

**ALAT PEMERAM PANILI
BERBASIS MIKROKONTROLER MC68HC908QB8
DENGAN ALGORITMA PID**

TUGAS AKHIR

Diajukan untuk melengkapi salah satu syarat untuk memperoleh gelar
Sarjana Teknik pada Jurusan Teknik Elektro Universitas Islam Indonesia
Yogyakarta



oleh :

Nama : Lalu Fauzan Satriadiningrat
No. Mahasiswa : 00 524 085

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

2007

HALAMAN PENGESAHAN PEMBIMBING

ALAT PEMERAM PANILI BERBASIS MIKROKONTROLER MC68HC908QB8 DENGAN ALGORITMA PID

TUGAS AKHIR

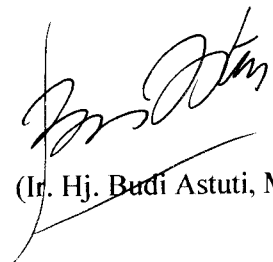
Disusun oleh :

Nama : Lalu Fauzan Satriadiningrat

No. Mahasiswa : 00 524 085

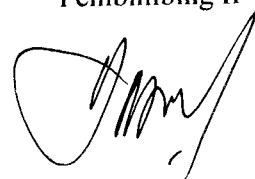
Yogyakarta, 1 Mei 2007

Pembimbing I



(Ir. Hj. Budi Astuti, MT)

Pembimbing II



(Yusuf Aziz Amrullah, ST)

HALAMAN PENGESAHAN PENGUJI
ALAT PEMERAM PANILI
BERBASIS MIKROKONTROLER MC68HC908QB8
DENGAN ALGORITMA PID

TUGAS AKHIR

Oleh :

Nama : Lalu Fauzan Satriadiningrat

No. Mahasiswa : 00 524 085

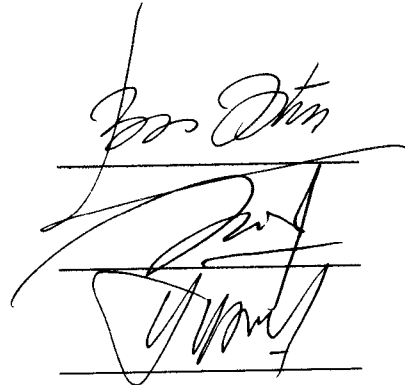
Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia
Jogjakarta, 26 April 2007

Tim Penguji

Ir. Hj. Budi Astuti, MT
Ketua

Dwi Ana Ratna Wati, ST
Anggota I

Yusuf Aziz Amrullah, ST
Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro

Fakultas Teknologi Industri
Universitas Islam Indonesia



Lalu Fauzan Satriadiningrat, ST, M.Sc

HALAMAN PERSEMBAHAN

*Seluruh daya upaya dan hasil dari skripsi ini
kupersembahkan untuk orang-orang yang
sangat kucintai dalam kefanaan dunia dan
kekalnya akhirat.*

- ✧ Untuk tetes alir darah BHETARE
PUTREJENGGALE & DATUMAS
PUTRE SILE DENDENG.*
- ✧ Inaq Toa'q (Almarhum) & Amaq Toa'q.*
- ✧ H.D. Muhammad Djumardi Silepati, Lale
Dewimas Fauziah.*
- ✧ Adinda Dinde Lale Putri Baiq Maryatun
Olyani & Pangeran Gagah Gede Lalu
Inder Tata Kayiripan.*

MOTTO

***CONQUER NOW
OR
DIE TRY'IN***

(Oxza)

KATA PENGANTAR



Assalamu'alaikum Wr.Wb.

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya yang diberikan kepada penyusun, shalawat serta salam kepada Rasulullah SAW beserta keluarga dan para pengikutnya sampai akhir zaman, sehingga Tugas Akhir ini dapat diselesaikan dengan baik. Tugas Akhir ini dilaksanakan sebagai prasyarat dalam menempuh jenjang Strata Satu (S-1) pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Selama mengerjakan Tugas Akhir dan dalam penyusunan laporan, tak lepas dari hambatan dan tantangan, namun berkat motivasi, informasi dan konsultasi dari berbagai pihak akhirnya semua dapat diatasi. Untuk itu tidak berlebihan, penyusun menyampaikan rasa hormat sebagai rasa ungkapan terima kasih kepada :

1. Inaq Toa'k (almarhum) & Amaq Toaq. *Eyes of mother, light of the sun, Beauty of mom's love.* Simpuhku diantara kekalutanku, do'aku air di saharaku.
2. Abah Mamiq & Inaq Uji. *Nggih Pngrabdem Plinggihm Datu Kaji.* Oa'q Tuan & Mak Njoe, *Matur Tampi Asih atas angen plungguhm.*
3. Kekasih hatiku Lisa Fatriani Becxk, selalu bersama dengan tawa dan senyum, lalui hidup dalam duka dan bahagia.
4. Adinda Dinde Lale Putri Baiq Maryatun Olyani & Pangeran Gagah Gede Lalu Inder Tata Kayiripan, *kaka'm* selalu ada untuk kalian.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR	vi
ABSTRAK	viii
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penyusunan Tugas Akhir	3
1.4 Batasan Masalah	4
1.5 Manfaat Penyusunan Tugas Akhir	5
1.6 Sistematika Penulisan	5
BAB II DASAR TEORI	
2.1 PID Digital	7
2.2 Penalaan Besaran (Tuning)	10
2.2.1 Metode <i>Ziegler-Nichols</i>	11
2.2.2 Metode <i>Quarter-Delay</i>	12

2.3	Tanggapan Transien	12
2.4	Sensor Suhu Termokopel	14
2.5	Penguat Operasional	15
2.6	Mikrokontroler MC68HC908QB8.....	17
2.6.1	Diagram Blok Mikrokontroler MC68HC908QB8.....	19
2.6.2	Organisasi Memori.....	19
2.3.2.1	<i>Random Access Memory (RAM)</i>	20
2.3.2.2	<i>Read Only Memory (ROM)</i>	20
2.6.3	Register-register.....	20
2.6.3.1	<i>Accumulator</i>	21
2.6.3.2	<i>Index Register</i>	21
2.6.3.3	<i>Stack Pointer (SP)</i>	22
2.6.3.4	<i>Program Counter (PC)</i>	22
2.6.3.5	<i>Condition Code Register</i>	23
2.6.4	<i>Timer Interface Module (TIM)</i>	24
2.6.5	<i>Analog to Digital Converter (ADC)</i>	27
2.7	Triac	30
2.8	Optokopler	31
2.9	Transistor Sebagai Saklar	32
2.10	Relai	32
2.11	Motor DC	33
2.12	<i>Liquid Crystal Display (LCD)</i>	34

BAB III PERANCANGAN DAN IMPLEMENTASI PERANGKAT KERAS
DAN PERANGKAT LUNAK

3.1 Perancangan Perangkat Keras.....	35
3.1.1 Pengkondisi Isyarat	36
3.1.2 Sistem Minimum Mikrokontroler MC68HC908QB8.....	37
3.1.3 <i>Analog to Digital Converter (ADC)</i>	37
3.1.4 <i>Driver Motor</i>	38
3.1.5 <i>Driver Heater</i>	39
3.1.6 Pengaturan Daya	40
3.2 Perancangan Perangkat Lunak (<i>Software</i>)	43
3.2.1 Gambaran Umum Program	43
3.2.2 Sub Rutin Kontrol <i>Heater</i>	44
3.2.2 Perancangan Pengendali PID	45
3.2.2 Sub Rutin Menaikkan dan Menurunkan Keranjang.....	50

BAB IV HASIL PENGAMATAN

4.1 Pengukuran	53
4.2 Pembahasan	54
4.2.1 Sensor Termokopel	56
4.2.2 Mikrokontroler MC68HC908QB8.....	56
4.2.3 <i>Driver Heater</i>	57
4.2.4 Tampilan (<i>Display</i>)	59
4.2.5 <i>Driver Motor</i>	60
4.3 Analisa Pengendali PID	60

BAB V PENUTUP

5.1 Kesimpulan68

5.2 Saran69

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Kendali PID	10
Gambar 2.2 Kurva proses reaksi <i>open-loop tuning</i>	11
Gambar 2.3 Metode pengukuran dengan termokopel	15
Gambar 2.4 Diagram blok <i>operational amplifier</i>	16
Gambar 2.5 Skematik penguat operasional	17
Gambar 2.6 Susunan kaki-kaki mikrokontroler MC68HC908QB8	15
Gambar 2.7 Diagram blok mikrokontroler MC68HC908QB8	19
Gambar 2.8 Register-register mikrokontroler MC68HC908QB8	21
Gambar 2.9 <i>Accumulator</i>	21
Gambar 2.10 <i>Index Register</i>	22
Gambar 2.11 <i>Stack Pointer</i>	22
Gambar 2.12 <i>Program Counter</i>	23
Gambar 2.13 <i>Condition Code Register</i>	23
Gambar 2.14 Diagram blok <i>Timer Interface Modul (TIM)</i>	26
Gambar 2.15 Diagram blok ADC internal mikrokontroler MC68HC908QB8 ...	28
Gambar 2.16 Konfigurasi Triac	30
Gambar 2.17 Optokopler	31
Gambar 2.18 Transistor sebagai saklar	32
Gambar 2.19 Simbol relai	33
Gambar 2.20 Diagram blok LCD	34
Gambar 3.1 Diagram blok alat pemeram panili	35
Gambar 3.2 Rangkaian pengkondisi isyarat	36

Gambar 3.3 Sistem minimum mikrokontroler MC68HC908QB8.....	37
Gambar 3.4 Rangkaian <i>driver motor</i>	39
Gambar 3.5 <i>Driver heater</i>	40
Gambar 3.6 Waktu tunda dan saat MT1 dan MT2 terhubung	41
Gambar 3.7 Grafik waktu tunda vs P/Pmax.....	42
Gambar 3.8 Bagan alir program utama	44
Gambar 3.9 Bagan alir sub rutin kontrol <i>heater</i>	45
Gambar 3.10 Bagan alir sub rutin motor naik turun	52
Gambar 4.1 Sudut picu 0°	58
Gambar 4.2 Sudut picu 20°	58
Gambar 4.3 Sudut picu 40°	58
Gambar 4.4 Sudut picu 90°	58
Gambar 4.5 Sudut picu 140°	59
Gambar 4.6 Sudut picu 160°	59
Gambar 4.7 Sudut picu 180°	59
Gambar 4.8 Respon PID pada nilai $K_p=25$, $K_i=50$, $K_d=100$	62
Gambar 4.9 Respon PID pada nilai $K_p=50$, $K_i=50$, $K_d=100$	63
Gambar 4.10 Respon PID pada nilai $K_p=75$, $K_i=50$, $K_d=100$	63
Gambar 4.11 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=25$	64
Gambar 4.12 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=50$	64
Gambar 4.13 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=75$	65
Gambar 4.14 Respon PID pada nilai $K_p=100$, $K_i=50$, $K_d=25$	65
Gambar 4.15 Respon PID pada nilai $K_p=100$, $K_i=50$, $K_d=50$	66

DAFTAR TABEL

Tabel 2.1 Fungsi kaki-kaki pada mikrokontroler MC68HC908QB8	18
Tabel 3.1 Waktu dan kondisi Motor	51
Tabel 4.1 Hasil pengukuran <i>Output</i> termokopel dan pengkondisi isyarat	53

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Panili (*Vanilla planifolia Andrews*) termasuk famili *Orchidaceae* (anggrek-anggrekan) yang menumpang secara epifit pada tanaman lain atau tonggak kayu dengan bantuan akar lekatnya. Panili terkenal karena aromanya yang khas yang disebabkan oleh adanya kandungan vanillin ($C_6H_8O_3$) dalam buahnya. Panili biasa dipergunakan untuk memberi aroma pada makanan dan minuman. Tanaman panili berasal dari negara Meksiko dan masuk ke Indonesia sekitar tahun 1819 dan mulai dibudidayakan besar-besaran pada tahun 1850. Dipasaran dunia, panili Indonesia terus laris karena kadar "valinenya" cukup tinggi (2,75%). Harga panili di pasaran Indonesia pada tahun 2006 penawarannya mencapai harga Rp. 800.000,-/kg kering siap pakai.

Kualitas panili dari Indonesia kurang dapat bersaing dengan negara-negara lain. Pengolahan panili di Indonesia umumnya dikerjakan dengan cara tradisional menggunakan drum dan tungku didih dan kurang memperhatikan prinsip dasar pengolahan panili sehingga menghasilkan panili dengan mutu tidak terlalu tinggi.

Pemanfaatan teknologi dengan sistem pengendalian merupakan solusi tepat untuk membantu petani panili agar mendapatkan mutu produk panili siap jual menjadi tinggi. Salah satu proses yang sangat penting dan sangat menentukan mutu dari panili adalah pemeraman tahap pertama setelah pemetikan yaitu pemeraman dengan cara merebus panili dengan suhu antara $50^{\circ} - 65^{\circ} C$. Tujuan

1.4 Batasan Masalah

Dalam pembuatan tugas akhir **Alat Pemeraman Panili Berbasis Mikrokontroler MC68HC908QB8 Dengan Algoritma PID** ini hanya membuat alat dan sistem pengendali dari proses pemeraman dan *prototype* dari alat tersebut. Batasan-batasannya adalah sebagai berikut :

1. Sistem kendali dari pemanas pemeram (*heater*) air ini menggunakan algoritma PID pada program mikrokontroler agar didapatkan suhu antara 50° sampai dengan 65° C untuk pemeraman. Pengendali ini kemudian akan mempertahankan suhu air pada *set point*.
2. Sistem kendali motor penggerak keranjang menggunakan pewaktu (*timer*) yang ada di dalam mikrokontroler, karena *output* dari pengendali motor ini diinginkan adanya gerakan naik dan turun dari keranjang. Durasi waktu pemeraman adalah selama 2.5 menit dan setiap 20 detik sekali keranjang naik turun.
3. Penampil suhu dengan *Liquid Crystal Display* (LCD)digunakan untuk melihat secara mudah dan akurat besarnya suhu yang ada dalam bak pemeram.
4. *Driver heater* digunakan sebagai pengendali elemen pemanas untuk menaikkan suhu air pada bak pemeram dengan menerjemahkan perintah-perintah mikrokontroler.
5. *Driver motor* digunakan sebagai penterjemah untuk perintah-perintah dari mikrokontroler sehingga aktifitas motor dapat dikendalikan sebagai penggerak keranjang perendam.

BAB II

DASAR TEORI

2.1 PID digital

Didalam suatu sistem kendali PID terdapat beberapa macam aksi kendali, diantaranya yaitu aksi kendali proporsional, aksi kendali integral dan aksi kendali derivatif. Masing-masing aksi kendali ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kendali proporsional mempunyai keunggulan risetime yang cepat, aksi kendali integral mempunyai keunggulan untuk memperkecil *error*, dan aksi kendali derivatif mempunyai keunggulan untuk memperkecil *derror* atau meredam *overshoot/undershoot*. Untuk itu agar dapat menghasilkan output dengan *risetime* yang tinggi dan *error* yang kecil ketiga aksi kendali ini digabung menjadi kendali PID.

Karakteristik umum yang digunakan dalam pengontrolan suatu sistem antara lain meliputi : stabilitas, akurasi, kecepatan respon dan sensitivitas.

- ❖ Dalam aksi kendali proporsional, output dari sistem kendali selalu sebanding dengan inputnya. Isyarat output merupakan penguatan dari isyarat kesalahan dengan faktor tertentu, faktor penguatan f ini merupakan konstanta proporsional dari sistem, yang dinyatakan dengan K_p , dimana K_p ini mempunyai respon yang tinggi/cepat.
- ❖ Dalam aksi kendali integral, output dari kendali ini selalu berubah selama terjadi penyimpangan, dan kecepatan perubahan output tersebut sebanding dengan penyimpangannya, konstantanya dinyatakan dengan K_i , dimana

Ki ini mempunyai sensitivitas yang tinggi, yaitu dengan cara mereduksi *error* yang dihasilkan dari isyarat *feedback*. Makin besar nilai Ki maka sensitivitasnya akan semakin tinggi, tetapi waktu yang dibutuhkan untuk mencapai kestabilan lebih cepat, demikian pula sebaliknya.

- ❖ Sedangkan aksi kendali derivatif (turunan) bekerja berdasarkan laju perubahan simpangan, sehingga jenis kendali ini selalu digunakan bersama-sama dengan kendali proporsional dan integral, konstantanya dinyatakan dengan Kd, dimana Kd ini mempengaruhi kestabilan dari sistem, karena aksi kendali ini dapat mereduksi *derror*.

Dengan adanya penggabungan aksi kendali PID ini maka diharapkan akan mendapat suatu respon yang mempunyai tingkat kestabilan yang tinggi.

