

BAB III

METODE PENELITIAN

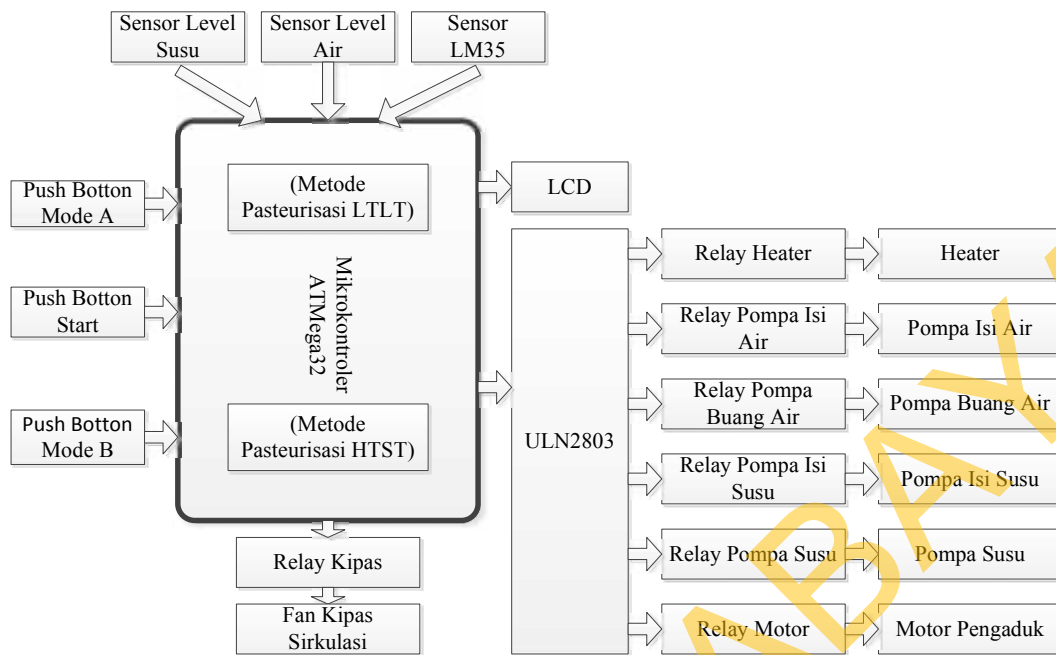
Pada bab ini akan dibahas mengenai perencanaan dan perangkaian perangkat keras (*hardware*) maupun pembuatan perangkat lunak (*software*). Dimana perangkat lunak tersebut akan diintegrasikan ke perangkat keras agar dapat bekerja untuk menjalankan perangkat keras. Perancangan perangkat keras dan perangkat lunak dilakukan dengan metode penelitian yang didasarkan pada studi keputusan berupa data – data literature dari masing – masing komponen, informasi dari internet dan konsep – konsep teoritis dari buku – buku penunjang.

Perancangan sebuah perangkat keras diperlukan sebelum proses perangkaian perangkat keras. Perancangan ini berguna supaya pengerjaan tahapan selanjutnya dapat dilakukan dengan lancar. Proses tahapan ini meliputi tahap perangkat keras, perangkat lunak dan penggabungan perangkat keras dan lunak.

Berikut perancangan perangkat keras, arsitektur dan perangkat lunak sistem:

3.1. Perancangan Perangkat Keras

Perancangan blok diagram perangkat keras pada sistem Rancang Bangun Prototipe Pasteurisasi Susu, secara garis besar digambarkan pada Gambar 3.1.



Gambar 3.1. Blok diagram keseluruhan system

Dari Gambar 3.1. dapat dilihat bahwa sistem secara keseluruhan terbagi menjadi beberapa bagian yaitu sebagai *Input* - an tombol *push button* yaitu “Start”, mode “A” dan Mode “B”, sensor – sensor yaitu sensor level air untuk mendeteksi air dipenampungan pemanasan air, sensor ketinggian susu di penampungan pemanasan susu dan sensor LM35 untuk temperatur, sedangkan untuk *Output* yaitu LCD, relay untuk *Heater*, pompa untuk mengisi air, isi susu dan pembuangan air, pomp untuk menyedot susu, motor DC sebagai motor pengaduk susu dan sebagai otak dari *Input* dan *Output* tersebut digunakan mikrokontroler ATmega32 yang berfungsi untuk mengatur jalannya sistem *Input* dan *Output*, dengan tujuan agar dapat menstabilkan proses pasteurisasi dan tidak melebihi batas maksimal ataupun minimal dari temperatur pemanasan yang ditentukan setiap metode pasteurisasi yang digunakan.

Pada *Minimum system* ATmega32 terdapat beberapa rangkaian pendukung yaitu rangkaian *reset* dan rangkaian *oscillator*. Pada rangkaian *reset* menggunakan

manual reset. Pada rangkaian *oscillator* menggunakan komponen kristal 8000000MHz sebagai clk (*clock*).

3.1.1. Rangkaian *Microcontroller*

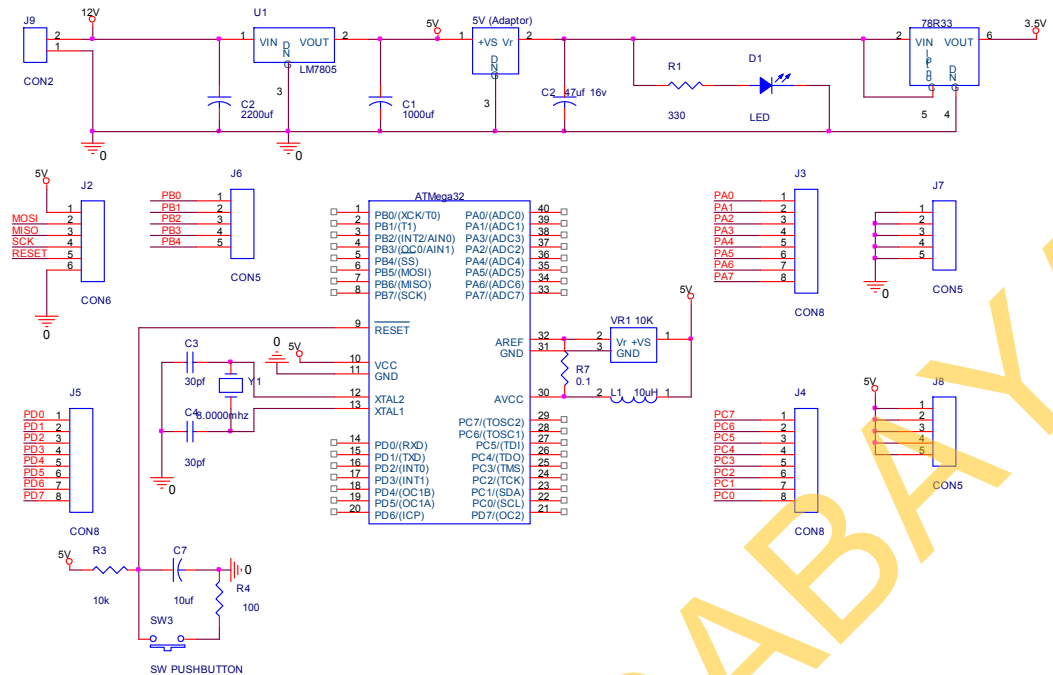
Pada proyek akhir ini dibuat piranti pengendali menggunakan *Microcontroller* keluaran AVR, yaitu ATmega32. Untuk mengaktifkan atau menjalankan *microcontroller* ini diperlukan rangkaian *minimum system*. Rangkaian *minimum system* tersebut terdiri rangkaian reset, rangkaian *oscillator*, rangkaian *power supply* dan rangkaian sistem *microcontroller*.

A. Rangkaian *minimum system microcontroller*

Untuk menjalankan *microcontroller* dibutuhkan sebuah rangkaian minimum agar *microcontroller* tersebut dapat bekerja dengan baik. Rangkaian *minimum microcontroller* terdiri dari rangkaian reset dan rangkaian *oscillator*.

Reset pada *microcontroller* ATmega32 terjadi dengan adanya logika *high* “1” selama dua cycle pada kaki RST pada *microcontroller* ATmega32. Setelah kondisi pin RST kembali *low*, maka *microcontroller* akan menjalankan program dari alamat 0000H. Dalam hal ini reset yang digunakan adalah *manual reset*.

Pada pin VCC diberi masukan tegangan operasi berkisar antara 4,5 volt sampai dengan 5,5 volt. Pin RST mendapat *Input* dari *manual reset*. Rangkaian *minimum system* dapat dilihat pada Gambar 3.2 berikut:

Gambar 3.2. Rangkaian *Minimum System*

Pin XTAL1 dan XTAL2 merupakan pin *oscillator* bagi *microcontroller* ATmega32. Pin XTAL1 berfungsi sebagai *Input* dan XTAL2 sebagai *Output oscillator*. *Oscillator* ini bisa berasal dari kristal atau dari keramik *resonator*. Seperti yang sudah terlihat di atas, pin XTAL1 dan XTAL2 dihubungkan dengan komponen XTAL sebesar 8000000 MHz. Pada proyek akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari kristal. Nilai C1 dan C2 masing-masing 33 pF.

$$\text{Clock Cycle} = \frac{12}{\text{Frekuensi Kristal}} = \frac{12}{8.000000 \times 10^6} = 1,5^{-6}$$

$$\text{Clock Cycle} = 1,5 \mu s$$

B. Perancangan *Interface I/O*

Rangkaian I/O dari *microcontroller* mempunyai kontrol direksi yang tiap bitnya dapat dikonfigurasi secara individual, maka dalam perancangan I/O yang digunakan ada yang berupa operasi port ada pula yang dikonfigurasi tiap bit I/O. Berikut ini akan diberikan konfigurasi dari I/O *microcontroller* tiap bit yang ada pada masing-masing port yang terdapat pada *microcontroller* :

1. Port A

Port A digunakan untuk *Input* Sensor Temperatur LM35.

2. Port B

Port B digunakan untuk *Input* Sensor Level Air, Kipas dan *Push Button*.

3. Port C

Port C digunakan untuk *Output* LCD.

4. Port D

Port D digunakan untuk *Output* Relay dan *Push Button*.

