoleh kayu bakar, pemerintah mengalihkan ke bahan bakar batubara. Bahan bakar batubara dan kayu harus dengan pembakaran tidak langsung, karena udara panas yang dihasilkan selain kotor juga rawan kebakaran. Selain itu waktu proses pengovenan relatif lama dan perlu pengawasan yang ekstra.

Melihat permasalahan di atas, kiranya amat mendesak untuk mendesain satu teknologi pengeringan oven (*curing-barn*) tembakau secara otomatis. Dengan menggunakan pengering oven otomatis diharapkan dapat memudahkan petani dalam proses pengeringan. Dalam penelitian ini, alat pengering oven otomatis dapat mengatur suhu secara otomatis. Kelebihan menggunakan oven otomatis ini yaitu alat pemanas tidak lagi menggunakan kompor, tetapi menggunakan element pemanas (*heater*) yang dikontrol secara otomatis. Selain itu alat oven otomatis ini menggunakan sensor kamera untuk mendeteksi tingkat kekeringan dari tembakau sehingga tidak lagi memerlukan bantuan manusia dalam memantau secara langsung tembakau kering atau tidak, kapan suhu dinaikkan dan kapan ventilasi dibuka. Dibandingkan dengan cara konvensional, proses pengeringan lebih singkat karena tidak ada kelalain dalam proses monitoring kapan suhu harus dinaikkan dan kapan suhu harus diturunkan.

### 1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini antara lain:

1. Bagaimana mendesain alat pengering tembakau untuk mempermudah petani?
2. Bagaimana merancang kontrol alat pengering yang stabil?
3. Bagaimana cara untuk membedakan tembakau sudah kering atau belum dalam proses pengeringan tembakau dalam oven (*curing-barn*)?

### 1.3 Tujuan

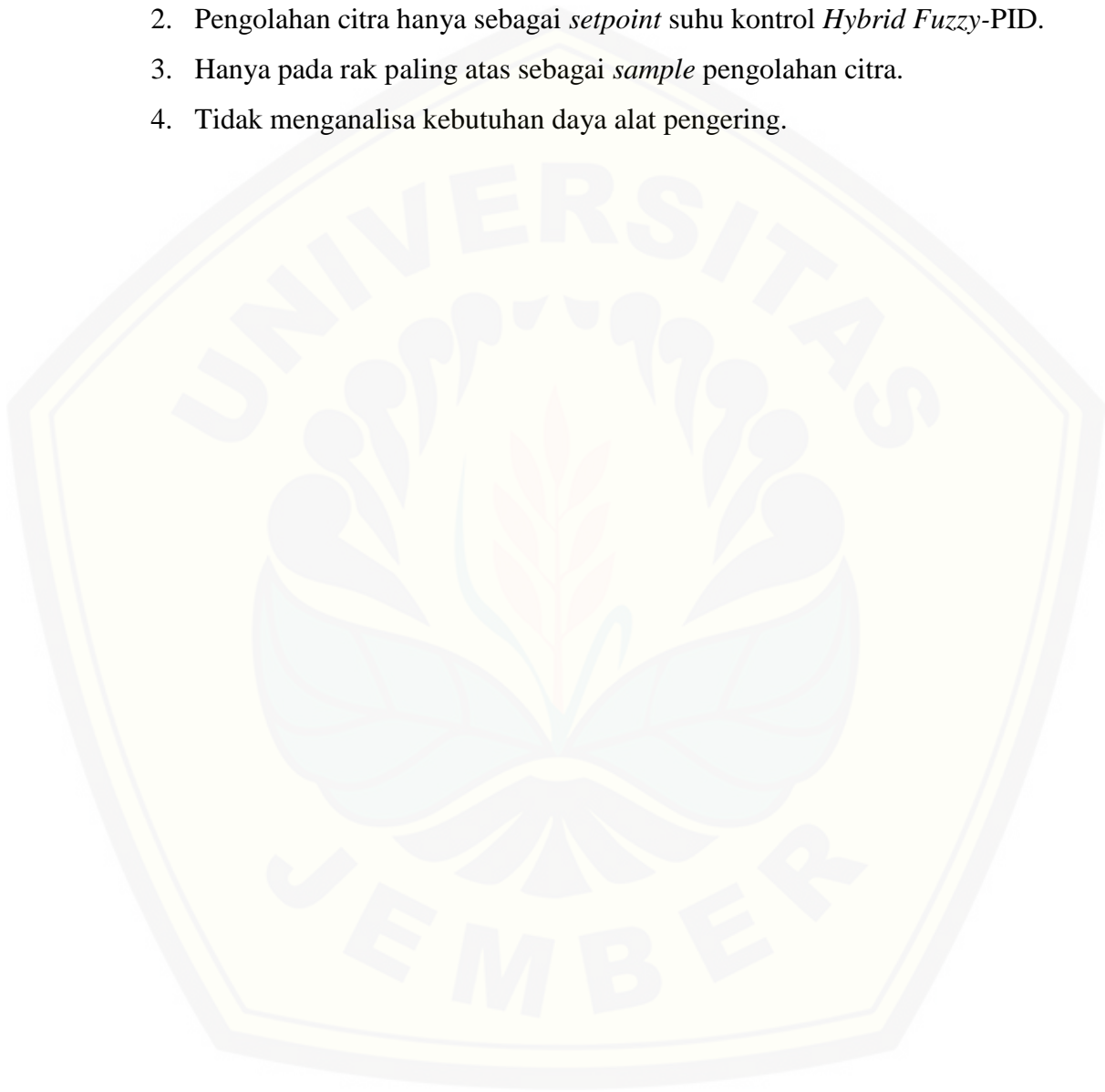
Tujuan dari penelitian ini antara lain:

1. Menghasilkan alat pengering tembakau otomatis.
2. Kontrol kestabilan suhu menggunakan *Fuzzy* dan PID.
3. Menerapkan pengolahan citra pada alat pengering untuk membedakan tingkat kekeringan daun tembakau.

#### 1.4 Batasan Masalah

Batasan Masalah dari penelitian ini antara lain:

1. Sitem kendali menggunakan kontrol *Hybrid Fuzzy*-PID untuk mengatur suhu.
2. Pengolahan citra hanya sebagai *setpoint* suhu kontrol *Hybrid Fuzzy*-PID.
3. Hanya pada rak paling atas sebagai *sample* pengolahan citra.
4. Tidak menganalisa kebutuhan daya alat pengering.



## BAB 2. TINJAUAN PUSTAKA

### 2.1 Tembakau

Tembakau adalah tanaman yang daunnya dimanfaatkan untuk bahan baku rokok atau untuk dihisap lewat hidung atau tembakau sedotan (*snuff*) atau tembakau kunyah (*chewing*). Daun tembakau terdapat kandungan nikotin yaitu, suatu zat aditif, dan juga sebagai bahan dasar untuk beberapa jenis insektisida. Di Indonesia, tembakau telah dikenal sebagai tanaman obat ataupun bahan halusinogen sejak 400 tahun yang lalu. (Balitas, 1994)

Tembakau merupakan hasil tanaman *Nicotiana tabacum L.* dengan daun sebagai bagian yang dipanen. Kultivar tembakau yang berasal dari spesies *Nicotiana tabacum L.*, sub genus *Tabacum*, genus *Nicotiana* dan famili *Solanaceae* telah berkembang luas. Tembakau tersebut telah banyak mengalami perkembangan dengan banyak melahirkan berbagai jenis tembakau baik berdasarkan morfologi, tipologi, adaptasi local, penggunaan dan musim tanamnya maupun berdasarkan cara pengolahan.

Terdapat beberapa ciri yang dimiliki oleh daun tembakau untuk bisa dilihat mutunya, antara lain warna dan panjang daun serta bentuk ujung daun. Kriteria dalam penilaian mutu tembakau yang terlebih dahulu dilihat adalah berdasarkan warna yaitu warna dasar (*value*) dan tingkat kecerahannya (*chroma*) yang ditentukan secara visual oleh *grader* yaitu seseorang orang yang dipercaya untuk menilai mutu atau *grade* daun tembakau. Faktor yang peting dalam menentukan jenis mutu tembakau adalah penilaian berdasarkan warna, karena warna merupakan sifat dasar yang dipengaruhi oleh baik atau buruknya proses pengolahan, posisi daun di batang, tua atau mudanya daun tembakau sewaktu dipanen, adanya gangguan iklim. (SNI 01 -7134-2006).





Gambar 2.1 Perkebunan Tembakau di Indonesia

(Sumber : [www.sampoerna.com](http://www.sampoerna.com))

Sebagai contoh kultivar tembakau dapat digolongkan berdasarkan adaptasi lokasi yaitu tembakau Virginia, Burley dan Turki. Secara umum tembakau di Indonesia dapat dibedakan menurut musim tanamnya yang terbagi menjadi dua jenis yaitu tembakau *Voor-Oogst* dan tembakau *Na-Oogst*.

1) Tembakau *Voor-Oogst*

Tembakau *Voor-Oogst* biasanya dinamakan tembakau musim kemarau atau *oneregend*. Jadi tembakau jenis ditanam pada akhir musim penghujan dan dipanen pada waktu musim kemarau.

2) Tembakau *Na-Oogst*

Tembakau *Na-Oogst* yaitu jenis tembakau yang ditanam akhir musim kemarau, kemudian dipanen atau dipetik pada musim penghujan.

Berdasarkan bentuk keringnya, tembakau dibedakan menjadi beberapa *type* tembakau diantaranya tembakau krosok (*leaf type*) dan tembakau rajangan (*slice type*). Tembakau rajangan merupakan tipe tembakau asli Indonesia sedangkan tembakau krosok adalah tembakau yang paling banyak terdapat di dunia.

Berdasarkan bentuk fisiknya, tembakau di Indonesia dipasarkan dalam dua wujud, yaitu:

1) Rajangan (*slicing type*)

Tembakau rajangan hanya terdapat di Indonesia saja. Tembakau ini diarajang terlebih dahulu sebelum dipasarkan, untuk selanjutnya dilakukan proses pengeringan dengan bantuan sinar matahari (*sun cured*). Berdasarkan tipe ukuran rajangannya, terbagi menjadi dua, meliputi: rajangan kasar dan sedang (*broad cut*) serta rajangan halus (*fine cut*). Berdasarkan warnanya, tembakau rajangan dibagi menjadi dua, rajangan kuning dan hitam.

2) Kerosok (*leaf type*)

Jenis tembakau yang paling banyak terdapat di dunia adalah tembakau kerosok. Setelah melalui proses pengeringan tembakau kerosok dipasarkan dalam bentuk lembaran daun,.

Harga tembakau krosok cenderung lebih mahal dari pada rajangan, sebab melalui tahapan yang panjang sebelum siap dipasarkan, mulai pengeringan hingga sortasi. Berdasarkan metode pengeringannya, tembakau krosok dibedakan menjadi:

- 1) *Air cured*, adalah suatu proses dalam mengeringkan tembakau dengan mengalirkan udara bebas atau angin. Metode pengeringan ini memerlukan bangunan khusus (*curing shed*). Pengeringan dengan metode ini akan menghasilkan tembakau dengan kadar gula rendah namun tinggi nikotin.
- 2) *Flue cured*, adalah proses pengeringan daun tembakau dengan mengalirkan udara panas melalui pipa (*flue*). Tembakau Virginia FC adalah tergolong tembakau yang diproses seperti ini. Prinsip pengeringan *flue cured* sangat sederhana, berkurangnya kelembaban secara perlahan selama 24 – 60 jam pertama (masa penguningan) diikuti hilangnya kadar air secara cepat hingga lamina mengering, yang diikuti mengeringnya gagang.
- 3) *Sun cured*, adalah suatu proses dalam mengeringkan tembakau dengan menggunakan bantuan sinar matahari secara langsung atau penjemuran. Proses penjemuran untuk tembakau krosok selama 7-10 hari. Metode ini juga dipakai untuk pengeringan tembakau Oriental, yang menghasilkan kadar gula dan

nikotin yang rendah. Sebagai contoh berdasarkan penggunaannya digolongkan tembakau cerutu, sigaret, kemyak, isap dan pipa.

- 4) *Fire cured*, adalah suatu proses dalam mengeringkan daun tembakau dengan cara mengalirkan asap dan panas di bagian bawah susunan daun tembakau. Berbeda dengan *flue cured*, dimana bara api tidak dibiarkan membara, melainkan dijaga agar tetap mengeluarkan asap. Dalam proses pengeringan *fire cured* agar menghasilkan asap yang cukup menggunakan bahan baku kayu akasia yang dicampur dengan bongkol dan ampas tebu. Sehingga diharapkan menghasilkan aroma yang harum dan manis. Pengeringan dengan metode ini akan menghasilkan tembakau dengan kadar gula rendah namun tinggi nikotin.

Tembakau Virginia dijual dalam wujud kering oven (*Curing*). Secara biologis proses *curing* merupakan proses melepaskan kadar air dari daun tembakau basah yang dipanen dalam keadaan hidup. Selama ini di beberapa petani ada yang berpendapat bahwa *curing* adalah proses pengeringan tembakau saja. Tidak menyadari bahwa sel-sel di dalam daun tersebut masih tetap hidup setelah dipanen. Tujuan *curing* sebenarnya yaitu untuk melepaskan kadar air 80 -90 % menjadi 10 - 15 %, perubahan warna dari zat hijau daun menjadi warna *orange* dengan aroma sesuai dengan standar tembakau yang diproses. Oleh sebab itu untuk mendapatkan hasil *curing*/omprongan tembakau yang baik, maka daun tembakau itu harus sudah masak dan seragam. Ciri-ciri daun yang sudah masak adalah :

- Warna daun tembakau sudah mulai hijau kekuningan, sebagian dari ujung dan tepi daun berwarna coklat.
- Tangkai daun berwarna hijau kuning, keputih-putihan.
- Daun/tulang daun dalam posisi mendatar.
- Kadang-kadang pada lembaran daun ada bintik-bintik coklat, sebagai lambang ketuaan.

Tabel 2.1 Proses *Flue-Curing*

Jalannya Pekerjaan	Derajat Panas	Banyaknya Jam	Pembaharuan Udara (ventilasi)	Lembab Relatif (RH)	Naiknya Derajat Panas Sejam
Memanaskan sampai kepanas penguningan .....	26-32	2	-	85-95	3
Tetap pada panas penguningan ke-1 (sampai $\pm 80\%$ dari daun-daun menjadi kuning).....	32	15-20	-	85-95	1
Naik dampai kepanas penguningan ke-2 .....	32-38	5	-	85-95	1
Tetap pada panas penguningan ke-2 (sampai hanya tinggal 1 baris halus berwarna hijau pada urat daun).....	38	$\pm 5$	-	85-95	3
Naik sampai kepanas penguningan ke-3 .....	38-40	1	-	80-90	-
Tetap pada panas penguningan ke-3 .....	40	3-4	(v)	$\pm 80$	3
Niak sampai kepanas penguningan ke-4 .....	40-43	1	(v)	$\pm 60$	-
Tetap pada panas penguningan ke-4 .....	43	$\pm 2$	v	45	1
Naik sampai kepanas <i>fixeer</i> ( <i>fixing temperature</i> ) .....	43-49	5	v	45	-
Tetap pada panas <i>fexeer</i> sampai ujung daun menjadi rapuh.....	49	10-30	vv	serendah rendah	1
Naik sampai kepanas pengeringan daun .....	49-60	10	vv	sama	-
Tetap sampai helai daun kering bear .....	60	$\pm 10$	vv	sama	3
Naik sampai kepanas pengeringan ibu tulang daun yang pertama .....	60-71	4	vv	sama	-
Tetap panas pengeringan ibu tulang daun yang pertama .....	71	1-5	-	sama	3
Naik sampai kepanas pengeringan penghabisan .....	71-72	4	-	sama	-

(v) : ventilasi dimana perlu; v : ventilasi sebagian, vv : ventilasi sepenuhnya  
 Sesudah diisi dengan daun hidup, maka dengan tidak dipanaskan, panasnya kebanyakan sudah meningkat  $32^{\circ}\text{C}$ . (Wanrooy, 1951)

## 2.2 Heating Element

*Electrical Heating Element* adalah sebuah elemen pemanas listrik yang menghasilkan panas yang bersumber dari kawat ataupun pita bertahanan listrik tinggi (*Resistance Wire*). Bahan untuk pembuatan element pemanas digunakan adalah niklin yang di kedua ujungnya dialiri arus listrik dan dilapisi oleh isolator listrik yang mampu meneruskan panas dengan baik hingga aman jika digunakan. Ada 2 macam jenis utama pada elemen pemanas listrik ini yaitu :

1. Elemen pemanas listrik bentuk dasar yaitu elemen pemanas yang menggunakan lapisan isolator listrik untuk melapisi *Resistance Wire*. Ada macam-macam elemen pemanas bentuk ini antara lain *Bank Channel heater*, *Ceramik Heater*, *Black Body Keramik Heater*, *Silica* dan *Quartz Heater*.
2. Elemen pemanas listrik bentuk lanjut yaitu suatu elemen pemanas yang menggunakan lapisan pipa atau lembaran plat logam untuk sebagai penyesuain terhadap penggunaan dari elemen pemanas tersebut. Bahan logam yang biasa digunakan yaitu tembaga, kuningan, *mild stell* dan *stainless stell*. Jenis element pemanas antara lain *Band*, *Nozzle*, *Stripe Heater*, *Tubular Heater* dan *Catridge Heater*.

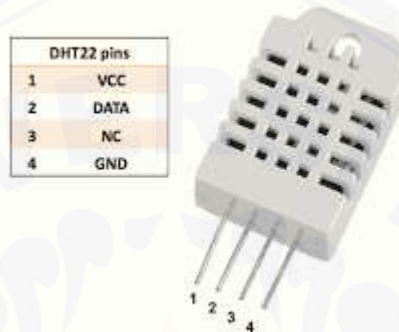


Gambar 2.2 Bentuk-Bentuk *Tubular Heater*

(Sumber : <http://www.hotwatt.com/tastraight.htm>)

### 2.3 Sensor DHT22

Modul DHT merupakan sensor kelembaban dan suhu yang terdiri dari dua bagian yaitu sensor kelembaban kapasitif dan thermistor. Sensor ini menggunakan IC pengendali keluaran sinyal digital sehingga sensor ini tidak memerlukan rangkaian pengendali sinyal dan ADC (Aosong, 2012). Modul DHT memiliki banyak jenis, salah satunya yaitu DHT22 (AM2302) seperti pada Gambar 2.3.