PID Digital pada dasarnya merupakan suatu proses dari suatu program yang dijalankan/diexecute dengan menggunakan komputer, dimana dengan memasukkan nilai *Setting Point* (SP) dan *Present Value* (PV), yang kemudian data yang didapatkan diproses sehingga error yang didapatkan sama dengan 0, atau nilai *Setting Point* = *Present Value*. Perhitungan ini hanya dilakukan dalam suatu algoritma yang dijalankan didalam perhitungan matematis didalam komputer. Sistem PID yang dijalankan berupa rumus matematis yang diterjemahkan dalam bentuk program. Sistem yang dikendalikan dari hasil pengolahan algoritmis ini akan diterjemahkan melalui *driver-driver* untuk pengendalian dalam bentuk sistem analog. Untuk dapat mengimplementasikan sistem kendali PID pada komputer, PID harus diubah ke dalam persamaan diskrit:

$$V_o = K_p e + K_i \int e dt + K_d \frac{d^2 e}{dt^2} \quad (2.1)$$

kemudian diturunkan

$$\frac{dV_o}{dt} = Kp e + Ki \frac{d}{dt} \left(\int e dt \right) + Kd \frac{d^2 e}{dt^2} \quad (2.2)$$

$$\frac{dV_o}{dt} = Kp e + Kie + Kd \frac{d}{dt} \left(\frac{de}{dt} \right) \quad (2.3)$$

dikali dengan T_s , sehingga

$$\frac{\Delta V_o}{T_s} = Kp \frac{\Delta e}{T_s} + Kie + Kd \frac{\Delta}{T_s} \left(\frac{\Delta e}{T_s} \right) \quad (2.4)$$

$$\Delta V_o = Kp \Delta e + Kie T_s + Kd \Delta \left(\frac{\Delta e}{T_s} \right) \quad (2.5)$$

Harga ΔV_o merupakan harga perubahan output yang didapat dari output sekarang dikurangi dengan output sebelumnya.

$$\Delta V_o = V_{on} - V_{on-1}, \text{ begitu juga pada perubahan error}$$

$$\Delta e = e_n - e_{n-1}$$

Sehingga persamaannya menjadi :

$$V_o - V_o(n-1) = Kp(e_n - e_{(n-1)}) + Kie_n T_s + \frac{Kd}{T_s} (\Delta e_n - \Delta e_{(n-1)}) \quad (2.6)$$

Pada kondisi akhir, perubahan Δ pada error sebelumnya dapat didistribusikan menjadi :

$$\Delta e_n = e_n - e_{(n-1)} \quad (2.7)$$

$$\Delta e_{n-1} = e_{(n-1)} - e_{(n-2)} \quad (2.8)$$

Kemudian disubstitusikan ke dalam persamaan, menjadi :

$$V_o = V_{o(n-1)} = Kp(e_n - e_{(n-1)}) + Kie_n T_s + \frac{Kd}{T_s} [(e_n - e_{(n-1)}) - (e_{(n-1)} - e_{(n-2)})] \quad (2.9)$$

Hasil akhir dari persamaan PID yaitu:

$$V_o = V_{o(n-1)} + Kp(e_n - e_{(n-1)}) + Kie_nTs + \frac{Kd}{Ts}(e_n - 2e_{(n-1)} + e_{(n-2)}) \quad (2.10)$$

Dimana : V_o = Output

V_{on-1} = Ouput sebelumnya

Kp = Konstanta Proporsional

Ki = Konstanta Integral

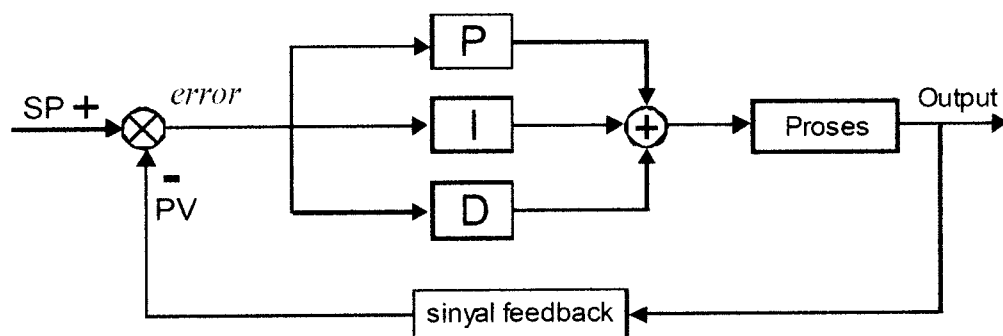
Kd = Konstanta Derivatif

e_n = Error saat n

e_{n-1} = Error saat n-1

e_{n-2} = Error saat n-2

Ts = Time Sampling



Gambar 2.1 Kendali PID

2.2 Penalaan Besaran PID (Tuning)

Aspek yang sangat penting dalam desain kontoler PID ialah penentuan parameter kendali PID supaya sistem *close loop* memenuhi kriteria performansi yang diinginkan. Hal ini disebut juga dengan *tuning* kendali. Ada beberapa metode penalaan besaran PID.

$$K_i = \frac{1}{T_i} = 2L \quad (2.12)$$

$$K_d = T_d = \frac{L}{2} \quad (2.13)$$

2.2.2 Metode *Quarter Delay*

Merupakan metode penalaan kendali kalang tertutup, dimana kendali dipasang pada mode otomatis, saat dilakukan penalaan konstanta yang dihasilkan metode ini biasanya terlalu sensitif. Ketika diterapkan pada proses yang memiliki derau pengukuran ataupun yang menggunakan katup yang sensitif.

Tahap-tahap penalaan yang dilakukan adalah sebagai berikut

1. Dengan kendali pada mode otomatis, atur semua konstanta penalaan kenilai tetapan yang sensitivitasnya paling sensitive. Pita proposional harus berada pada nilainya yang tertinggi, integral time harus berada pada nilainya yang tertinggi dan derivative time juga harus berada pada nilainya yang tinggi.
2. Buatlah perubahan step yang paling kecil pada set point kendali dan rekam pegukuran kendali hingga keluaran stabil.
3. Ubah *set point* kembali ke nilai awal dan rekam pengukuran seperti sebelumnya.

2.3 Tanggapan Transien

Tanggapan Transien adalah tanggapan sistem yang berlangsung dari awal sampai akhir atau hingga sistem stabil, pada sistem orde 2 tanggapan transien

dari proses pemeraman adalah untuk menghentikan proses respirasi yang terjadi di dalam buah, mematikan sel-sel buah panili tanpa mengurangi aktifitas dan kadar enzim dalam buah. Proses pemeraman menggunakan alat perebus yang dengan medium air tawar dengan suhu $50^{\circ} - 65^{\circ} \text{ C}$. Untuk meletakkan panili digunakan keranjang dari kawat.

Pemanfaatan teknologi pengendalian juga digunakan pada sistem pemanasan air sehingga suhu air perebusan tetap terjaga. Penggunaan teknologi algoritma PID (*Proportional Integral Derivative*) dipilih sebagai salah satu solusi penunjang untuk menjaga suhu air tetap pada kondisi optimal untuk perebusan. Pemilihan teknologi PID adalah karena kendali PID adalah kendali yang cukup handal dan efisien, keunggulan kendali PID antara lain tanggapannya cepat, memiliki keadaan tunak yang mantap dan offsetnya kecil. Algoritma PID juga cukup efektif untuk lingkungan kendali yang tetap, namun kinerjanya kurang dalam sistem yang lingkungannya berubah-ubah atau tidak linear. Kelemahan lain kendali PID adalah kurang dapat diandalkan untuk mengendalikan plant yang kompleks dan sulit dicari model matematisnya. Walaupun memiliki keterbatasan tersebut namun pada saat ini didunia industri sistem pengendalian PID masih populer digunakan..

Dari paparan diatas kita dapat melihat beberapa permasalahan yang dihadapi oleh pembudidaya panili diantaranya adalah:

1. Manajemen pembudidayan panili.
2. Kurangnya informasi dan pendidikan tentang pengolahan panili.

3. Pada panen dalam skala besar dengan proses pemeraman manual menghadapi resiko penurunan mutu yang jauh lebih besar.
4. Belum adanya alat yang otomatis dan mudah digunakan oleh para petani untuk sistem pemeraman.

Dari berbagai permasalahan para pembudidaya ini akan di buat sebuah solusi tentang pemeraman buah panili. Alat ini bekerja secara otomatis dan dapat digunakan oleh para petani. Alat ini di beri nama “ **ALAT PEMERAM PANILI BERBASIS MIKROKONTROLER MC68HC908QB8 DENGAN ALGORITMA PID** “

1.2 Rumusan Masalah

Dari latar belakang dan dasar pemikiran diatas dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana membuat suatu peralatan pemeram panili yang praktis dan handal.
2. Bagaimana membuat alat pemeram yang mudah digunakan dan dapat mempertahankan mutu dari panili.

1.3 Tujuan Penyusunan Tugas Akhir

Tujuan dari pembuatan peralatan sistem elektronika ini adalah untuk membuat alat pemeram panili yang mudah digunakan, efisien dan handal.

terbagi menjadi tiga berdasarkan konstanta peredamnya yaitu sistem kurang teredam ($\xi < 1$), teredam kritis ($\xi = 1$) dan teredam lebih ($\xi > 1$).

Pada tanggapan transient ini ada beberapa parameter yang perlu diketahui yaitu

- Waktu tunda (*delay time*) adalah waktu yang diperlukan sistem untuk mencapai nilai akhir.
- Waktu naik (*rise time*) adalah waktu yang diperlukan sistem untuk naik dari 10% sampai 90% nilai akhir.
- Waktu puncak (*peak time*) adalah waktu yang diperlukan sistem untuk mencapai puncak pertama kali.
- Persen Overshoot perbandingan nilai puncak maksimum dengan nilai akhir, yang dinyatakan dalam bentuk.

$$\% \text{ OS} = \frac{C_{\max} - C_{\text{akhir}}}{C_{\text{akhir}}} \times 100\%$$

Overshoot ini hanya terjadi pada sistem yang kurang teredam ($\xi < 1$).

- Waktu penetapan (*settling time*) adalah waktu yang diperlukan sistem untuk mencapai nilai $\pm 2\%$ dari nilai keadaan tunak (*steady state*).
- Kesalahan keadaan tunak (*steady state error*) adalah perbedaan antara keluaran yang didapat saat tunak dengan nilai yang diinginkan.

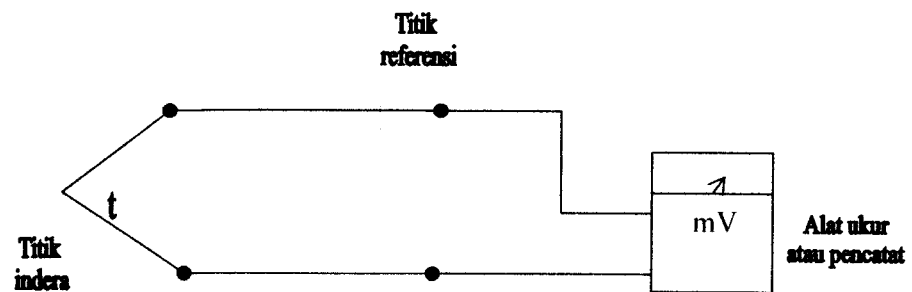
2.4 Sensor Suhu Termokopel

Sensor temperatur merupakan bagian dari sistem yang mengubah besaran suhu menjadi isyarat listrik. Sensor digunakan untuk mendeteksi perubahan suhu. Suhu yang sudah dideteksi tersebut kemudian akan dikirim ke bagian pengkondisi isyarat.

Pada tahun 1821 ahli fisika berkebangsaan Jerman bernama Thomas Johann Seebeck menemukan bahwa tiap-tiap konduktor seperti logam yang diberikan gradient terhadap perubahan suhu akan menghasilkan perbedaan tegangan. Hal ini sekarang disebut efek *thermoelectric*. Dengan menggabungkan dua jenis logam yang berbeda kemudian salah satu ujung diberikan perubahan suhu dan ujung yang lain diberikan pengukur tegangan akan menghasilkan perbedaan tegangan tiap derajat perubahan suhu. Kedua ujung termokopel yang merupakan gabungan dari ujung pengindra dan ujung panas diletakkan didalam media yang akan diukur suhunya dalam penelitian ini adalah air. Jika diantara kedua ujung tersebut terdapat perbedaan suhu, suatu ggl yang menyebabkan arus didalam rangkaian akan dihasilkan. Besarnya suhu ggl termal yang dihasilkan tergantung pada bahan kawat yang digunakan dan pada selisih suhu yang terukur pada titik sambung.

Termokopel strukturnya terdiri dari sepasang bahan konduktor logam yang berbeda dihubungkan bersama-sama pada satu ujung (sebagai ujung pengindra atau ujung panas) dan berakhir pada ujung yang lain (titik referensi atau ujung dingin) yang dipertahankan pada suhu konstan yang diketahui (suhu referensi). Bila antara ujung pengindra dan titik referensi terdapat perbedaan suhu, suatu

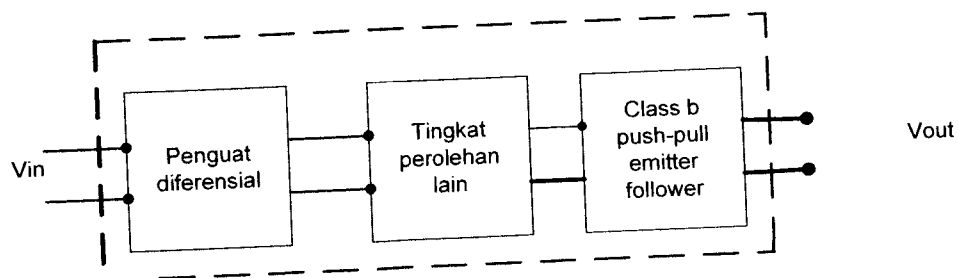
GGL yang menyebabkan arus dalam rangkaian akan dibangkitkan. Bila titik referensi ditutup oleh sebuah alat ukur, maka penunjukan alat ukur tersebut akan sebanding dengan selisih suhu antara ujung panas dan titik referensi. Efek termolistrik yang diakibatkan oleh potensial-potensial kontak pada titik-titik sambung ini dikenal sebagai efek *seebeck*. Metode pengukuran dengan Termokopel ditunjukkan pada Gambar 2.3.



Gambar 2.3 Metode pengukuran dengan termokopel

2.5 Penguat Operasional

Operasional amplifier adalah salah satu komponen yang bersifat multifungsi dalam sistem analog yang dapat menguatkan isyarat yang memiliki frekuensi dari 0 Hz hingga beberapa megaHertz. Pada awalnya op-amp didesain dengan menggunakan tabung hampa. Dengan perkembangan teknologi rangkaian *solid-state* menggantikan op-amp berbasis tabung. Pemanfaatan op-amp sangat bervariasi. Sebagai contoh dengan menghubungkan dua buah resistor eksternal, perolehan tegangan dan bandwidth penguatan operasional dapat diatur sesuai dengan kebutuhan. Lebih jauh lagi dengan komponen eksternal lainnya, dapat dibangun pula converter bentuk gelombang, osilator, filter aktif dan lainnya.

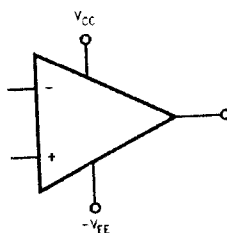


Gambar 2.4 Diagram blok operational amplifier

Pada blok diagram terlihat bahwa blok pertama adalah sebagai tingkat penguat diferensial, diikuti dengan tingkat perolehan lain, dan sebuah pengikut emiter dorong-tarik kelas B. Karena penguat diferensial berada pada tingkat pertama, ini menentukan karakteristik masukan penguat operasional. Pada kebanyakan penguat operasional, keluarannya berujung tunggal ini dirancang untuk memiliki nilai tetap sebesar nol. Dengan demikian, tegangan masukan nol secara ideal akan menghasilkan nol pula.

Gambar 2.5 adalah simbol skematik dari penguat operasional. Yang memiliki 2 (dua) buah masukan nonpembalikan dan pembalikan dan sebuah keluaran berujung tunggal. Idealnya, simbol ini berarti bahwa penguat operasional memiliki perolehan tegangan yang tak terhingga, impedansi masukan tak terhingga, dan impedansi keluaran nol.

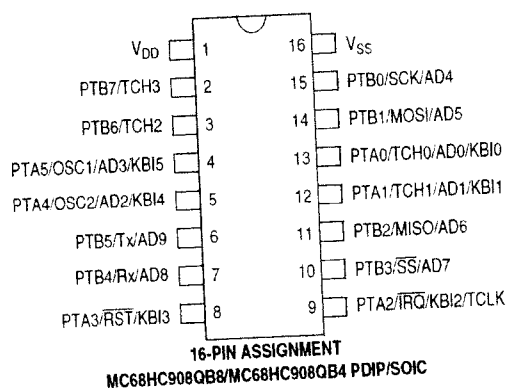
Pada perancangan ini bagian pengkondisi isyarat dari sistem ini adalah menggunakan 3 (tiga) buah penguat operasional LM 358. Kelebihan dari LM 358 adalah penguatan operasional yang presisi dengan memiliki dua buah penguat operasional sekaligus yang dikemas dalam satu IC.



Gambar 2.5 Skematik penguat operasional.

2.6 Mikrokontroler MC68HC908QB8

Mikrokontroler MC68HC908QB8 adalah salah satu dari keluarga MC68HC08 merupakan Mikrokontroler yang dirancang untuk aplikasi pengontrolan yang memiliki *Flash Erasable and Programmable Read Only Memori* (EPROM) didalamnya dan dilengkapi dengan ROM, RAM, I/O dan fasilitas tambahan yaitu analog to digital converter pada satu chip (*one chip microcomputer*). Mikrokontroler ini diproduksi oleh *Motorola semiconductor* yang sekarang dikenal dengan *freescall semiconductor*. Diagram susunan kaki dan simbol logika untuk Mikrokontroler MC68HC908QB8 ditunjukkan pada Gambar 2.6.



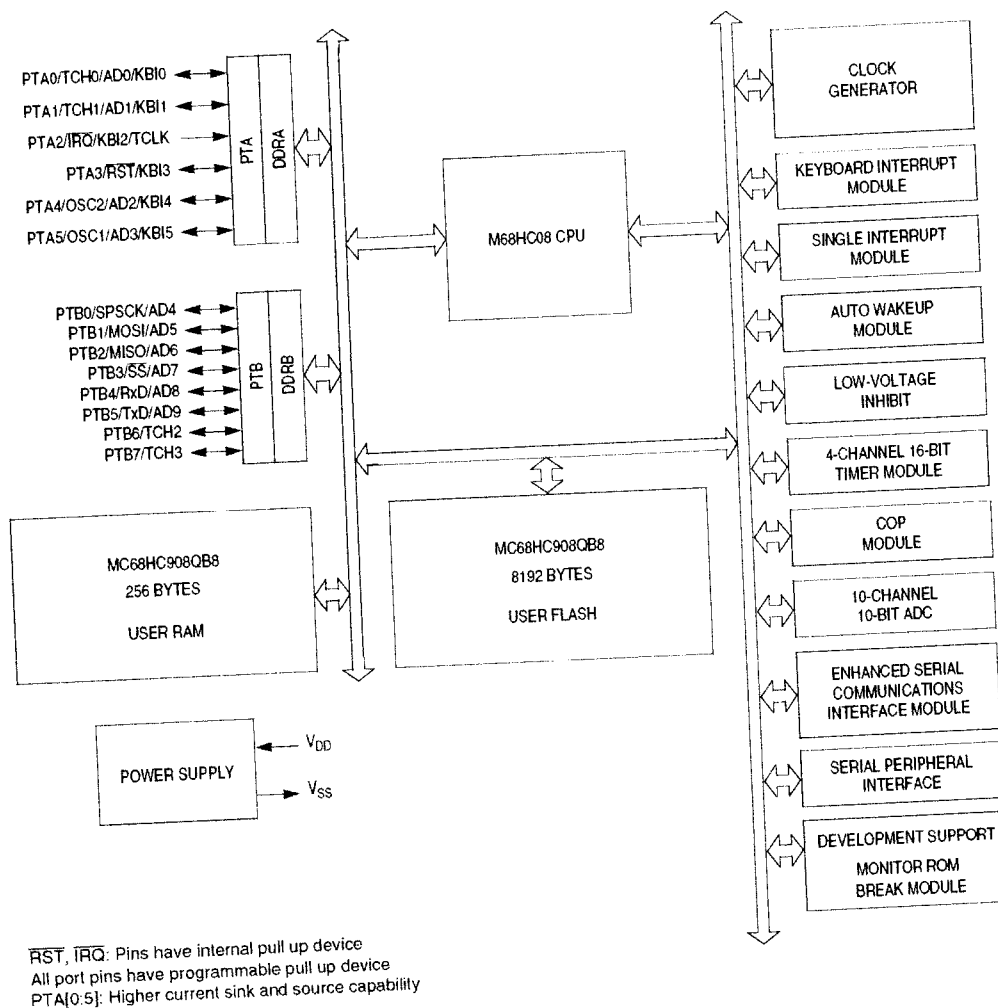
Gambar 2.6 Susunan kaki Mikrokontroler MC68HC908QB8

Tabel 2.1 Fungsi kaki-kaki pada Mikrokontroler MC68HC908QB8

Pin Name	Description	Input/Output
V _{DD}	Power supply	Power
V _{SS}	Power supply ground	Power
PTA0	PTA0 — General purpose I/O port	Input/Output
	TCH0 — Timer Channel 0 I/O	Input/Output
	AD0 — A/D channel 0 input	Input
	KBI0 — Keyboard interrupt input 0	Input
PTA1	PTA1 — General purpose I/O port	Input/Output
	TCH1 — Timer Channel 1 I/O	Input/Output
	AD1 — A/D channel 1 input	Input
	KBI1 — Keyboard interrupt input 1	Input
PTA2	PTA2 — General purpose input-only port	Input
	IRQ — External interrupt with programmable pullup and Schmitt trigger input	Input
	KBI2 — Keyboard interrupt input 2	Input
	TCLK — Timer clock input	Input
PTA3	PTA3 — General purpose I/O port	Input/Output
	RST — Reset input, active low with internal pullup and Schmitt trigger	Input
	KBI3 — Keyboard interrupt input 3	Input
PTA4	PTA4 — General purpose I/O port	Input/Output
	OSC2 — XTAL oscillator output (XTAL option only) RC or internal oscillator output (OSC2EN = 1 in PTAPUE register)	Output Output
	AD2 — A/D channel 2 input	Input
	KBI4 — Keyboard interrupt input 4	Input
PTA5	PTA5 — General purpose I/O port	Input/Output
	OSC1 — XTAL, RC, or external oscillator input	Input
	AD3 — A/D channel 3 input	Input
	KBI5 — Keyboard interrupt input 5	Input
PTB0	PTB0 — General-purpose I/O port	Input/Output
	SPSCK — SPI serial clock	Input/Output
	AD4 — A/D channel 4 input	Input
PTB1	PTB1 — General-purpose I/O port	Input/Output
	MOSI — SPI Master out Slave in	Input/Output
	AD5 — A/D channel 5 input	Input
PTB2	PTB2 — General-purpose I/O port	Input/Output
	MISO — SPI Master in Slave out	Input/Output
	AD6 — A/D channel 6 input	Input
PTB3	PTB3 — General-purpose I/O port	Input/Output
	SS — SPI slave select	Input
	AD7 — A/D channel 7 input	Input
PTB4	PTB4 — General-purpose I/O port	Input/Output
	RxD — ESC1 receive data I/O	Input/Output
	AD8 — A/D channel 8 input	Input
PTB5	PTB5 — General-purpose I/O port	Input/Output
	TxD — ESC1 transmit data I/O	Output
	AD9 — A/D channel 9 input	Input
PTB6	PTB6 — General-purpose I/O port	Input/Output
	TCH2 — Timer channel 2 I/O	Input/Output
PTB7	PTB7 — General-purpose I/O port	Input/Output
	TCH3 — Timer channel 3 I/O	Input/Output

2.6.1 Diagram Blok Mikrokontroler MC68HC908QB8

Diagram blok Mikrokontroler MC68HC908QB8 ditunjukkan pada Gambar 2.7 dibawah ini.