Untuk perancangan *interface Input* dan *Output* pada *microcontroller* yang lebih mendetil dapat dilihat pada tabel 3.1 dibawah :

Tabel 3.1. Perancangan *interface Input/Output*

Port	Alokasi Port pada Hardware
PortA.0	<i>Input</i> LM35
PortB.1	<i>Input</i> Level Air Batas Atas A
Port B.2	<i>Input</i> Level Air Batas Bawah A
Port B.3	<i>Input</i> Level Air Batas Atas B
Port B.4	<i>Push Button</i> Mode B
PortB.5	<i>Input</i> Level Air Batas Bawah B
Port B.6	Relay Kipas
PortC.0	-
PortC.1	-
PortC.2	D7 LCD
PortC.3	D6 LCD
PortC.4	D5 LCD
PortC.5	D4 LCD

PortC.6	EN LCD
PortC.7	RS LCD
PortD.0	<i>Push Button</i> Mode Start
PortD.1	<i>Push Button</i> Mode A
PortD.2	Pin 8 ULN2803 (relay pomp susu)
PortD.3	Pin 7 ULN2803 (relay motor)
PortD.4	Pin 6 ULN2803 (relay isi air)
PortD.5	Pin 5 ULN2803 (relay buang air)
PortD.6	Pin 4 ULN2803 (relay isi susu)
PortD.7	Pin 3 ULN2803 (relay <i>Heater</i>)

C. Program Downloader

Untuk melakukan proses *downloading* program dalam format *.HEX* dari komputer ke dalam *memory* program *internal microcontroller*, penulis menggunakan kabel *downloader* dengan *interface* USB. USB Downloader adalah sebuah alat sederhana yang bisa penulis gunakan untuk berkomunikasi pertukaran data, dari komputer ke mikrokontroler ataupun sebaliknya dengan menggunakan jalur data port USB. Dimana Gambar dan Skematiknya sebagai berikut :

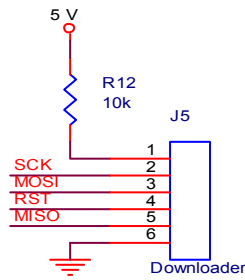


Gambar 3.3 Gambar *USB Downloader*

Gambar 3.4 : Skematik USB *Downloader*
(Sumber : <http://www.fischl.de/usbaspl/>)

USB Downloader merupakan *downloader* yang komponen utamanya adalah mikrokontroler atmega8 yang digunakan sebagai protokol komunikasi serial berdasarkan program (*firmware*) yang telah kita tanamkan terlebih dahulu ke chip atmega8. Firmware disini didefinisikan sebagai suatu program atau perangkat lunak (biasanya berupa file dengan ekstensi *.hex) yang ditanamkan pada memory non volatile (biasanya dalam mikrokontroler atmega8 pada memori flash atau EEPROM) dan tidak dapat diubah (permanen) kecuali ada perubahan fisik dari hardware. Firmware tersebut digunakan sebagai bentuk protocol atau tugas khusus yang akan dijalankan dalam hal ini pada USBASP digunakan sebagai protokol komunikasi serial SPI (serial programming interface) antara PC dan mikrokontroler target.

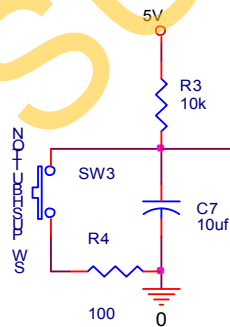
Sedangkan untuk konektor *downloader* pada *microcontroller* ATmega32 dapat dilihat pada Gambar 3.5 berikut:



Gambar 3.5. Konektor *Downloader* pada *Microcontroller* ATmega32

D. Rangkaian *Reset*

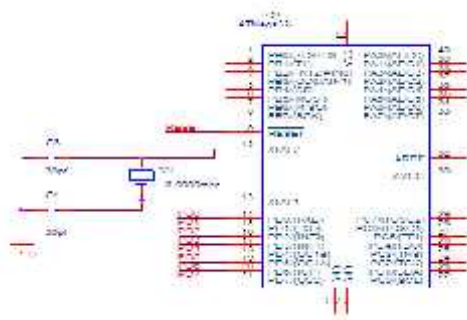
Reset pada *microcontroller* ATmega32 terjadi dengan adanya logika *high* “1” selama dua *cycle* pada kaki RST pada *microcontroller* ATmega32. Setelah kondisi pin RST kembali *low*, maka *microcontroller* akan menjalankan program dari alamat 0000H. Dalam hal ini reset yang digunakan adalah *manual reset*. Rangkaian *reset* dapat dilihat pada Gambar 3.6 berikut :



Gambar 3.6. Rangkaian Reset

E. Rangkaian *Oscillator*

Pin XTAL1 dan XTAL2 merupakan pin *oscillator* bagi *microcontroller* ATmega32. Pin XTAL1 berfungsi sebagai *Input* dan XTAL2 sebagai *Output oscillator*. *Oscillator* ini bisa berasal dari kristal atau dari keramik *resonator*. Rangkaian *oscillator* dapat dilihat pada Gambar 3.7 berikut:



Gambar 3.7. Rangkaian *Oscillator*

Pada proyek akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari kristal. Nilai C1 dan C2 masing-masing 33 pF.

3.1.2. Rangkaian Pendukung.

Pada proyek ini menggunakan beberapa komponen pendukung untuk membantu kinerja *Input* dan *Output*. Komponen pendukung ini memiliki rangkaian dan penjelasan sebagai berikut :

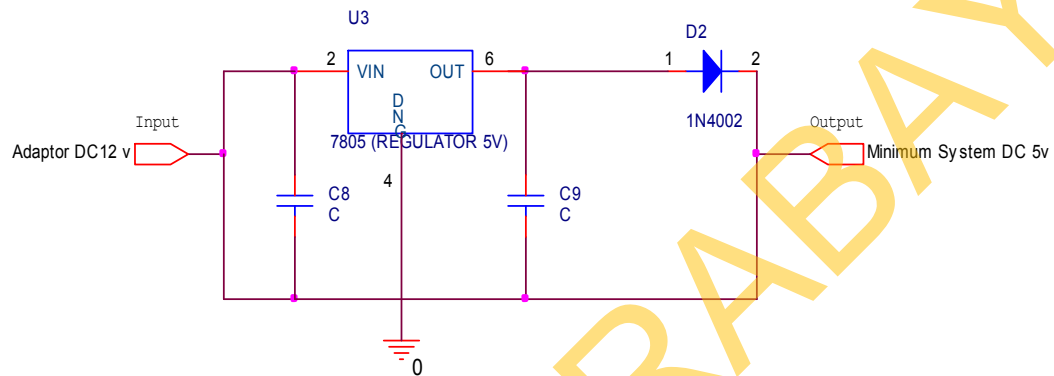
A. Regulator

Pada bagian regulator untuk proyek ini menggunakan 2 komponen regulator yaitu regulator untuk keluaran 5v dan 3,3v dimana dalam proyek ini menggunakan IC 7805 dan IC 78R33 dimana penggunaannya untuk menstabilkan tegangan dengan keluaran 5v dengan regulator 7805 dan tegangan keluaran 3,3v dengan regulator 78R33.

1. Rangkaian Regulator *Output* 5v IC7805A

Rangkaian ini berfungsi untuk Catu daya. Catu daya merupakan pendukung utama bekerjanya suatu sistem. Catu daya yang biasa digunakan untuk menyuplai tegangan sebesar 5 Volt adalah catu daya DC yang

memiliki keluaran +5 volt. Catu daya ini digunakan untuk mensuplay tegangan sebesar 5 volt. IC 7805 (IC regulator) digunakan untuk menstabilkan tegangan searah. Kapasitor digunakan untuk mengurangi tegangan kejut saat pertama kali saklar catu daya dihidupkan. Sehingga keluaran IC regulator 7805 stabil sebesar 5 volt DC. Rangkaian regulator terlihat pada gambar dibawah ini.



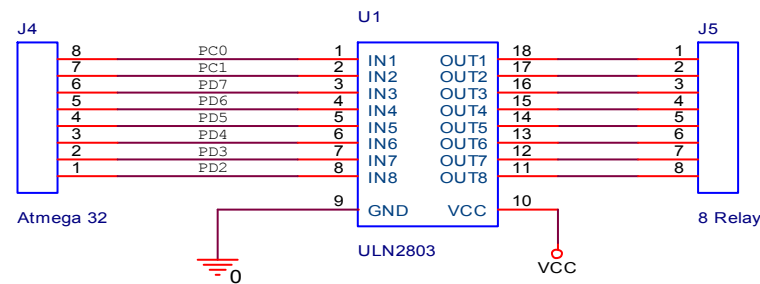
Gambar 3.8. Rangkaian Regulator 5v

2. Rangkaian Regulator *Output* 3,3v IC78R33

Rangkaian ini memiliki fungsi sama dengan rangkaian diatas, namun untuk fungsinya berbeda, rangkaian regulator ini berfungsi untuk mensuplay tegangan 3.3v untuk penggunaan alat pomp air elektrik.....(Gambar dapat dilihat pada lampiran Skematik)

B. Rangkaian ULN2803

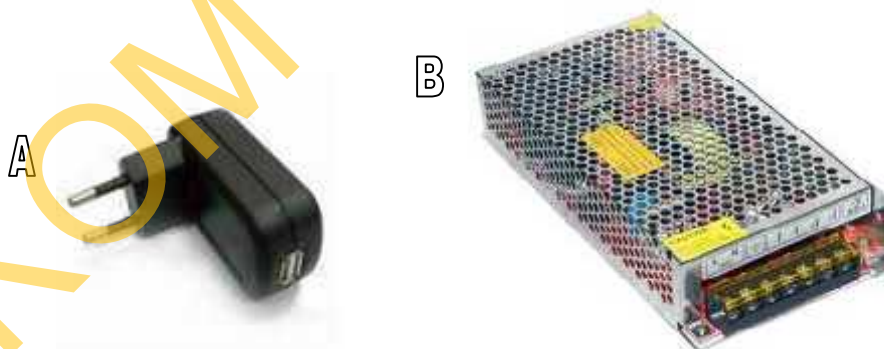
Rangkaian ULN2803 disini memiliki fungsi untuk driver dari relay dimana pada proyek ini menggunakan 8 buah relay dengan memfokuskan 6 buah relay utama untuk fungsi penting dalam proses pasteurisasi diantaranya relay untuk *Heater*, relay pompa air elektrik 3.3v untuk susu, relay pomp susu, relay pomp pengisi air, relay pomp air pembuangan dan relay motor.