Gambar 2.3 Sensor DHT22

(Sumber : <http://www.instructables.com>)

DHT-22 atau dikenal dengan versi AM2302 adalah sebuah *single chip* sensor yang dapat mengukur suhu dan kelembaban relatif dengan multi modul sensor yang *output*-nya sudah terkalibrasi secara digital. DHT-22 memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat. Dalam modul DHT 22 terdapat komponen sensor kelembaban, sensor suhu NTC (atau termistor) dan IC di sisi belakang sensor.

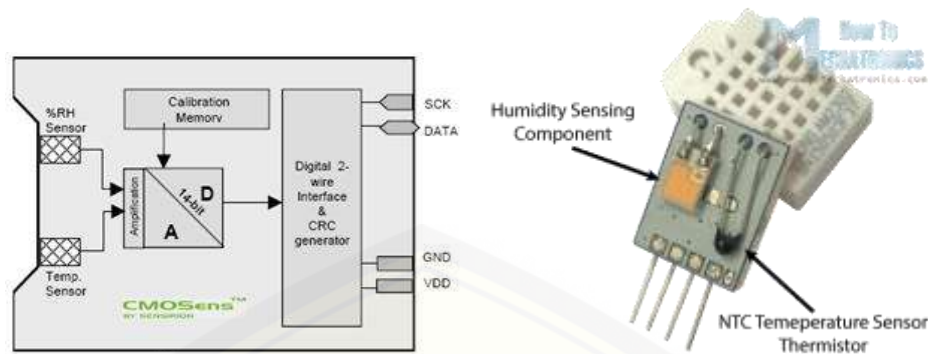
Sensor DHT22 memiliki *range* pengukuran yang luas yaitu  $-40^{\circ}\text{C}$  sampai  $125^{\circ}\text{C}$  dengan tingkat akurasi  $\pm 0.5^{\circ}\text{C}$  untuk suhu dan 0 sampai 100% dengan tingkat akurasi  $\pm 5\%$  untuk kelembaban. Sensor ini memiliki *output* digital (*single-bus*) dengan akurasi yang tinggi.

Ada dua spesifikasi dimana DHT11 lebih baik dari pada DHT22 yaitu tingkat sampling untuk DHT11 adalah 1Hz atau satu membaca setiap detik, sedangkan sampling DHT22 adalah 0,5Hz atau satu pembacaan setiap dua detik dan juga DHT11 memiliki ukuran tubuh lebih kecil dibandingkan DHT22. Tegangan operasi kedua sensor adalah dari 3 sampai 5 volt, sedangkan arus maksimum yang digunakan saat mengukur adalah 2.5mA.

Cara kerja DHT22 dalam mengukur suhu yaitu dengan menggunakan komponen NTC atau thermistor. Istilah "NTC" berarti "*Negative Temperature Coefficient*" yang berarti resistansi menurun seiring dengan kenaikan suhu. Thermistor merupakan sebuah resistor variabel yang mana akan mengalami perubahan resistansi ketika mengalami perubahan suhu. Sensor ini dibuat dari bahan semikonduktor seperti keramik atau polimer untuk memberi perubahan resistansi lebih besar dengan hanya sedikit perubahan suhu. Perubahan resistansi ini diukur dan diolah oleh IC yang membuatnya siap dibaca oleh mikrokontroler. IC tersebut merupakan OTP (*One Time Programming*) memory yang fungsinya hampir sama dengan EPROM. Dengan OTP *program memory* inilah koefisien kalibrasi disimpan. Sehingga ketika internal sensor mendeteksi sesuatu, maka module ini menyertakan koefisien tersebut dalam kalkulasinya.

Pada pengukuran suhu data yang dihasilkan adalah 14 bit, sedangkan untuk kelembaban data yang dihasilkan adalah 12 bit. Sensor DHT22 memiliki ADC (*Analog to Digital Converter*) di dalamnya sehingga keluaran data DHT22 sudah terkonversi dalam bentuk data digital dan tidak memerlukan ADC eksternal dalam pengolahan data pada mikrokontroler. Sehingga ketika sensor bekerja secara otomatis akan menghasilkan data yang sudah dikalkulasi berupa besar derajat suhu udara dan persen kelembaban relatif yang terukur. Sensor ini dapat ditransmisikan melalui medium dengan panjang maksimum 20 meter.

Pada proses pengiriman data keluaran dari temperatur dan kelembaban pada pin data sensor DHT22 yaitu secara bergantian sesuai dengan *clock* yang diberikan oleh mikrokontroler agar sensor dapat bekerja. Sistem sensor yang digunakan DHT22 untuk mengukur suhu dan kelembaban adalah dengan sistem komunikasi *bidirectional 2-wire*. Selain itu sistem sensor ini mempunyai 1 jalur data yang digunakan untuk perintah pengalamatan dan pembacaan data.



Gambar 2.4 diagram blok DHT22  
(Sumber : HowToMechatronics.com)

Tabel 2.2 Spesifikasi sensor DHT22

Model	DHT22
Power supply	3.3-6V DC
Output signal	Digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	Humidity 0-100%RH; temperature -40 ~ 80 °C
Accuracy	Humidity $\pm 2\%$ RH (Max $\pm 5\%$ RH); temperature $< \pm 0.5$ °C
Resolution or sensitivity	Humidity 0.1%RH
Repeatability	Humidity $\pm 1\%$ RH
Humidity hysteresis	$\pm 0.3\%$ RH
Long-term stability	$\pm 0.3\%$ RH/year
Sensing period	Average: 2s
Interchangeability	Fully interchangeable
Dimensions	Small size 14x18x5.5mm; big size 22x28x5mm

Dalam akuisisi data sensor DHT22, mikrokontroler melakukan pengiriman *low signal* berturut turut selama 18ms melalui jalur komunikasi data, kemudian mengembalikannya kondisi sinyal *high*. Jadi dapat diartikan bahwa sensor DHT22 melakukan permintaan akuisisi data. Selanjutnya, sensor DHT22 akan mengirimkan *feedback* berturut turut *low signal* selama 18 mikro detik melalui jalur komunikasi data. Jadi hal ini diartikan sebagai respon permintaan akuisisi data oleh mikrokontroler. Jadi DHT22 akan mulai mengirimkan data sensor secara serial.



## 2.4 Fun/Kipas Aksial

*Axial Fan* adalah sebuah alat yang berfungsi menghasilkan aliran fluida gas searah dengan poros kerja kipas tersebut. Kipas aksial adalah kipas yang banyak digunakan dalam kehidupan sehari-hari dan paling banyak digunakan karena kemudahan desain serta harga yang lebih ekonomis dari pada kipas jenis lain. Kipas aksial mempunyai desain yang tidak terlalu rumit serta dapat menghasilkan *flow* yang besar. Kipas aksial biasa digunakan sebagai alat pendingin pada berbagai keperluan seperti pendingin CPU, pendingin *Amplifier* hingga pendingin mesin kendaraan bermotor menggunakan kipas tipe aksial.



Gambar 2.5 Kipas Aksial

(Sumber : <https://es.aliexpress.com>)

Kipas aksial banyak digunakan di dunia industri. Pada pembangkit listrik tenaga uap juga menggunakan kipas aksial untuk men-*supply* udara dalam jumlah banyak yang dibutuhkan untuk proses pembakaran pada *furnace* boiler. Kipas ini memiliki dua tingkat (*stage*) *impeller*, yang kedua-duanya dapat diatur besar bukaan *pitch*-nya. Hal tersebut berfungsi untuk mengatur jumlah (debit) udara yang akan dikirim untuk proses pembakaran.

## 2.5 Kamera

Kamera digunakan sebagai sensor untuk mengambil informasi citra dari tembakau. Spesifikasi kamera Havit HV-V622 yang digunakan sebagai berikut:

Tabel 2.3 Spesifikasi kamera Havit HV-V622

<b>Interface to PC</b>	High Speed USB 2.0
<b>Image Sensor</b>	1/6" CMOS, 640x480 pixels
<b>Resolution</b>	800x480, 352x600, 640x288, 320x240 Capture Format
<b>Lens Features</b>	F=2.4, f=3.5mm
<b>Frame Rate</b>	Max 30fps
<b>Focus Range</b>	Automatic focus 10 cm to infinity
<b>Viewing Angel</b>	60 degrees
<b>Still Image</b>	Up to 16 Megapixel, 4608 x 3456
<b>LED Lightning Support</b>	Available by 3 LED with Trimmer Control
<b>Operating System Support</b>	PC with Windows (2000 / XP / Vista/7)



Gambar 2.6 A4Tech PK-710G

(Sumber : <https://market.yandex.ru>)

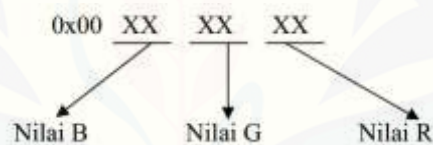
*Webcam* ini sangat mudah diarahkan kemana saja secara manual dengan fungsi *pivot* pada bagian *base*-nya. Pemasangan sangat mudah karena tidak membutuhkan *driver*.

## 2.6 Pengolahan Citra

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra atau gambar dilakukan secara digital menggunakan komputer

### 2.6.1. *Color Image* atau RGB (*Red, Green, Blue*).

Pada *color image* ini setiap piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika masing-masing warna memiliki *range* 0-255, maka totalnya adalah  $255^3 = 16.581.375$  variasi warna berbeda. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti Gambar 2.7, variabel 0x00 menyatakan angka dibelakangnya adalah hexadecimal.



Gambar 2.7 Nilai Warna RGB dalam Hexadesimal



Gambar 2.8 Komposisi Warna RGB

(Sumber : Suhendra, 2016)

### 2.6.2 *Black and White*

Citra digital *black and white* (*grayscale*) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 *byte*. Citra *grayscale* diperoleh dari hasil konversi rata-rata dari nilai *r*, *g*, dan *b*, dengan demikian maka persamaannya dapat dituliskan pada Persamaan 2.1.

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.1)$$

Dimana  $I_R(x, y)$  merupakan nilai piksel *Red* titik  $(x, y)$ ,  $I_G(x, y)$  nilai piksel *Green* titik  $(x, y)$ ,  $I_B(x, y)$  nilai piksel *Blue* titik  $(x, y)$  sedangkan  $I_{BW}(x, y)$  adalah nilai piksel *black and white* titik  $(x, y)$ .

### 2.6.3 *Binary Image*

Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (*finger print*), atau gambar arsitektur. *Binary image* merupakan hasil pengolahan dari *black and white image*, dengan menggunakan fungsi pada Persamaan 2.2 dan dalam bentuk *floating point* menggunakan Persamaan 2.3.

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (2.2)$$

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (2.3)$$

Dimana  $I_{BW}(x, y)$  merupakan nilai piksel *Gray* titik  $(x, y)$ ,  $I_{Bin}(x, y)$  nilai piksel *Binary* titik  $(x, y)$ , sedangkan  $T$  adalah nilai *threshold*.

## 2.7 Arduino Mega 2560

Arduino merupakan platform bersifat *open-source hardware* yang biasa digunakan dalam pembuatan prototipe elektronik. Arduino ditujukan bagi para desainer, seniman dan siapapun yang tertarik dalam membuat obyek inovatif

Arduino Mega 2560 adalah salah satu jenis *board* mikrokontroler berbasis arduino menggunakan chip ATmega 2560. Arduino Mega memiliki 54 pin *input/output* digital dengan 15 pin *input* diantaranya dapat digunakan sebagai *output* PWM, pin SDA dan pin SCL, 16 pin *input* analog dan 4 pin sebagai *UART (port serial hardware)*. Menggunakan osilator kristal 16 MHz, selain itu terdapat koneksi USB, *jack power*, *ICSP header*, dan tombol *reset*.

Untuk menjalankan Arduino Mega yaitu hanya menghubungkan *board* Arduino Mega ke komputer dengan menggunakan kabel USB atau dengan tegangan DC 7-12volt pada *jack power*. Sedangkan untuk memprogramnya menggunakan aplikasi IDE Arduino.

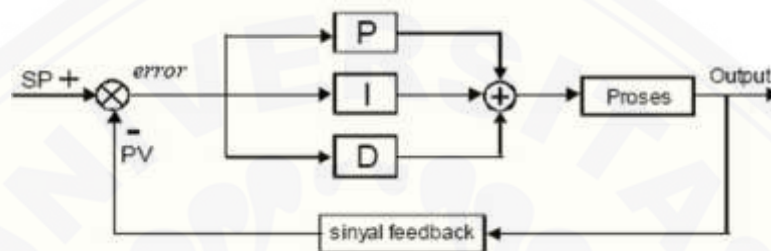


Gambar 2.9 *Board* Arduino Mega 2560

(Sumber : Mahesta, 2015)

## 2.8 Sistem Kendali PID (*Proportional Integral Derivative*)

Sistem kendali PID (*Proportional Integral Derivative*) adalah sistem kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*feedback*) pada sistem tersebut. Sistem kendali PID terdiri dari tiga kontrol yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*). Masing-masing kontrol tersebut memiliki kelebihan dan kekurangan. Blok diagram sistem kendali PID ditunjukkan pada Gambar 2.10.



Gambar 2.10 Blok Diagram Pengendali PID

(Sumber : Bachri, 2004)

Adapun persamaan sistem kendali PID adalah :

$$PID = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.4)$$

Dengan :

PID = output dari kendali PID

$K_p$  = konstanta *proporsional*

$K_i = K_p/T_i$

$T_i$  = konstanta *integral*

$K_d = K_p \cdot T_d$

$T_d$  = konstanta *derivatif*

$e(t) = error$

Untuk menghasilkan respon yang maksimal terhadap masukan dalam perancangan sistem kendali PID yaitu dengan mengatur masing-masing parameter P, I dan D.

### 2.8.1 Cara Mentuning Manual Parameter PID

Cara mentuning yaitu cara dengan coba-coba merubah parameter-parameter PID dengan melihat tabel parameter PID. Dengan menganalisa respon yang dihasilkan, nilai-nilai  $K_p$ ,  $K_i$ , dan  $K_d$  bisa diubah-ubah sesuai dengan Tabel 2.4. Parameter PID ditunjukkan pada Tabel 2.4.

Tabel 2.4 Parameter PID

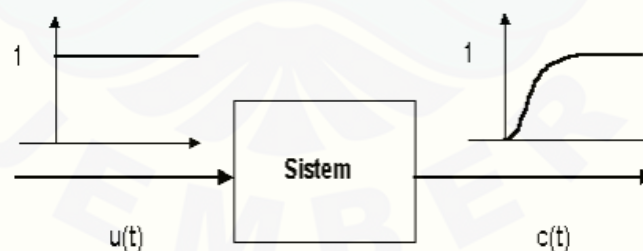
Parameter	Rise Timer	Overshoot	Setting Time	S-S Error
$K_p$	Berkurang	Bertambah	0	Berkurang
$K_i$	Berkurang	Bertambah	Bertambah	Menghilangkan
$K_d$	Minor Change	Berkurang	Berkurang	Minor Change

### 2.8.2 Metoda Ziegler-Nichols

Metode *Ziegler-Nichols* adalah salah satu metode mencari parameter-parameter PID. Metoda ini memiliki 2 tipe, yaitu tipe 1 (*open loop*) dan tipe 2 (*closed loop*).

#### 2.8.2.1 Ziegler-Nichols tipe 1 (*open loop*)

Dalam metoda ini, digunakan sistem *open loop*. Sistem diberi input step sehingga respon *open loop* terbentuk.



Gambar 2.11 Sistem Diberi *Input Step*

(Sumber : [www.slideshare.net](http://www.slideshare.net))

Dari respon *open loop* yang dihasilkan, parameter-parameter ZN tipe 1 ( $L$  dan  $T$ ) dapat didapatkan. Proses desain menentukan parameter  $L$  dan  $T$  ditunjukkan pada Gambar 2.12.



Gambar 2.12 Proses Desain Penentuan Parameter L dan T  
(Sumber : Bachri, 2004)

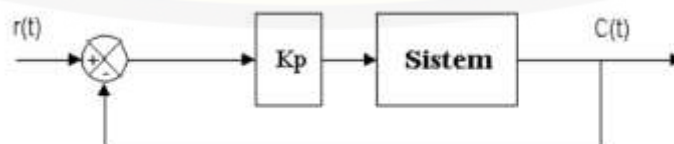
Setelah parameter L dan T didapatkan, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dicari dengan menggunakan rumus-rumus parameter PID untuk metoda ZN tipe 1. Tabel 2.5 menunjukkan tabel parameter PID untuk ZN tipe 1.

Tabel 2.5 Parameter PID untuk ZN tipe 1

Type Of Controller	$K_p$	$T_i$	$T_d$
P	$T/L$	$\infty$	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	0.51

### 2.8.2.2 Ziegler-Nichols tipe 2 (*closed loop*)

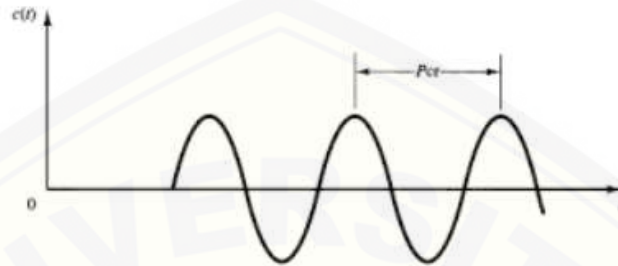
Dalam metoda ZN tipe 2, digunakan sistem *closed loop*. Namun yang digunakan hanya  $K_p$  saja. Sistem dibuat hingga berosilasi terus menerus dengan mengatur besarnya nilai  $K_p$ .



Gambar 2.13 Sistem *closed loop* dengan Menggunakan  $K_p$  Saja  
(Sumber : [www.slideshare.net](http://www.slideshare.net))



Besarnya nilai  $K_p$  saat respon sistem berosilasi terus menerus merupakan nilai  $K_{cr}$ . Dari respon yang dihasilkan, parameter lain ZN tipe 2 selain  $K_{cr}$ , yaitu  $P_{cr}$  dapat dicari. Proses desain menentukan parameter  $P_{cr}$  ditunjukkan pada gambar 2.14.