Gambar 2.7 Diagram blok Mikrokontroler MC68HC908QB8

2.6.2 Organisasi Memori

Organisasi memori merupakan fasilitas pada Mikrokontroler MC68HC908QB8 yang telah tersedia didalamnya. Organisasi memori ini terdiri dari RAM dan ROM.

2.6.2.1. *Random Access Memory (RAM)*

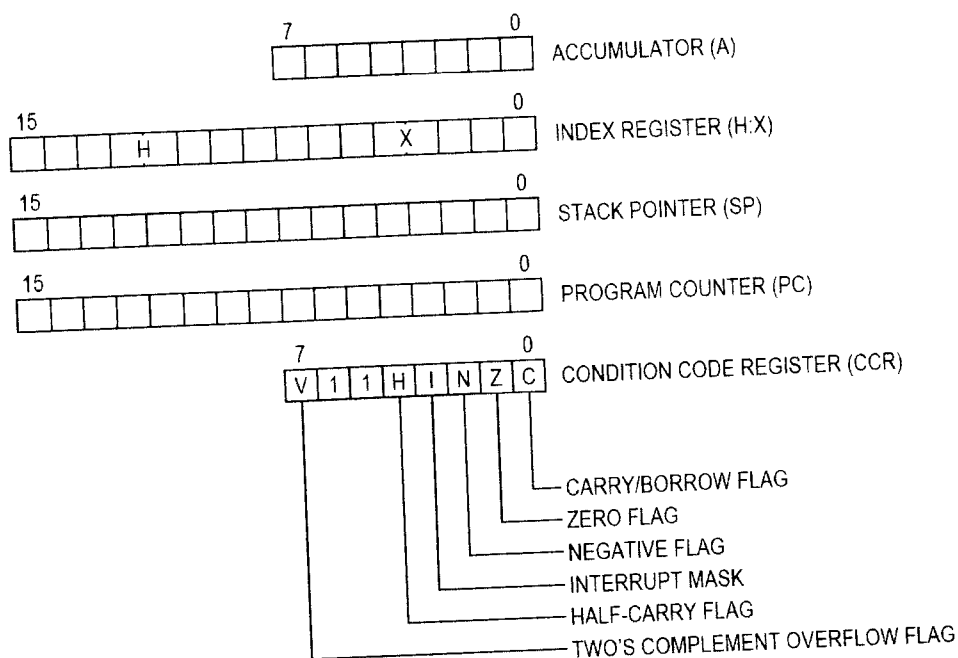
RAM adalah jenis memori yang dapat dibaca dan ditulis digunakan untuk menyimpan memori data dan bersifat *volatile* (terhapus jika catu daya terputus). Pada Mikrokontroler MC68HC908QB8 disediakan RAM internal sebesar 128 byte.

2.6.2.2 *Read Only Memory (ROM)*

ROM merupakan memori yang dipergunakan untuk menyimpan data sebagai dasar pengoperasian Mikrokontroler. ROM bersifat *non volatile* (data tetap tersimpan walupun catu daya terputus). Pada Mikrokontroler MC68HC908QB8 telah memiliki ROM internal yang digunakan untuk menyimpan program selama tidak melebihi kapasitas, dimana kapasitas ROM internal sebesar 4 Kbyte dan I/O port yang tersedia sebanyak 13 saluran bidireksional dan 1 pin khusus untuk input saja.

2.6.3 *Register-register*

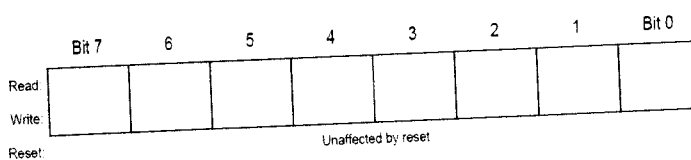
Pada Mikrokontroler seri MC68HC908QB8 register-register fungsi khusus masih mengacu pada core CPU MC68HC05. Masing-masing register ini dapat dilihat pada gambar berikut.



Gambar 2.8 Register-register Mikrokontroler MC68HC908QB8

2.6.3.1 Accumulator

Accumulator adalah register 8-bit untuk keperluan umum. CPU menggunakan register accumulator untuk menyimpan operand dan hasil dari operasi logika atau aritmatika.

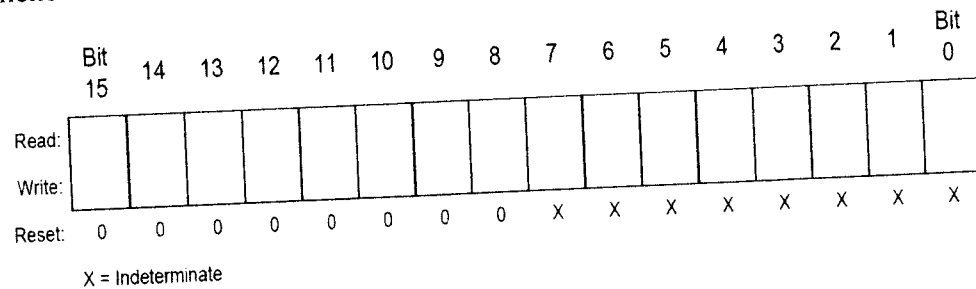


Gambar 2.9 Accumulator (A)

2.6.3.2 Index Register

Index register adalah register 16-bit, terbagi atas index register H yang merupakan index register byte teratas (*Upper Byte Index Register*) dan index

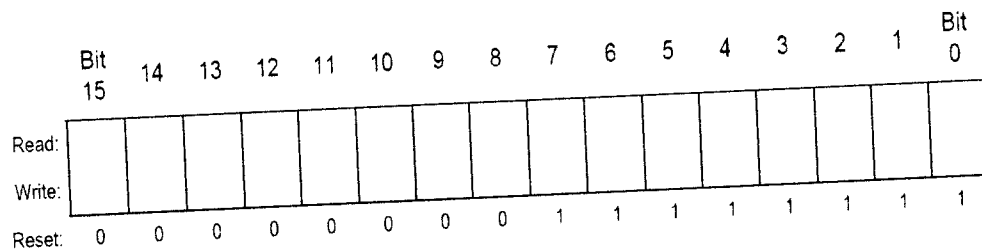
register X (*Low Order Byte*), serta mencabang ruang alamat memori program sampai 64 Kbyte. H:X adalah yang menghubungkan 16-bit *index register*. Pada mode *Indexed addressing*, CPU menggunakan isi dari *index register* untuk menentukan kondisi alamat dari operand.



Gambar 2.10 *Index Register (H:X)*

2.6.3.3 *Stack Pointer (SP)*

Stack Pointer (SP) adalah register 16-bit menunjuk ke alamat RAM internal yang terhubung dengan operasi tertentu untuk menyimpan dan mengambil data secara cepat. Pada kondisi reset, stack pointer berada di alamat \$00FF.

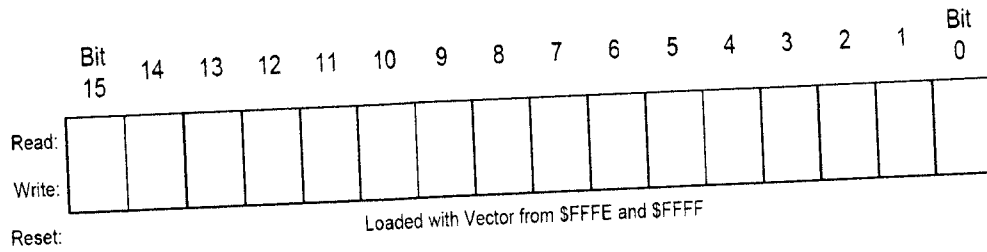


Gambar 2.11 *Stack Pointer (SP)*

2.6.3.4 *Program Counter*

Program Counter adalah register 16-bit yang berisi alamat-alamat dari instruksi atau operand yang akan dibaca oleh CPU. *Program counter* secara

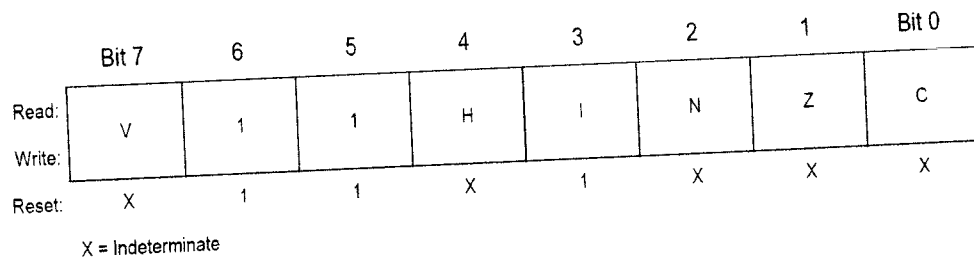
otomatis menaikkan nilai alamat memori setelah instruksi atau operand di eksekusi oleh CPU.



Gambar 2.12 Program Counter (PC)

2.6.3.5 Condition Code Register

Condition Code Register adalah register 8-bit yang berisi *interrupt mask* dan 5 buah flag yang mengindikasikan hasil eksekusi instruksi. Bit ke 5 dan Bit ke 6 secara permanen di set pada logika 1.



Gambar 2.13 Condition Code Register (CCR)

Berikut penjelasan singkat tentang flag yang ada pada *condition code register*.

- *Overflow Flag (V)*

CPU akan menentukan *overflow flag* pada kondisi dua buah komplement overflow pada logika 1 dengan instruksi cabang BGT, BGE, BLE dan BLT.

- *Half Carry Flag (H)*
Flag ini digunakan pada operasi aritmatika *binary coded desimal (BCD)*. CPU akan menentukan *half carry flag* pada kondisi *carry* terjadi antara bit-3 dan bit-4 dalam operasi *add-without-carry (ADD)* atau *add-with-carry (ADC)*.
- *Interrupt Mask (I)*
Flag ini di fungsikan pada logika 0 yang berfungsi untuk menolak semua *interrupt* sehingga CPU tetap akan melanjutkan eksekusi pada program sampai instruksi *interrupt mask* berlogika 1.
- *Negative Flag (N)*
CPU akan menentukan *negative flag* pada operasi aritmatika, operasi logika atau prosedur manipulasi data yang menghasilkan nilai negatif.
- *Zero Flag (Z)*
Flag ini akan di set oleh CPU pada kondisi operasi aritmatika, operasi logika atau prosedur manipulasi data yang menghasilkan nilai \$00.
- *Carry/Borrow Flag (C)*
CPU akan menentukan *Carry Flag* pada prosedur operasi penambahan sehingga lebih dari bit-7 pada *accumulator* atau pada kondisi operasi pembagian yang membutuhkan nilai peminjaman.

2.6.4 *Timer Interface Module (TIM)*

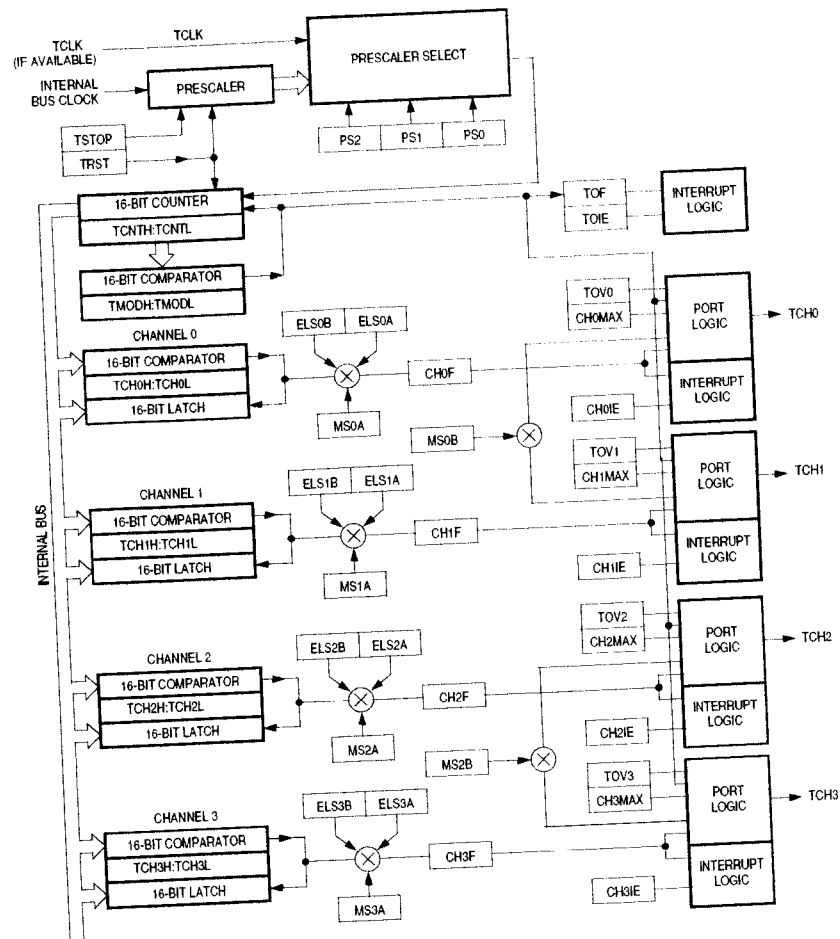
Mikrokontroler MC68HC908QB8 mempunyai 4 channel pewaktu/pencacah (*timer counter*) 16 bit yang digunakan untuk pengukuran

interval waktu, lebar pulsa dan berfungsi juga sebagai modulasi lebar pulsa (PWM).

TIM ini juga memiliki fitur-fitur seperti :

- Dua *channel* perbandingan input dan output
- Pembangkit sinyal PWM *buffer* dan *unbuffered*
- Frekuensi clock input yang dapat di diprogram sesuai kecepatan bus internal,
- Beroperasi dengan sistem *free running*
- Pin toggle pada kondisi salah satu *channel overflow*
- Bit stop dan reset pada pencacah

Pin yang digunakan oleh TIM yaitu pin PTA0/TCH0, PTA1/TCH1, PTB6/TCH2 dan PTB7/TCH3. Gambar 2.14 merupakan arsitekur dari TIM. Komponen utama adalah *counter* 16-bit yang dapat beroperasi secara *free running* atau terkendali. TIM *counter* berfungsi sebagai referensi untuk umpan input dan perbandingan dengan output. TIM dikendalikan oleh register TMODH:TMODL yang menyediakan nilai TIM. Pada *software* juga dapat dilakukan pembacaan nilai *TIM counter* tanpa mengganggu proses pewaktuan. Kedua TIM Channel dapat diprogram agar independen sebagai input saja atau sebagai komparator *output*.



Gambar 2.14 Diagram blok *Timer Interface Module (TIM)*

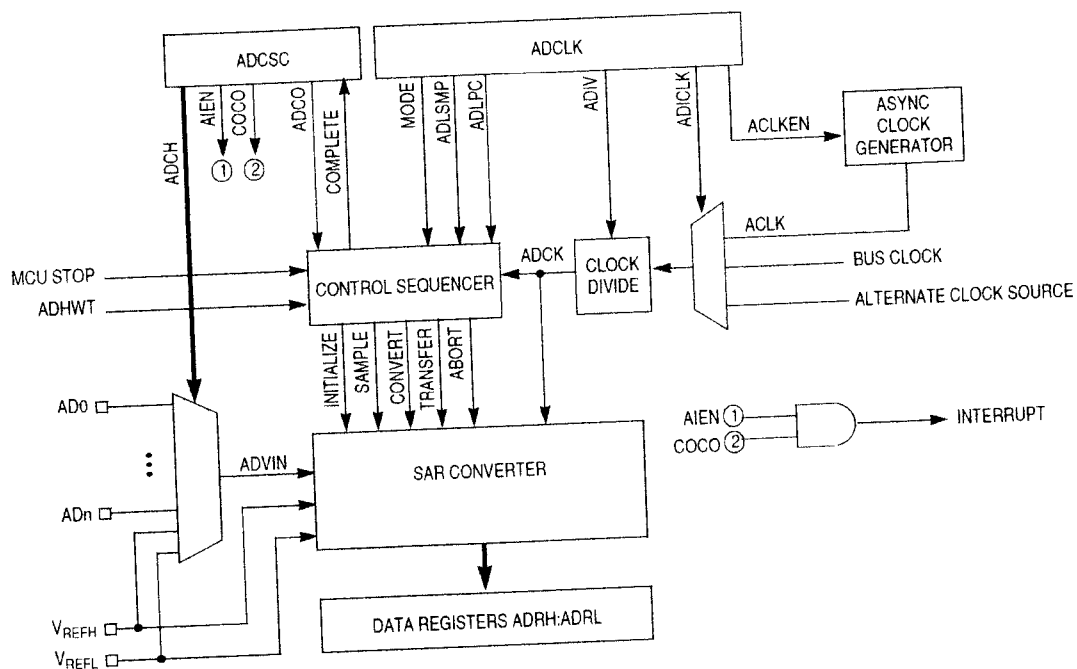
Register kendali I/O dan monitoring pada TIM adalah sebagai berikut:

- *TIM status and control register (TSC)*
- *TIM control registers (TCNTH:TCNTL)*
- *TIM counter modulo registers (TMODH:TMODL)*
- *TIM channel status and control registers (TSC0 and TSC1)*
- *TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)*

2.6.5 ADC (*Analog to Digital Converter*)

Piranti-piranti dan sistem-sistem logika hanya mengenal isyarat digital. Tetapi, kebanyakan isyarat seperti isyarat yang dihubungkan dengan transduser mempunyai bentuk analog. Sebelum diumpankan ke piranti digital isyarat-isyarat ini harus diubah ke dalam bentuk digital oleh pengubah analog ke digital atau *analog to digital converter (ADC)*. Pada transduser, isyarat analog dari sensor suhu diubah menjadi isyarat digital menggunakan ADC. Banyak sekali aplikasi yang dapat digunakan pada rangkaian konverter digital, salah satunya adalah untuk mengukur besarnya suhu didalam bak penampung air.