Gambar 3.9. Rangkaian ULN2803

C. Power Supply 12V 10A dan Adaptor 5V

Pada perancangan alat ini dibutuhkan Power Supply dengan Ampere yang cukup besar dikarenakan lonjakan tegangan yang tidak stabil pada saat beberapa alat yang bekerja secara bersamaan. Digunakannya power supply dan adaptor bertujuan agar supplay tegangan 5V tidak mengganggu tegangan yang 12 V, tetapi pada dasarnya tidak seperti itu, akan tetapi dalam kasus permasalahan yang dihadapi memerlukan penggunaan adaptor dan power supply untuk memaksimalkan kinerja dari alat.



Gambar 3.10. (A) adaptor 5v. (B) power supply 12v

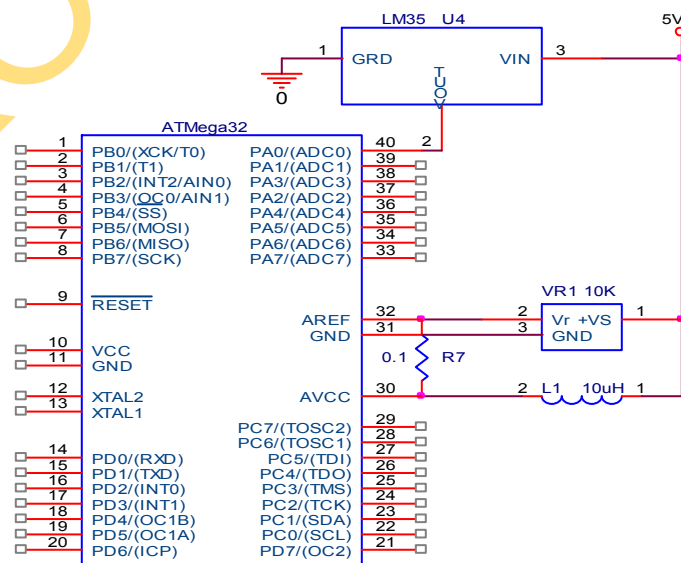
3.1.3. Rangkaian Input

Pada proyek ini terdapat rangkaian *Input* yaitu sensor temperature dan sensor Level air Optocoupler dimana masing – masing komponen memiliki fungsi

yang berbeda – beda untuk menunjang kinerja alat dalam proyek ini. Dimana penjelasan dan rangkaiannya sebagai berikut :

A. Rangkaian *Input* Sensor Temperatur LM35

Rangkaian *Input* sensor temperatur LM35 memiliki fungsi untuk mengetahui temperatur yang terdapat pada ruang yang dipanasi pada proyek ini. Rangkaian *Input* LM35 menggunakan ADC yang telah terdapat pada mikrokontroler. Sensor LM35 ini terhubung pada pin A(0) pada mikrokontroler, yang didalamnya telah terdapat ADC(0), dimana *Output* tegangan keluaran sangat berubah - ubah berpadanan dengan perubahan suhu. Sensor ini berfungsi sebagai pengubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar 10 mV /°C yang berarti bahwa kenaikan suhu 1° C maka akan terjadi kenaikan tegangan sebesar 10 mV, dengan tegangan catu daya yang digunakan adalah 5V, dengan mensetting keluaran Vref sebesar 1V dimaksudkan agar kenaikan temperatur tidaklah secara cepat sehingga menjadi tidak akurat. Gambar rangkaian Sensor Temperatur LM35 dapat dilihat pada gambar 3.12.



Gambar 3.11. Rangkaian LM35

Untuk mengetahui setiap kenaikan tersebut dibutuhkan pembanding yang teruji keakuratannya, dimana digunakan suatu alat pembanding untuk mendapatkan nilai temperature yang sesuai, agar proses pasteurisasi dapat berjalan dengan baik. Dengan demikian didapatkan rumus untuk menentukan persentase error pada sistem ini yaitu

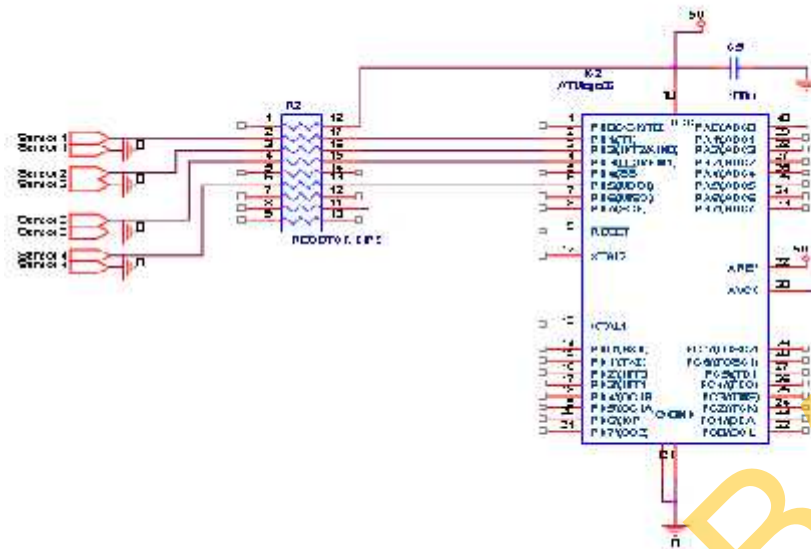
$$\%Error = \frac{HasilPengukuran - NilaiAkurasi}{NilaiAkurasi} \times 100\%$$

Keterangan :

1. Hasil Pengukuran : Nilai yang terukur pada sistem (temperatur sistem °C).
2. Nilai Akurasi : Nilai yang terukur pada alat ukur (temperatur thermometer °C).

B. Rangkaian *Input Sensor Level Air*

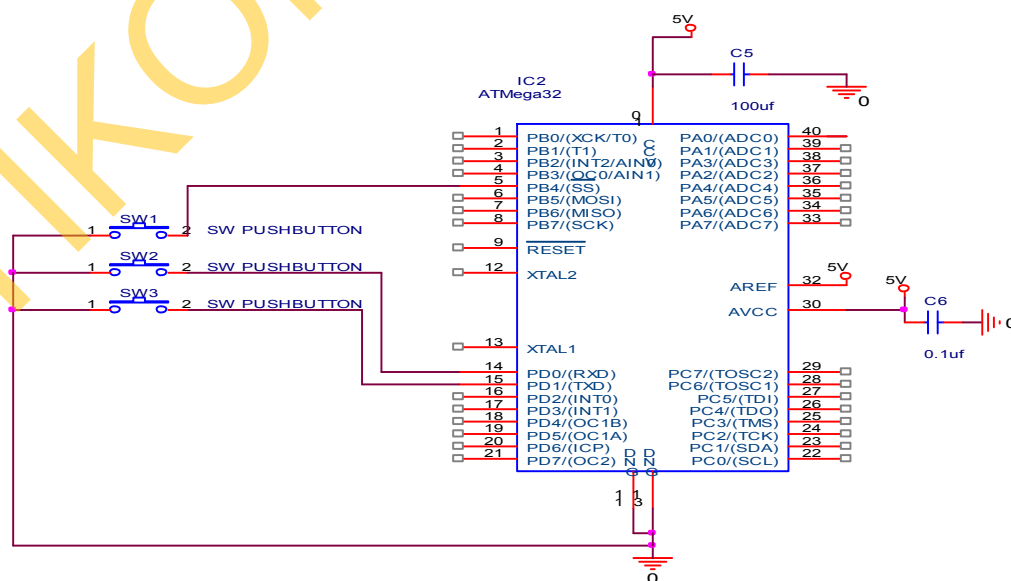
Rangkaian sensor level air ini menggunakan jenis sensor yang menyerupai pelampung atau jenis *float switch water level* dimana salah satu ujungnya dialiri tegangan catu daya 5v dan dilengkapi dengan rangkaian *pull-up* yang difungsikan agar nilai *Output* yang keluar agar memiliki nilai 0 atau 1, dimana prinsip kerja alat ini yaitu, saat sensor level air bernilai 1 apabila switch akan terdorong oleh air yang naik ke permukaan dan tersentuh ujung katup pada sensor maka pin akan *High* atau kondisi “*open*” dan begitu sebaliknya akan bernilai “0” apabila menjauh dari katup atas dengan kondisi “*close*” atau *low*. Gambar 3.12 Rangkaian Level Air.



Gambar 3.12. Rangkaian Level Air.

C. *Push Button Mini switch.*

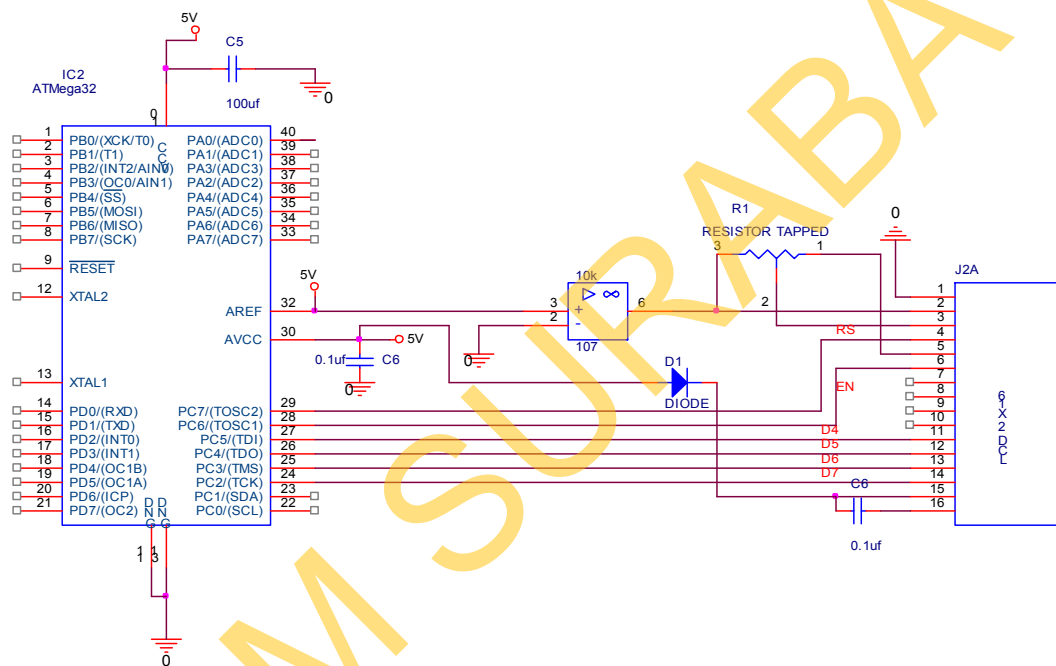
Push Button disini memiliki fungsi untuk memfungsikan alat dan untuk pemilihan mode pasteurisasi yang diinginkan user. Prinsip kerja *Push Button* adalah memiliki fungsi sama seperti saklar *push-on* yaitu akan terhubung pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat – saat katup tidak ditekan.