Gambar 2.14 Proses Mendesain Menentukan Parameter  $P_{cr}$

(Sumber : [www.slideshare.net](http://www.slideshare.net))

Setelah parameter  $K_{cr}$  dan  $P_{cr}$  didapatkan, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dihitung dengan menggunakan rumus-rumus parameter PID untuk ZN tipe 2. Tabel 2.6 menunjukkan tabel parameter PID untuk ZN tipe 2.

Tabel 2.6 Parameter PID Untuk ZN Tipe 2

Type Of Controller	$K_p$	$T_i$	$T_d$
P	$0.5K_{cr}$	$\infty$	0
PI	$0.45K_{cr}$	$(1/1.2)P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

## 2.9 Kontrol Fuzzy Logic

### 2.9.1 Teori Himpunan Logika Fuzzy

Dalam bahasa Inggris, *fuzzy* mempunyai arti kabur atau tidak jelas. Jadi, logika *fuzzy* adalah suatu logika yang menghimpun unsur-unsur ketidakpastian (kabur).

Pada sistem logika biasa dikenal dengan dua kemungkinan yaitu salah bernilai 0 atau benar bernilai 1, logika seperti ini disebut juga dengan logika tegas (*crisp*). Sedangkan sistem logika *fuzzy* mengenal diantara salah (0) dan benar (1) yang dinyatakan kedalam derajat kebenaran dimana nilainya antara 0 (nol) sampai 1 (satu).

Himpunan fuzzy adalah pengelompokan berdasarkan variabel *linguistik* yang dinyatakan dengan fungsi keanggotaan. Keanggotaan suatu himpunan nilainya dapat diketahui berdasarkan derajat keanggotaan yaitu antara 0 sampai dengan 1.

### 2.9.2 Fungsi Keanggotaan

Keanggotaan (*membership function*) adalah suatu pemetaan suatu input ke dalam derajat keanggotaan dalam bentuk kurva yang nilainya antara 0 sampai dengan 1. Cara untuk mendapatkan derajat keanggotaan yaitu melalui pendekatan fungsi. Ada bermacam-macam fungsi yang biasa digunakan diantaranya:

1. representasi linear
2. representasi segitiga
3. representasi trapesium
4. representasi kurva bentuk bahu
5. representasi kurva S
6. representasi bentuk lonceng

### 2.9.3 Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy

Ada 3 operator dasar himpunan *fuzzy*, yaitu sebagai berikut.

a. Operator *AND*

Operator ini berhubungan dengan operasi interseksi pada himpunan.

$$\mu_{A \wedge B} = \min(\mu_A[x], \mu_B[y])$$

b. Operator *OR*

Operator ini berhubungan dengan operasi union pada himpunan.

$$\mu_{A \vee B} = \max(\mu_A[x], \mu_B[y])$$

c. Operator *NOT* Operator ini berhubungan dengan operasi komplemen pada himpunan.

$$\mu_{\neg A} = 1 - \mu_A[x]$$

### 2.9.4 Fuzzyfikasi

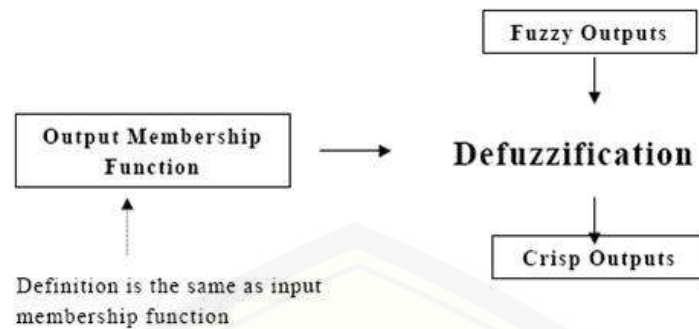
Proses *fuzzyfikasi* adalah suatu proses mengubah dari *variable non fuzzy* (variabel numerik) menjadi *variable fuzzy* (variabel linguistik). Dalam proses *fuzzyfikasi* nilai *input* yang berbentuk *variable numeric* sebelum diolah harus dirubah terlebih dahulu kedalam variabel *linguistik* sehingga menjadi informasi *fuzzy*. Dengan kata lain *fuzzyfikasi* adalah pemetaan titik-titik numerik (*crisp point*).

### 2.9.5 Penentuan *Rule Base*

*Rule base* adalah pembuat aturan *fuzzy* yang berhubungan dengan keadaan masukan dan keluaran sehingga dapat diambil keputusan untuk mendapatkan aksi keluaran dari kontrol. Pembuatan *rule base* berfungsi untuk mencari suatu nilai *fuzzy output* dari *fuzzy input*.

### 2.9.6 Defuzzyfikasi

*Defuzzifikasi* adalah proses pemetaan dari hasil aksi kontrol *inferensi fuzzy* ke aksi kontrol *non-fuzzy*. Jadi *defuzzifikasi* adalah proses pemetaan kembali dari himpunan-himpunan *fuzzy* untuk mendapatkan nilai tegasnya kembali (*crisp*).

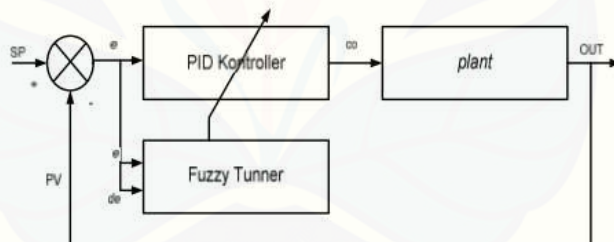


Gambar 2.15 Defuzzyfikasi.

(Sumber : fahmizaleeits.wordpress.com)

### 2.10 Hybrid Fuzzy-PID

*Hybrid Fuzzy* PID adalah metode pengontrolan yang menggabungkan dua buah kontroler yaitu kontroler *fuzzy* dan kontroler PID sehingga dinamakan sebagai *Hybrid Fuzzy* PID. Kontroler ini memiliki masukan sinyal *error* dan sinyal  $\Delta e$  (*delta error*). Struktur kontroler *Hybrid Fuzzy* PID ditampilkan pada Gambar 2.16



Gambar 2.16 Hybrid Fuzzy-PID

(Sumber : Happiyanto, 2016)

**BAB 3. METODE PENELITIAN**

**3.1 Waktu dan Tempat**

3.1.1 Tempat

Adapun tempat penelitian, pengujian dan analisis dilakukan secara umum dilakukan di :

Tempat : Lab. Elektronika dan Terapan Fakultas Teknik Universitas Jember

Alamat : Jl. Slamet Riyadi No. 62, Patrang, Kabupaten Jember

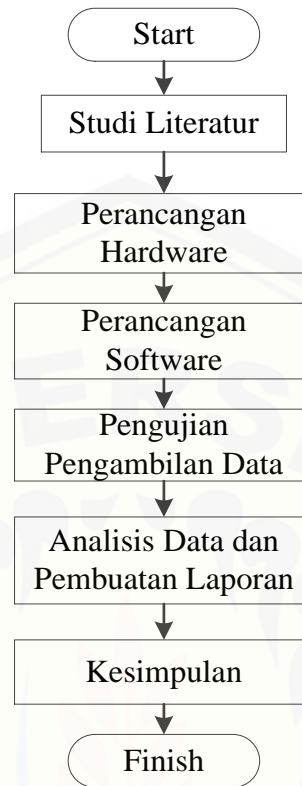
3.1.2 Waktu

Adapun waktu penelitian, pengujian dan analisis dilakukan secara umum dilakukan Desember 2016 sampai dengan April 2017.

Tabel 3.1 *Timieline* Penelitian

No	Kegiatan	Bulan															
		Januari				Februari				Maret				April			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Persiapan	■															
2	Studi Literatur		■	■	■												
3	Perancangan <i>Elektrisel</i> dan Oven				■	■	■	■									
4	Pembelian Alat, Bahan					■	■	■	■								
5	Pembuatan Alat Peningkat Tembakau					■	■	■	■	■	■	■					
6	Pengujian													■	■	■	■
7	Analisis Data													■	■	■	■
8	Kesimpulan															■	■

### 3.2 Tahap Penelitian



Gambar 3.1 *Flowchart* Penelitian

Tahap penelitian “Rancang Bangun Alat Pengering Tembakau Otomatis dengan Metode Pengolahan Citra Digital Menggunakan Kontrol *Hybrid Fuzzy-PID*” adalah sebagai berikut :

1. Studi literatur

Tahap pertama melalui penelitian ini dilakukan dengan mencari literatur yang terkait dengan penelitian ini, dengan tujuan agar perancangan dan pelaksanaan penelitian dapat sesuai dengan yang diinginkan. Literatur diperoleh dari buku-buku dan jurnal yang berhubungan dengan sistem pengering tembakau, kontrol PID dan *Fuzzy logic* serta yang berhubungan dengan perancangan alat pengering tembakau maupun sistem kontrol. Selain studi literatur, juga melakukan wawancara secara langsung pada pakarnya yaitu seorang petani tembakau yang juga melakukan proses pengovenan tembakau.

## 2. Perancangan *Hardware*

Tahap kedua adalah proses perancangan *hardware* baik dari alat dan bahan yang digunakan, desain alat pengering tembakau, rancangan bentuk dan cara kerja masing-masing komponen yang digunakan dan desain elektriknya.

## 3. Perancangan *Software*

Tahap ketiga adalah proses perancangan *plan* dan sistem kontrol *hybrid fuzzy-PID*, pengolahan citra digital, sensor suhu, program dan arduino. Tahap menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  serta menentuka *membership function* dari logika *fuzzy* yang akan digunakan pada *plan* agar mencapai terget yang di inginkan.

## 4. Pengujian dan Pengambilan Data

Tahap keempat adalah pengujian terhadap komponen – komponen yang kritis seperti pengujiana sensor suhu dan pengujian sensor kamera. Pengujian ini sangat penting karena dapat mempengaruhi tingkat keberhasilan dari proses pengeringan.

## 5. Analisa Data dan Pembuatan Laporan

Tahap ke enam adalah melakukan analisa pada data yang diperoleh dari data sebelumnya. Pengambilan data dilakukan dari hasil pengujian kontrol PID dan kontrol *fuzzy* dengan *monitoring* kamera. Sehingga di harapkan menghasilkan keluaran yang diinginkan dengan menganalisa masing-masing percobaan.

## 6. Penarikan Kesimpulan

Tahap kedepan adalah tahap terakhir pada penelitian yaitu memberikan kesimpulan dari pengujian dan analisa data yang diperoleh.

### 3.3 Perancangan Perangkat Keras

#### 3.3.1 Alat dan Bahan

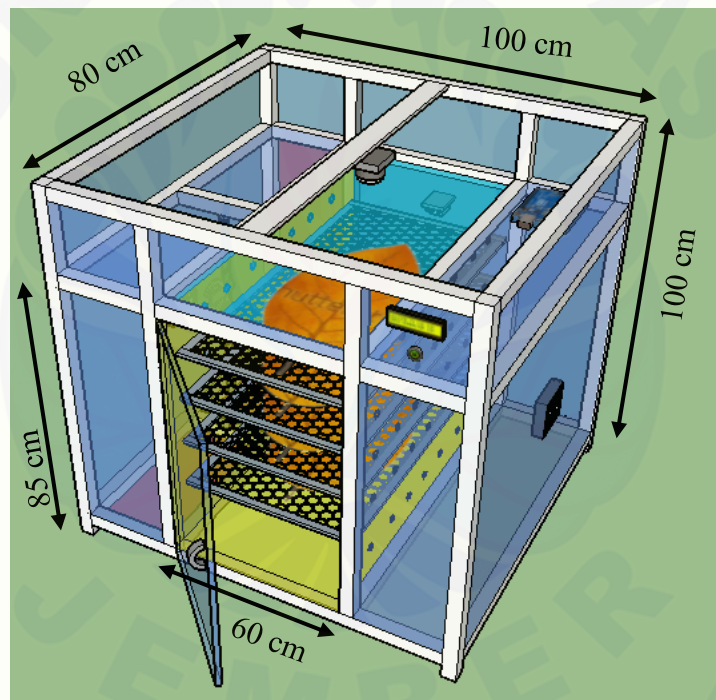
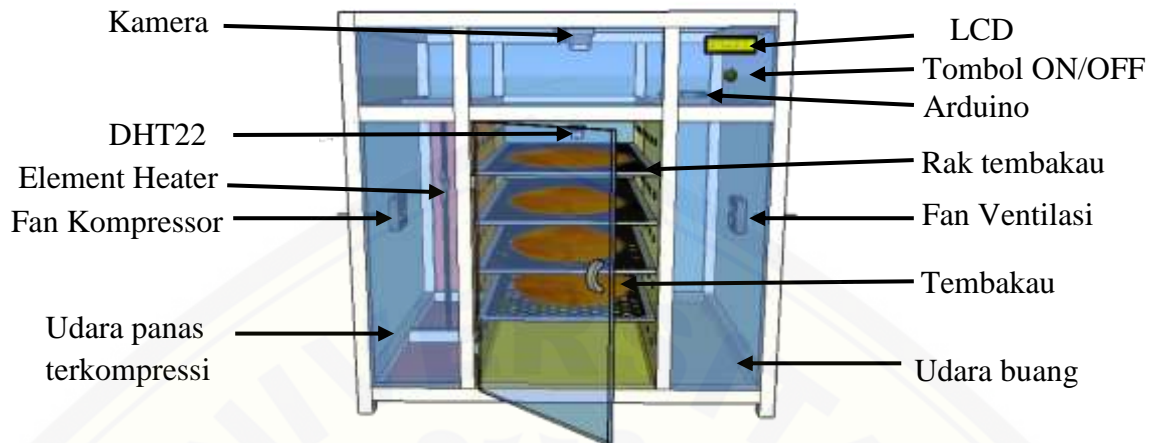
Dalam penelitian ini digunakan peralatan-peralatan sebagai penunjang pembuatan dan pengujian tugas akhir. Sedangkan untuk penjelasan beberapa komponen lebih lengkapnya sebagaimana dijelaskan dalam BAB 2. Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut.

Tabel 3.2 Daftar Peralatan yang Digunakan

No	Nama Alat	Spesifikasi Teknis dan Penjelasan
1	<i>Switching Power Supply</i>	2 buah <i>Power Supply</i> , <i>Input Voltage</i> AC 85 - 265V 50 / 60Hz, <i>Output Voltage</i> 12V DC, 20A, dan 240W. <i>Output Voltage</i> 12V DC, 2A, dan 24W.
2	<i>Arduino Controller</i>	Arduino Mega2560. Atmega 2560 <i>microcontroller</i> , <i>Input voltage</i> 7-12V, 5V <i>Electric current</i> : 500mA, 3.3 <i>Electric current</i> : 50mA, 54 <i>Digital I/O Pins</i> (15 PWM <i>output</i> ), 16 <i>Analog Inputs</i> , 256 KB <i>Flash Memory</i> , 16 <i>Clock Speed</i>
3	Sensor Suhu	DHT22. <i>Input voltage</i> 3.3-6V, <i>Digital signal via single-bus</i> , <i>Operating renge temperature</i> -40 ~ 80 °C,
4	LCD	LCD 1602 + I2C, <i>Voltage</i> 5V DC, <i>Display</i> 2-lines, 16- <i>characters</i>
5	Modul RTC	DS1307 I2C <i>Real Time Clock chips</i> (RTC), LIR2032 <i>rechargeable lithium battery</i> , 1Hz <i>output pin</i> , 56 Bytes <i>of Non-volatile memory available to user</i>
6	<i>Micro SD Chard</i>	6 pins (GND, VCC, MISO, MOSI, SCK, CS), <i>standard communication interface SPI</i> , <i>power supply</i> 4.5V ~ 5.5V, 3.3V <i>voltage regulator circuit board</i> ,
7	<i>Fan</i>	2 buah ukuran 12x12cm: <i>Electric current</i> 0,12A 10 buah ukuran 5x5cm: <i>Electric current</i> 0,09A
8	<i>Webcam</i>	16 Megapixel, <i>Rosolution</i> 800x480, 352x600, 640x288, 320x240. PC with Windows (2000 / XP / Vista/7), <i>High Speed</i> USB 2.0
9	<i>Heater Element</i>	2 buah, <i>Power</i> 190W, <i>Voltage</i> 220V, <i>Frekuensi</i> 50Hz
10	<i>LED Strip</i>	10 buah, <i>Voltage</i> 12 V
11	Laptop	Toshiba Satelite C40-A

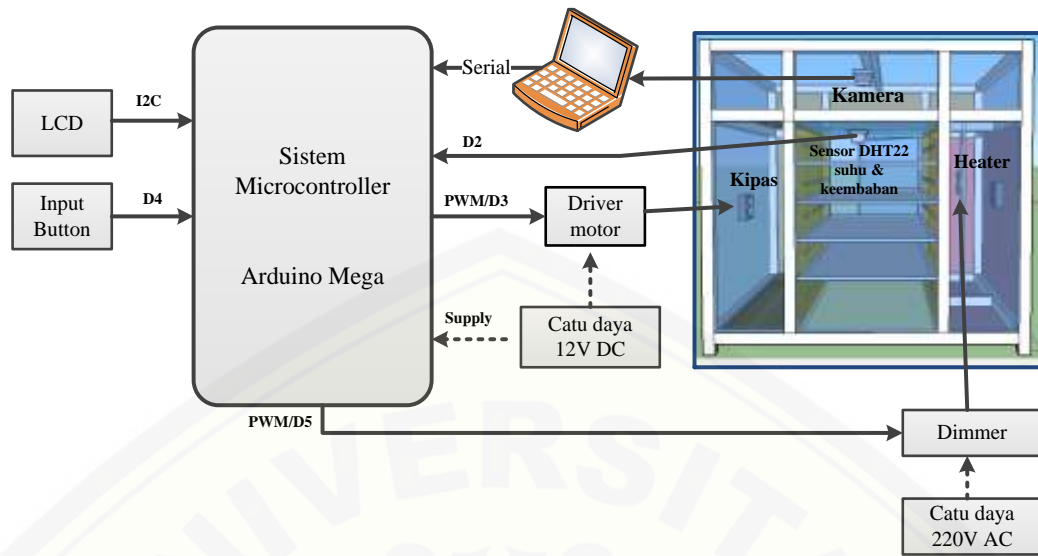


### 3.3.2 Desain Alat



Gambar 3.2 Desain Oven Tembakau

Alat pengering tembakau yang digunakan berupa lemari almunium yang dilengkapi dengan pemanas didalamnya menggunakan *heating element* dan *fan*/kipas sebagai pengatur ventilasi. Kamera diletakan sebelah atas dari lemari untuk mengambil citra dari tembakau dan arduino diletakkan sebelah atas dari alat sebagai pemrosesan dan PC/laptop diletakna disamping alat pengolahan citra yang diambil dari kamera. Rancangan bentuk alat dapat dilihat pada Gambar 3.2.