ADC yang digunakan adalah fasilitas yang sudah ada disediakan didalam Mikrokontroler MC68HC908QB8 sehingga tidak diperlukan lagi ADC eksternal. Pada ADC internal Mikrokontroler MC68HC908QB8 dapat dilakukan proses konversi secara *single* ataupun *continuous*. Konversi tegangan analog ke digital dengan menggunakan metode pendekatan berurutan atau beruntun (*Successive Approximation Register*) dengan resolusi 10-bit dan waktu konversi sesuai dengan kecepatan proses pada unit Mikrokontroler. Pendekatan beruntun adalah metoda yang paling banyak digunakan. Metode ini didasari pada pendekatan isyarat *input* dengan kode biner dan kemudian berturut-turut memperbaiki pendekatan ini untuk setiap bit pada kode sampai didapatkan pendekatan yang paling baik. Metode SAR digunakan untuk menyimpan kode biner pada setiap tahapan dalam proses.



Gambar 2.15 Diagram blok ADC internal Mikrokontroler MC68HC908QB8

Seperti yang terlihat pada Gambar 2.17, ADC ini menggunakan saluran yang sama dengan saluran input output sehingga pada pemrograman dapat dipilih dengan piranti lunak untuk mengaktifkannya. Pada keluaran input multiplexer (ADVIN) akan di konversi ke oleh SAR CONVERTER kemudian disimpan pada data register (ADRH dan ADRL).

Spesifikasi ADC pada dasarnya adalah sama dengan spesifikasi pada DAC, yaitu diantaranya :

1. Resolusi

Resolusi adalah perubahan masukan terkecil yang menyebabkan berubahnya *ouput*. Resolusi pada sebuah ADC dapat dianggap sebagai kesalahan kuantisasi. Semakin banyak jumlah bit yang mengkode isyarat maka kesalahan kuantisasi ini semakin dapat direduksi. Kesalahan

kuantisasi secara umum adalah sebesar \pm LSB. Perumusan resolusi secara umum adalah sebagai berikut :

$$\begin{aligned} \% \text{ Resolusi} &= (1 / \text{jumlah step total}) \times 100\% \\ &= (1 / (2^n - 1)) \times 100\% \end{aligned} \quad (2.4)$$

dengan n = jumlah bit

2. Akurasi

Keakuratan sebuah ADC tergantung pada resolusi, tetapi tergantung pada keakuratan komponen-komponen penyusunnya, seperti : komparator, resistor dan level *amplifier*-nya.

3. Waktu Konversi

Waktu konversi yaitu waktu yang dibutuhkan untuk mengubah isyarat analog ke isyarat digital. Waktu konversi (*conversion timing*) ini sangat penting karena *bandwidth* isyarat input merupakan hal yang sangat signifikan dalam perhitungan waktu konversi.

Modul ADC Internal MC68HC908QB8 merupakan modul konversi tegangan analog ke digital dengan spesifikasi :

- Masukan analog yang berbagi dengan input output
- Resolusi 10-bit dengan ketelitian \pm 1 angka LSB
- Tegangan *input* maksimum 5 volt, 3mA
- Model pendekatan beruntun (*successive approximation register*).
- Tidak perlu pengatur tegangan nol dan tegangan maksimum.

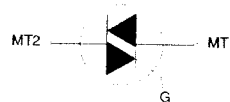
maka ketelitian modul ADC ini dapat ditentukan dengan persamaan :

$$\begin{aligned}
 \text{Ketelitian (V)} &= \frac{V_{\text{ref}}}{\text{Resolusi}} && (2.5) \\
 &= \frac{5}{2^{10} - 1} \\
 &= 0,0048 \text{ volt atau } 4,8 \text{ mV} \\
 &= \text{dibulatkan menjadi } 5 \text{ mV}
 \end{aligned}$$

Dengan ketelitian 5 mV maka keluaran pengkondisi isyarat dapat disesuaikan agar data yang didapat pada pengkondisi isyarat dapat sesuai dengan range data dari ADC Internal MC68HC908QB8. Toleransi ketelitian yang lebih besar dari tingkat kenaikan tegangan yang diukur akan menyebabkan kesalahan dalam pengukuran. Untuk menghindari kesalahan tersebut maka tingkat kenaikan yang diukur harus dikuatkan dengan menggunakan rangkaian amplifier sehingga tingkat kenaikan tegangan berada di atas toleransi.

2.7 Triac

Triac adalah komponen semikonduktor yang terbuat dari silikon. Triac merupakan salah satu keluarga thyristor yang mempunyai kemampuan mengantarkan arus 2 arah (*bi directional*). Triac tersusun dari P-N *junction* yang kompleks atau secara teori triac terdiri dari 2 buah SCR yang digabungkan. Triac mempunyai 3 buah kaki yaitu MT1, MT2 dan *gate* (G). Dalam aplikasinya triac biasa digunakan sebagai *switch* untuk beban atau tegangan yang tinggi.



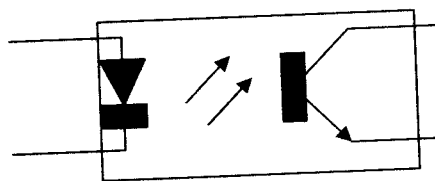
Gambar 2.16 Konfigurasi Triac

Tegangan picu positif yang masuk pada kaki *gate* akan menyebabkan MT2 akan menghambat arus yang melewatinya dan menyebabkan kaki MT1 mengalirkan arus listrik. Hal ini juga berlaku saat kaki *gate* mendapatkan picu arus yang lebih negatif atau 0 maka MT2 akan mengalirkan arus dan MT1 menahan arus.

2.8 Optokopler

Optokopler termasuk dalam transduser optik, dimana komponen ini terdiri dari LED Infra merah sebagai sumber cahaya dan phototransistor sebagai penyensor. Pada umumnya optokopler diproduksi dalam bentuk paket plastik dan dapat diberi lensa atau filter untuk menaikkan kepekaannya.

Dalam sebuah sistem digital optokopler banyak digunakan sebagai saklar digital dan juga sebagai isolator diantara kedua *circuit*. Hal ini disebabkan karena dengan terpisahnya antara jalur cahaya dengan penerima cahayanya sehingga menyebabkan terjadinya sebuah transfer energi listrik dalam satu arah, dari IRED ke photo detektor, sambil mempertahankan isolasi listrik diantara kedua *circuit*, seperti terlihat pada gambar dibawah ini.

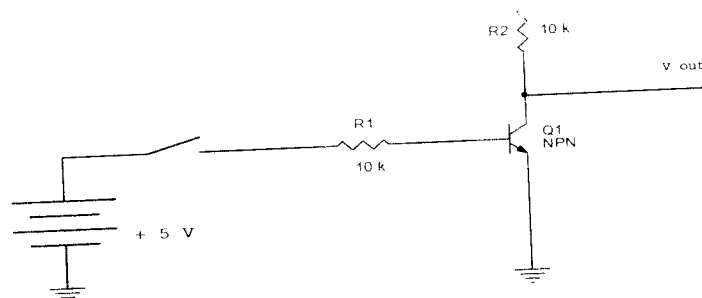


Gambar 2.17 Optokopler

2.9 Transistor Sebagai Saklar

Bias basis dalam rangkaian berguna dalam rangkaian-rangkaian digital karena rangkaian tersebut biasanya dirancang untuk beroperasi didaerah jenuh dan *cut off*. Oleh karena itu, rangkaian bias basis ini memiliki tegangan keluaran rendah ataupun tegangan keluaran tinggi.

Berikut ini adalah sebuah contoh penggunaan rangkaian berbias basis untuk saklar diantara daerah jenuh dan *cut off*. Gambar 2.19 menunjukkan sebuah contoh dari sebuah transistor yang difungsikan sebagai saklar. Ketika saklar tertutup transistor dalam kondisi *hard saturation* maka tegangan keluaran sekitar 0 V. Ketika saklar terbuka, arus basis turun menjadi nol dan mengakibatkan arus kolektor turun menjadi nol. Dengan tidak adanya arus yang melalui resistor 10k, semua tegangan catu kolektor akan melalui terminal kolektor-emiter. Oleh karena itu, tegangan keluaran naik menjadi + 5 V.



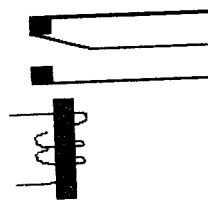
Gambar 2.18 Transistor sebagai saklar.

2.10 Relai

Relai merupakan sebuah saklar yang digerakkan oleh adanya medan magnet yang menarik salah satu *pin*. Gambar 2.16 adalah simbol Relai, dimana

Relai mempunyai 5 *pin*. 2 *pin* merupakan ujung kumparan yang melilit sebuah logam atau besi dan 3 *pin* lagi merupakan *pin-pin* terminal.

Cara kerja alat ini, bila ada arus yang melewati kumparan maka besi akan berubah menjadi medan magnet yang akan menarik *pin* terminal di atasnya. Dengan tertariknya salah satu *pin* oleh medan magnet yang ditimbulkan oleh kumparan maka *pin* tersebut akan menyambung dengan *pin* satunya dan kebalikannya ketika tidak ada arus yang melewatinya.



Gambar 2.19 Simbol relai

2.11 Motor DC

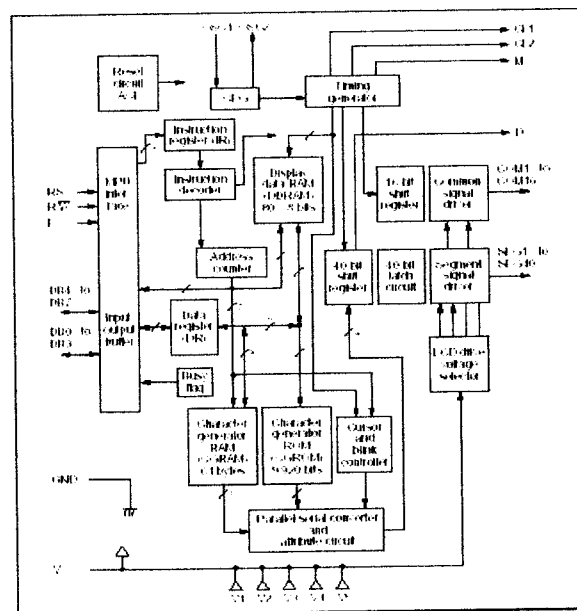
Motor DC adalah suatu sistem mesin yang berfungsi mengubah tenaga listrik arus searah (listrik DC) menjadi gerak atau tenaga mekanis. Motor arus searah atau motor DC banyak digunakan pada peralatan elektronik, misalnya *tape player*, kipas angin portabel, mainan anak, *power window* dan *central lock* pada mobil. Motor ini digerakkan oleh baterai.

Prinsip kerja motor arus searah (DC) berdasarkan pada penghantar yang mengalirkan arus dan ditempatkan tegak lurus pada medan magnet, cenderung bergerak tegak lurus terhadap medan. Besarnya gaya yang diberikan untuk

menggerakkan, berubah sebanding dengan kekuatan medan magnet, besarnya arus yang mengalir tergantung pada jenis penghantar dan panjang penghantar.

2.12 Tampilan LCD (*Liquid Crystal Display*)

Sebagai penampil data digunakan LCD. LCD yang digunakan disini adalah jenis LCD M1632 dengan 16x2 karakter. Dalam LCD tersebut data yang sifatnya sementara dari DD RAM dan CG RAM akan disimpan dalam *Data Register (DR)*. Sedangkan kode-kode instruksi, informasi alamat data display dan pembangkit karakter akan disimpan dalam *Instruction Register*. Pada saat *busy flag* menunjukkan bahwa LCD berada dalam operasi internal yang tidak bisa diterima oleh instruksi berikutnya.



Gambar 2.20 Diagram blok LCD M1632

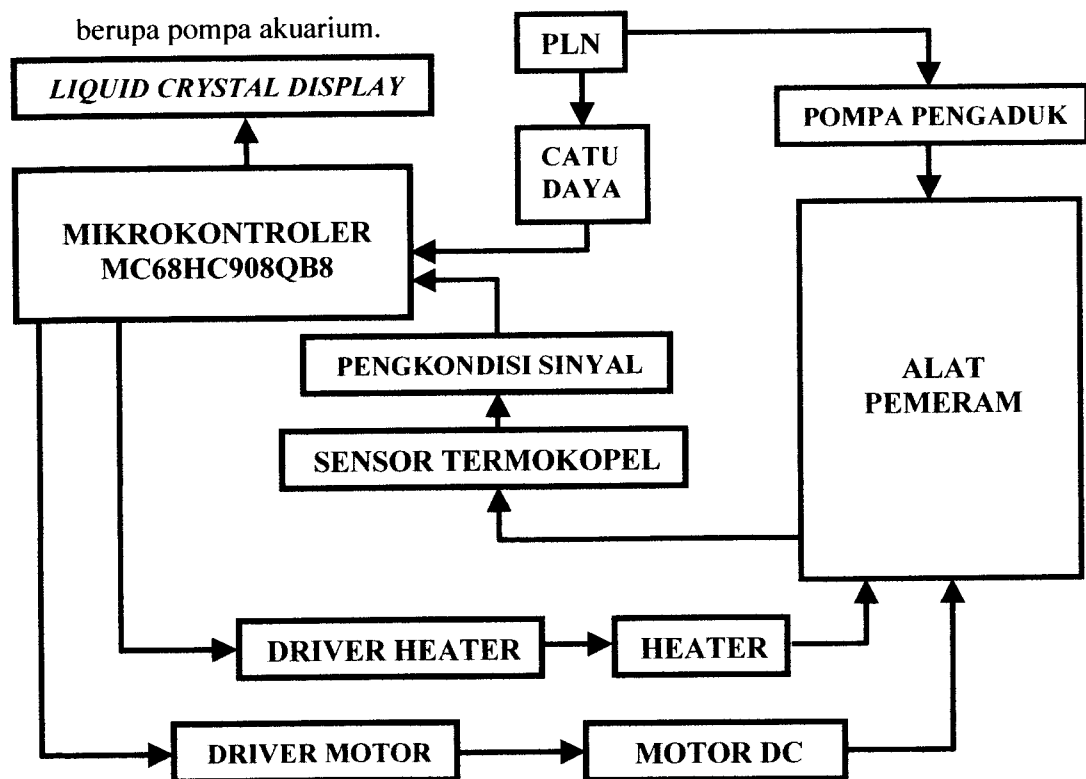
Sebagai pembangkit susunan kode karakter adalah CG ROM, karakter yang dibangkitkan adalah kode karakter 8 bit dan menyediakan 192 pola karakter.

BAB III

**PERANCANGAN DAN IMPLEMENTASI PERANGKAT KERAS SERTA
PERANGKAT LUNAK**

3.1 Perancangan Perangkat Keras

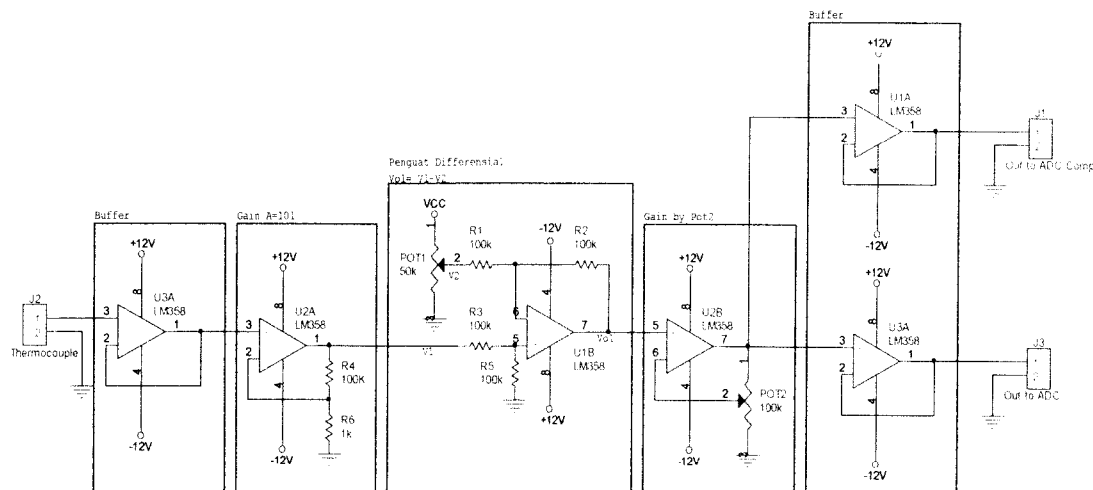
Diagram blok Alat Pemeram Panili Berbasis Mikrokontroler MC68HC908QB8 Dengan Algoritma PID terbagi menjadi beberapa blok diagram yaitu : pengkondisi isyarat, mikrokontroler sebagai pengendali utama dari sistem, penampil *liquid crystal display* sebagai penampil data suhu , *driver motor* sebagai pengendali *motor* dan *driver heater* sebagai pengendali *heater*. Agar didapatkan suhu air yang merata didasar bak pemeram diberikan pompa pengaduk berupa pompa akuarium.



Gambar 3.1 Blok diagram alat pemeram panili

3.1.1 Pengkondisi Isyarat

Unit pengkondisi isyarat ini berfungsi menguatkan sinyal keluaran dari sensor suhu. Sensor suhu berupa termokopel yang mengeluarkan tegangan dalam orde miliovolt, sehingga untuk dapat dibaca ADC perlu dikuatkan lagi.