Gambar 3.13. Rangkaian *Push Button*.

3.1.4. Rangkaian Output

A. Modul Display (LCD)

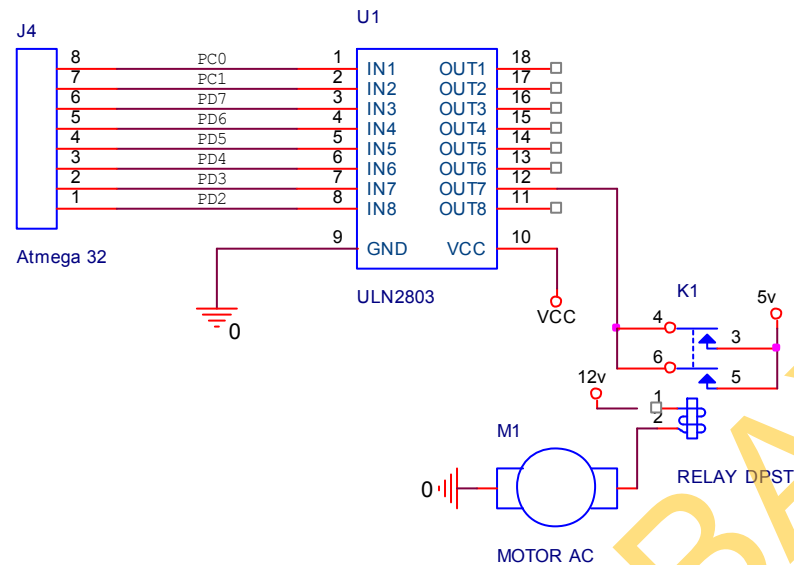
Modul *display* merupakan modul yang berfungsi untuk menampilkan menu, intruksi – intruksi program yang akan dijalankan, informasi waktu dan temperatur saat proses pasteurisasi dilakukan yang dikirim oleh mikrokontroler. Informasi tersebut ditampilkan pada sebuah LCD 16 x 2.



Gambar 3.14 Rangkaian LCD

B. Relay Motor

Motor DC digunakan sebagai pengaduk cairan susu agar cairan susu didalam tabung pemanas cepat mencapai titik temperatur panas yang di inginkan. Motor yang digunakan menggunakan tegangan 12v dan arus 1A. untuk menjalankan motor DC ini memerlukan perantara antara mikrokontroler dan motor driver. Dimana sebagai driver motor digunakan relay seperti yang dijelaskan diatas sehingga memenuhi tegangan dan arus motor.

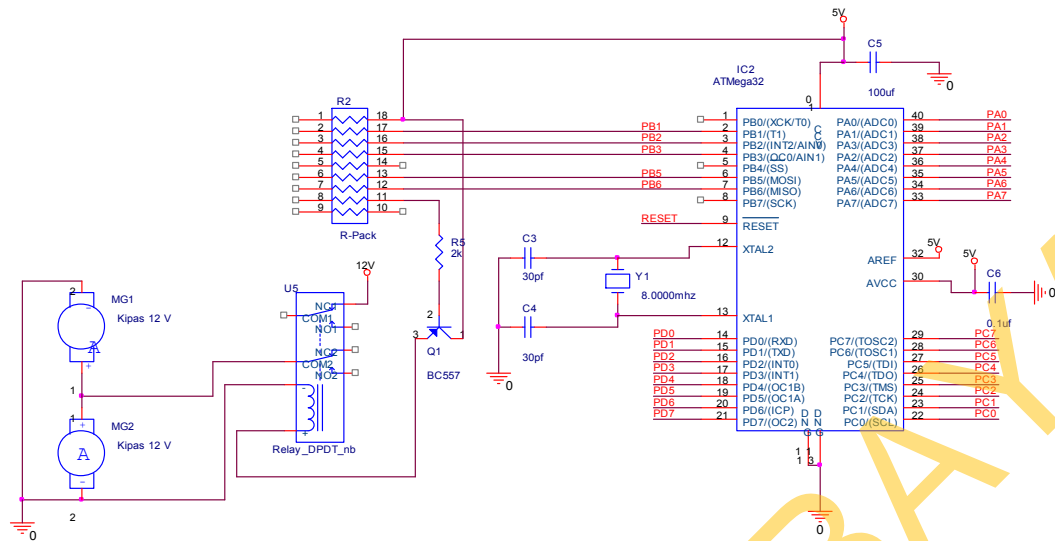


Gambar 3.15 Rangkaian Motor DC.

Kecepatan motor DC yang dikontrol dengan rangkaian diatas akan memiliki torsi maksimum dan kecepatan maksimum. Pada rangkaian interface antara motor DC dengan mikrokontroler seperti pada rangkaian diatas, data kontrol motor DC dari mikrokontroler diberikan ke driver ULN2803 untuk menggerakan relay yang pada akhirnya relai *ON* dan motor DC mendapat *supply* tegangan melalui kontaktor relay.

C. Relay Kipas

Kipas digunakan sebagai prototype pendingin ruangan yang dipanaskan, yang difungsikan sebagai pendingin ruangan apabila panas yang dihasilkan kompor melebihi temperatur yang diinginkan maka kipas akan menyala sampai temperatur ruangan tersebut sesuai yang diinginkan.



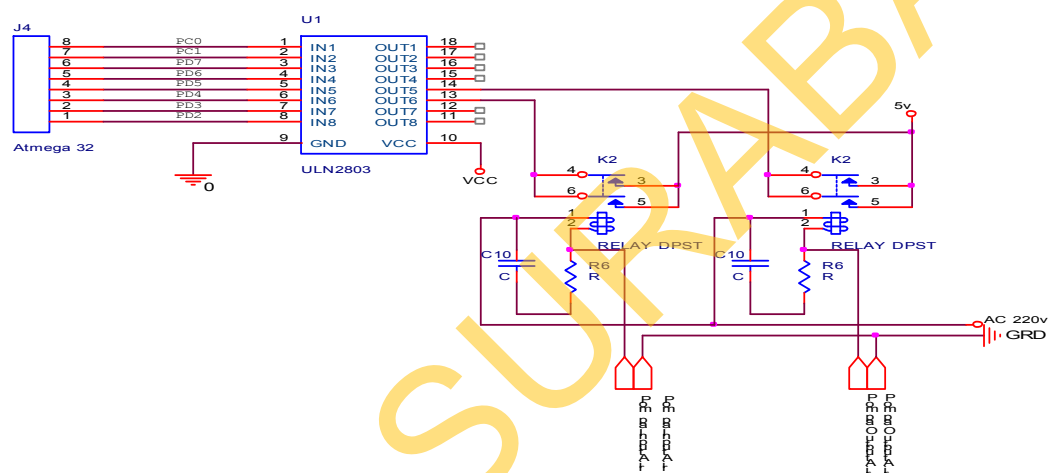
Gambar 3.16 Rangkaian Relay Kipas.

Prinsip kerja kipas ini menggunakan prinsip logika *high* dan *low* pada pin mikrokontroler digunakan untuk trigger pada rangkaian transistor (Transistor BC557) sebagai saklar. Logika *high* akan dijadikan bias maju untuk pin basis transistor sehingga transistor aktif ketika kaki *colector* dan *emitter* dibias maju. Untuk menghasilkan kondisi *on* / *off* seperti saklar, transistor dioperasikan pada salah satu titik kerja, yaitu titik saturasi dan *cut-off*. Saat nilai saturasi, transistor akan seperti sebuah saklar yang tertutup (*on*) sehingga arus dapat mengalir ke kolektor menuju emitor. Sedangkan pada kondisi *cut-off*, transistor seperti sebuah saklar terbuka (*off*) sehingga tidak ada arus yang mengalir ke emitor. Jadi saat logika *high* atau bernilai 5 v maka relay akan aktif dan kipas akan berputar dan jika logika *low* atau bernilai 0 v maka relay *non-aktif* dan kipas akan berhenti.

D. Relay Pomp Air *Input* dan *Output*

Relay pompa air ini sistem kerjanya sama seperti sama seperti pada kinerja motor DC diatas menggunakan ULN2803 sebagai driver untuk

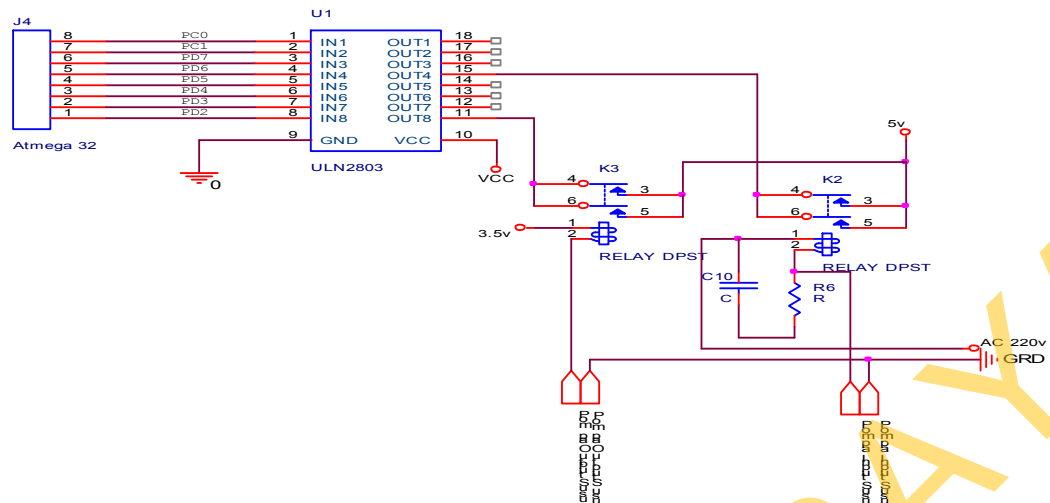
menjalankan pompa. Dimana pada rangkaian interface antara pompa air dengan mikrokontroler seperti pada gambar 3.17 gambar rangkaian relay pompa air, data kontrol pompa air dari mikrokontroler diberikan ke driver ULN2803 untuk menggerakkan relay yang pada akhirnya relai *ON* dan pompa air mendapat supply tegangan melalui kontaktor relay. Pompa akan *ON* apabila arus tegangan 5 v masuk dan berlogika *high* atau bernilai 1 dan pompa akan *OFF* dengan kondisi sebaliknya.