Gambar 3.3 Diagram Blok Perangkat Keras



Bagian Depan Oven



Bagian Dalam Oven



Ruang Pemanas



Rangkaian kontroller dan driver kipas



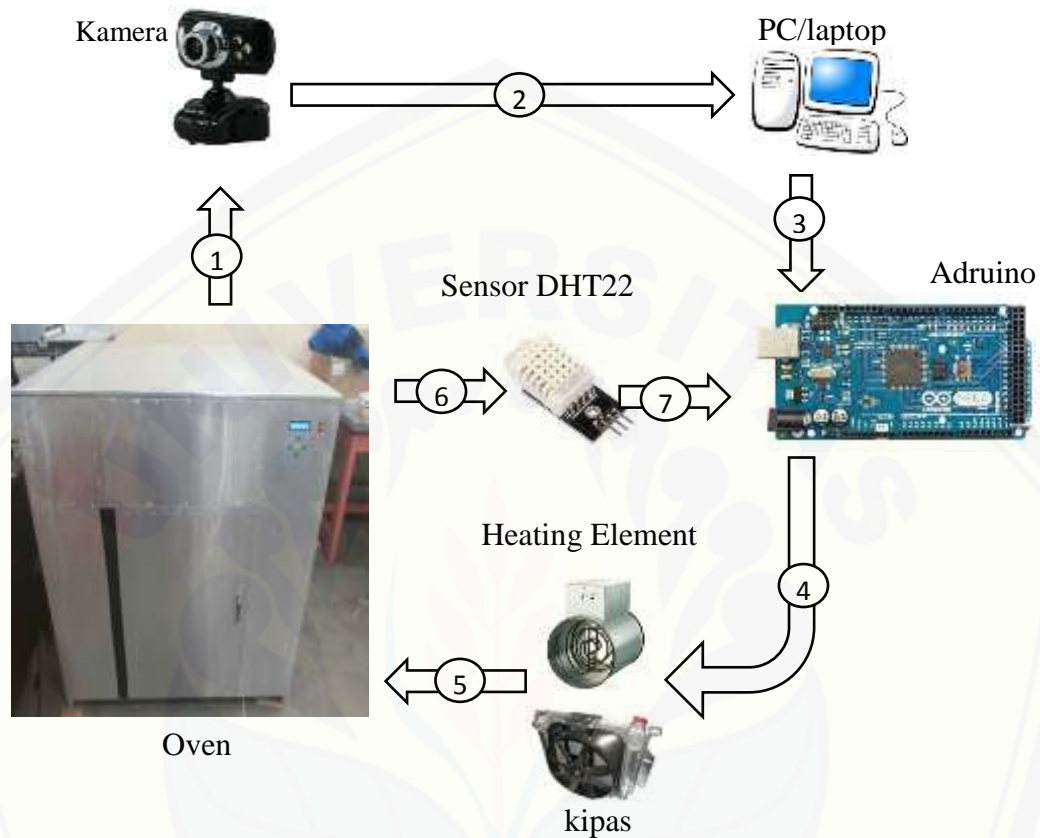
Rangkaian LCD dan tombol



Penempatan kamera, LED dan daun tembakau

Gambar 3.4 Oven Tembakau

### 3.3.3 Rancangan Bentuk dan Cara Kerja Alat



Gambar 3.5 Diagram alur alat pengering tembakau otomatis

Keterangan :

1. Pengambilan data citra tembakau
2. Pengiriman data citra tembakau untuk di *extraksi* di komputer
3. Pengiriman data perbandingan binary untuk di olah di arduino
4. Arduino mengontrol suhu
5. Keuaran panas dan udara dari kipas
6. Pembacaan sensor suhu
7. Pengiriman data hasil pembacaan sensor suhu

Gambar 3.5 menjelaskan bahwa informasi citra tembakau diambil menggunakan kamera. Kamera yang digunakan berupa *wabcam* dengan resolusi 16Mp. Informasi citra berupa perubahan warna dari tembakau selama pengeringan. Informasi citra yang diambil dari kamera dikirim ke komputer untuk diolah. Setelah informasi citra selesai diolah oleh komputer menghasilkan data *binary* nantinya akan dikirim ke arduino untuk diproses. Arduino akan mengolah data *binary* untuk dibandingkan dengan *setpoint*. Dari jumlah *binary* tembakau dapat diketahui tembakau sudah kering atau belum. Apabila tembakau masih belum kering maka arduino akan mengontrol aktuator *heater* dan kipas berdasarkan pembacaan sensor DHT22 untuk mengetahui suhu. Indikator tembakau dinyatakan kering yaitu berwarna *orange*. Apabila tembakau sudah dinyatakan kering maka Arduino akan mematikan mesin pengering.

### 3.3.4 Desain Elektrik

#### 3.3.4.1 Rangkaian Sensor DHT22

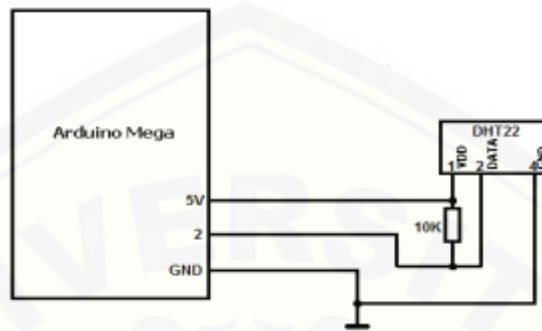
Sensor DHT22 dihubungkan dengan mikrokontroler Arduino Uno, yang berperan sebagai perangkat akuisisi data. Sensor DHT22 berkomunikasi dengan mikrokontroler melalui *single wire* atau *one wire two way serial interface*. Komunikasi antara sensor DHT22 dan Arduino Mega menggunakan pin 23 digital input. Sementara catu daya sensor dihubungkan dengan pin *power 5V* dan *ground*. Gambar 3.6 menunjukkan tampilan interkoneksi sistem antara Arduino Mega dengan sensor DHT22.

*Listing* program arduino :

```
1| #include <DHT.h>
2| #define DHTPIN 23
3| #define DHTTYPE DHT22
4| DHT dht(DHTPIN, DHTTYPE);
5| float temp= dht.readTemperature();
```

Pada program arduino terdapat *library* sendiri untuk sensor DHT. *Listing* program pada baris ke-1 merupakan *library* sensor DHT, pada baris ke-2 merupakan inisialisasi dari variabel DHTPIN sebagai pin 23 dari sensor DHT, pada baris ke-3 merupakan inialisasi DHTTYPE sebagai tipe sensor dari DHT, pada alat

ini menggunakan sensor DHT22 atau dikenal dengan AM2302. Pada baris ke-4 penentuan tipe dan pin untuk sensor DHT. Pada baris ke-5 menyimpan nilai suhu yang terbaca sensor kedalam variabel *temp* yang bertipe data *floating point* (desimal).

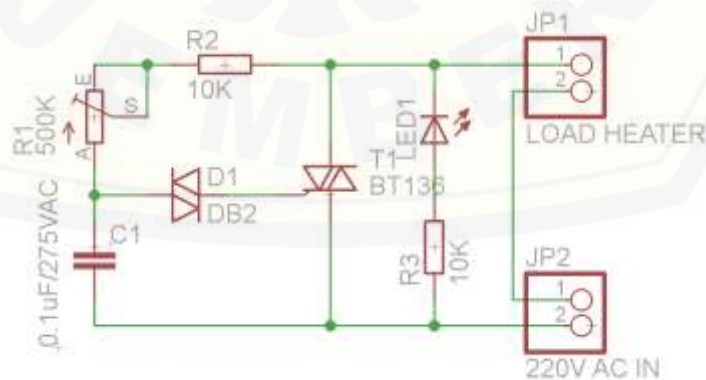


Gambar 3.6 Rangkain DHT22 dengan Arduino

### 3.3.4.2 Rangkain Driver Heater

Pada penelitian ini tungku listrik digunakan untuk pemanas ruangan/*drying*. Proses *drying* secara umum dapat diartikan sebagai proses menghilangkan sejumlah air (dalam jumlah sedikit) yang terkandung dalam suatu material. Pada proses *drying*, air biasanya dihilangkan dalam wujud uap dengan bantuan gas panas.

Alat yang digunakan dalam proses pembuatan gas panas yaitu *heating element* dengan bantuan *fan*/kipas agar udara bebas terkompresi (*compressed air*) sehingga menjadi gas panas. Udara terkompresi hasil dari *compressor* akan masuk ke ruang *oven*.

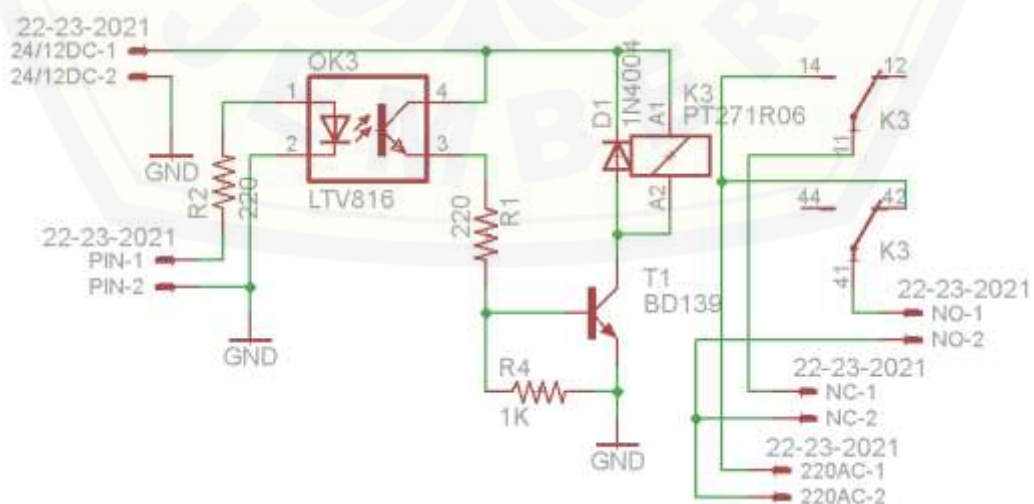


Gambar 3.7 Rangkain Dimmer

Desain alat pemanas udara ini menggunakan element dengan daya 190 watt dan menggunakan tegangan AC 220 volt. *Fan* untuk mengkompres udara menggunakan *fan* dengan tengan dc 12 volt.

Untuk mengatur besar kecilnya intensitas element pemanas, dibutuhkan rangkain kontrol *drimer*. Komponen pengendali *drimer* berupa potensio 500K dan TRIAC BT136. Sistem pengendalian tegangan AC yaitu menggunakan servo untuk memutar potensio (500K) pada *dimer* dengan *mapping* PWM (*pulse width modulation*) masukan 0-255 ke dalam bentuk sudut servo (0-180). Semakin besar lebar pulsa maka akan semakin besar sudut servo sehingga semakin besar tingkat kepanasannya yang dikeluarkan, sebaliknya semakin kecil lebar pulsa maka tingkat kepanasannya akan semakin kecil.

Alat pengendali *Element Heater* ini menggunakan sistem PWM didalam mengendalikan tegangan AC, terdapat mikrokontroller ATmega2560 sebagai *device* pemberi sinyal PWM. Besar nilai PWM sama dengan PWM untuk kontrol kipas untuk pemanas. Penggunaan servo hanya berfungsi untuk memutar tuas potensio untuk mengendalikan TRIAC BT136 yang mana dapat mengendalikan tingkat pemanas element AC 220 volt. Agar *heater* tidak ada kebocoran arus ketika digunakan makan dibuat rangkain pemutus arus AC seperti Gambar 3.8. Ketika PWM bernilai 0 makan relay akan memutus arus AC sedangkan apabila PWM lebih besar dari 0 maka *relay* akan aktif.



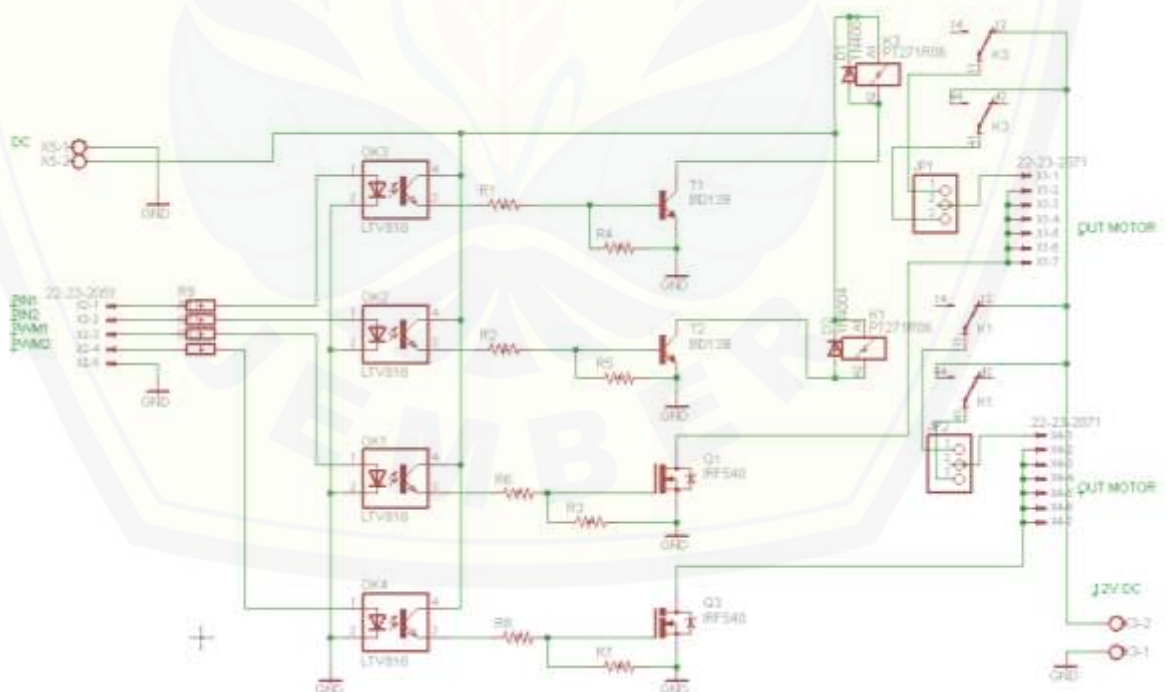
Gambar 3.8 Rangkain *Relay Dimmer*

### 3.3.4.3 Rangkain Driver Motor Ventilasi

Motor DC yang dipakai pada penelitian ini adalah kipas angin motor DC. Seperti namanya, kipas angin motor DC ini berfungsi sebagai aktuator yang menyalurkan udara dari oven. Pengendalian motor DC ini menggunakan PWM dari arduino. Kipas angin motor DC dapat dilihat dalam Gambar 3.9.



Gambar 3.9 Kipas angin motor DC



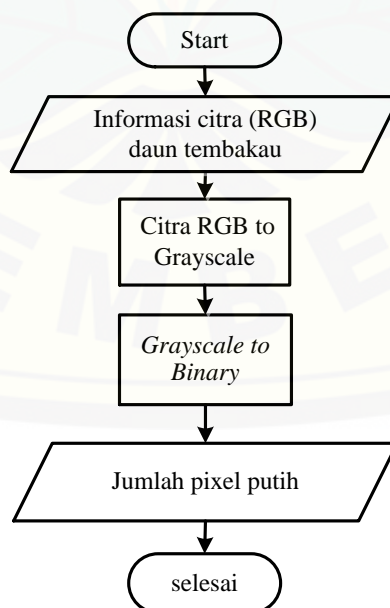
Gambar 3.10 Rangkain Driver Motor

### 3.4 Perancangan Perangkat Lunak

#### 3.4.1 Sistem Pengolahan Citra

Proses pengeringan dengan sistem oven terdapat beberapa tahap perubahan warna daun tembakau. Perubahan warna dari daun tembakau selama proses pengeringan mulai dari hijau, kuning dan *orange*. Dalam pengamplifikasinya sistem hanya mendeteksi jumlah warna hijau saja atau kuning saja. Untuk mendeteksi warna kuning, semakin banyak warna kuning (semakin banyak jumlah pixel warna putih citra biner) maka daun tembakau semakin kering. Sedangkan untuk mendeteksi warna hijau, semakin sedikit warna hijau (semakin sedikit jumlah pixel warna putih citra biner) maka daun tembakau semakin kering. Daun yang baru dipanen memiliki warna hijau dan memiliki warna kuning apabila telat dipanen.

Dalam pengolahan citra dari warna daun hijau dengan nilai RGB tertentu diubah ke tingkat *grayscale* terendah yang berarti citra hanya mengandung informasi hitam dan putih pada pixel penyusunnya yang biasa disebut citra biner. Salah satu operasi biner adalah mendapatkan jumlah pixel dari suatu citra yang sudah ditentukan. Jumlah pixel biner sebagai indikator tingkat kematang dari preses pengovenan.