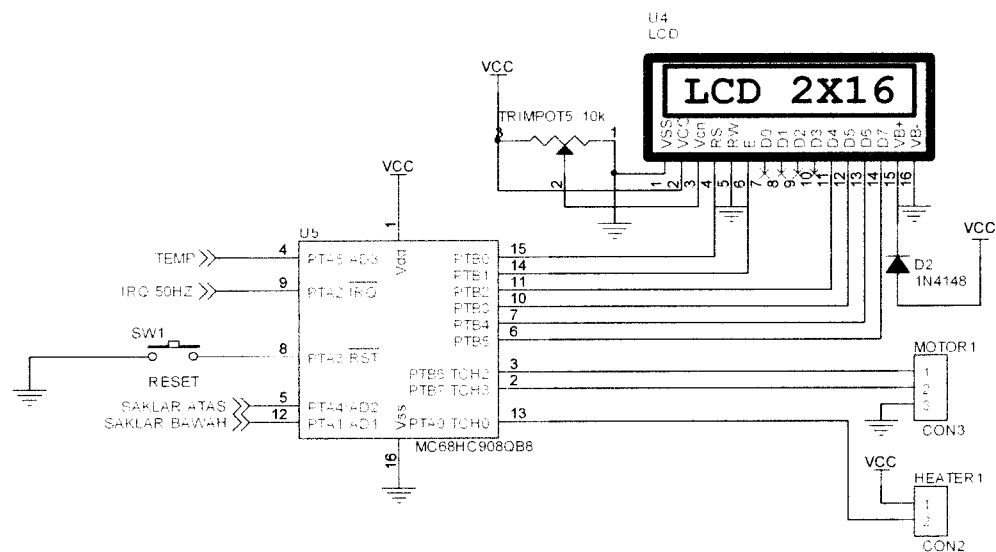


Gambar 3.2 Rangkaian pengkondisi isyarat

Input untuk pengkondisi isyarat adalah keluaran dari termokopel kemudian masuk ke buffer dulu sebelum diumpankan ke penguat noninverting op-amp LM 358. Penguat bagian pertama setelah diberikan buffer ini dikuatkan dulu 101 kali kemudian baru diberikan pada penguat kedua yaitu penguat diferensial. Sinyal keluaran penguat pertama dibandingkan dengan keluaran tegangan referensi dari Vcc. Kemudian keluaran dari penguat diferensial ini dipasang rangkaian penguat berikutnya yang memiliki penguatan yang dapat diubah-ubah agar nilai tegangan dapat sesuai dengan lebar pulsa dari data pengkondisi isyarat maka pada output LM 358 sebuah potensiometer. Besarnya penguatan didapat dengan mengubah nilai potensiometer yang diperlukan pada saat kalibrasi output untuk diumpankan ke input ADC.

3.1.2 Sistem Minimum mikrokontroler MC68HC908QB8

Pada sistem minimum MC68HC908QB8 ini ditampilkan *port-port* yang digunakan sebagai masukan dan keluaran. Sistem minimum ini digunakan sebagai rangkaian utama dari perancangan sistem elektronis ini. Mikrokontroler MC68HC908QB8 mempunyai 12 buah *port bi directional* atau dapat digunakan sebagai masukan dan keluaran. Sedangkan untuk *pin* PTB6 digunakan untuk memberikan keluaran ke *driver motor* naik dan *pin* PTB7 digunakan untuk keluaran *driver motor* turun. *Pin* PTA0 digunakan untuk memberikan keluaran ke *driver heater*. *Pin* PTA1 dan PTA4 digunakan untuk saklar batas atas dan batas bawah. Gambar dari sistem minimum ini dapat dilihat pada Gambar 3.3 berikut



Gambar 3.3 Sistem minimum mikrokontroler MC68HC908QB8

3.1.3 ADC (Analog to Digital Converter)

Pengkonversi analog ke digital (*Analog to Digital Converter*) digunakan untuk menerjemahkan atau mengubah besaran analog yang terbaca oleh

pengkondisi isyarat menjadi sinyal digital sehingga dapat diproses oleh mikrokontroler . Dalam perancangan ini memakai ADC internal mikrokontroller yang mempunyai kemampuan konversi dengan pengaturan *output* 8-bit. ADC ini menggunakan salah satu kanal *input*.

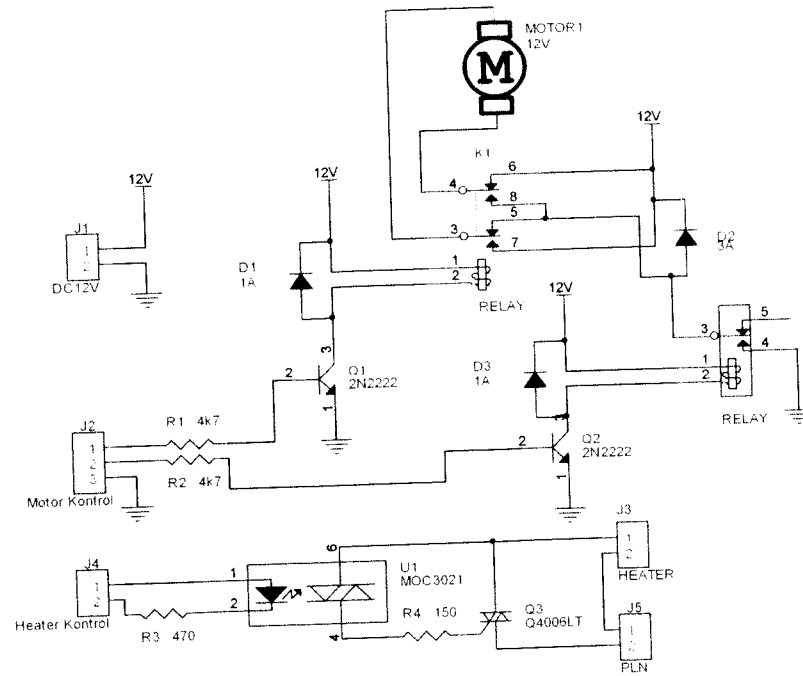
3.1.4 *Driver Motor*

Driver yang digunakan untuk penggerak motor DC 12 volt adalah rangkaian pembalik fasa yang terdiri dari 1 buah *relay* 12 volt. Prinsip kerja dari rangkaian ini adalah menukarkan fasa pada input *motor* sehingga gerakan *motor* berbalik. Pada Gambar 3.4 dapat dilihat adanya transistor NPN yang berfungsi sebagai saklar. Saat PTB5 diberi logika tinggi maka arus akan melewati R1 dan menuju basis sehingga menjenuhkan *emitor* dan *colector* sehingga arus mengalir menuju ground dan *relay* bekerja menukarkan fasa motor.

Arus yang diinginkan mengalir pada basis besarnya :

$$\begin{aligned}
 I &= (V-0.7)/R && (3.1) \\
 &= (5-0.7) / 4700 \\
 &= 0.91 \text{ mA}
 \end{aligned}$$

Saat PTB4 diberi logika tinggi arus akan mengalir menuju R2 dan menuju transistor yang menyebabkan transistor menjadi jenuh sehingga arus dapat mengalir ke *ground* yang membuat *relay* dan motor bekerja.

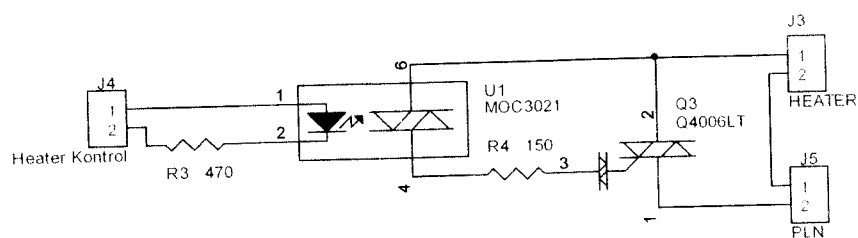


Gambar 3.4 Rangkaian *driver motor*

3.1.5 *Driver Heater*

Pengendali pemanas pada perancangan ini menggunakan TRIAC atau dioda 2 arah yang dihubungkan ke optotriac MOC 3021 atau IC *non zero crossing triac driver* yang berfungsi untuk menyekat antara tegangan rendah dari mikrokontroler dan tegangan tinggi dari jala-jala listrik. Pada perancangan ini *heater* yang digunakan 7 buah *heater* air masing-masing dengan daya 300 watt sehingga triac yang digunakan adalah triac dengan seri Q4010LT yang mempunyai kemampuan menyekat daya maksimal sebesar 4000 watt (dalam *data sheet* 400 awal berarti volt dan angka terakhir berarti arus jadi dalam hitungan nilai maksimal adalah $400 \times 10 = 4000$ watt).

Pada rangkaian *driver heater* Gambar 3.5 dapat dilihat bahwa prinsip kerja dari rangkaian ini adalah sebagai pengontrol fasa pada jalur catu daya AC. PTA0 diberi logika rendah maka arus akan mengalir ke *led* yang ada didalam optotriac dan menjenuhkan triac yang ada pada optotriac dan membuat triac membuka sekatan dan menghidupkan *heater*. Dengan mengatur *timing* pada pembukaan sekatan tegangan triac akan diperoleh hasil pengendalian jumlah tegangan yang masuk menuju heater. Sehingga suhu air dapat dikendalikan.

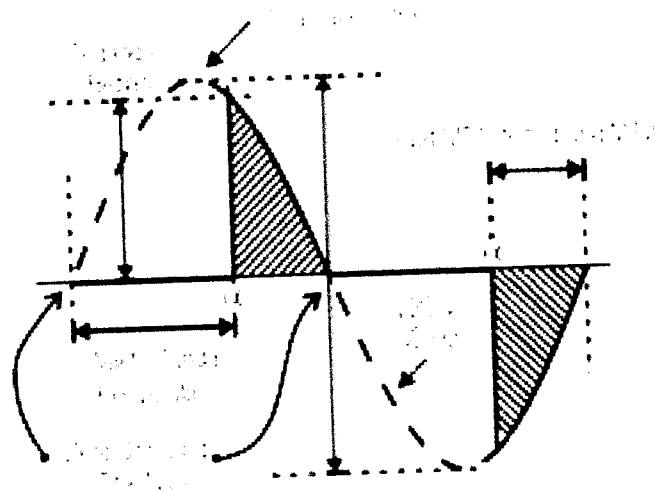


Gambar 3.5 *Driver Heater*

3.1.6 Pengaturan Daya

Untuk menjaga suhu air pada bak perendam tetap pada *set point* digunakan algoritma PID pada perangkat lunak. Algoritma PID menentukan jumlah daya yang dikirimkan ke pemanas. Pengaturan ini dilakukan dengan memberikan arus *gate* sesaat pada triac agar MT1 dan MT2 terhubung (lihat Gambar 2.17 simbol triac). MT1 terhubung terus ke MT2 selama arus beban yang mengalir lebih besar dari arus minimum (*holding current*). Mengingat sumber daya yang dipakai berasal dari tegangan bolak-balik, pada daerah *titik nol* (*zero crossing*) dari tegangan bolak balik (lihat Gambar 3.6), arus beban yang mengalir akan mengecil sampai kurang dari arus minimum yang diperlukan, akibatnya hubungan antara

MT1 dan MT2 akan terputus dengan sendirinya. Daya yang disalurkan ke beban tergantung pada lamanya MT1 terhubung ke MT2 setiap setengah periode tegangan sinus dari jala-jala listrik, yakni bagian yang di-arsir dalam Gambar 3.6, pada saat-saat itulah beban menerima daya.

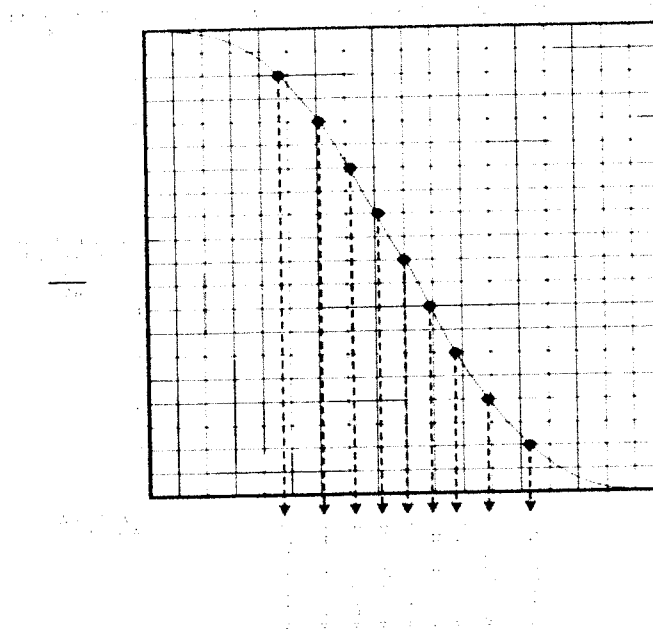


Gambar 3.6 Waktu tunda dan saat MT1 dan MT2 terhubung.

Dengan demikian, daya yang disalurkan ke beban bisa diatur dengan mengatur waktu tunda saat penyulutan TRIAC, terhitung mulai saat tegangan sinus jala-jala listrik mencapai titik nol. Teknik pengaturan daya semacam ini dikatakan sebagai teknik *phase control*.

Grafik pada Gambar 3.7 menggambarkan hubungan antara waktu tunda penyulutan dengan daya yang disalurkan (P) dibagi dengan daya maksimum (P_{max}). Dengan waktu tunda 0 mili-detik, P/P_{max} mencapai 1.0 artinya semua daya disalurkan ke beban. Untuk frekuensi jala-jala listrik 50 Hz, waktu tunda maksimum adalah 10 mili-detik, saat itu P/P_{max} bernilai 0 artinya tidak ada daya yang disalurkan.

Hubungan waktu tunda dengan nilai P/P_{max} tidak linear, skala bagian bawah dari grafik Gambar 3.7 memperlihatkan nilai waktu tunda untuk memperoleh berbagai nilai P/P_{max} .



Gambar 3.7 Grafik waktu tunda vs P/P_{max}

Agar bisa menentukan waktu tunda dengan tepat untuk mendapatkan hasil pengaturan daya yang akurat, mikrokontroler harus mengetahui saat titik nol (*zero crossing*) dari tegangan jala-jala listrik. Dengan rangkaian pemantau titik nol tegangan jala-jala listrik yang dibentuk dengan transistor NPN sebagai interup. Pada tegangan yang sudah disearahkan ketika melewati transistor NPN akan dikuatkan sehingga menjadi gelombang kotak yang diumpankan ke kaki PTA3/IRQ mikrokontroller MC68HC908QB8. Setiap kali tegangan jala-jala listrik mulai meninggalkan titik nol, gelombang kotak akan berubah dari '1' menjadi '0' yang merupakan sinyal permintaan interupsi bagi mikrokontroller MC68HC908QB8.

3.2 Perancangan Perangkat Lunak (*Software*)

3.2.1 Gambaran Umum Program

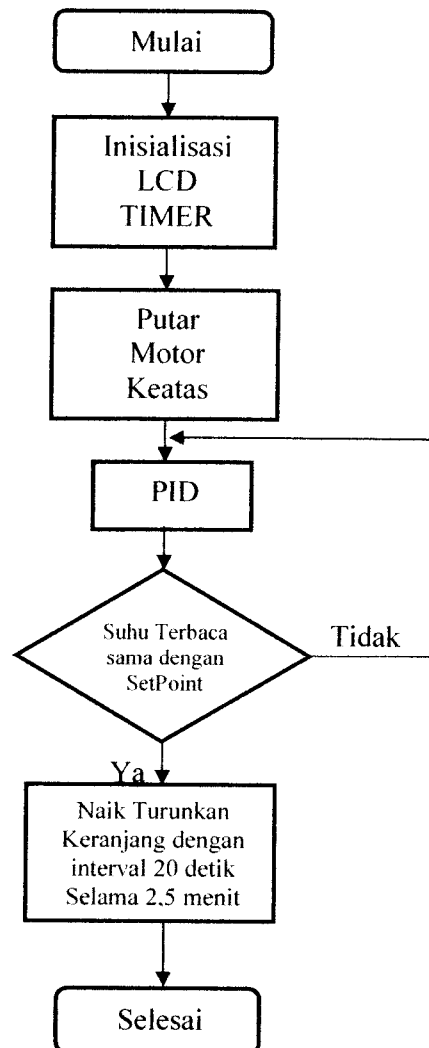
Perangkat lunak pada Alat Pemeram Panili Berbasis Mikrokontroler MC68HC908QB8 Dengan Algoritma PID ini berfungsi untuk mengendalikan keseluruhan kerja dari alat ini. Dalam diagram alir utama perangkat lunak dapat dilihat aliran kerja dari perangkat lunak yang dimulai dari langkah inialisasi sampai dengan program berakhir.

Program dimulai dengan inialisasi LCD dan Timer kemudian dengan mengecek posisi keranjang agar keranjang ketika proses dimulai posisi keranjang berada diatas untuk pengisian beban yaitu panili. Kemudian langkah berikutnya adalah memanaskan air sehingga tercapai suhu setpoint dan menggerakkan keranjang turun naik dengan interval 20 detik selama 2.5 menit.

Walapun siklus pemeraman berakhir pengendali suhu tetap bekerja mempertahankan suhu air tetap pada setpoint untuk melakukan proses pemeraman berikutnya. Hal ini agar ketika melakukan pemeraman berikutnya dengan suhu yang tetap terjaga, pemeraman dapat dilakukan berulang-ulang.

Program utama juga memiliki sub rutin kendali yang merupakan bagian dari program utama yaitu sub rutin kendali motor yang berfungsi sebagai pengendali gerak motor untuk menaikkan dan menurunkan keranjang kedalam bak perendam dan sub rutin kendali suhu yaitu sub rutin untuk mengendalikan suhu air pemeraman berupa algoritma PID yang akan mengendalikan jumlah daya yang diberikan ke *heater* melalui *driver heater*.

Bagan alir program utama dapat dilihat sebagai berikut:

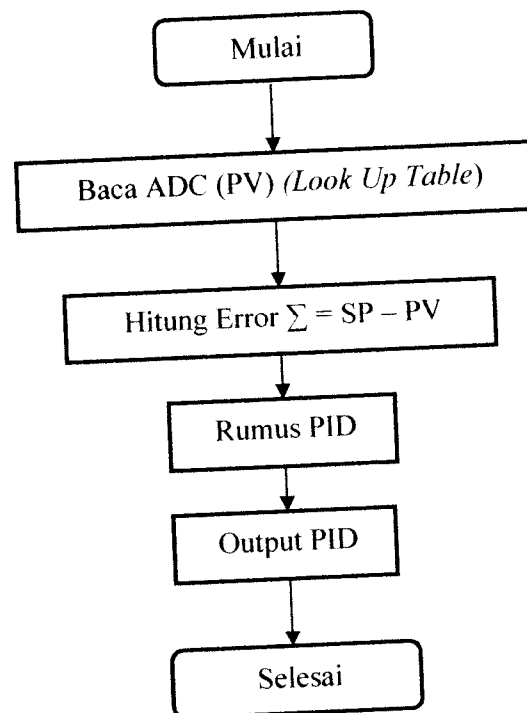


Gambar 3.8 Bagan Alir Program Utama

3.2.2 Sub Rutin Kontrol Heater

Heater yang digunakan untuk menaikkan suhu air dikendalikan oleh sistem kontrol PID dengan memanfaatkan sudut picu pada triac. Untuk memicu triac digunakan referensi frekuensi dari PLN yaitu 50Hz sehingga titik picu dapat

ditentukan sehingga daya diumpungkan ke heater dapat dikendalikan. Ketika pembacaan ADC yang bekerja dalam kondisi *free running* diperoleh nilai yang dieksekusi oleh kontrol algoritma PID pada start awal kurang dari nilai SP (*Set Point*) maka sudut picu akan dilakukan pada sudut 0° . PID terus akan mempertahankan suhu pada set point dengan mengatur sudut picu sesuai dengan jumlah *error* dan *feedback*.