Gambar 3.17 Rangkaian Pompa Air *input* dan Air *Output*

E. Relay Pomp Susu *Input* dan *Output*

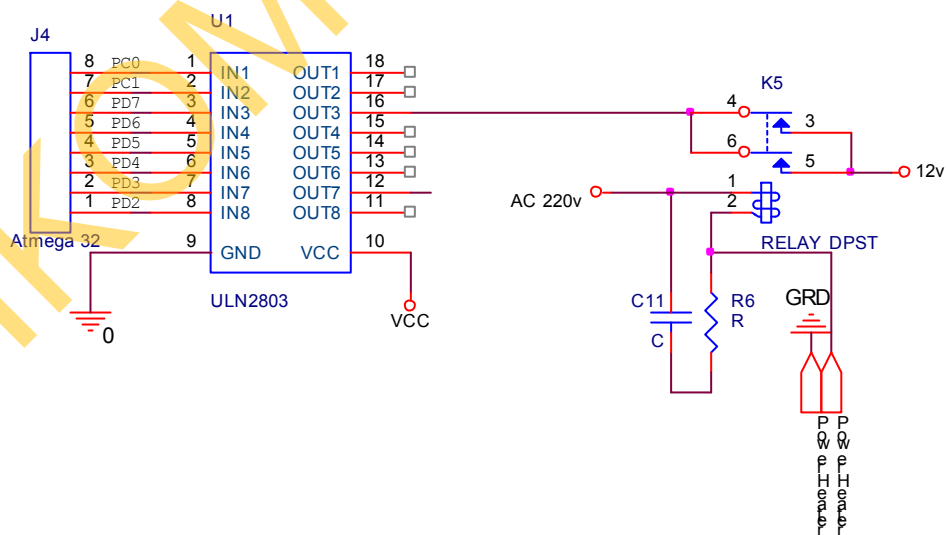
Untuk pompa susu pada gambar 3.18 dibawah, cara kerjanya hamper sama dengan pompa air, hanya saja yang sebagi pembeda disini adalah pompa *Output* susu, pompa *Output* susu disini menggunakan alat pompa sedot air pada gallon air minum kemasan. Dimana pada dasarnya menggunakan baterai yang mengkonsumsi daya 3.5v sehingga untuk mengaktifkanya memerlukan suplay daya 3.5 v dari regulator 3.5v sesuai pada gambar 3.9 rangkaian regulator diatas.



Gambar 3.18. Rangkaian Susu *Input* dan Susu *Output*

F. *Heater*

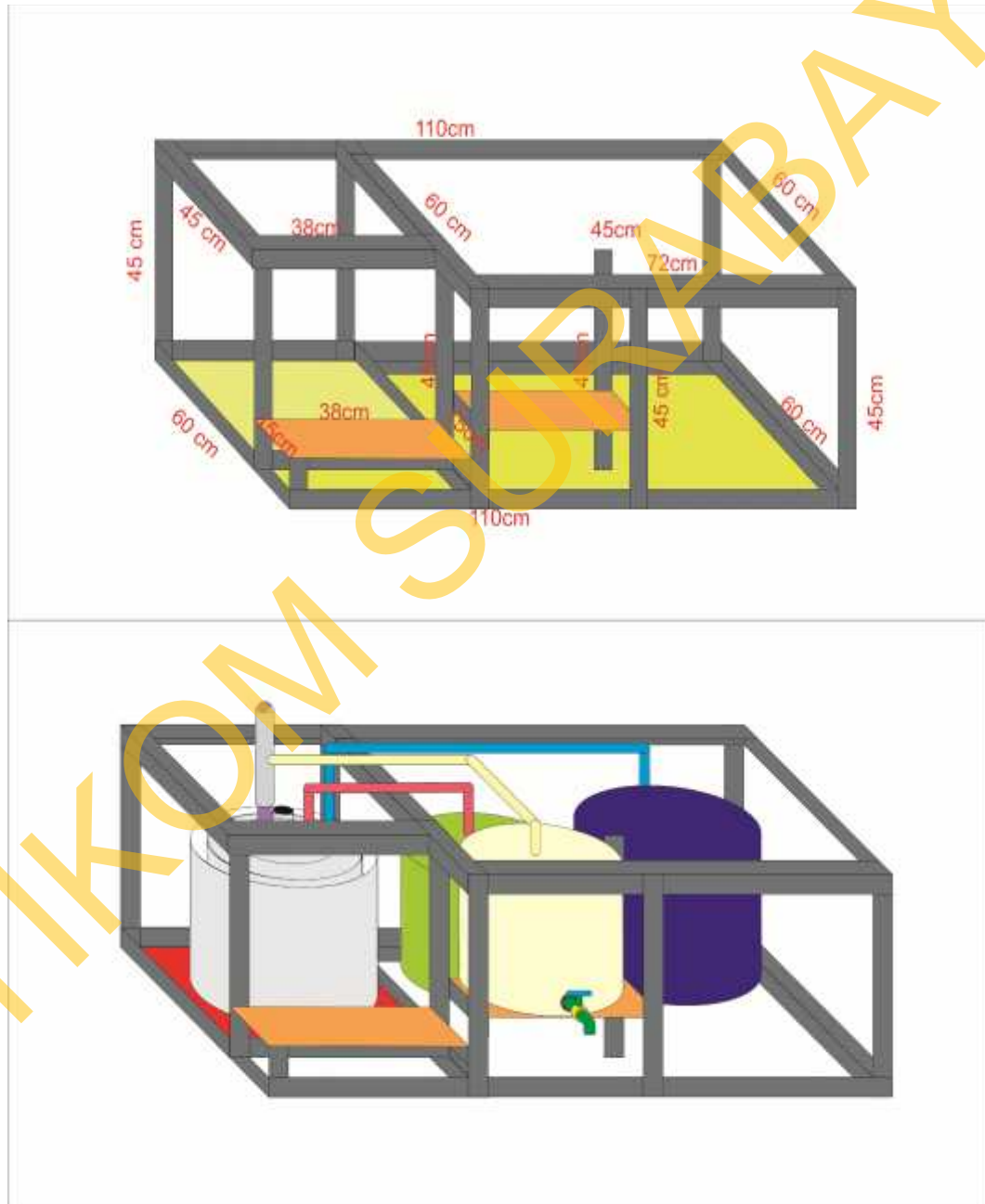
Rangkain *Heater* sama seperti rangkaian – rangkaian lainnya hanya saja dalam penggunaannya disini menggunakan 4 *Heater* untuk melakukan proses pemanasan, 4 *Heater* tersebut di parallel ke dalam 1 relay dimana relay yang digunakan yaitu relay 16A. Masing – masing *Heater* memiliki daya 150 Watt sehingga dapat ditotal menjadi 600 Watt,



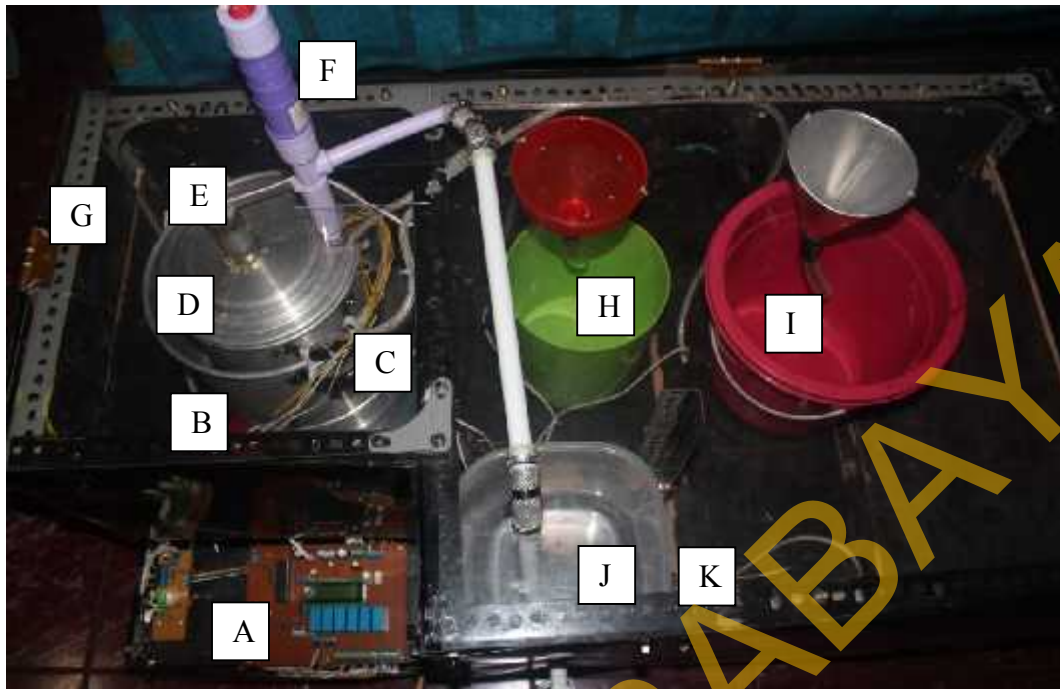
Gambar 3.19. Rangkaian *Heater*.

3.2. Perancangan Arsitektur Sistem

Perancangan arsitektur sistem dalam hal ini adalah maket untuk komponen – komponen elektronika, dimana terbuat dari besi sebagai kerangka utama, acrilik sebagai penutup atas, papan kayu sebagai alas, dan mika sebagai penutup luar yang dirancang seperti gambar 3.20.