Gambar 3.11 *Flowchart* Pengolahan Citra Digital



Pada proses pengolahan citra digital daun tembakau, ada beberapa tahapan untuk mengolah citra hasil pendeteksian oleh *webcame* dalam bentuk citra RGB hingga menjadi citra biner sehingga dapat diproses oleh mikrokontroler.

a. Input citra

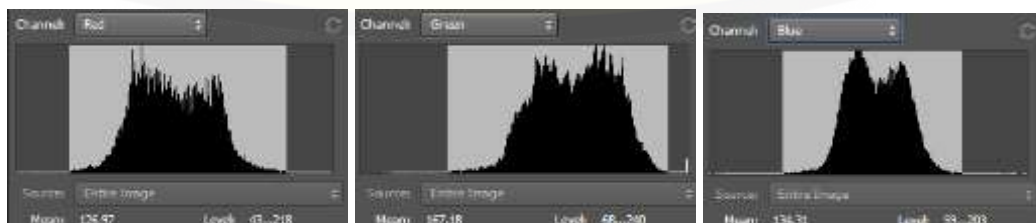
Citra yang di inputkan adalah citra daun tembakau yang dengan model RGB yang terdeteksi oleh *webcame*. Selanjutnya citra RGB di konversi ke dalam model warna citra *grayscale*. Sehingga hanya memiliki satu nilai keabuan dalam tiap piksel.

b. Indek RGB

Seperti diketahui bahwa tiap citra yang berwarna memiliki nilai indeks warna RGB yang bermacam-macam. Perbedaan prosentase indeks RGB membuat suatu citra menjadi berwarna merah, hijau, biru, kuning dan sebagainya. Semakin tinggi indeks warnanya, maka citra tersebut akan semakin terang. Begitu pula sebaliknya, semakin kecil nilai indeks warnanya, maka citra akan semakin gelap. Dalam penelitian ini, ketiga nilai indeks tersebut akan dipisahkan dan akan dijadikan parameter dalam melakukan klasifikasi. Tampak pada Gambar 5 citra latih ke 120 bahwa ekstraksi G lebih terang dibanding R dan B yang berarti nilai indeks RGB nya tinggi.



Gambar 3.12 Warna RGB daun tembakau



a

b

c

Gambar 3.13 a) Histogram Red, b) Histogram Green dan c) Hitogram Blue

Tabel 3.3 Indeks warna RGB daun tembakau

Warna	Nilai Indeks Warna	
	Minimum	Maksimum
Red	43	218
Green	68	240
Blue	59	203

## c. Nilai Mean Dan Standar Deviasi

Ekstraksi fitur histogram yang dilakukan dalam penelitian ini adalah ekstraksi nilai rata-rata (mean) dan nilai penyimpangan (standar deviasi) tiap-tiap nilai indeks RGB, sehingga didapatkan 3 nilai untuk mean dan 3 nilai untuk standar deviasi untuk masing-masing citra. Nilai mean menunjukkan nilai intensitas rata-rata tiap indeks warna RGB. Sedangkan nilai standar deviasi menunjukkan penyimpangan maksimum dari nilai intensitas rata-rata tiap indeks warna RGB. nilai mean dan standar deviasi memperlihatkan kualitas tiap-tiap daun tembakau.

Rumus rata-rata (Mean):

$$\mu = \frac{x_1 + x_2 + \dots + x_n}{N}$$

dimana:

$\mu$  = Rata-rata (Mean)

$x_n$  = Sekelompok nilai pixel

$N$  = Total keseluruhan pixel

Rumus standar deviasi :

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$$

dimana :

$S$  = Standart deviasi

$n$  = Total keseluruhan pixel

$X_i$  = nilai pixel yang ke  $i$

$\bar{X}$  = Rata-rata (Mean)



Gambar 3.14 Nilai Mean dan Standart deviasi warna

#### d. Normalisasi

Biasanya jumlah pixel pada citra hasil dari pembacaan *wabcame* sangat besar tergantung dari resolusi kamera, sehingga bila ditampilkan di layar bisa melebihi batas layar yang diizinkan. Pada penelitian ini *wabcame* memiliki resolusi 640x480 pixels dengan jumlah seluruh pixel  $N=307200$ . Untuk kebutuhan tampilan pada layar, proses normalisasi dilakukan terlebih dahulu. Tujuan dari normalisasi citra adalah mengurangi resolusi citra yang berguna saat proses pengenalan citra dan juga meningkatkan akurasi pengenalan. Proses yang digunakan pada tahap normalisasi ini adalah proses penskalaan citra.

*Scaling* atau penskalaan pada citra disebut juga *image zooming*, yaitu proses untuk mengubah ukuran citra asli (*zoom in* / memperbesar ukuran citra asli atau *zoom out* / memperkecil ukuran citra asli). Proses perubahan ukuran resolusi citra dibutuhkan untuk menyesuaikan resolusi citra masukan dengan resolusi citra *template*. Perubahan resolusi dengan ukuran awal 640x480 menjadi 320x240 pixel sehingga mengalami penurunan jumlah pixel dari 307200 menjadi 76800.

e. Citra *Grayscale*

Citra grayscale yaitu citra yang setiap pixelnya merupakan sampel tunggal, yaitu informasi intensitas. Citra jenis ini hanya terbentuk dari warna abu-abu yang memiliki intensitas berbeda. Untuk melakukan perubahan suatu gambar *fullcolor* (RGB) menjadi suatu citra *grayscale* (gambar keabuan), metode yang umum digunakan, yaitu:

$$(R + G + B)/3$$

dimana :

R : Unsur warna merah

G : Unsur warna hijau

B : Unsur warna biru

Nilai yang dihasilkan dari persamaan diatas akan di *input* ke masing-masing unsur warna dasar citra *grayscale*. Proses perubahan citra RGB ke *grayscale* ditunjukkan pada Gambar 3b



a) Citra RGB

b) Citra *Grayscale*

Gambar 3.15 Perubahan citra RGB ke *Grayscale*

f. *Thresholding*

Pada operasi ini nilai pixel yang memenuhi syarat ambang batas dipetakan ke suatu nilai yang di kehendaki. Dalam hal ini syarat ambang batas dan nilai yang di kehendaki disesuaikan dengan kebutuhan.

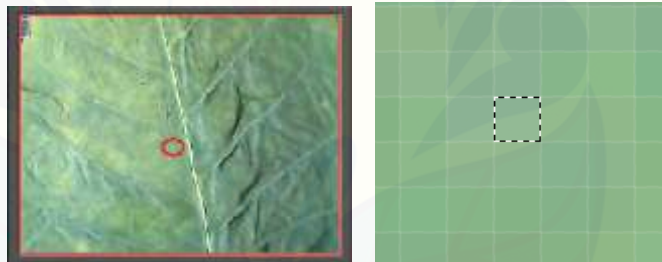
*Thresholding* digunakan untuk memisahkan bagian citra yang dibutuhkan, menghilangkan bagian citra yang tidak dibutuhkan. Dalam pelaksanaannya *thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara intensitas objek objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*.

*Thresholding* digunakan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang berada pada *range threshold*  $T_1$  dan  $T_2$  sebagai latar depan dan yang diluar *range* dari nilai *threshold*  $T_1$  dan  $T_2$  sebagai latar belakang atau sebaliknya.

g. Proses Segmentasi Citra

Segmentasi citra adalah proses pengolahan citra yang bertujuan memisahkan wilayah (region) objek dengan wilayah latar belakang agar objek mudah dianalisis dalam rangka mengenali objek yang banyak melibatkan persepsi visual.

Sebelum proses segmentasi citra akan ditentukan dahulu nilai *threshold* (T) menggunakan metode *thresholding*. Penentuan *Threshold* membutuhkan bantuan *user* untuk memasukkan nilai ambang dengan mengambil *sample* warna dari pixel tertentu sebagai *Threshold*.



Gambar 3.16 Pengambilan *sample* pixel warna hijau dengan nilai R=143, G=184 dan B=152 pada koordinat  $I_{rgb}(147,113)$

Dengan pengambilan sample warna pada pixel tertentu maka akan diperoleh nilai masing warna R, G dan B. Simple warna pada koorinat tersebut nantinya di konversi menjadi warna *grayscale* dengan rumus:

$$I_g(x,y) = \frac{I_R(x,y) + I_G(x,y) + I_B(x,y)}{3}$$

Contoh dengan mengambil *sample* seperti pada Gambar 3.16 dengan nilai R=143, G=184 dan B=152 maka diperoleh nilai  $I_g(x,y)$  sebagai berikut :

$$I_g(147,113) = \frac{143+184+152}{3} = 160$$

Dimana  $I_R(x,y)$  merupakan nilai pixel *Red* pada titik  $(x,y)$ ,  $I_G$  merupakan nilai pixel *Green* pada titik  $(x,y)$  dan  $I_B$  merupakan nilai pixel *Blue* pada titik  $(x,y)$  sedangkan  $I_g$  adalah nilai pixel *grayscale* pada titik  $(x,y)$ .

Nilai  $I_g$  dijadikan nilai *Threshold*  $T_0$  bagi citra keseluruhan. Citra *grayscale* yang mempunyai nilai yang sama dengan  $T_0 = 160$  akan dijadikan warna putih yang merupakan representasi dari sebuah pixel yang warnanya diloloskan. Sedangkan selain citra yang nilainya tidak sama dengan  $T_0$  dijadikan warna hitam yang merupakan representasi dari pixel yang warnanya tidak diloloskan.

$$I_g(x,y) = \begin{cases} 255 & I_g(x,y) = T_0 \\ 0 & I_g(x,y) \neq T_0 \end{cases}$$

Dengan menggunakan fungsi *InRange()* pada *Visual Studio* maka *range Threshold* dapat di perlebar dengan memasukkan nilai  $T_k$ . Nilai  $T_k$  digunakan untuk konstanta penjumlahan dan pengurangan terhadap *Threshold*  $T_0$  sehingga diperoleh akan diperoleh nilai ambang  $T_1$  dan  $T_2$  sebagai batas atas dan batas bawah dari ambang *Threshold*.

$$T_1 = T_0 - T_k$$

$$T_2 = T_0 + T_k$$

Jadi besar *range Threshold* keseluruhan  $T_{rg}$  yaitu :

$$T_{rg} = T_2 - T_1 \text{ atau } T_{rg} = 2 T_k$$

$$I_g(x,y) = \begin{cases} 255 & T_1 \leq I_g(x,y) \leq T_2 \\ 0 & \text{diluar } T_1 \text{ dan } T_2 \end{cases}$$

Misal nilai  $T_k = 20$  maka diperoleh  $T_1$ ,  $T_2$  dan  $T_{rg}$  :

$$T_1 = 160 - 20 = 140$$

$$T_2 = 160 + 20 = 180$$

$$T_{rg} = 180 - 140 = 40$$

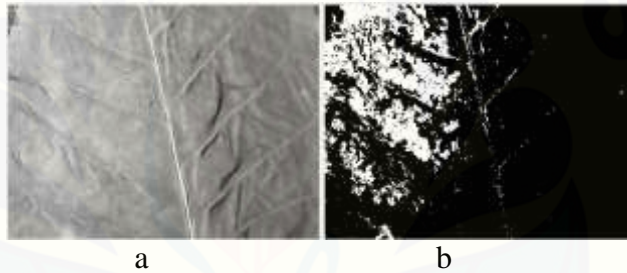
Jadi nilai intensitas semua piksel yang berada diantara *range threshold*  $T_1$  dan  $T_2$  sebagai latar depan dan yang diluar *range* dari nilai *threshold*  $T_1$  dan  $T_2$  sebagai latar belakang.

#### h. Citra Biner

Konversi dari citra *Grayscale* ke citra biner dilakukan dengan operasi pengambangan (*thresholding*). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap *pixel* ke dalam 2 kelas, hitam dan putih. Setiap *pixel* di dalam citra dipetakan ke dua nilai, 1 atau 0 dengan fungsi pengambangan:

$$I_{Bin}(x, y) = \begin{cases} 1 & T_1 \leq I_g(x, y) \leq T_2 \\ 0 & \text{diluar } T_1 \text{ dan } T_2 \end{cases}$$

yang dalam hal ini,  $f_{Bin}(i, j)$  adalah citra biner, dan  $T$  adalah nilai ambang yang dispesifikasikan. Dengan operasi pengambangan tersebut, objek dibuat berwarna terang (1 atau putih) sedangkan latar belakang berwarna gelap (0 atau hitam).



Gambar 3.17 a) Citra *grayscale* dan b) Citra biner

Citra biner yang berwarna terang yang mempunyai nilai biner 1 merepresentasikan citra warna tertentu yang terdeteksi sesuai *threshold*. Citra biner yang bernilai 1 dari seluruh citra dijumlah kemudian dikirim ke mikrokontroller melalui komunikasi serial.

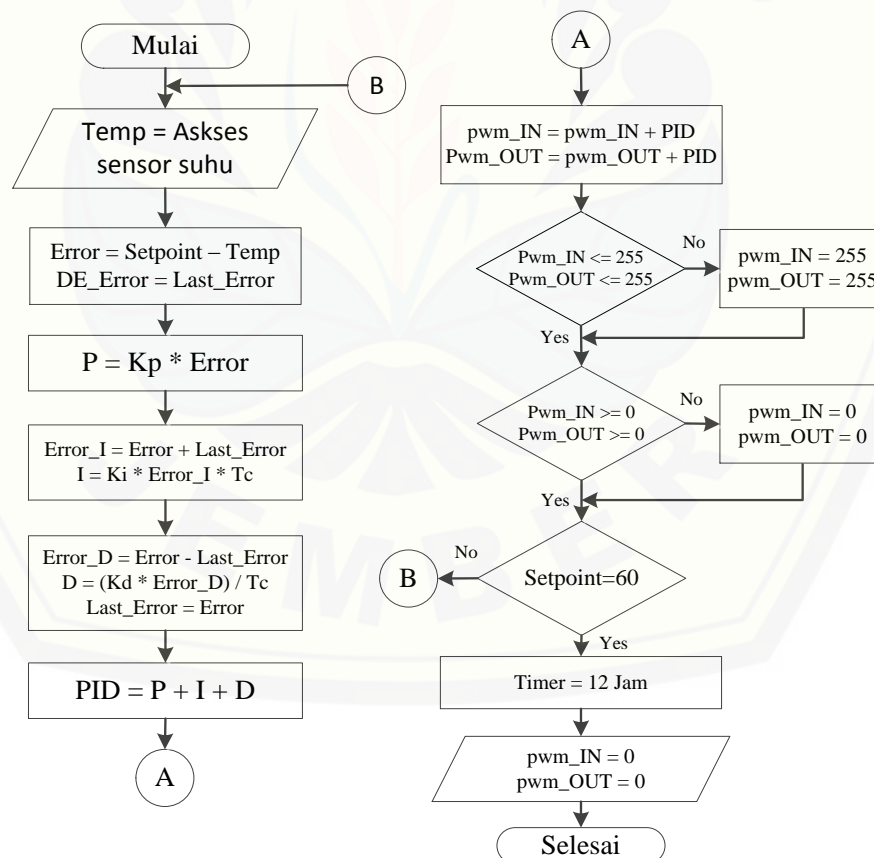
Pada pengolahan citra digital menggunakan *software* Microsoft Visual Studio. Menggunakan *platform* EmguCV yaitu *cross platform* yang terdapat dalam .NET untuk *library* pengolahan citra pada Intel OpenCV. EmguCV ini mengikuti fungsi yang terdapat pada OpenCV yang diambil dari .NET. Jadi *compatible* dengan bahasa pemrograman VB, VC++, C#, IronPython dan sebagainya. Program ini bersifat *opensource*. Pada pengolahan citra digital dibagi menjadi tiga yaitu *color image*, *black and white image* dan *binary image*.

### 3.4.3 Sistem Kontrol PID

#### a. Perancangan Kontrol PID

Perancangan kontrol PID mengacu pada Persamaan 2.4. Untuk membuat kontrol PID diperlukan algoritma program kontrol PID agar menjadi kontrol yang sesuai dengan yang diharapkan. Algoritma ini nantinya akan dimasukkan kedalam program arduino. Pada Gambar 3.14 merupakan *flowchart* dari algoritma program kontrol PID untuk rancang bangun kontrol PID pada alat pengering tembakau menggunakan arduino.

Hasil respon kendali PID adalah salah satunya ditentukan oleh parameter  $K_p$ ,  $K_i$  dan  $K_d$ . Metode yang digunakan untuk pengujian adalah metode coba-coba (*trial and error*) beberapa sampel nilai, dimana nilainya akan disimpan pada SD card dalam bentuk angka sehingga nantinya dapat di analisa dalam bentuk grafik.



Gambar 3.18 *Flowchart* Program Kontrol PID



Hal yang paling penting dari kontrol PID adalah menentukan parameter  $K_p$ ,  $K_i$  dan  $K_d$ . Pada program Arduino,  $pwm\_IN$  sebagai keluaran dari kontrol PID untuk mengatur intensitas panas *heater* dan kecepatan putaran kipas meniup udara panas yang masuk oven, sedangkan  $pwm\_OUT$  sebagai keluaran kontrol PID mengatur kecepatan kipas untuk meniup udara keluar dari oven. Pada penelitian ini untuk menentukan parameter-parameter PID menggunakan metode manual tuning yaitu dengan cara *trial and error*.

b. Metode *Manual Tunning*

Metode *manual tuning* adalah metode untuk mencari nilai parameter-parameter kontrol PID dengan cara *trial and error*. Dalam pencarian nilai parameter PID mengacu pada masing-masing karakteristik dari setiap parameter yang dicari berdasarkan pada Tabel 2.4 pada BAB sebelumnya. Pada penelitian ini dilakukan *manual tuning* parameter PID dengan langkah sebagai berikut.

- 1). Memberikan *gain* parameter *proportional* sehingga *plan* memberikan nilai *output* yang mencapai *setpoint*. Namun dengan memberikan parameter ini akan meningkatkan *overshoot* pada *output* dan *error steady-state* masih tinggi.
- 2). Memberikan nilai parameter integral sehingga nilai *error steady-state* dapat dikurangi. Namun pemberian parameter ini akan meningkatkan *overshoot* dan *settling time* meningkat.
- 3). Memberikan nilai parameter *derivative* sehingga nilai *overshoot* dan *settling time* dikurangi. Namun untuk memberikan nilai parameter ini tidak diajarkan terlalu tinggi apabila respon kontrol sangat cepat karena dapat membuat sistem lebih tidak stabil.