Gambar 3.9 Bagan alir sub rutin kontrol *heater*

3.2.3 Perancangan Pengendali PID

Perancangan sistem yang digunakan sebagai pengendali ini menggunakan rumus PID Digital sebagai berikut.

$$V_o = V' + K_p (e_0) + K_i (\text{Sumerror}) + K_d (e_0 - e_1)$$

Dimana : V_o	= Output PID
V'	= Output PID sebelumnya
K_p	= Konstanta Proporsional
K_i	= Konstanta Integral
K_d	= Konstanta derivatif
e_o	= Error sekarang
e_i	= Error sebelumnya

Output PID berupa sudut picu yaitu sinyal yang dikirimkan ke *driver heater*. Sinyal ini menentukan *timing* dari teknik *phase control* untuk mengirimkan daya ke beban. Penentuan jumlah daya ke beban diperoleh dari perhitungan rumus PID dengan masukan sinyal *feedback* yang didapatkan dari selisih nilai *Setting Point* (SP) dan *Present Value* (PV). Untuk mendapatkan nilai K_p , K_i , K_d dilakukan *tuning PID* dengan metode *trials and errors* yang selanjutnya nilai tersebut dimasukkan dalam rumus PID digital. Hasil pembacaan ADC yang merupakan level digital di sesuaikan dengan tabel suhu yang telah dikalibrasi terlebih dahulu. Selisih antara hasil pembacaan dan nilai SP yaitu *error*(e_o) merupakan input rumus PID. Kendali Proporsional memberikan kontribusi sebagai kelipatan error sebesar ketetapan P. Kendali Integral merupakan jumlah error dikalikan dengan ketetapan I, dan Kendali Diferensial merupakan selisih antara error saat ini dan error sebelumnya dikalikan dengan ketetapan D. Semakin kecil error, maka nilai PID akan semakin kecil dan jika terjadi terus maka akan semakin stabil, atau error akan 0, dan suhu yang terbaca

akan sama dengan *setpoint*. Jika terjadi perubahan suhu, maka sistem akan kembali memberi tanggapan untuk membuat suhu kembali stabil.

Misalnya diberikan contoh operasi aksi kendali suhu pada suhu 47° dengan *set point* 50°. Pada suhu 47° diambil sebagai nilai awal untuk memasukkan ke rumus PID digital. Dengan metode *trials and errors* dimisalkan nilai konstanta $K_p=10$, $K_i=1/100$ dan $K_d=10$ maka perhitungannya adalah :

$$\begin{aligned} V_o &= V' + K_p (e_o) + K_i (\text{Sumerror}) + K_d (e_o - e_1) \\ &= 0 + 10 (3) + 3/100 + 10 (3 - 3) \\ &= 30 + 0.03 + 0 \\ &= 30.03 \end{aligned}$$

Dengan nilai perhitungan diatas diperoleh nilai PID 30.03 kemudian dilanjutkan dengan menentukan timing waktu tunda untuk *driver heater*. Dengan frekuensi bus mikrokontroler 3.2MHz sebagai perhitungan clock untuk timer. Perhitungan setengah siklus sinus adalah selama 3.2MHz tetapi pada percobaan efektifitas penyulutan penuh diperoleh pada nilai 2.5MHz. Dari perolehan nilai PID dimasukkan ke rumus perhitungan waktu tunda, yaitu :

$$\begin{aligned} t &= (250 - \text{PID}) * 100 \\ t &= (250 - 30.03) * 100 \\ t &= 21997 \end{aligned}$$

Dengan referensi interupsi PLN 50Hz sebagai *starting point* maka timer akan mulai menghitung sejumlah 21997 detak dan mengirim sinyal ke *driver heater*. Sinyal ini yang akan mengaktifkan optokopler yang akan mengaktifkan triac agar membuka sekatan sehingga arus mengalir menuju *heater*. Pada clock

dengan nilai 25000 maka sama dengan sudut penyulutan 180° , sehingga nilai masing-masing nilai sudut penyulutan dapat diperoleh dengan perhitungan :

$$\alpha = \frac{180 * t}{25000}$$

$$\alpha = \frac{180 * 21997}{25000}$$

$$\alpha = 158.58^\circ$$

Dengan menggunakan perhitungan yang sama pada suhu 48° nilai *timing* penyulutan juga dapat dilakukan dengan perhitungan sebagai berikut :

$$\begin{aligned} V_o &= V^* + K_p (e_0) + K_i (\text{Sumerror}) + K_d (e_0 - e_1) \\ &= 30.03 + 10 (2) + 5/100 + 10 (2 - 3) \\ &= 30.03 + 20 + 0.05 - 10 \\ &= 40.08 \end{aligned}$$

Kemudian mencari *timing* penyulutan.

$$t = (250 - \text{PID}) * 100$$

$$t = (250 - 40.08) * 100$$

$$t = 20992$$

Sudut penyulutannya dapat diperoleh dengan perhitungan berikut,

$$\alpha = \frac{180 * t}{25000}$$

$$\alpha = \frac{180 * 20992}{25000}$$

$$\alpha = 151.14^\circ$$

Begitu juga dengan nilai suhu 49° dapat dilakukan perhitungan berikut,

$$\begin{aligned} V_o &= V' + K_p (e_o) + K_i (\text{Sumerror}) + K_d (e_o - e_1) \\ &= 40.08 + 10(1) + 6/100 + 10(1 - 2) \\ &= 40.08 + 10 + 0.06 - 10 \\ &= 40.14 \end{aligned}$$

Kemudian mencari timing penyulutan.

$$\begin{aligned} t &= (250 - \text{PID}) * 100 \\ t &= (250 - 40.14) * 100 \\ t &= 20986 \end{aligned}$$

Sudut penyulutannya dapat diperoleh dengan perhitungan berikut,

$$\begin{aligned} \alpha &= \frac{180 * t}{25000} \\ \alpha &= \frac{180 * 20986}{25000} \\ \alpha &= 151.09^\circ \end{aligned}$$

Kemudian pada suhu 50° dapat dilakukan perhitungan berikut,

$$\begin{aligned} V_o &= V' + K_p (e_o) + K_i (\text{Sumerror}) + K_d (e_o - e_1) \\ &= 40.14 + 10(0) + 6/100 + 10(0 - 1) \\ &= 40.14 + 10 + 0.06 - 10 \\ &= 30.08 \end{aligned}$$

Kemudian mencari timing penyulutan.

$$\begin{aligned} t &= (250 - \text{PID}) * 100 \\ t &= (250 - 30.08) * 100 \\ t &= 21992 \end{aligned}$$

Sudut penyulutannya dapat diperoleh dengan perhitungan berikut,

$$\alpha = \frac{180 * t}{25000}$$

$$\alpha = \frac{180 * 21992}{25000}$$

$$\alpha = 158.34^\circ$$

Output dari perhitungan rumus PID yang dieksekusi dalam mikrokontroler mempengaruhi tingkat daya yang diperoleh oleh *heater*. Perubahan-perubahan ini mempunyai pengaruh pada output tegangan sehingga tegangan yang dihasilkan oleh *driver heater* berubah-ubah.

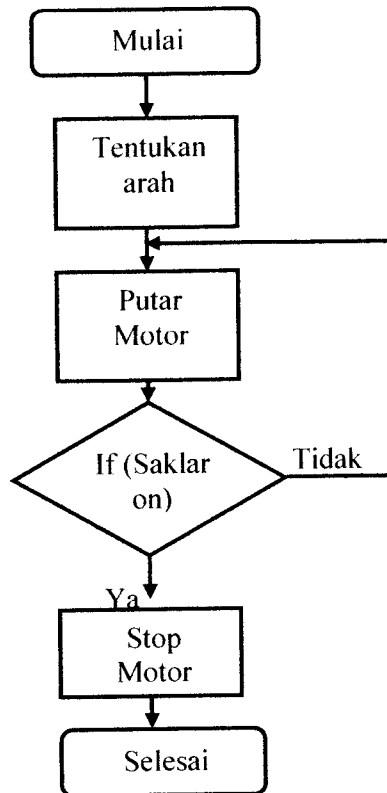
3.2.3 Sub Rutin Menaikkan dan Menurunkan Keranjang

Sub rutin ini adalah sub rutin terakhir dalam sistem elektronis ini. Sub rutin ini bekerja setelah syaratnya terpenuhi yaitu setelah suhu mencapai *set point*. Sub rutin ini bekerja mengendalikan lamanya waktu naik dan turun untuk keranjang. Waktu tempuh dari kondisi normal ke bagian bawah tungku dididih adalah 1 (satu) detik.

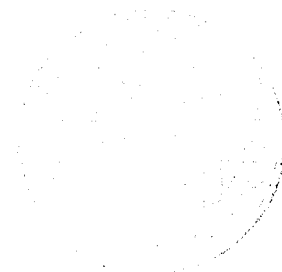
Karena waktu pemeraman hanya 2,5 menit atau 150 detik maka terjadi 7 kali siklus. Dalam Tabel 3.1 dapat dilihat bahwa terjadi 14 kali kejadian naik turun dari motor penggerak. Dalam *software* perintah-perintah untuk mengendalikan motor ini dalam sub rutin ini ditujukan untuk membangkitkan pewaktu dan menge-*set* kapan memberikan logika rendah (0) atau logika tinggi (1) pada *pin* PTB4 dan *pin* PTB5.

Tabel 3.1 Waktu dan kondisi motor

Langkah	Waktu (detik ke)	Kondisi Motor
1.	0	Turun
2.	20	Naik
3.	21	Turun
4.	41	Naik
5.	42	Turun
6.	62	Naik
7.	63	Turun
8.	83	Naik
9.	84	Turun
10.	104	Naik
11.	105	Turun
12.	125	Naik
13.	126	Turun
14.	146	Naik



Gambar 3.10 Bagan alir Sub Rutin Motor Naik Turun



BAB IV

HASIL PENGAMATAN

4.1 PENGUKURAN

Pengukuran dilakukan beberapa tahap dengan mengukur beberapa titik output, antara lain output sensor dan output pengkondisi isyarat (penguat). Cara pengukuran ditunjukkan seperti pada Gambar 2.1 untuk termokopel. Adapun hasil dari pengukuran ditunjukkan oleh tabel 4.1 berikut ini:

Tabel 4.1 Hasil pengukuran output sensor termokopel, pengkondisi isyarat dan tampilan LCD.

Suhu (°C)	Termokopel (mV)	Pengkondisi Isyarat (V)	Tampilan LCD
28	0.1	0.962	28
29	0.1	1.141	29
30	0.1	1.300	30
31	0.1	1.410	31
32	0.2	1.562	32
33	0.2	1.695	33
34	0.3	1.807	34
35	0.3	1.958	35
36	0.4	2.128	36
37	0.4	2.230	37
38	0.5	2.327	38
39	0.5	2.462	39
40	0.6	2.570	40
41	0.6	2.740	41
42	0.7	2.801	42
43	0.7	2.902	43
44	0.8	3.110	44
45	0.8	3.202	45
46	0.9	3.382	46
47	0.9	3.500	47
48	1.0	3.624	48
49	1.0	3.680	49
50	1.1	3.835	50

Lanjutan Tabel 4.1 Hasil pengukuran output sensor termokopel, pengkondisi isyarat dan tampilan LCD

Suhu ($^{\circ}\text{C}$)	Termokopel (mV)	Pengkondisi Isyarat (V)	Tampilan LCD
51	1.1	4.001	51
52	1.2	4.650	52
53	1.2	4.200	53
54	1.3	4.350	54
55	1.3	4.490	55
56	1.4	4.620	56
57	1.4	4.720	57
58	1.5	4.880	58
59	1.5	4.920	59
60	1.6	4.040	60
61	1.6	5.160	61
62	1.7	5.340	62
63	1.7	5.440	63
64	1.8	5.490	64
65	1.8	5.560	65

Keluaran dari pengkondisi isyarat akan diumpankan ke ADC untuk diubah ke isyarat digital. Isyarat digital yang dihasilkan oleh ADC kemudian di olah oleh mikrokontroler sebagai masukan untuk algoritma PID untuk mempertahankan *set point suhu* yang telah ditentukan (50°).

4.2 Pembahasan

Pengukuran suhu air dilakukan dimulai dengan suhu 28°C karena rata-rata suhu air pada suhu kamar saat pengambilan data percobaan berada pada suhu 28°C sehingga set awal pengambilan data suhu air pada suhu tersebut. Output termokopel yang berorde milivolt dikuatkan lagi untuk memperoleh nilai dengan orde Volt menggunakan sistem pengkondisi isyarat. Penguatan ini diperlukan agar

ADC dapat membaca input suhu tersebut karena jangkauan tegangan ADC adalah mulai dari 0 Volt sampai 5 Volt.

Input untuk pengkondisi isyarat adalah keluaran dari termokopel kemudian masuk ke penguat non inverting op-amp LM 358. Gambar rangkaian pengkondisi isyarat ini dapat dilihat pada Gambar 3.2 Penguat non inverting pada op-amp LM 358 bagian pertama berfungsi sebagai penyangga atau *buffer*. Isyarat keluaran termokopel di diberikan pada bagian penguat pertama LM 358 kemudian dikuatkan dan dibandingkan dengan tegangan referensi dari Vcc dengan pengaturan pada potensio. IC LM 358 berikutnya digunakan untuk menguatkan keluaran dari LM 358 yang berfungsi sebagai *buffer*. Karena ADC mempunyai lebar data 255-bit untuk melebarkan pulsa dari data pengkondisi isyarat maka pada output LM 358 dipasang rangkaian penguat berikutnya yang diberi sebuah potensiometer. Besarnya penguatan didapat dengan mengubah nilai potensiometer yang diperlukan pada saat kalibrasi output dari LM 358 untuk diumpankan ke input ADC. Penempatan potensiometer pada LM 358 bertujuan agar diperoleh penguatan yang dapat dikalibrasi dengan kebebasan maksimal. Hal ini bertujuan untuk respon sistem pada kondisi yang berbeda pengkalibrasian ulang tidak terlalu mengalami kesulitan.

Hasil output pengkondisi sinyal yang diperoleh menunjukkan hasil penguatan yang linier. Perolehan ini merupakan hasil yang diharapkan agar ADC dapat bekerja dengan baik pada mode free running,

4.2.1 Sensor termokopel

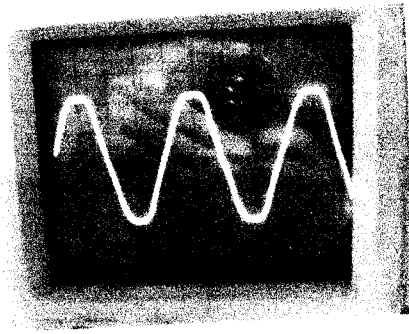
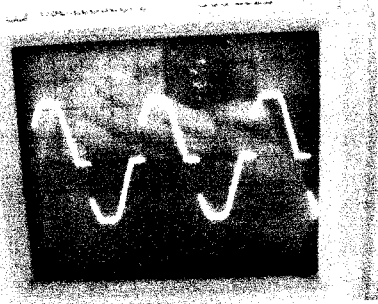
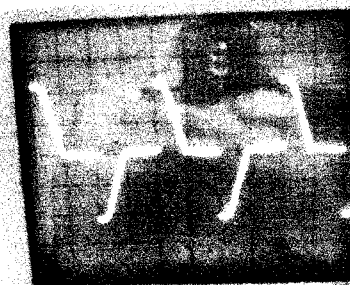
Sensor suhu termokopel berfungsi untuk mengubah suhu menjadi energi listrik. Kenaikan tiap $^{\circ}\text{C}$ suhu air yang terbaca oleh sensor suhu termokopel akan berpengaruh pada tegangan output dari sensor suhu termokopel tersebut. Semakin besar suhu air berarti tegangan output pada termokopel akan bertambah besar dan begitu juga sebaliknya apabila suhu air yang terbaca oleh sensor semakin kecil maka tegangan output pada termokopel semakin kecil pula, pengukuran dilakukan dengan cara memasukkan termometer dan ujung termokopel yang telah dihubungkan kedalam air yang dipanaskan, kemudian dua ujung lainnya yang tidak terhubung disambung dengan alat pengukur multimeter. Metode pengukuran seperti ini memiliki kekurangan serius terutama ketelitiannya rendah, karena termokopel hanya dapat menyalurkan daya yang sangat terbatas untuk menggerakkan alat pencatat atau multimeter.

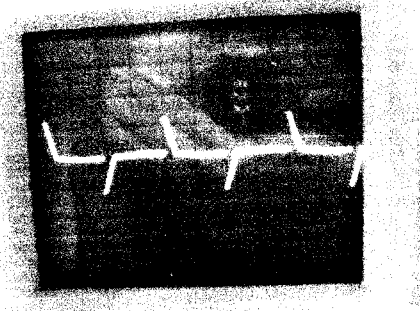
4.2.2 Mikrokontroler MC68HC908QB8

Mikrokontroler yang merupakan pusat pengendali dari keseluruhan sistem telah bekerja dengan baik. Ini terlihat dari telah bekerjanya alat sesuai yang diinginkan. *Port-port* yang *digunakan* dari mikrokontroler yang terhubung dengan rangkaian penampil Liquid Crystal Display bekerja dengan baik. Proses perhitungan algoritma PID untuk pengendalian suhu dilakukan secara perangkat lunak. Keluaran dari proses algoritma ini menentukan isyarat perintah (*command signal*) ke bagian *driver heater*, yang kemudian isyarat ini digunakan untuk mengendalikan daya ke *heater*.

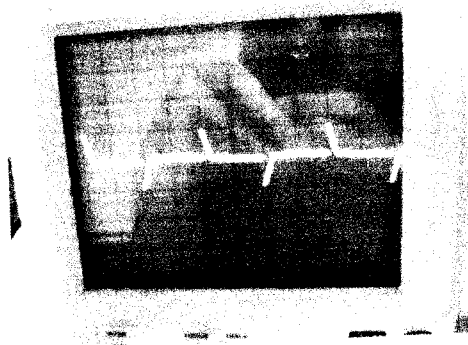
4.2.3 *Driver heater*

Driver heater yang difungsikan sebagai pengkondisi suhu yang akan dikontrol telah bekerja dengan baik, hal ini ditunjukkan pada saat pengujian alat, apabila suhu terbaca lebih besar mendekati set point (50^0) maka daya *heater* akan diatur sesuai dengan tingkat *error*. Begitu juga dengan suhu yang terbaca lebih kecil dari set point (50^0) maka *heater* akan disuplai daya lagi oleh driver sesuai tingkat *error* dan osilasi yang terjadi. *Driver* ini dikendalikan oleh mikrokontroler melalui pin PTA0 yang diberi logika 0 maka akan menyalakan led sehingga kaki basis transistor mendapat trigger dan akan menjenuhkan transistor hal ini menyebabkan kaki kolektor dan emitor dari transistor terhubung, keadaan ini akan membuat pemicu dari triac mendapat tegangan dari jala-jala PLN. Sedangkan ketika diberi logika 1 dari mikrokontroler, transistor akan memutuskan hubungan karena kaki basis tidak mendapat trigger. Perhitungan sudut picu dari triac dilakukan oleh mikrokontroler dengan memfungsikan timer yang disinkronisasi dengan frekuensi PLN. Gambar rangkaian dari *driver* untuk *heater* dapat dilihat pada Gambar 3.5. Hasil pengamatan sudut picu pada triac dengan output isyarat dengan osiloskop dapat dilihat pada Gambar 4.2 sampai dengan Gambar 4.8

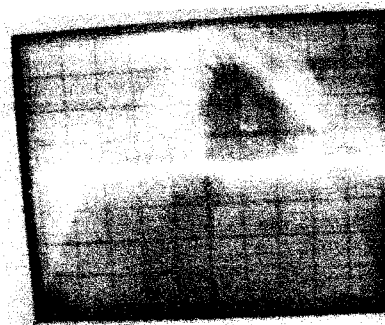
Gambar 4.2 Sudut Picu 0° Gambar 4.3 Sudut Picu 20° Gambar 4.4 Sudut Picu 40° Gambar 4.5 Sudut Picu 90°



Gambar 4.6 Sudut Picu 140°



Gambar 4.7 Sudut Picu 160°



Gambar 4.8 Sudut Picu 180°

4.2.5 Tampilan (*display*)

Media *output* kontrol ini menggunakan peraga *liquid crystal display* yang berfungsi untuk menampilkan suhu yang terbaca. Dalam perancangan ini untuk menampilkan data digunakan perintah-perintah dalam software.