Gambar 3.20 Arsitektur Sistem.



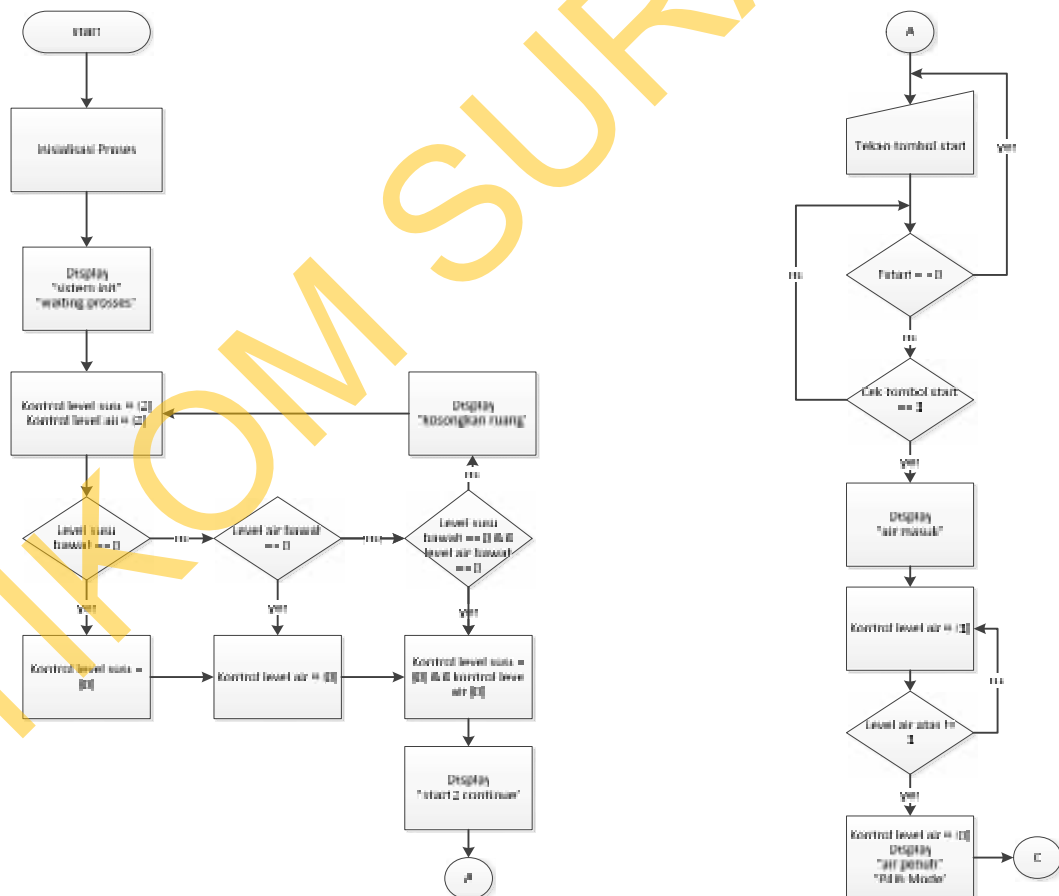
Gambar 3.21 Sistem Keseluruhan.

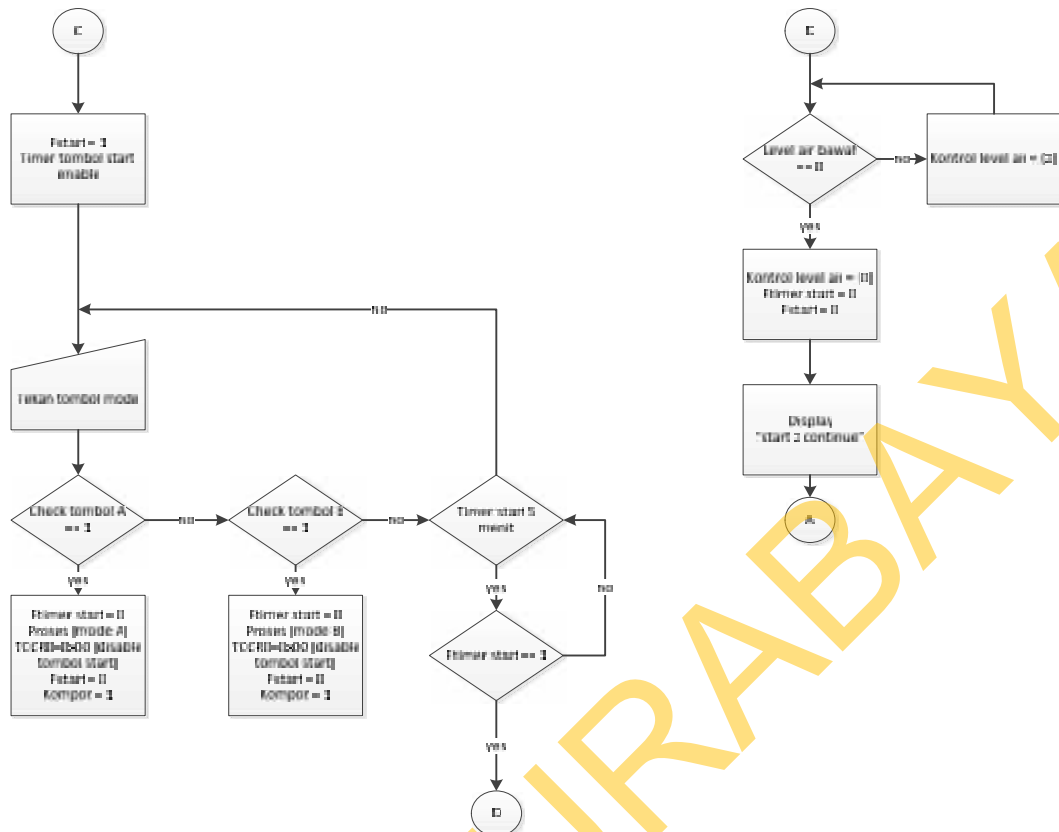
Keterangan :

- A. Board Utama
- B. *Heater*
- C. Panci Pemanas Air (Terdapat Sensor Ketinggian Air)
- D. Panci Pemanas Susu (Terdapat Sensor Ketinggian Susu dan Sensor Temperatur LM35)
- E. Motor Pengaduk
- F. Pompa Penyedot (*Output* susu)
- G. Kipas Sirkulasi
- H. Penampung Susu (Terdapat Pompa input susu)
- I. Penampung Air (Terdapat Pompa input air)
- J. Penampung Susu
- K. *Power Suppy* dan *Adaptor*.

3.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk memperoleh, menampilkan data temperatur dan waktu untuk pemodelan Pasteurisasi yang digunakan. Perancangan perangkat lunak terbagi dalam beberapa *device* sistem antara lain : program baca level air, program baca temperatur, program hitung pemodelan pasteurisasi, program *dispay* LCD, dan program untuk penggerak komponen – komponen pembantu lainnya seperti, motor pengaduk, *Heater* (pemanas), kipas sirkulasi, pompa penyedot susu, pompa untuk mengeluarkan air, pompa mengisi air dan pompa mengisi susu. Diagram alir perangkat lunak secara umum dapat dilihat pada Gambar 3.22. berikut.





Gambar 3.22 Diagram alir program secara umum.

Pada gambar di atas merupakan pembacaan diagram alir program keseluruhan secara umum, dimana saat awal pembacaan, program akan melakukan inisialisasi fungsi –fungsi awal yang telah dibuat. Dimana saat sistem melakukan inisialisasi, sistem melakukan pengecekan seluruh perintah yang akan dijalankan seperti timer, interupsi, port – port yang digunakan dan yang lainnya diakhiri dengan `display` perintah “*system init*” “*waiting prosses.*”. Pada saat pengecekan awal sistem terjadi, dilakukan pengecekan dengan menjalankan fungsi dari kontrol level air dan kontrol level susu, dimana saat kontrol level air pada sensor level air bagian bawah dan kontrol level susu pada sensor level susu bawah yang terdapat pada ruang pemanasan berlogika “1” atau “*open*” maka sistem akan menjalankan perintah dengan `display` “mengkosongkan ruang” dengan penyesuaian komponen *Output* untuk pengosongan ruang pemanasan yaitu pompa air keluar dan pompa