### 3.5.3 Sistem *Fuzzy Logic*

Pada perancangan kontrol *fuzzy* pada sistem pengering tembakau dirancang dengan perhitungan secara sistematis. Dimana *logika fuzzy* akan implementasikan secara langsung kedalam sebuah *embended system* yang berupa Arduino Mega. Tahap pembuatan *fuzzy logic control* ini dilakukan melalui beberapa tahap yaitu dimulai dari tahap *fuzzyfikasi* dimana terdiri dari pembuatan *membership function*,

kemudian dilanjutkan ketahap *inferent* dimana pada tahap ini merupakan proses pembuatan *rule base* kemudian ke tahap *defuzzyfikasi* yaitu proses *output* yang diperoleh dari keseluruhan proses.

Algoritma *Fuzzy Logic* memiliki beberapa metode untuk merepresentasikan hasil logika *fuzzy* yaitu metode Tsukamoto, Sugeno dan Mamdani. Metode Sugeno yang dipilih untuk mengatur perilaku respon kontrol PID karena metode *fuzzy logic* sugeno lebih efisien dalam perhitungan dibandingkan dengan metode mamdani dan *error rate* pada metode Sugeno lebih kecil dari pada Tsukamoto. Metode Sugeno dipilih untuk mencari respon suhu tercepat dengan merubah parameter Kp, Ki dan Kd. Nilai heuristik pada algoritma *fuzzy* ini adalah fungsi optimasi yang menjadikan kontrol PID lebih baik. *Fuzzy Logic* metode Sugeno merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk IF – THEN, dimana output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear.

a. Tahapan *Fuzzyfikasi*

Tahap awal pembuatan *fuzzy* diawali dengan proses *fuzzyfikasi* dimana pada tahap ini dilakukan terhadap *error* dan *delta error* dari hasil kontrol PID, yaitu dengan membuat himpunan keanggotaan untuk *error* dan *delta error*, kemudian melakukan perhitungan keanggotaan untuk menentukan derajat keanggotaan masing-masing nilai *error* dan *delta error*. Berikut program arduino pembacaan *error* dan *delta error*.

```
error = SetPoint-temp; //error
de_error = last_error; //delta error
fuzzyRutin(error,de_error); //input fuzzy
```

Pada tahap ini terlebih dahulu ditentukan besarnya *setpoint* yang digunakan untuk membuat *membership function* dan juga pada proses perhitungan *error* dan *delta error*. Untuk besarnya nilai *setpoint* yang digunakan yaitu 32, 38, 40, 43, 49 dan 60, dimana nilai ini menjadi acuan dalam kontrol *fuzzy* agar kontrol tetap stabil.

Dalam sistem pengendalian ini bertujuan untuk mengendalikan suhu, maka kendali berdasarkan *error* ( $e$ ) suhu dan perubahan *error* suhu atau *delta error* suhu ( $de$ ). Kontroller mengindera sinyanya dari sensor suhu dan menerima *input* berupa suhu referensi dari hasil pengolahan citra yang berupa jumlah *pixel* dikonversi menjadi suhu referensi. Kontroller memproses selisih suhu aktual dan suhu referensi menjadi  $e(t)$  dan memcatanya. Kontroller ini kemudian memproses selisih *error* ini dengan nilai *error* suhu sebelumnya  $e(t-1)$  dan menjadikannya perubahan *error* suhu  $de(t)$ . Hasil perhitungan *error* dan *delta error* ini kemudian diproses sesuai dengan tahapan pada blok sistem kendali *fuzzy*. Berikut program membuat himpunan keangotaan *error* dan *delta error* arduino.

```
//member function ERROR
mf[0][0] = FuzzyMember(err,-3, -3, -2); //NB
mf[1][0] = FuzzyMember(err,-2.5, -1.5, -0.5); //NM
mf[2][0] = FuzzyMember(err,-1, 0, 1); //Z
mf[3][0] = FuzzyMember(err, 0.5, 1.5, 2.5); //PM
mf[4][0] = FuzzyMember(err,2, 3, 3); //PB
//mamber function DELTA ERROR
mf[0][1] = FuzzyMember(derror,-3, -3, -2); //NB
mf[1][1] = FuzzyMember(derror,-2.5, -1.5, -0.5); //NM
mf[2][1] = FuzzyMember(derror,-1, 0, 1); //Z
mf[3][1] = FuzzyMember(derror,0.5, 1.5, 2.5); //PM
mf[4][1] = FuzzyMember(derror,2, 3, 3); //PB

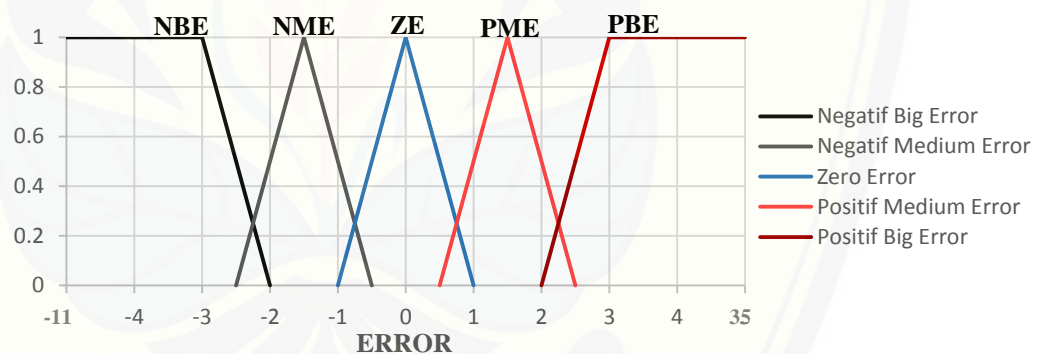
if (err<= -3){mf[0][0] = 1;}
else if (err>=3){ mf[4][0] = 1;}
if (derror<= -3){mf[0][1] = 1;}
else if (derror>=3){mf[4][1] =1;}
```

Setelah membuat himpunan keangotaan *error* dan *delta error*, kemudian menentukan derajat keangotaan masing-masing nilai *error* dan *delta error*.

```
//mencari alfa predikat FUZZY
int x1 = 1;
for(int a1 = 0; a1 <=4; a1 = a1 +1)
for(int a2 = 0; a2 <=4; a2 = a2 +1)
{
    apred[x1] = min(mf[a1][0],mf[a2][1]);
    x1 = x1 + 1;}
```

Pada kendali *fuzzy logic* nilai *input* berupa *error* yang didapat dari besarnya nilai *setpoint* dikurangi dengan nilai suhu aktual yang terbaca oleh sensor suhu. Untuk besarnya nilai *setpoint* berubah-ubah berada di rentang 25°C sampai 60°C.

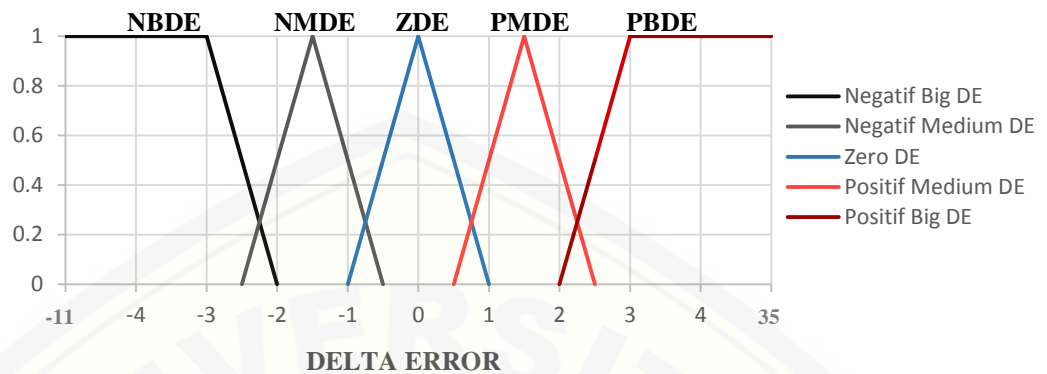
Dari penentuan *input error* terlebih dilakukan beberapa kemungkinan yang terjadi untuk besarnya nilai *error* yang nantinya dijadikan sebagai batas terbesar dan juga terkecil untuk daerah batas *crisp*. Beberapa *error* yang terjadi yaitu dengan nilai *error* tertinggi sebesar 35 ketika pembacaan sensor suhu masih pada kondisi suhu ruangan 25°C. Sedangkan untuk kemungkinan nilai *error* terendah sebesar -11 dimana hal ini terjadi ketika pembacaan sensor suhu sebesar 60°C dengan *setpoint* 49°C. Untuk pemberian *label* pada *membership function* pada *input error* yaitu menggunakan NBE atau *Negatif Big Error*, NME atau *Negatif Medium Error*, ZE atau *Zero Error*, PME atau *Positif Medium Error* dan PBE atau *Positif Big Error*.



Gambar 3.19 *Membership function input error*

Untuk nilai *delta error* didapat dari nilai turunan terhadap waktu ( $de/dt$ ) atau nilai *error* sebelumnya. Daerah batas *crisp* ditentukan berdasarkan beberapa kemungkinan yang terjadi. Dari beberapa percobaan yang dilakukan untuk nilai *delta error* tertinggi yaitu sebesar sebesar 35 sedangkan untuk kemungkinan nilai *error* terendah sebesar -11 dimana hal ini terjadi ketika pembacaan sensor suhu sebesar 60°C. Untuk pemberian *label membership function* pada *input error* yaitu menggunakan NBDE atau *Negatif Big Delta Error*, NMDE atau *Negatif*

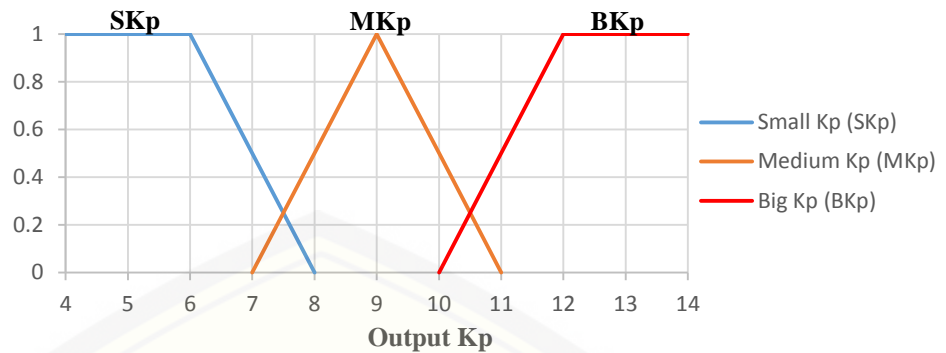
*Medium Delta Error*, ZDE atau *Zero Delta Error*, PMDE atau *Positif Delta Medium Error* dan PBDE atau *Positif Big Delta Error*.



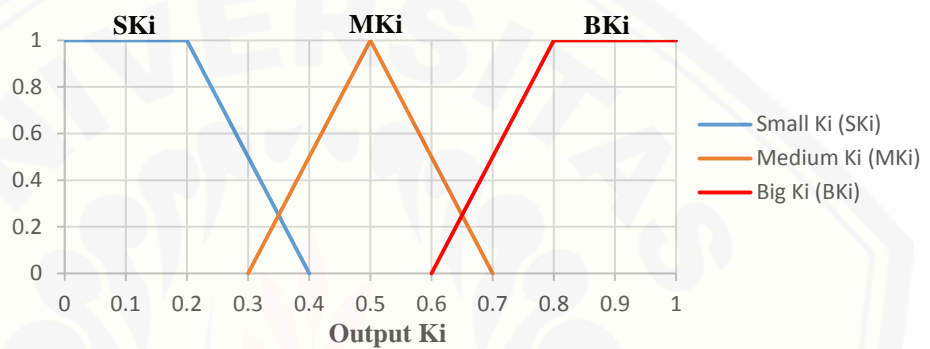
Gambar 3.20 *Membership function input delta error*

Pada *membership function control*, untuk menentukan daerah batas *crisp* berdasarkan dari *output* yang nantinya diinginkan. Karenan *setpoint* suhu berubah-ubah maka digunakan kontrol *fuzzy logic* ini difungsikan untuk *tuning* nilai  $K_p$ ,  $K_i$  dan  $K_d$  pada kontrol PID yang sudah dilakukan percobaan sebelumnya. Jadi *output* yang diinginkan dari *fuzzy logic* ini ada tiga yaitu *output* nilai  $K_p$ ,  $K_i$  dan  $K_d$ . *Output* dari ketiganya nantinya akan dijadikan konstanta dari kontrol PID. Besarnya nilai *error* dan *delta error* akan mempengaruhi nilai dari konstanta  $K_p$ ,  $K_i$  dan  $K_d$ .

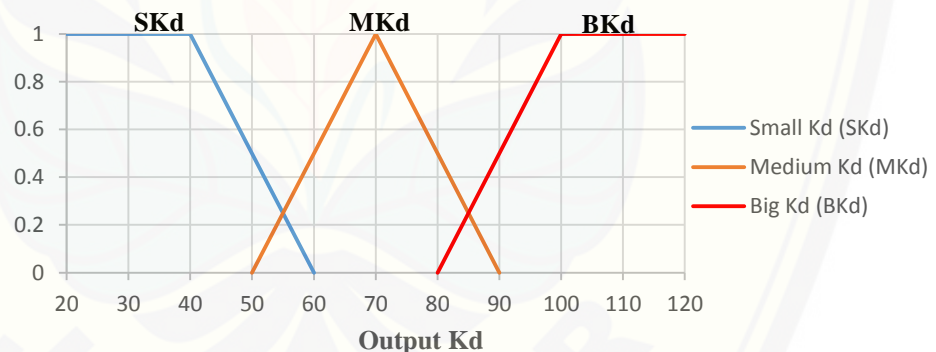
Dari pengujian kontrol PID sebelumnya diperoleh data nilai  $K_p$ ,  $K_i$  dan  $K_d$ . Untuk *output*  $K_p$  daerah batas *crisp* yang digunakan yaitu 4 hingga 14. Sedangkan untuk pemberian *label membership function* pada *output*  $K_p$  yaitu menggunakan SK<sub>p</sub> atau *Small K<sub>p</sub>*, MK<sub>p</sub> atau *Medium K<sub>p</sub>* dan BK<sub>p</sub> atau *Big K<sub>p</sub>*. Untuk *output*  $K_i$  daerah batas *crisp* yang digunakan yaitu 0 hingga 1. Sedangkan untuk pemberian *label membership function* pada *output*  $K_i$  yaitu menggunakan SK<sub>i</sub> atau *Small K<sub>i</sub>*, MK<sub>i</sub> atau *Medium K<sub>i</sub>* dan BK<sub>i</sub> atau *Big K<sub>i</sub>*. Sedangkan untuk *output*  $K_d$  daerah batas *crisp* yang digunakan yaitu 20 hingga 120. Untuk pemberian *label membership function* pada *output*  $K_d$  yaitu menggunakan SK<sub>d</sub> atau *Small K<sub>d</sub>*, MK<sub>d</sub> atau *Medium K<sub>d</sub>* dan BK<sub>d</sub> atau *Big K<sub>d</sub>*.



Gambar 3.21 Membership function output Kp



Gambar 3.22 Membership function output Ki



Gambar 3.23 Membership function output Kd

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada beberapa fungsi yang bisa digunakan yaitu segitiga dan trapesium. Fungsi-fungsi ini dipilih sesuai dengan kebutuhan.

Pada representasi kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear). Kurva segitiga mempunyai himpunan yang hanya memiliki satu titik derajat keanggotaan paling tinggi (satu) sedangkan lainnya

bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah (nol). Pada fungsi segitiga ini bertujuan untuk memfokuskan input satu pada daerah agar tetap stabil. Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Fungsi ini dipilih karena pada rentang yang mempunyai nilai derajat keanggotaan paling tinggi mempunyai respon yang sama.

b. Tahap *Inferensi*

Inferensi dimulai dengan memetakan nilai derajat keanggotaan dari masing-masing *error* dan *delta error* ke sebuah tabel yang berisi basis aturan yang dipilih (*rule base*) dan menggunakan sebuah metode untuk pemilihan nilai dari derajat keanggotaan. Dengan demikian terlebih dahulu harus dibuat matrik atau tabel *rule base*. Setelah langkah ini maka proses *inferensi* selesai dan untuk selanjutnya adalah langkah *defuzifikasi* untuk menentukan besarnya nilai Kp, Ki dan Kd yang dikirim ke kontrol PID. Berikut program arduino untuk *rule base* keluaran Kp.

```
for(int rp = 1; rp <= 25; rp = rp + 1){
  switch(rp) {
    case 1 : {MKPrule(rp);BKIrule(rp);MKDrule(rp);break;}
    case 2 : {SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
    case 3 : {SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
    case 4 : {SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
    case 5 : {SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
  }
}
```

Dalam membuat *membership function* untuk masing-masing *input* dan *output*, dibutuhkan beberapa kali percobaan untuk mendapatkan hasil dari *output* kontrol yang sesuai dengan target. Beberapa percobaan yang sering mengalami kesulitan yaitu pada saat menentukan *rule base*. Hal ini dikarenakan pada tahapan tersebut merupakan pembuatan aturan dari setiap kondisi yang terjadi dalam suatu proses sehingga jika dalam pembuatan aturan atau *rule base* tidak sesuai dengan kondisi yang terjadi, maka bisa saja terjadi *output* yang dihasilkan tidak sesuai dengan target yang diinginkan.