4.2.6 *Driver motor*

Pengontrol untuk naik turun dari motor pada perancangan ini menggunakan relay yang digunakan sebagai rangkaian penukar fase. Dari sistem bahwa di kedua relay ujung NO (*Normally Open*) dan NC (*Normally Close*) masing-masing dihubungkan dengan motor sehingga pertukaran fasa oleh relay bekerja saat mendapatkan suplai arus dari mikrokontroler.

Pengujian dari rangkaian ini dengan cara memberikan arus pada kaki resistor $4k7 \Omega$ untuk menukar fasa dari pin PTB6 dan pin PTB7 untuk menhidupkan motor. Dari hasil pengujian respon motor bekerja dengan baik sesuai yang diharapkan..

4.3. Analisa Pengendali PID

Respon yang diamati adalah parameter **Td** (waktu tunda), **Tr** (rise time), **Tp** (waktu puncak), dan **Mp** (*overshoot* maksimum). Pada tanggapan transien ini ada beberapa parameter yang perlu diketahui yaitu

- Waktu naik (*rise time*) adalah waktu yang diperlukan sistem untuk naik dari 10% sampai 90% nilai akhir.
- Waktu penetapan (*settling time*) adalah waktu yang diperlukan sistem untuk mencapai nilai $\pm 2\%$ dari nilai keadaan tunak *steady state* atau nilai *error* terkecil.

- Waktu puncak (*peak time*) adalah waktu yang diperlukan sistem untuk mencapai puncak pertama kali.
- *Persen Overshoot* perbandingan nilai puncak maksimum dengan nilai akhir yang dinyatakan dalam bentuk

$$\% OS = \frac{C_{\max} - C_{\text{akhir}}}{C_{\text{akhir}}} \times 100\%$$

Percobaan dilakukan dengan parameter-parameter konstanta Kp, Ki, Kd dengan nilai yang berbeda-beda dengan metode *trials and errors*. Metode ini digunakan agar diperoleh perbandingan-perbandingan dengan set parameter yang berbeda. Dari parameter-parameter ini kemudian dimasukkan ke rumus PID digital. Kemudian dilakukan percobaan-percobaan pada tiap parameter.

Nilai konstanta PID dimasukkan pada deklarasi variabel pada awal program PID yang nanti ketika nilai error diperoleh kemudian dimasukkan dengan konstanta PID yang telah ditetapkan oleh program. Berikut listing program utama perhitungan PID.

....

```

{
    if(PL50HzFlag) /** Interup**/
    ADSCR = 67; /** konversi ADC**/
    while(!ADCF1ag);ADCF1ag=0;
    PV=ADR; /** nilai konversi=Pv**/
    PV=TabelSuhu[PV]; /** Look-table**/

    e1=e0; /** nilai error**/
    e0=50-PV;
    if(!e0)MotorFlag=1;/** bila PV=SP ke
    sumerror+=e0; /** subrutin motor **/

    PID+=KP*e0; /** rumus PID **/
    PID+=Sumerror/Ki;
    PID+=Kd*(e0-e1);

    alpha=(250-PID)*100; /**hitung timer**/

```

```

puti_lcd(0x08,PV);    /**tampilkan di LCD**/
puti_lcd(0x48,(250-PID)*18/25);

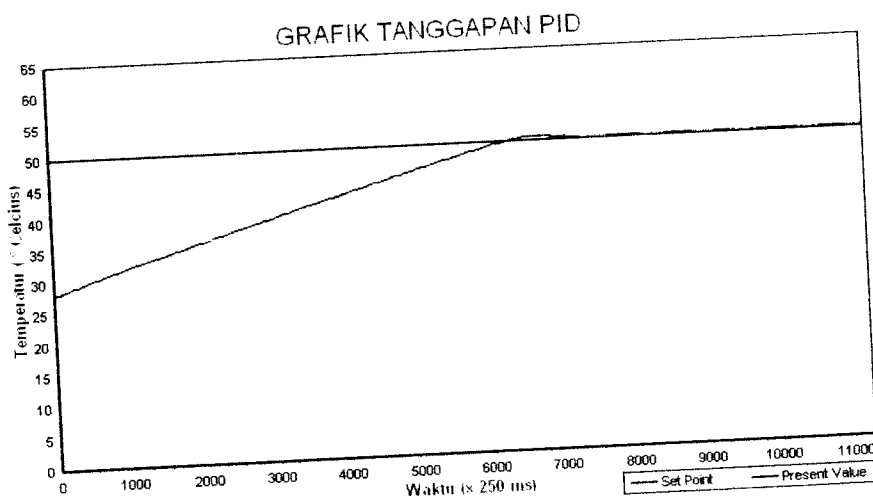
PL50HzFlag=0;

}
....

```

Pengambilan data dilakukan dengan menggunakan alat antar muka yang dipinjam dari klub robot yang menggunakan komunikasi antar muka berbasis mikrokontroler AVR dengan antarmuka serial kemudian data yang diperoleh diolah dengan program Delphi. Berikut hasil perolehan hasil percobaan dengan nilai konstanta PID yang bervariasi.

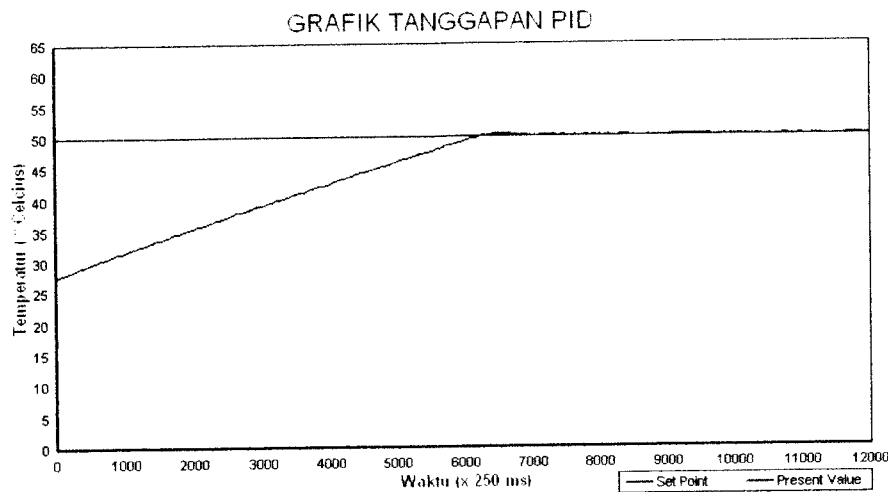
- Percobaan dengan nilai $K_p=25$, $K_i=50$, $K_d=100$



Gambar 4.10 Respon PID pada nilai $K_p=25$, $K_i=50$, $K_d=100$

Dari respon sistem diatas diperoleh nilai *rise time* = 10 menit 48 detik, *peak time* = 13 menit 19 detik dan *settling time* = 14 menit 47 detik.

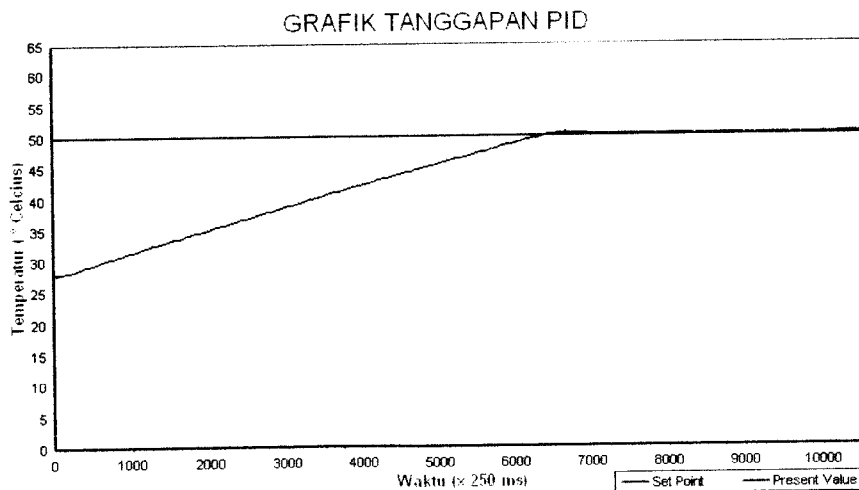
- Percobaan dengan nilai $K_p=50$, $K_i=50$, $K_d=100$



Gambar 4.11 Respon PID pada nilai $K_p=50$, $K_i=50$, $K_d=100$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 12 detik, *peak time* = 12 menit 52 detik dan *settling time* = 15 menit 17 detik.

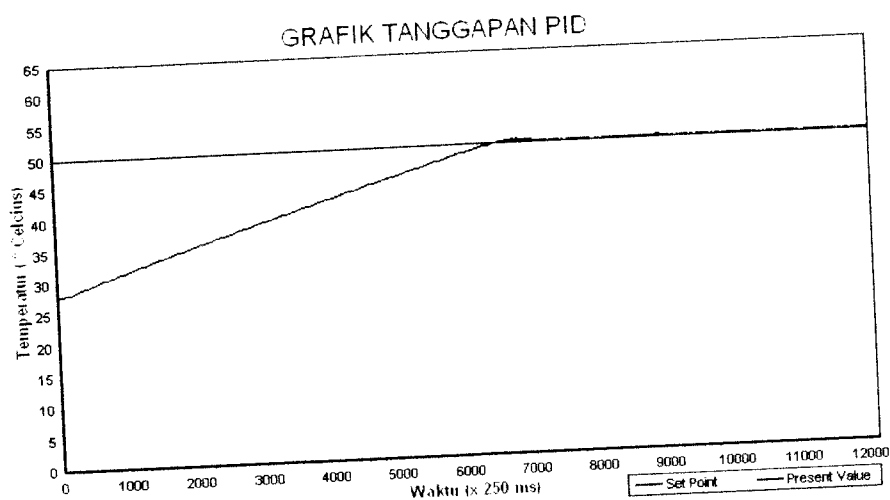
- Percobaan dengan nilai $K_p=75$, $K_i=50$, $K_d=100$



Gambar 4.12 Respon PID pada nilai $K_p=75$, $K_i=50$, $K_d=100$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 22 detik, *peak time* = 13 menit 28 detik dan *settling time* = 14 menit 6 detik

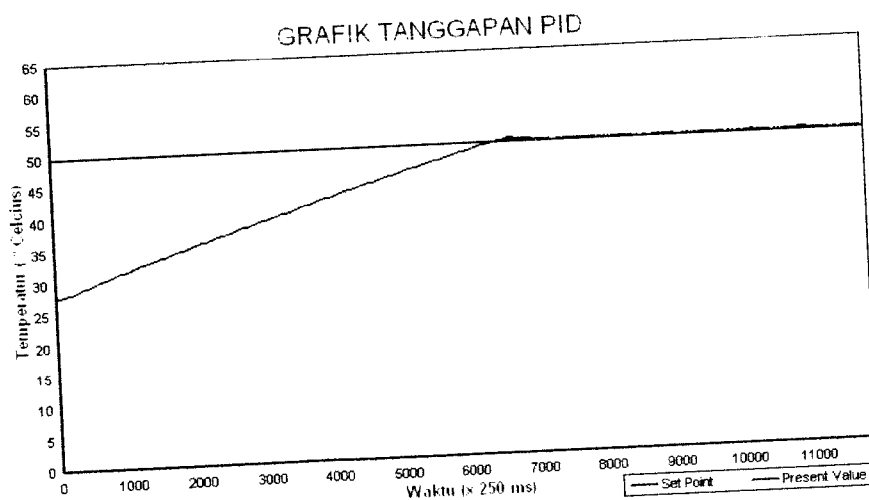
- Percobaan dengan nilai $K_p=50$, $K_i=100$, $K_d=25$



Gambar 4.11 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=25$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 40 detik, *peak time* = 13 menit 36 detik dan *settling time* = 14 menit 25 detik.

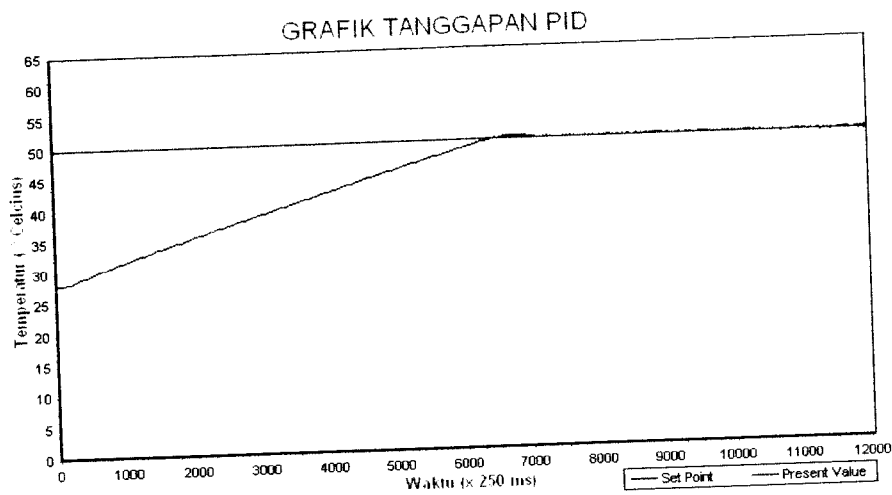
- Percobaan dengan nilai $K_p=50$, $K_i=100$, $K_d=50$



Gambar 4.13 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=50$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 25 detik, *peak time* = 13 menit 21 detik dan *settling time* = 15 menit 02 detik.

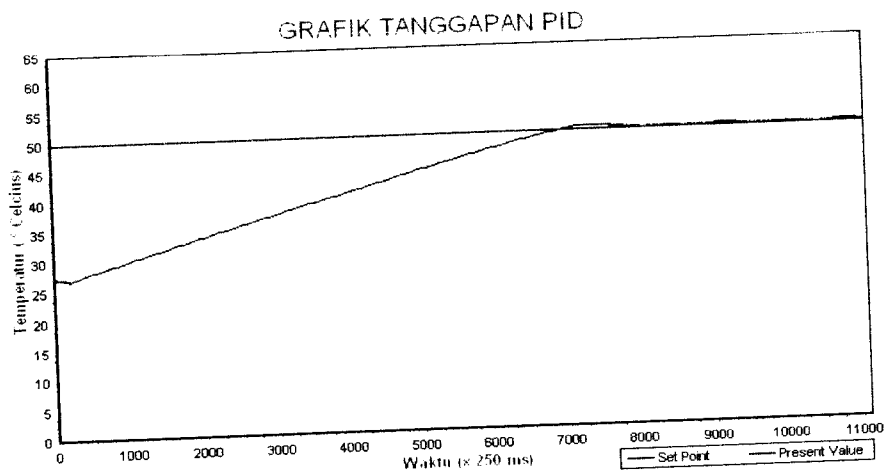
- Percobaan dengan nilai $K_p=50$, $K_i=100$, $K_d=75$



Gambar 4.14 Respon PID pada nilai $K_p=50$, $K_i=100$, $K_d=75$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 31 detik, *peak time* = 13 menit 38 detik dan *settling time* = 14 menit 24 detik.

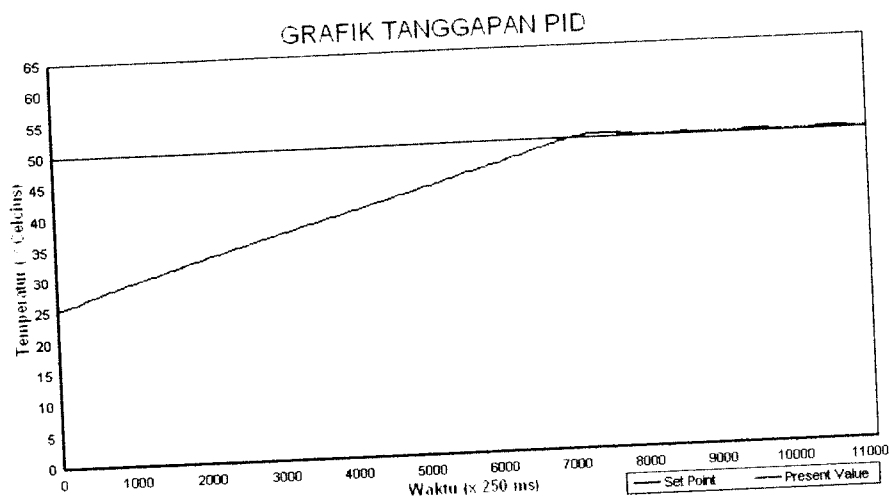
- Percobaan dengan nilai $K_p=100$, $K_i=50$, $K_d=25$



Gambar 4.15 Respon PID pada nilai $K_p=100$, $K_i=50$, $K_d=25$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 41 detik, *peak time* = 13 menit 36 detik dan *settling time* = 14 menit 31 detik.

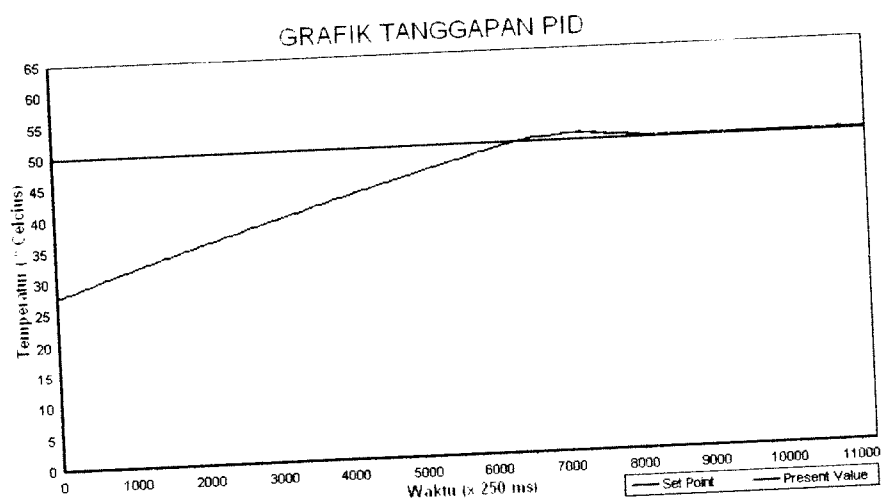
- Percobaan dengan nilai $K_p=100$, $K_i=50$, $K_d=50$



Gambar 4.16 Respon PID pada nilai $K_p=100$, $K_i=50$, $K_d=50$

Dari respon sistem diatas diperoleh nilai *rise time* = 12 menit 05 detik, *peak time* = 13 menit 36 detik dan *settling time* = 14 menit 25 detik.

- Percobaan dengan nilai $K_p=100$, $K_i=50$, $K_d=75$



Gambar 4.17 Respon PID pada nilai $K_p=100$, $K_i=50$, $K_d=75$

Dari respon sistem diatas diperoleh nilai *rise time* = 11 menit 24 detik, *peak time* = 14 menit 5 detik dan *settling time* = 16 menit 32 detik.