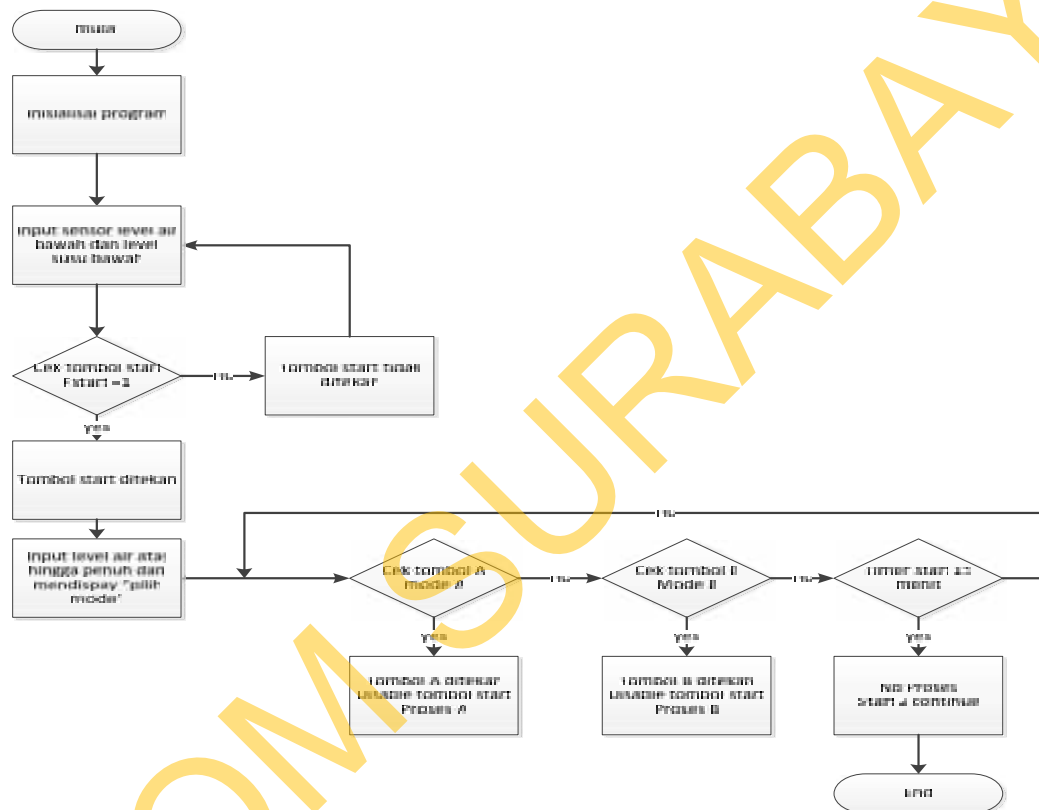
susu keluar akan *aktif* hingga sensor mendeteksi bahwa air dan susu benar – benar kosong. Jika sudah sensor tidak mendeteksi cairan yang ada dalam tabung atau berlogika “0” atau “close” maka dilanjutkan dengan mendisplay perintah “start 2 continue”. Dengan mendisplay perintah “start”, dimaksudkan agar user melakukan penekanan tombol “start”. Jika tombol “start” tidak ditekan sistem akan menunggu hingga terjadi penekanan “fstart == 0”, dan sebaliknya jika dilakukan penekanan maka sistem akan membaca penekanan tombol *start* “fstart == 1” dengan perintah selanjutnya yaitu pengisian ruang pemanasan air dengan display “air masuk” dan disesuaikan dengan *Output* dari pompa air masuk akan *aktif*, selanjutnya mengisi air pada ruang pemanasan air dan menunggu hingga sensor level air atas mendeteksi kondisi “open” atau “1” dengan mendisplay kondisi “air penuh” disesuaikan dengan pompa air masuk akan berhenti mengisi dan dilanjutkan dengan mendisplay “pilih mode” dengan diaktifkannya *timer* tombol “start” selama 5 menit, ketika salah satu tombol mode mendapat inputan penekanan, maka setiap inputan tombol mode akan menjalankan masing – masing prosesnya seperti proses (mode A) dan proses (mode B) dengan dimulainya proses pemanasan.

Apabila salah satu tombol mode tidak ditekan maka sistem akan kembali melakukan ke proses awal dimana akan terjadi pengosongan ruang dengan aktifnya pompa air keluar dan menunggu kondisi sensor level air bawah bernilai “0” atau “close” yang disesuaikan dengan berhentinya pompa air keluar dan mendisplay “start 2 continue”.

3.3.1. Program Mikrokontroler

A. Program Tombol.

Diagram alir untuk mengetahui input penekanan tombol untuk melakukan proses menjalankan alat dan memilih mode pasteurisasi yang diinginkan user, dapat dilihat pada gambar 3.23. berikut.



Gambar 3.23. Diagram alir pembacaan input penekanan tombol.

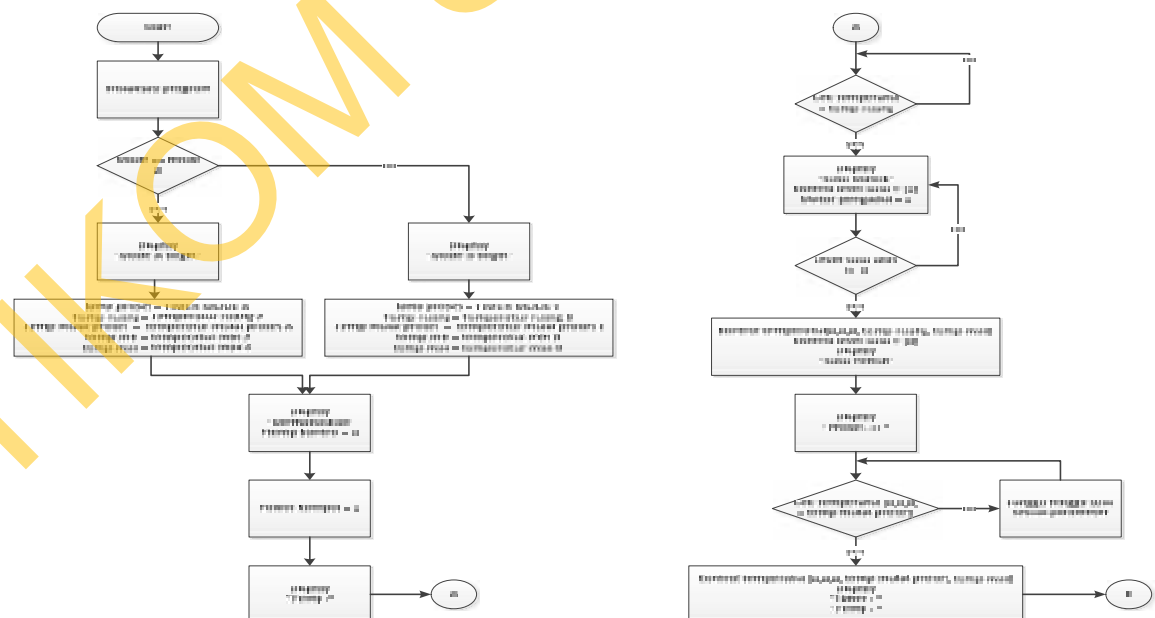
Pada gambar 3.23. Diagram alir pembacaan input penekanan tombol dimulai dengan inputan dari sensor level susu bawah dan level air bawah, dikondisikan jika dalam media penampungan untuk pemanasan terdapat susu dan air maka dikosongkan terlebih dahulu. Setelah itu tombol “Start” akan berfungsi. Setelah penekanan tombol “Start”, pompa akan *aktif* untuk mengisi media penampungan untuk pemanasan hingga level air atas menerima inputan. *timer* tombol “Start” akan *aktif* selama 5 menit. Apabila salah satu tombol mode “A”

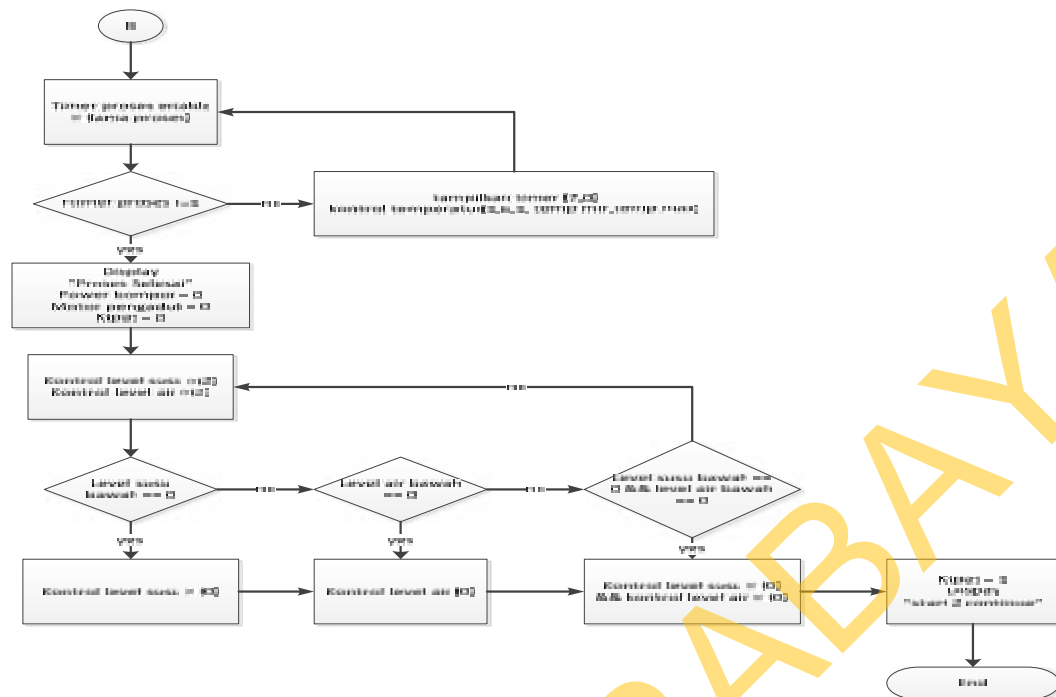
atau mode “B” ditekan maka tombol “*Start*” akan disable dan proses selanjutnya berlangsung, dan apabila tombol mode tidak ditekan selama 5 menit maka air akan keluar kembali dan proses kembali ke awal. Berikut potongan program pembacaan tombol yaitu :

```
system_init();
while (1)
{
    if (fstart == 0)
    {
        if (cek_tombol_start() == 1)
        {
            lcd_clear();
            lcd_putsf("Air Masuk");
            control_level_air(1);
            while(level_air_atas != 0);
            control_level_air(0);
        }
        ..... (lanjutan program bisa dilihat di halaman
lampiran )
    }
}
```

B. Program Mode A dan Mode B

Diagram alir untuk mengetahui mode mana yang akan diinputkan user dapat dilihat pada gambar 3.24 berikut.





Gambar 3.24. Diagram Alir Mode A dan Mode B.