Tabel 3.4 Rule output Kp

DE \ E	NBE	NME	ZE	PME	PBE
NBDE	MKp	MKp	BKp	BKp	BKp
NMDE	SKp	MKp	MKp	BKp	BKp
ZDE	SKp	SKp	MKp	MKp	BKp
PMDE	SKp	SKp	SKp	MKp	BKp
PBDE	SKp	SKp	SKp	MKp	BKp

BKp= Big Kp

MKp= Medium Kp

SKp= Small Kp

Tabel 3.5 Rule output Ki

DE \ E	NBE	NME	ZE	PME	PBE
NBDE	BKi	BKi	SKi	BKi	BKi
NMDE	BKi	MKi	SKi	MKi	BKi
ZDE	BKi	MKi	SKi	MKi	BKi
PMDE	BKi	MKi	SKi	MKi	BKi
PBDE	BKi	BKi	SKi	BKi	BKi

BKi= Big Ki

MKi= Medium Ki

SKi= Small Ki

Tabel 3.6 Rule output Kd

DE \ E	NBE	NME	ZE	PME	PBE
NBDE	MKd	MKd	BKd	MKd	SBd
NMDE	SKd	BKd	BKd	MKd	SKd
ZDE	SKd	BKd	BKd	MKd	SKd
PMDE	SKd	BKd	BKd	BKd	SKd
PBDE	SKd	BKd	BKd	BKd	SKd

BKd= Big Kd

MKd= Medium Kd

SKd= Small Kd

Keterangan :

NBE = Negatif Big Error

NBDE = Negatif Big Delta Error

NME = Negatif Medium Error

NMDE = Negatif Medium Delta Error

ZE = Zero Error

ZDE = Zero Delta Error

PME = Positif Medium Error

PMDE = Positif Medium Delta Error

PBE = Positif Big Error

PBDE = Positif Big Delta Error



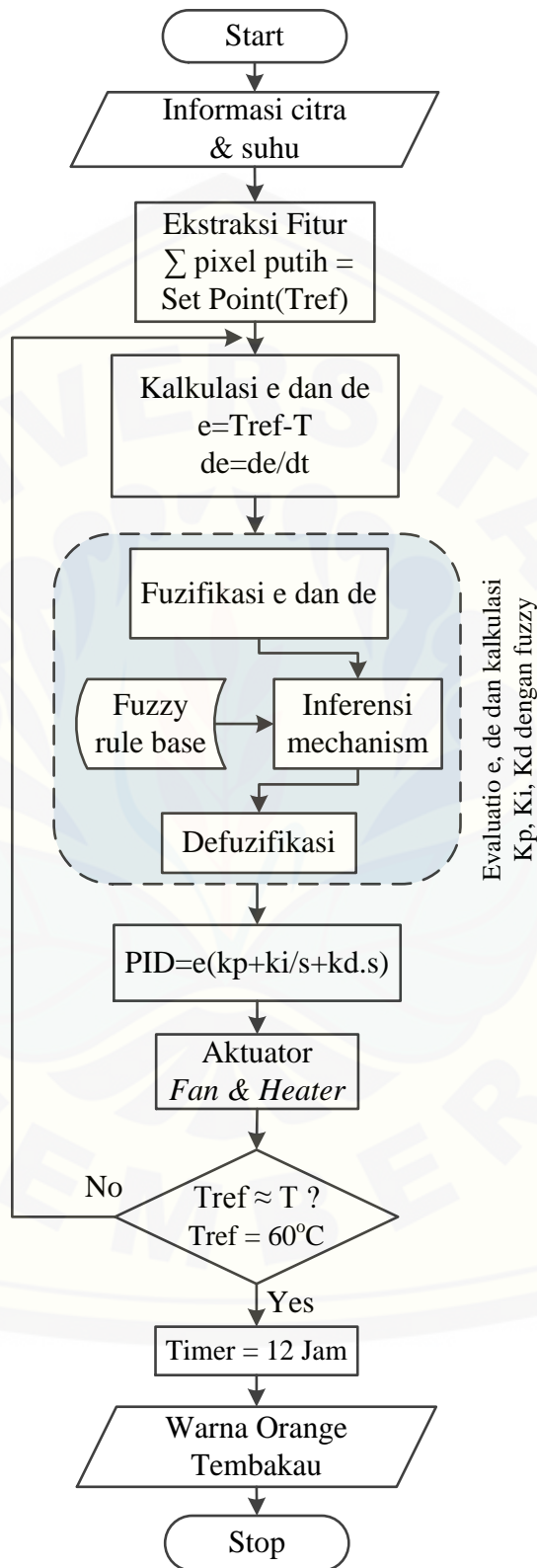
c. Tahap Defuzifikasi

*Defuzifikasi* dilakukan dengan memetakan nilai-nilai  $U$  ke himpunan *defuzifikasi* yang dinyatakan dengan persamaan-persamaan garis. Berbeda dengan *fuzifikasi*, *defuzifikasi* akan menghasilkan luasan-luasan yang dibatasi dengan tinggi masing-masing nilai  $U$  dan garis yang dihasilkan dari pemetaan ke *ruel base*. Berikut program arduino untuk *defuzifikasi* dengan output  $K_p$ .

```
//Defuzifikasi
atasP =0, bawahP = 0, zxP = 0;
for (int pp = 1; pp <= 25; pp = pp +1)
{
  atasP = atasP + (apred[pp] * zp[pp]);
  bawahP = bawahP + apred[pp];
}

zxP = atasP / bawahP;
Kp = zxP;
```

3.5 Flowchart



Gambar 3.24 Flowchat Kendali Fuzzy-PID Suhu

### 3.6 Metode Pengujian dan Pengumpulan Data

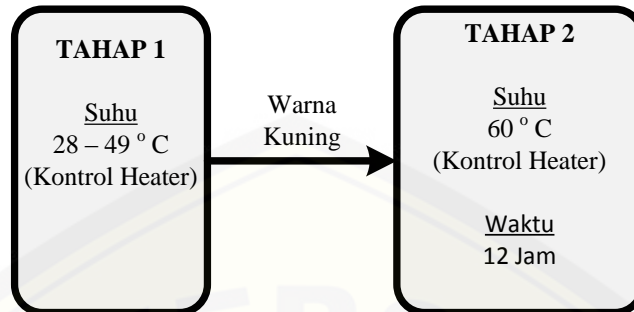
Pengujian dilakukan dengan cara menguji dari tiap-tiap komponen dari alat diantaranya arduino, kamera, DHT22, *dimmer*, kipas dan *elemen heater*. Pengujian alat pengering dilakukan dengan cara menguji pemanas pada alat pengering. Alat pemanas menggunakan elemen pemanas yang dikontrol menggunakan sistem *fuzzy*-PID untuk mengatur suhu pada alat pengering. Pengujian pengambilan informasi citra dilakukan dengan cara menguji kemampuan mengambil citra dari kamera. Pengujian pengolahan informasi citra dilakukan dengan cara mengambil citra tembakau untuk mengetahui tembakau sudah kering atau belum. Pengujian alat secara keseluruhan dilakukan dengan cara mengambil informasi citra tembakau pada alat pengering hingga tembakau dinyatakan sudah kering yaitu berwarna *orange*.

### 3.7 Metode Analisis Data

Analisis dari alat pengering tembakau otomatis dengan metode pengolahan citra digital menggunakan kontrol *Fuzzy*-PID diharapkan hasil pengeringan tembakau dalam bentuk krosok menghasilkan tembakau yang bermutu. Tembakau yang berkualitas berdasarkan warna. Dalam penggunaan citra digital diharapkan memberikan kelebihan dalam mengatur suhu berdasarkan ekstraksi jumlah pixel warna kuning atau hijau, sehingga tidak lagi menggunakan mata manusia untuk mengetahui tingkat kematangan dari daun tembakau.

Dalam proses pengeringan tembakau, selalu terjadi proses perubahan warna. Untuk mengetahui kondisi tembakau apakah sudah kering sempurna atau kering sebagian, arduino mempartisi jumlah pixel biner yang dikirim oleh PC untuk membagi menjadi beberapa *setpoint* suhu. Apabila dalam fase pengeringan dan dalam waktu yang sudah ditentukan tembakau masih terdapat warna hijau atau kuning maka proses pengeringan bisa dihentikan. Apabila dalam kondisi seperti ini masih dilanjutkan dikhawatirkan akan merusak tembakau itu sendiri. Tentunya dalam menentukan pembatasan waktu telah dilakukan percobaan sebelumnya dalam fase pengikatan warna. Maka dari itu dalam tahap 2 pengeringan dibatasi

waktu selama 12 jam untuk memproses. Apabila melebihi batas waktu maka sistem akan tidak bekerja.



Gambar 3.25 Diagram Blok Fase Pengeringan Sistem

Penggunaan kontrol *Fuzzy*-PID diharapkan mampu mengurangi kesalahan dan kerusakan daun tembakau selama proses pengeringan. Dengan menggunakan kelebihan masing-masing kontrol *Fuzzy* dan PID sehingga dapat mempermudah dalam mengatur suhu dalam *oven* karena dalam proses pengeringan daun tembakau sangat kritis terhadap perubahan suhu.

Data untuk pengujian pengeringan tembakau menggunakan satu lembar daun tembakau untuk diambil datanya sebagai *sample* melalui pengolah citra digital berdasarkan perubahan warna daun tersebut.

## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil yang telah diperoleh dalam penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem kontrol *Hybrid Fuzzy*-PID yang digunakan sebagai kontrol kestabilan suhu oven (*curing-barn*) menghasilkan tingkat kestabilan yang cukup bagus dibandingkan hanya menggunakan kontrol PID saja seperti yang ditunjukkan grafik respon kendali *Hybrid Fuzzy*-PID pada Gambar 4.7. Sistem kontrol *Hybrid* dapat memperbaiki *rise time* ketika *setpoint* 60°C.
2. Sistem pengolahan citra digital sangat efektif dalam membantu menganalisa proses pengeringan tembakau menjadi sistem pengeringan otomatis.
3. Dalam proses pengeringan tembakau, tembakau yang baik untuk dioven yaitu tembakau yang masih hijau basah.

### 5.2 Saran

Dengan harapan penelitian ini dapat membuahkan hasil yang maksimal dan menjadi lebih baik lagi di kemudian hari, penulis memberikan saran sebagai berikut:

1. Alat pengering tembakau ini masih perlu perbaikan karena masih belum dapat mengontrol kelembapan dan kecepatan aliran udara dalam oven keran ini merupakan parameter yang diperlukan dalam proses pengeringan tembakau.
2. Karena proses pengeringan cukup lama maka dengan menggunakan pengolahan citra digital dengan menggunakan *software* Visual Studio belum efektif untuk dijadikan sistem oven otomatis karena membutuhkan perantara PC yang selalu aktif untuk mengolah citra.

**DAFTAR PUSTAKA**

- Abdallah, F. 1970. Canto bacco quality be measured. Lockwood Publishing Company, Inc., New York. 74pp
- Anonim. 2010a. Kebijakan Pengembangan Industri Hasil Tembakau (IHT) dan Pemanfaatan Penggunaan Dana Bagi Hasil Cukai Hasil Tembakau. Ditjen Industri Agro dan Kimia. Kementerian Perindustrian, Jakarta.
- Aosong Electronics Co, (2012), *Temperature and Humidity Module. DHT22.Product Manual*, Guangzhou, China.
- Damardjati, D.S., Trim, D.S., Haryono (1992). Improving Rice Quality by Using A Solar Supplemented Dryer for Paddy Drying. *Proceeding of 14th Asean Seminar on Grain Post Harvest Technology, Manila Philipinnes*, pp. 101 - 112.
- Ichsani, D., Setiawan, A., Rohadi, Y. (2001). Drying Process of Cloves Using Solar Dryer Combined With A Heat Exchanger. *The International Conference on Renewable Energy for Regional Development, Denpasar, Bali, 28-31 August*.
- Indrajit, Richardus E., Djokopranoto, Richardus., (2003), *Manajemen Persediaan*, PT Gramedia Widiasaranan Indonesia, Jakarta.
- Haryati N. 2011. Analisis permintaan dan penawaran tembakau Besuki Na-Oogst di kabupaten Jember [tesis]. Jember: Sekolah Program Pascasarjana, Universitas Jember.
- <http://artikel-teknologi.com/macam-macam-kipas-fan/> (terakhir dilihat 26 November 2016).
- <http://penjualheater.blogspot.co.id/p/tentang-heater.html> (terakhir dilihat 26 November 2016).
- <http://www.hotwatt.com/tastraight.htm> (terakhir dilihat 26 September 2016 ).
- Kusumadewi, Sri, 2003, “*Artificial Intelegence Teknik dan Aplikasinya*”. Jogjakarta: Graha Ilmu
- Logika\_Fuzzy, [http://k12008.widyagama.ac.id/ai/diktatpdf/Logika\\_Fuzzy.pdf](http://k12008.widyagama.ac.id/ai/diktatpdf/Logika_Fuzzy.pdf), 19 November 2016
- Manuel Llanos Company.1985. The quality of tobacco and its physical and chemical compcition (I). *Tabak Journal International*. 6:485-486.
- McAndrew Alasdair, (2004), *An Introduction to Digital Image Processing with Matlab. Notes for SCM2511 Image Processing 1*, School of Computer Science and Mathematics Victoria University of Technology.
- Murhawi. 1990. Teknik Budidaya Tembakau. Surabaya: Penebar Swadaya

- Peedin, G.F. 1999. Production practices of flue-cured tobacco. *In Tobacco Production, Chemistry, and Technology*, D. Layten Davis & M.T. Nielsen (eds), CORESTA, Univ. Press. Cambridge, Britain. pp: 66–69.
- Srzednicki, George, Driscoll, Robert, H., Xinghe, Niu., Shuanglin, Wang., Jinfeng, Ju. (2001). Effect of In Store Drying on Energy Use in Grain Industry In China. *The International Conference on Renewable Energy for Regional Development, Denpasar, Bali, 28-31 August*.
- SwitchDoc Labs, *iBeacon Rasp berry Pi Scanner in Python.*, <http://www.switchdoc.com/2014/08/ibeacon-raspberry-pi-scanner-python/>. Diakses: 13 Neovember 2016 jam 13:14.
- Tirtosastro, S., Hastono, A.B., Darmono. (2003). Perekayasaan Instalasi Pemanfaatan Udara Panas Buang Pada Pengovenan Tembakau Virginia. *Jurnal Littri*. Vol. 9. No. I, Maret 2003, hal. 17-24.
- Tjiptono, Fandy dan Anastasia, Diana, (1996), *Total Quality Management*, Yogyakarta : Andi Offset.
- Tso, T.C. 1972. *Physiology and biochemistry of tobacco plants*. Dowden Hutchinson and Ross, Inc., Stroudsburg. 393pp.



# **LAMPIRAN**



**E. Listing Program Arduino Mega 2560**

```

#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SD.h>
#include <SPI.h>
#include "RTClib.h"
#include <Servo.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3,
POSITIVE);
char derajat = 0xDF;
#define DHTPIN 23
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
RTC_DS1307 rtc;
File myFile;
int pinCS = 53;
Servo myservo;

float inK=00.00;
float ratainK;
float SetPoint=00.00;
float fKp=0.00, fKi=0.00, fKd=0.00;
float error = 0, de_error=0, last_error=0;
float error_I=0,error_D=0;
float out_P=0,out_I=0,out_D=0,out_PID=0;
float Tc=0.01;
int lastSetMenit;
float
rataTemp,rataOutPID,rataError,rataLastError,rataDError;
const int K2 = 4;
const int K1 = 3;
const int heat=43;
const int Up = 25;
int lastState = 0;
int IT=0;
int chk;
float hum,temp;
float T=0;
int vKipas1=0,vKipas2=0;
float pwm_IN, pwm_OUT;
int vServo=0;;
float pwm;
float atasP =0, bawahP = 0, zxP = 0;
float atasI =0, bawahI = 0, zxI = 0;