Respon kinerja PID sesuai dengan yang diharapkan sehingga suhu tetap terjaga pada *set point* dengan nilai error keadaan tunak yang kecil. Motor bergerak ketika suhu mencapai 50°C dan melakukan perendaman selama 20 detik kemudian naik lagi dengan sela 1 detik turun lagi merendam panili selama 20 detik lagi dengan 7 kali siklus perendaman. Pada kondisi *steady state* kinerja PID menunjukkan pengaturan daya yang optimal terlihat dengan penunjukan intensitas cahaya lampu sebagai monitoring daya yang dikirim ke heater. Sistem *phase control* akan lebih baik lagi jika menggunakan resolusi ADC yang lebih tinggi misalnya ADC dengan resolusi 10-bit keatas sehingga aksi integral dan derivatif pergerakannya akan lebih halus.

BAB V

PENUTUP

Berdasarkan dari proses perancangan, pembuatan, pengamatan dan hasil uji sistem yang dibuat, maka dapat diambil beberapa kesimpulan dan saran untuk kemajuan, perbaikan dan pengembangan dari aplikasi sistem.

5.1 Kesimpulan

Dari proses perancangan, implementasi, analisis kinerja sistem elektronis dan pengujian sistem dapat diambil beberapa kesimpulan sebagai berikut:

1. Kendali PID kinerjanya cukup bagus dalam sistem kendali suhu atau plant yang bersifat linier.
2. Untuk mendapatkan hasil pengendalian PID yang optimal pada suhu parameter-parameter konstanta PID yang tepat adalah $K_p=25$, $K_i=50$, dan $K_d=100$ karena memiliki *rise time* yang singkat dan dengan *settling time* yang lebih cepat.
3. Keluaran dari kendali PID adalah penjumlahan dari 3 bagian yaitu proposional dengan error, integral dengan jumlah error dan derivatif dengan selisih error.

5.2 Saran

Untuk pengembangan pengendalian dengan PID selanjutnya, penulis menyarankan:

1. Dalam merancang hardware, alat ukur, komponen dan perangkat lain yang digunakan harus dipilih yang mempunyai kualitas bagus untuk memperkecil error.
2. Sistem pengendalian ini dapat dikembangkan dengan ditambahkan kendali fuzzy untuk memperbaiki output dari kendali PID.
3. Peningkatan resolusi dari ADC akan meningkatkan kinerja PID dalam mengendalikan pengaturan daya pada *heater*.

DAFTAR PUSTAKA

- Budioko, T., 2005, *Belajar Dengan Mudah Dan Cepat Pemrograman Bahasa C Dengan SDCC Pada Mikrokontroler AT89X051/AT89C51* Gava Media, Yogyakarta.
- Cooper, W. D., 1985, *Instrumentasi Elektronik dan Teknik Pengukuran edisi ke- 2*, Erlangga , Jakarta.
- Ludhantono, A. A., 2005, *Alat Pemeram Panili Berbasis Mikrokontroler AT89C51*, Skripsi, Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia, Yogyakarta.
- Malvino, A. P. (terjemahan), 2004, *Prinsip-prinsip Elektronika*, Salemba Teknika, Jakarta.
- Rismunandar; Sukma, E. S., 2002, *Bertanam Panili Edisi Revisi*, Penebar Swadaya, Jakarta.
- Setiawan, I., 2006, *Perancangan Sistem Kendali Motor DC Dengan PID Berbasis Mikrokontroler AT89S52*, Skripsi, Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia, Yogyakarta.
- Simpson, C. D., 1996, *Industrial Electronic*, Prentice-Hall, USA.
- www.freescale.com


```

#include "Cpu.h"
#include "Events.h"
#include "LcdData.h"
#include "LcdRs.h"
#include "LcdE.h"
#include "MotorArahPort.h"
#include "MotorPowerPort.h"
#include "HeaterPort.h"
#include "PL50Hz.h"
#include "SulutAlpha.h"
#include "AdcPort.h"
#include "SaklarAtas.h"
#include "SaklarBawah.h"
#include "Reset.h"

const byte TabelSuhu[256]={
  {23, 23, 23, 23, 24, 24, 24, 24, 24, 24, 25, 25, 25, 25, 25, 25, 26, 26, 26,
  26, 26, 26, 27, 27, 27, 27, 27, 27, 28, 28, 28, 28, 28, 28, 28, 29, 29, 29, 29,
  29, 29, 30, 30, 30, 30, 30, 31, 31, 31, 31, 31, 31, 31, 32, 32, 32, 32, 32, 32,
  33, 33, 33, 33, 33, 34, 34, 34, 34, 34, 34, 34, 35, 35, 35, 35, 35, 35, 36, 36,
  36, 36, 36, 36, 37, 37, 37, 37, 37, 37, 37, 38, 38, 38, 38, 38, 38, 39, 39, 39,
  39, 39, 40, 40, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 41, 42, 42, 42, 42, 42,
  43, 43, 43, 43, 43, 44, 44, 44, 44, 44, 44, 45, 45, 45, 45, 45, 45, 46, 46, 46,
  46, 46, 46, 47, 47, 47, 47, 47, 47, 48, 48, 48, 48, 48, 48, 49, 49, 49, 49, 49,
  49, 50, 50, 50, 50, 50, 50, 51, 51, 51, 51, 51, 51, 52, 52, 52, 52, 52, 52,
  53, 53, 53, 53, 53, 54, 54, 54, 54, 54, 54, 55, 55, 55, 55, 55, 55, 56, 56, 56,
  56, 56, 56, 57, 57, 57, 57, 57, 58, 58, 58, 58, 58, 58, 59, 59, 59, 59, 59,
  60, 60, 60, 60, 60, 60, 61, 61, 61, 61, 61, 61, 62, 62, 62, 62, 62, 62, 63,
  63, 63, 63, 63, 63, 64, 64, 64, 64, 64, 64, 64, 65, 65, 65, 65, 65, 65}

void delay(unsigned int i)
{
  int j;
  while(i){for(j=0;j<100;j++){i--};}
}

#define Stop 0
#define Bawah 1
#define Atas 2

void MotorPutar(byte val)
{
  if(val==Atas)
  {
    MotorPowerPort_PutVal(1);
    MotorArahPort_PutVal(1);
  }
  else if(val==Bawah)
  {
    MotorPowerPort_PutVal(1);
    MotorArahPort_PutVal(0);
  }
  else
  {
    MotorPowerPort_PutVal(0);
  }
}

}

bool AdcFlag=0;
bool IntFlag=0;
bool MotorFlag=0;
word Alpha=0;

void main(void)
{
  int i=0,n=0, m=0;
  byte j=0;
  signed long e0=0;
  signed long e1=0;
  signed long SumError=0;
  signed long PID=0;
}

```

```

unsigned char PV=0;
unsigned char PV1=0;
long AdcData=0;
word Kp=10;
word Ki=15;
word Kd=10;

init_lcd();
puts_lcd(0x00,"Oxza Corporation");
puts_lcd(0x40,"Electrical E-UII");
delay(2000);
puts_lcd(0x00,"Loading.....");
puts_lcd(0x40," MC68HC908QB8 ");

awal:
Alpha=25000;
n=0;
m=0;
delay(2000);

MotorPutar(Bawah);
while (SaklarBawah_GetVal());
MotorPutar(Atas);
while (SaklarAtas_GetVal());
MotorPutar(Stop);

puts_lcd(0x00,"Suhu =      C");
pos_lcd(0x0e);putc_lcd(0xdf);
puts_lcd(0x40,"      ");
pos_lcd(0x4e);putc_lcd(0xdf);

for(;;)
{
  AdcData=0;
  for(i=0;i<1000;i++){
    AdcFlag=0;
    ADSCR = 67;
    while(!AdcFlag);
    AdcData+=ADR;
  }

  PV=(AdcData/1000)&0xff;
  PV1=TabelSuhu[PV];

  e1=e0;
  e0=163-PV;
  if (!e0 && n==0)
  {
    n=1;
    j=101;
  }
  SumError=SumError+e0;
  if(SumError<-250)SumError=-250;
  else if(SumError>250)SumError=250;

  PID+= Kp*e0
        +SumError/Ki
        +Kd*(e0-e1);

  if(PID<1)PID=0;
  else if(PID>250)PID=250;

  Alpha=((250-PID)*100)&0xffff;

  puti_lcd(0x08,PV1);
  puti_lcd(0x48,(250-PID)*18/25);

  if(n==1 && j>100)
  {
    MotorPutar(Bawah);
    n=2;
  }
}

```

```

}
else if(n==2 && SaklarBawah_GetVal()==0)
{
    MotorPutar(Stop);
    n=3;
    j=0;
}
else if(n==3 && j>100)
{
    MotorPutar(Atas);
    n=4;
}
else if(n==4 && SaklarAtas_GetVal()==0)
{
    MotorPutar(Stop);
    n=1;
    j=101;
    m++;
}
if(m==7)n=5;

if(n==1 || n==3)j++;
if(!Reset_GetVal()) break;
}

puts_lcd(0x00,"Reset.....");
puts_lcd(0x40,"");
puts_lcd(0x00,"Loading lagi euy");
puts_lcd(0x40,"");
delay(2000);
goto awal;

}

extern word Alpha;
extern bool IntFlag;
char nInt=0;
word SulutTimer;
void PL50Hz_OnInterrupt(void)
{
    nInt++;
    if(nInt==25)
    {
        nInt=0;
        IntFlag=1;
    }
    SulutTimer=Alpha;
    if(SulutTimer<20)SulutTimer=20;
    TCH0H = TMODH = (*(TWREG*)&SulutTimer).b.high;
    TCH0L = TMDL = (*(TWREG*)&SulutTimer).b.low;
    TSC_TSTOP = 0;
}

void ADC_TEMP_OnEnd(void)
{
    ADCFlag=1;
}

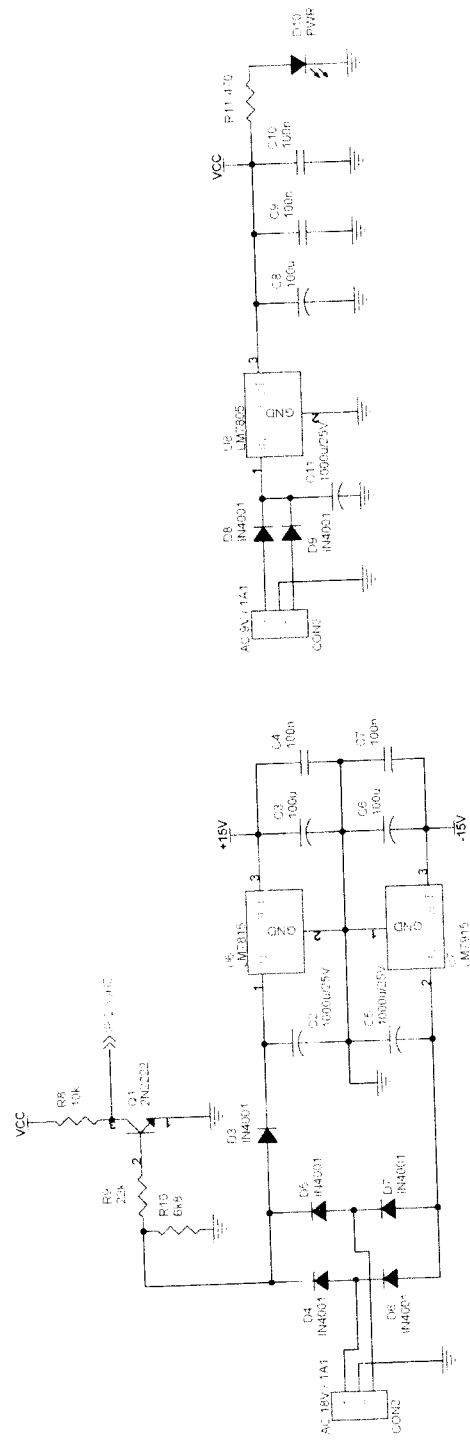
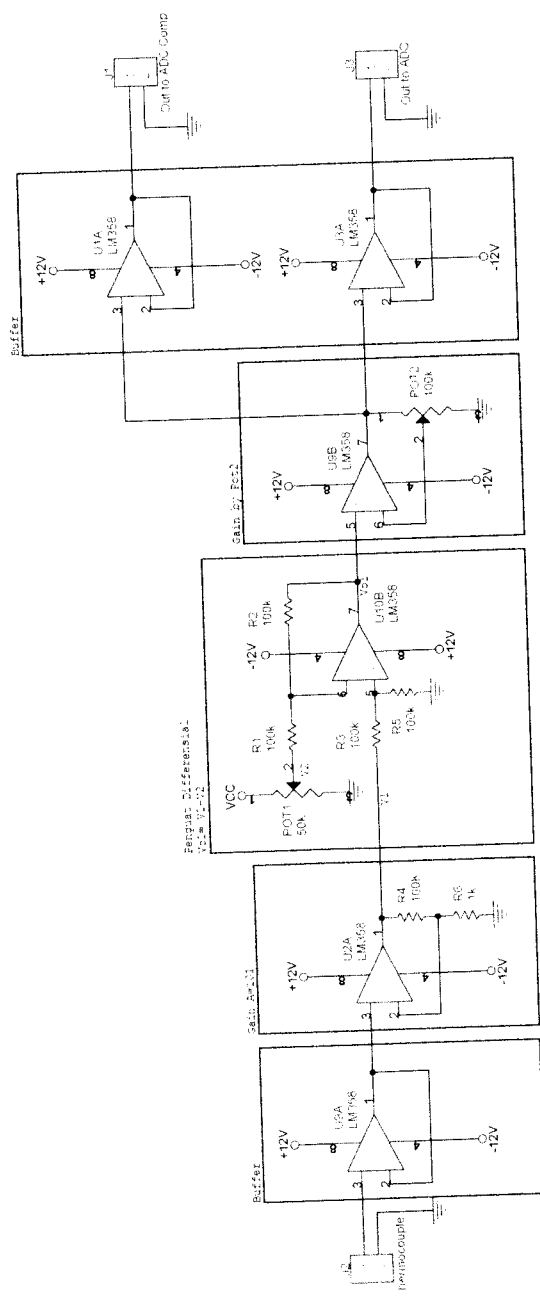
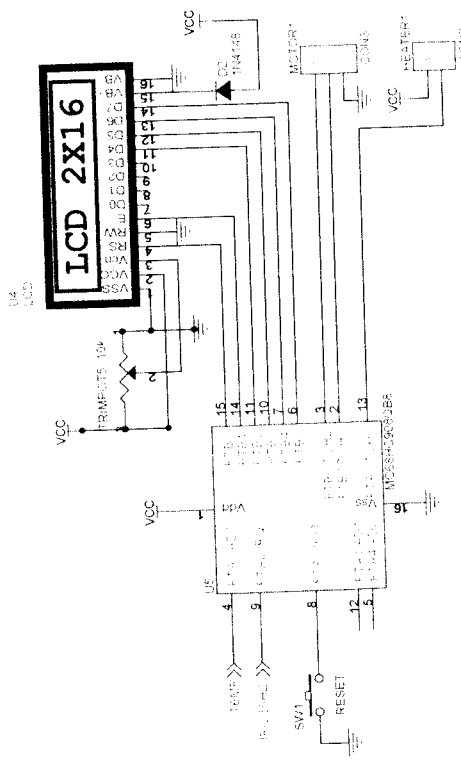
static SudutPenyulutan;
void Tim0Sulut_OnInterrupt(void)
{
    PortHeater_PutVal(0);
    PortHeater_PutVal(1);

    TSC_TSTOP = 1;

    TCH0H = TMODH = (*(TWREG*)&SudutPenyulutan).b.high;
    TCH0H = TMODH = (*(TWREG*)&SudutPenyulutan).b.low;
}

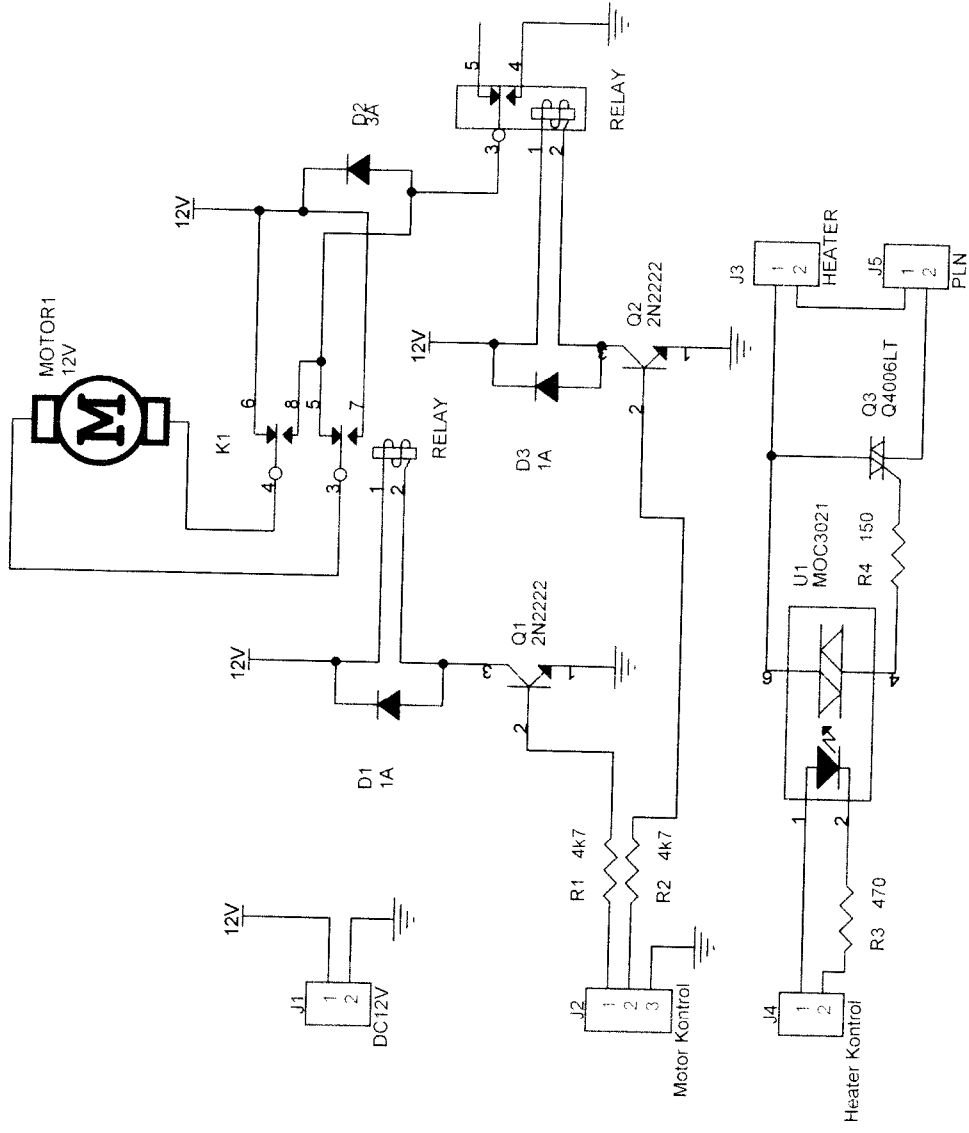
```

ALAT PEMERAM PANILI



Title		OXZA Corporation	
Size		Document Number	
Rev		<Doc>	
Date	Monday, October 09, 2006	Sheet	1 of 1

DRIVER MOTOR & HEATER



Title		OXZA Corp.	
Size	A	Document Number	<Doc>
Date:	Thursday, April 19, 2007	Sheet	1 of 1
		Rev	