Pada gambar 3.24 Diagram Alir Mode A dan Mode B diproses melalui penekanan tombol mode oleh user sendiri sesuai kebutuhan waktu pasteurisasi yang diinginkan. Dimana untuk tombol Mode A waktu yang di sediakan untuk Mode pasteurisasi jenis LTLT (*Low Temperature Long Time*) dengan waktu pasteurisasi selama 30 menit temperatur 60°C - 63°C dan untuk tombol Mode B dengan mode pasteurisasi jenis HTST (*High Temperatur Short Time*) dengan waktu pasteurisasi selama 15 detik temperatur 71°C - 75°C . berikut potongan program pembacaan tombol Mode A dan Mode B.

```

#define TIMER_MODE_A 1800 // 30 MENIT
#define TIMER_MODE_B 15 // 15 DETIK

void proses(u8 mode)
{
    u16 lama_proses;
    float temp_ruang, temp_mulai_proses, temp_min, temp_max;

    // --- PEMILIHAN MODE ---
    lcd_clear();
    if (mode == MODE_A)

```

```

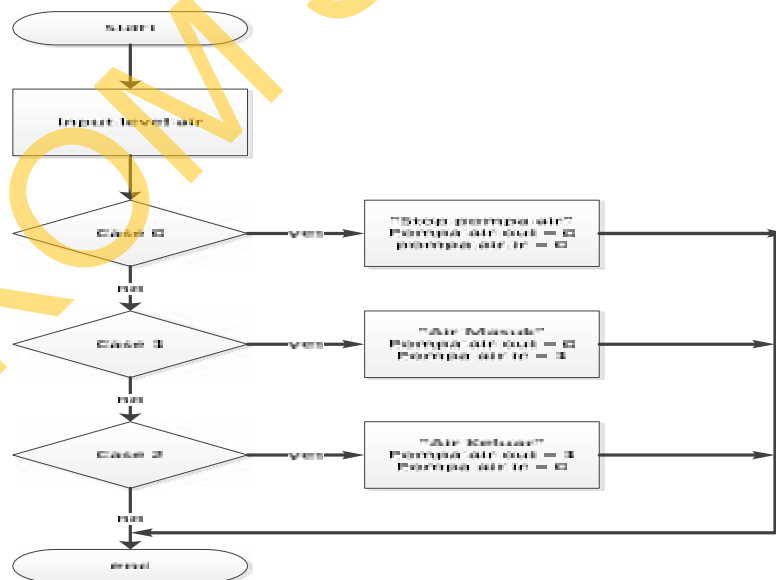
{
    lcd_putsf("MODE A BEGIN!");
    lama_proses = TIMER_MODE_A;
    temp_ruang = TEMPERATUR_RUANG_A;
    temp_mulai_proses = TEMPERATUR_MULAI_PROSES_A;
    temp_min = TEMPERATUR_MIN_A;
    temp_max = TEMPERATUR_MAX_A;
}
else
{
    lcd_putsf("MODE B BEGIN!");
    lama_proses = TIMER_MODE_B;
    temp_ruang = TEMPERATUR_RUANG_B;
    temp_mulai_proses = TEMPERATUR_MULAI_PROSES_B;
    temp_min = TEMPERATUR_MIN_B;
    temp_max = TEMPERATUR_MAX_B;
}
..... (lanjutan program bisa dilihat di halaman

```

lampiran)

C. Program Pompa Air.

Diagram alir untuk mengaktifkan dan menon-aktifkan pompa air masuk dan keluar, dimana diagram alirnya dapat dilihat pada gambar 3.25. sebagai berikut.



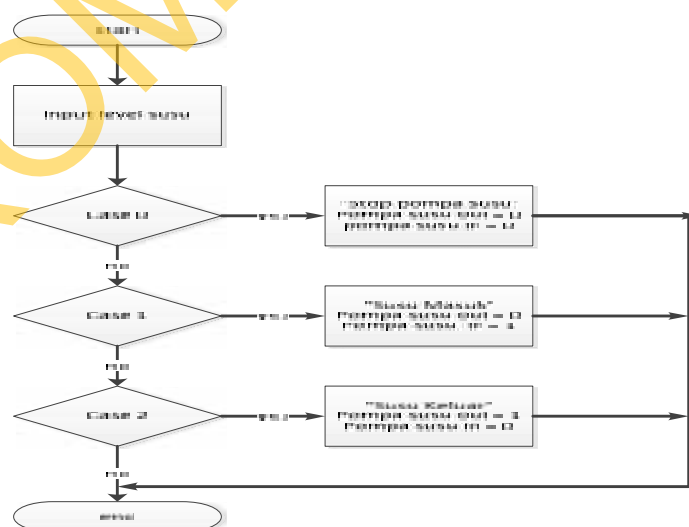
Gambar 3.25. Diagram alir Pompa air.

Pada gambar 3.25. Diagram alir Pompa air, menunjukkan bahwa proses kinerja pompa air, pada *case* (0) posisi pompa air non-aktif berarti pompa tidak mendapat inputan untuk melakukan proses. Sedangkan *case* (1) pompa air masuk akan *aktif*, dan *case* (2) pompa air keluar akan *aktif*. Berikut potongan program pembacaan pompa air sebagai berikut:

```
void control_level_air(u8 mode_control)
{
    switch (mode_control)
    {
        // stop pompa air
        case 0 :
        {
            pompa_air_out = 0;
            pompa_air_in = 0;
            break;
        } ..... (lanjutan program bisa dilihat di halaman lampiran )
    }
}
```

D. Program Pompa Susu.

Program alir pompa susu berfungsi untuk mengaktifkan dan menonaktifkan pompa susu masuk dan keluar, dimana diagram alirnya dapat dilihat pada gambar 3.26. sebagai berikut.



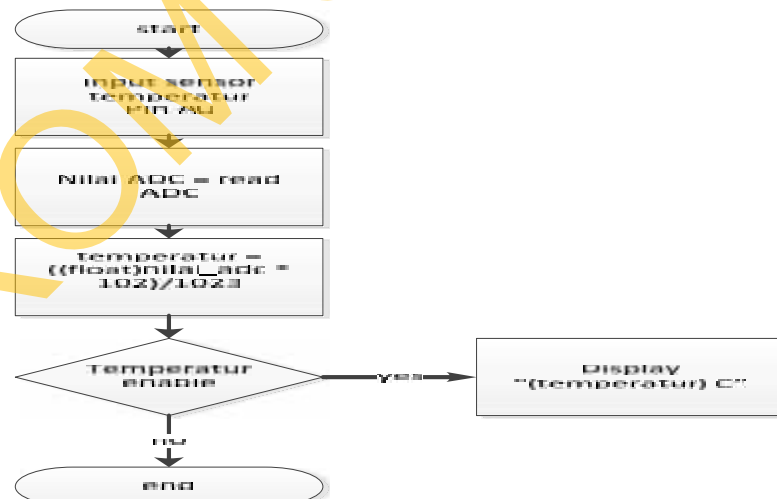
Gambar 3.26. Diagram Alir Pompa Susu.

Pada gambar 3.26. Diagram Alir Pompa Susu proses kinerjanya sama seperti pompa air diatas, dimana untuk *case* (0) menonaktifkan pompa susu *input* maupun *Output*, *case*(1) mengaktifkan pompa susu masuk, dan *case* (2) mengaktifkan pompa susu keluar. Berikut potongan program pembacaan diagram alir pompa susu sebagai berikut :

```
void control_level_susu(u8 mode_control)
{
    switch (mode_control)
    {
        case 0 :
        {
            pompa_susu_out = 0;
            pompa_susu_in = 0;
        } ..... (lanjutan program bisa dilihat di halaman lampiran )
    }
}
```

E. Program Pembacaan Temperatur.

Diagram alir untuk pembacaan temperatur yang berdasarkan inputan sensor temperatur LM35 dengan *Output* temperatur “C” (*cencius*) terdapat pada gambar 3.27.



Gambar 3.27. Diagram Alir Pembacaan Temperatur.

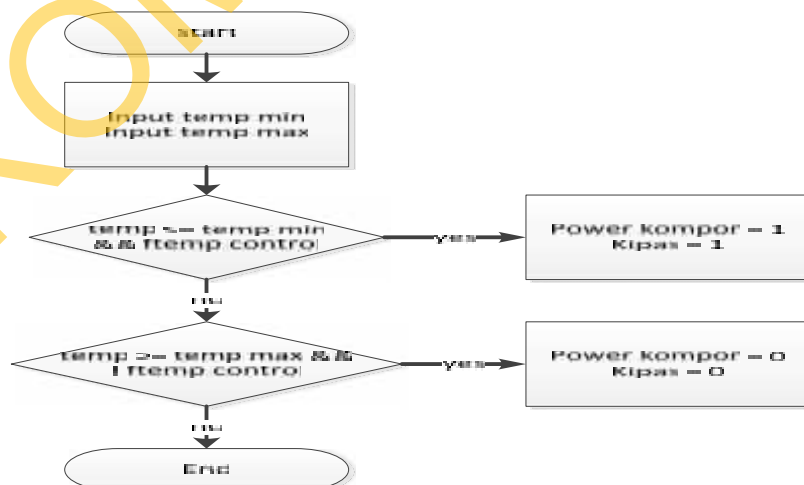
Pada gambar 3.27. Diagram Alir Pembacaan Temperatur menunjukkan bahwa apabila sensor mendapat *input* temperatur maka outpunya akan masuk ke

ADC, dimana nilai ADC sama dengan Read ADC. Nilai ADC merupakan data digital yang telah dihasilkan dari konversi ADC. Untuk mendapatkan nilai dari temperatur maka $((\text{float}) \text{nilai_adc} * 102) / 1023$ dimana nilai dari 102°C itu didapat dari nilai v_{ref} yang telah di *setting* 1000mV atau nilai maksimal dan nilai minimal 0mV 2°C. Saat proses pembacaan temperatur *aktif* atau *enable* nilai temperatur akan ditampilkan dengan nilai temperatur dengan satuan *celcius*. Berikut potongan program dari pembacaan temperatur.

```
float cek_temperatur(const u8 enable, const u8 baris, const u8
kolom)
{
    int nilai_adc = read_adc(0);
    s8 str[6];
    float temperatur = ((float)nilai_adc * 102)/1023;
}..... (lanjutan program bisa dilihat di halaman lampiran )
```

F. Program Kipas.

Diagram alir kipas merupakan pembacaan *input* dari sensor temperatur, menghasilkan *Output* kipas dapat dilihat pada gambar 3.28.



Gambar 3.28. Diagram Alir Kipas.

Pada gambar 3.28. Diagram Alir Kipas diatas, kipas akan *non-aktif* apabila kurang dari temperatur minimal dan temperatur *control* yang sudah di *setting*. Maka *Power* kompor = 1, berarti *heater* dalam kondisi memanaskan. Apabila sebaliknya kipas akan *aktif*, jika melebihi temperatur maksimal dan tidak sesuai dengan temperatur *control* maka *Power* kompor = 0 dimana *Power* kompor = 0 itu berfungsi untuk menon-aktifkan kompor. Potongan program sebagai berikut.

```
void control_temperatur(const u8 enable, const u8 baris, const u8
kolom, float t_min, float t_max)
{
    float temp = cek_temperatur(enable, baris, kolom);
    if (temp <= t_min && ftemp_control)
    {
        Power_kompor(0);
        kipas = 1;
        ftemp_control = 0;
    }
}..... (lanjutan program bisa dilihat di halaman lampiran )
```