```

```
float atasD =0, bawahD = 0, zxD = 0;
float apred[26],mf [5][3], zp[26],zi[26],zd[26];
float ccp,cxp;
float cci,cxi;
float ccd,cxd;

void SetTime(){
long pix=inK;
if(pix >= 5500){ SetPoint=32.00;}
else if(pix >= 2500){ SetPoint=38.00;}
else if(pix >= 1000){ SetPoint=40.00;}
else if(pix >= 400){ SetPoint=49.00;}
else if(pix <= 401){ SetPoint=60.00;}
}
void baca_serial(){
    if(Serial.available()){
        Serial.flush();
        if(Serial.read()=='#'){
            inK=Serial.parseFloat();
        }
        Serial.flush();
    }
}
void klibDHT(){
    hum = dht.readHumidity();
    T = dht.readTemperature();
    temp= (1.043*T) - 1.809;
}
void dataLogPenting(){
    DateTime now = rtc.now();
    klibDHT();
    myFile = SD.open("Log1.txt", FILE_WRITE);
    if (myFile) {
        myFile.print(rataTemp);
        myFile.print("\t");
        myFile.print(SetPoint);
        myFile.print("\t");
        myFile.print(rataOutPID);
        myFile.print("\t");
        myFile.print(rataError);
        myFile.print("\t");
        myFile.print(rataDError);
        myFile.print("\t");
        myFile.print(now.minute(), DEC);
        myFile.print("\t");
        myFile.println((long)inK);
        myFile.close(); }
}
```

```
else {Serial.println("error buka SD Chard Log1");}
}

void lcdDHT22(){
  DateTime now = rtc.now();
  lcd.setCursor(0,0);
  lcd.print("PV:");
  lcd.print(temp);
  lcd.print(derajat);
  lcd.print('C');
  lcd.print(" ");
  lcd.setCursor(11,0);
  lcd.print(now.hour(), DEC);
  lcd.print(':');
  lcd.print(now.minute(), DEC);
  lcd.setCursor(0,1);
  lcd.print("SP:");
  lcd.setCursor(3,1);
  lcd.print(SetPoint);
  lcd.print(derajat);
  lcd.print("C");
  lcd.setCursor(11,1);
  lcd.print(" ");
  lcd.setCursor(11,1);
  lcd.print((long)inK);
}

void lcdOUT(){
  lcd.setCursor(0,0); //SetPoint
  lcd.print("Kp:");
  lcd.setCursor(3,0);
  lcd.print(" ");
  lcd.setCursor(3,0);
  lcd.print(fKp);
  lcd.setCursor(9,0); //out sERVO
  lcd.print("Ki:");
  lcd.setCursor(12,0);
  lcd.print(" ");
  lcd.setCursor(12,0);
  lcd.print(fKi);
  lcd.setCursor(9,1); //OUT Kipas
  lcd.print("Kd:");
  lcd.setCursor(12,1);
  lcd.print(" ");
  lcd.setCursor(12,1);
  lcd.print(fKd);
  lcd.setCursor(0,1);
```

```
        lcd.print("E :");
        lcd.setCursor(3,1);
        lcd.print("    ");
        lcd.setCursor(3,1);
        lcd.print(error);
    }

void lcdOUT2() {
    lcd.setCursor(0,0); //SetPoint
    lcd.print("Kd:");
    lcd.setCursor(3,0);
    lcd.print("    ");
    lcd.setCursor(3,0);
    lcd.print(fKd);
    lcd.setCursor(0,1); //Baca Kamera
    lcd.print("SV:");
    lcd.setCursor(3,1);
    lcd.print("    ");
    lcd.setCursor(3,1);
    lcd.print(vServo);
    lcd.print(derajat);
    lcd.setCursor(9,0); //out SERVO
    lcd.print("KS1:");
    lcd.setCursor(13,0);
    lcd.print("    ");
    lcd.setCursor(13,0);
    lcd.print(vKipas1);
    lcd.setCursor(9,1); //OUT Kipas
    lcd.print("KS2:");
    lcd.setCursor(13,1);
    lcd.print("    ");
    lcd.setCursor(13,1);
    lcd.print(vKipas2);
}

void bacaTombol3() {
    int state=0;
    int g = digitalRead (Up);
    if (g == HIGH) {state = 1;}
    if (state != lastState) {
        if (state== 1) {ItemUp2();}
        lastState = state;
    }
    delay(5);
}

void displayTotal() {
    int pilih=IT;
    if(pilih==0) {lcd.clear();IT++;}
```

```
else if(pilih==1){lcdDHT22();}
else if(pilih==2){lcd.clear();IT++;}
else if(pilih==3){lcdOUT();}
else if(pilih==4){lcd.clear();IT++;}
else if(pilih==5){lcdOUT2();}
}
void ItemUp2(){IT++; if(IT>=6){IT=0;}delay (100);}

void PID(){
error=SetPoint-temp;          //error
de_error=last_error;
fuzzyRutin(error,de_error);
out_P=fKp*error;
error_I=error+last_error;    //kontrol I
out_I=fKi*error_I*Tc;
error_D=error-last_error;   //kontrol D
out_D=(fKd*error_D)/Tc;
last_error=error;
out_PID=out_P+out_I+out_D;   //kontrol PID
}
void Control(){
pwm_IN=pwm_IN+out_PID;
if(pwm_IN<=00.00){pwm_IN=00.00;}
if(pwm_IN>=255.00){pwm_IN=255.00;}
pwm_OUT=pwm_OUT-out_PID;
if(pwm_OUT<=00.00){pwm_OUT=00.00;}
if(pwm_OUT>=255.00){pwm_OUT=255.00;}
}
void driverKipas(){
vKipas1 = (int)pwm_IN;
analogWrite(K1, vKipas1);
vKipas2 = (int) pwm_OUT;
analogWrite(K2, vKipas2);
}
void sServo(){
vServo = map((int)pwm_IN, 0, 255, 0, 180);
myservo.write(vServo);
delay(15);
if (vServo<=0){digitalWrite (heat,LOW);}
else {digitalWrite (heat,HIGH);}
}
void setup(){
Serial.begin(9600);
rtc.begin();
if (! rtc.begin()) { Serial.println("Couldn't find
RTC"); while (1);}
if (! rtc.isrunning()) {
```

```
    Serial.println("RTC lost power, lets set the
time!");
    rtc.adjust(DateTime(F(__DATE__), 0,0,0));
    }
    pinMode(pinCS, OUTPUT);
    if (SD.begin()){Serial.println("SD card bisan
digunakan");}
    else{ Serial.println("SD card initialization
failed"); return;}
    myFile = SD.open("Log1.txt", FILE_WRITE);
    myFile.println(" //////////baru////////////////////");
    myFile.close();
    lcd.begin(16,2);
    pinMode(Up, INPUT);
    pinMode(heat, OUTPUT);
    dht.begin(); // DHT
    myservo.attach(45); // servo
    myservo.write(0);
    delay(15);
    ItemUp2();
}

void loop(){
    klibDHT();
    baca_serial();
    displayTotal();
    // SetTime();
    PID();
    Control();
    driverKipas();
    sServo();
    rataTemp          =(rataTemp+temp)/2;
    rataOutPID        =(rataOutPID+out_PID)/2;
    rataError         =(rataError+error)/2;
    rataDError        =(rataDError+de_error)/2;
    DateTime now = rtc.now();
    int setMenit=(now.hour()*60)+now.minute();
    if (setMenit!=lastSetMenit){dataLogPenting();}
    lastSetMenit=setMenit;
    bacaTombol3();

}

void fuzzyRutin(float err, float derror){
    //member function ERROR
    mf[0][0] = FuzzyMember(err,-3, -3, -2); //NB
    mf[1][0] = FuzzyMember(err,-2.5, -1.5, -0.5); //NM
```

```

mf[2][0] = FuzzyMember(err,-1, 0, 1); //Z
mf[3][0] = FuzzyMember(err, 0.5, 1.5, 2.5); //PM
mf[4][0] = FuzzyMember(err,2, 3, 3); //PB
    //mamber function dERROR
mf[0][1] = FuzzyMember(derror,-3, -3, -2); //NB
mf[1][1] = FuzzyMember(derror,-2.5, -1.5, -0.5); //NM
mf[2][1] = FuzzyMember(derror,-1, 0, 1); //Z
mf[3][1] = FuzzyMember(derror,0.5, 1.5, 2.5); //PM
mf[4][1] = FuzzyMember(derror,2, 3, 3); //PB

if (err<= -3){mf[0][0] = 1;}
else if (err>=3){ mf[4][0] = 1;}
if (derror<= -3){mf[0][1] = 1;}
else if (derror>=3){mf[4][1] =1;}

//mencari alfa predikat FUZZY
int x1 = 1;
for(int a1 = 0; a1 <=4; a1 = a1 +1)
for(int a2 = 0; a2 <=4; a2 = a2 +1)
{
    apred[x1] = min(mf[a1][0],mf[a2][1]);
    x1 = x1 + 1;
}
for(int rp = 1; rp <= 25; rp = rp + 1){
switch(rp) {
case 1  :{MKPrule(rp);BKIrule(rp);MKDrule(rp);break;}
case 2  :{SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
case 3  :{SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
case 4  :{SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
case 5  :{SKPrule(rp);BKIrule(rp);SKDrule(rp);break;}
case 6  :{MKPrule(rp);BKIrule(rp);MKDrule(rp);break;}
case 7  :{MKPrule(rp);MKIrule(rp);BKDrule(rp);break;}
case 8  :{SKPrule(rp);MKIrule(rp);BKDrule(rp);break;}
case 9  :{SKPrule(rp);MKIrule(rp);BKDrule(rp);break;}
case 10 :{SKPrule(rp);BKIrule(rp);BKDrule(rp);break;}
case 11 :{BKPrule(rp);SKIrule(rp); BKDrule(rp);break;}
case 12 :{MKPrule(rp);SKIrule(rp); BKDrule(rp);break;}
case 13 :{MKPrule(rp);SKIrule(rp); BKDrule(rp);break;}
case 14 :{SKPrule(rp);SKIrule(rp); BKDrule(rp);break;}
case 15 :{SKPrule(rp);SKIrule(rp); BKDrule(rp);break;}
case 16 :{BKPrule(rp);BKIrule(rp); MKDrule(rp);break;}
case 17 :{BKPrule(rp);MKIrule(rp); MKDrule(rp);break;}
case 18 :{MKPrule(rp);MKIrule(rp); MKDrule(rp);break;}
case 19 :{MKPrule(rp);MKIrule(rp); BKDrule(rp);break;}
case 20 :{MKPrule(rp);BKIrule(rp); BKDrule(rp);break;}
case 21 :{BKPrule(rp);BKIrule(rp); SKDrule(rp);break;}
case 22 :{BKPrule(rp);BKIrule(rp); SKDrule(rp);break;}

```

```
case 23 :{BKPrule(rp);BKIrule(rp); SKDrule(rp);break;}
case 24 :{BKPrule(rp);BKIrule(rp); SKDrule(rp);break;}
case 25 :{BKPrule(rp);BKIrule(rp); SKDrule(rp);break;}
    } }

//Defuzufikasi
atasP =0, bawahP = 0, zxP = 0;
atasI =0, bawahI = 0, zxI = 0;
atasD =0, bawahD = 0, zxD = 0;
for (int pp = 1; pp <= 25; pp = pp +1)
{
    atasP = atasP + (apred[pp] * zp[pp]);
    atasI = atasI + (apred[pp] * zi[pp]);
    atasD = atasD + (apred[pp] * zd[pp]);
    bawahP = bawahP + apred[pp];
    bawahI = bawahI + apred[pp];
    bawahD = bawahD + apred[pp];
}
zxP = atasP / bawahP;
zxI = atasI / bawahI;
zxD = atasD / bawahD;
fKp = zxP;
fKi = zxI;
fKd = zxD;

if (fKp>=14.00){fKp=14.00;}
else if (fKp<=4.00){fKp=4.00;}
if (fKi>=1.00){fKi=1.00;}
else if (fKi<=0.00){fKi=0.00;}
if (fKd>=120.00){fKd=120.00;}
else if (fKd<=20.00){fKd=20.00;}

}

float FuzzyMember(float a, float b, float c, float d){
float hasil;
if(a <= b || a >= d){hasil = 0;}
else if(a > b && a<=c){hasil = (a - b)/(c-b);}
else if(a > c && a < d){hasil = (d - a)/(d-c);}
return hasil;
}

//Rule
void SKPrule(int x){
    ccp = 4 + (apred[x] * 2); // KURVA NAIK
    cxp = 8 - (apred[x] * 2); // KURVA TURUN
    zp[x] = (ccp+cxp)/2; //NS
}
```



```
void MKPrule(int x){
    ccp = 7 + (apred[x] * 2); // KURVA NAIK
    cxp = 11 - (apred[x] * 2); // KURVA TURUN
    zp[x] = (ccp+cxp)/2; //NS
}
void BKPrule(int x){
    ccp = 10 + (apred[x] * 2); // KURVA NAIK
    cxp = 14 - (apred[x] * 2); // KURVA TURUN
    zp[x] = (ccp+cxp)/2; //ZR
}
void SKIrule(int x){
    cci = 0 + (apred[x] * 0.2); // KURVA NAIK
    cxi = 0.4 - (apred[x] * 0.2); // KURVA TURUN
    zi[x] = (cci+cxi)/2; //NS
}
void MKIrule(int x){
    cci = 0.3 + (apred[x] * 0.2); // KURVA NAIK
    cxi = 0.7 - (apred[x] * 0.2); // KURVA TURUN
    zi[x] = (cci+cxi)/2; //NS
}
void BKIrule(int x){
    cci = 0.6 + (apred[x] * 0.2); // KURVA NAIK
    cxi = 1 - (apred[x] * 0.2); // KURVA TURUN
    zi[x] = (cci+cxi)/2; //ZR
}
void SKDrule(int x){
    ccd = 20 + (apred[x] * 20); // KURVA NAIK
    cxd = 60 - (apred[x] * 20); // KURVA TURUN
    zd[x] = (ccd+cxd)/2; //NS
}
void MKDrule(int x){
    ccd = 50 + (apred[x] * 20); // KURVA NAIK
    cxd = 90 - (apred[x] * 20); // KURVA TURUN
    zd[x] = (ccd+cxd)/2; //NS
}
void BKDrule(int x){
    ccd = 80 + (apred[x] * 20); // KURVA NAIK
    cxd = 120 - (apred[x] * 20); // KURVA TURUN
    zd[x] = (ccd+cxd)/2; //ZR
}
}
```

## B. Listing Program VB

```
Imports Emgu.CV.UI
Imports Emgu.CV
Imports Emgu.CV.Structure
Imports Emgu.CV.CvEnum

Imports System.IO
Imports System.IO.Ports
Imports System.Threading

Public Class Form1
    Dim CamDevice As New Capture(1)
    Dim ImgGray As New Image(Of Gray, Byte)(320, 240)
    Dim ImgBiner As New Image(Of Gray, Byte)(320, 240)
    Dim ImgRGB As New Image(Of Bgr, Byte)(320, 240)
    Dim x As Integer
    Dim y As Integer
    Dim red As Integer
    Dim green As Integer
    Dim blue As Integer
    Dim lebar_temp As Integer
    Dim lebar As Integer
    Dim data_kirim As Single
    Dim str As String = 0
    Dim kunci As Integer = 0

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

        CamDevice.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEIGHT, 240)

        CamDevice.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WIDTH, 320)
        For Each sp As String In My.Computer.Ports.SerialPortNames
            ComboBox1.Items.Add(sp)
        Next
        ComboBox1.SelectedIndex = 0
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        ImgRGB = CamDevice.QueryFrame.Flip(Emgu.CV.CvEnum.FLIP.NONE)
        ImgGray = ImgRGB.Convert(Of Gray, Byte)()
        ImgBiner = ImgRGB.InRange(New Bgr(blue - NumericUpDown1.Value,
green - NumericUpDown2.Value, red - NumericUpDown3.Value), New Bgr(blue +
NumericUpDown1.Value, green + NumericUpDown2.Value, red +
NumericUpDown3.Value))
        '------(mendapatkan lebar
dengan menghitung jumlah piksel yang bernilai 1)
        For i = 0 To 240 - 1 ' 320 - 1 'lebar2 - 1 'gambarHSV.Height - 1
            For j = 0 To 320 - 1 '240 - 1 'tinggi - 1 'gambarHSV.Width - 1
                If ImgBiner.Data(i, j, 0) = 255 Then
                    lebar = lebar + 1
                ElseIf ImgBiner.Data(i, j, 0) = 0 Then
                    lebar = lebar + 0
                End If
            Next
        Next
```

```
Next
ImageBox1.Image = ImgRGB
ImageBox2.Image = ImgGray
ImageBox3.Image = ImgBiner
TextBox1.Text = red
TextBox2.Text = green
TextBox3.Text = blue
TextBox6.Text = lebar
lebar = 0

red = 104 'hijau
green = 164
blue = 137
'red = 175 'kuning
'green = 204
'blue = 147
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Timer1.Enabled = True
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Timer1.Enabled = False
End Sub
Private Sub ImageBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ImageBox1.Click
Dim P As Point = ImageBox1.PointToClient(Cursor.Position)
x = P.X
y = P.Y
TextBox4.Text = x
TextBox5.Text = y
blue = ImgRGB.Data(y, x, 0)
green = ImgRGB.Data(y, x, 1)
red = ImgRGB.Data(y, x, 2)
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
data_kirim = Convert.ToSingle(TextBox6.Text)
str = Convert.ToString(data_kirim)
str = str + "#"
TextBox7.Text = str
SerialPort1.Write(str)
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
On Error GoTo err
If SerialPort1.IsOpen Then
SerialPort1.Close()
Timer2.Enabled = False
Label10.Text = "Disconnected"
Button3.Text = "Buka PORT"
Else
SerialPort1.BaudRate = 9600
SerialPort1.PortName = ComboBox1.Text
SerialPort1.Open()
Timer2.Enabled = True
```

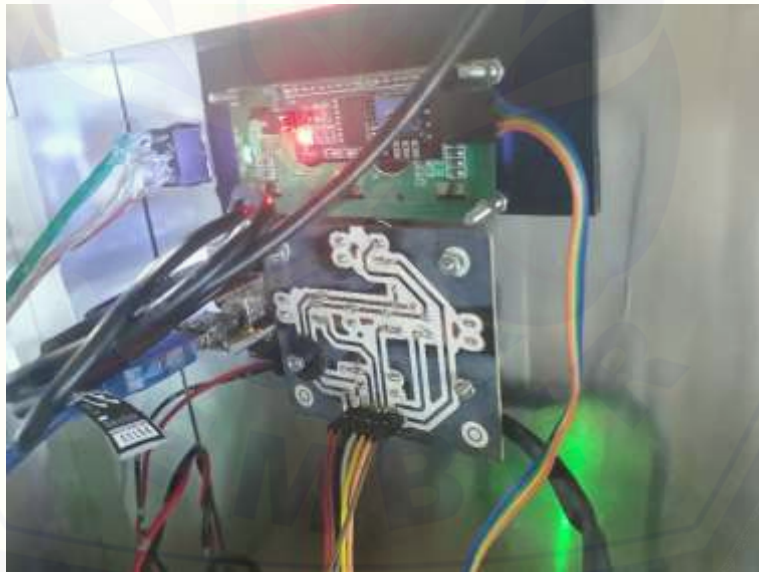
```
        Label10.Text = "Connected"
        Button3.Text = "Tutup PORT"
    End If
    Exit Sub
err:   MsgBox("Serial Error")
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    data_kirim = Convert.ToSingle(TextBox6.Text)
    str = Convert.ToString(data_kirim)
    str = "#" + vbTab + str
    TextBox7.Text = str
    SerialPort1.Close()
    SerialPort1.Open()
    If SerialPort1.IsOpen Then
        SerialPort1.Write(str)
    End If
    Dim file As System.IO.StreamWriter
    file =
My.Computer.FileSystem.OpenTextFileWriter(Environment.GetFolderPath(Environ
ment.SpecialFolder.MyDocuments) + "\Tembakau19_60.txt", True)
        file.WriteLine "[" + FormatDateTime(Now, vbLongTime) + "]" + vbTab
+ TextBox7.Text)
        file.Close()
    End Sub
End Class
```

C. Gambar *Hardware*



Gambar Kontroller, *Driver* Motor dan *Dimmer*



Gambar Rangkain LCD dan Tombol



Gambar Bagian Dalam Oven



Gambar Ruang Pemanas



Gambar Keseluruhan Alat Pengering

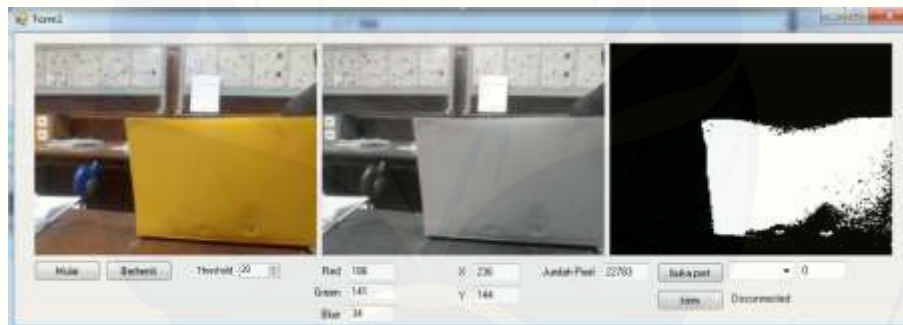


Gambar Pengambilan Data



Gambar Panen Daun Tembakau dengan Petani

#### D. Gambar Software



Gambar Aplikasi Pengolahan Citra