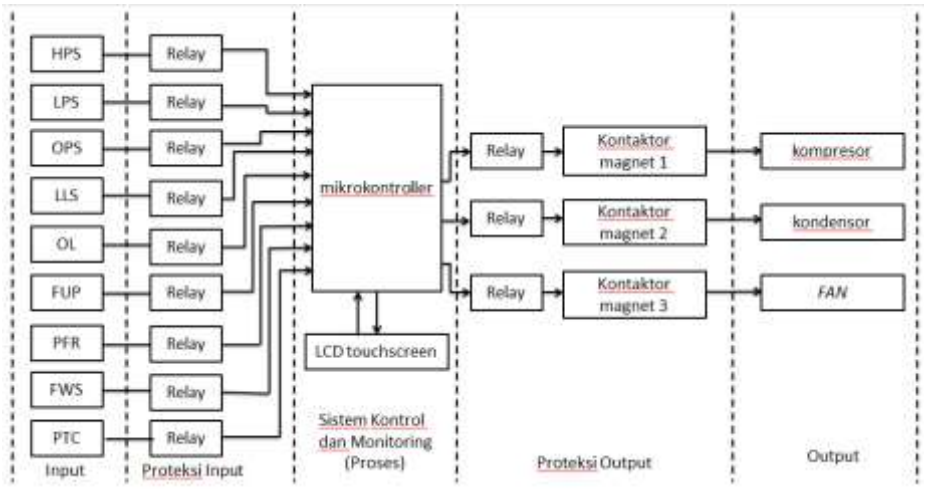
BAB III

PERANCANGAN ALAT

Secara umum, proyek akhir ini terdiri dari perancangan dan pemasangan *hardware* serta perancangan dan pembuatan algoritma pengontrolan motor induksi 1 phasa menggunakan mikrokontroller, untuk mendapatkan hasil yang sesuai dengan keinginan atau pengaturan yang sudah dilakukan sebelumnya maka pengontrolan motor dilakukan secara *real time*, sehingga setiap terjadi perubahan beban atau *set point*, sedikit banyak akan mengakibatkan perubahan pada parameter beban motor.

3.1 Perancangan Sistem dan Prinsip Kerja Sistem

Perancangan dan pembuatan perangkat keras ini meliputi pembuatan mekanik dan rangkaian elektronika yaitu terdiri dari beberapa bagian seperti motor induksi 1phasa, sensor, kontator magnet, relay, modul mikrokontroller atau arduino mega dan LCD *touchscreen* 3.5in *display*.

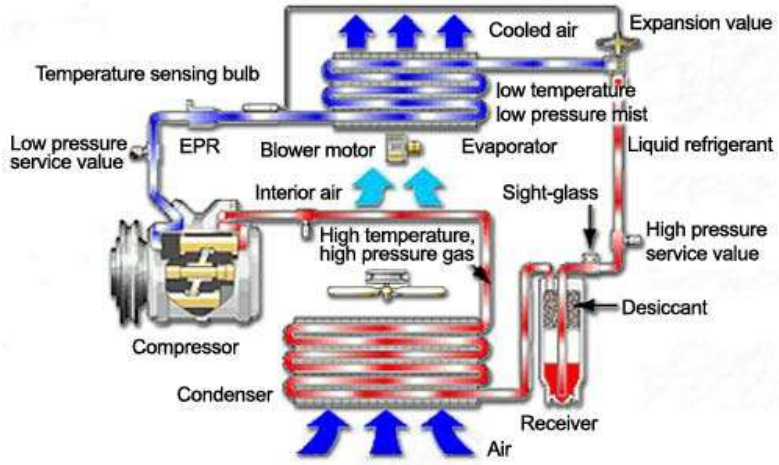


Gambar 3.1 Blok diagram Sistem Kontrol *Chiller*

Penjelasan dari blok diagram di atas adalah sebagai berikut. Sistem akan membaca apakah semua inputan berada dalam kondisi normal atau tidak, jika inputan tidak dalam posisi normal maka sistem tidak akan bekerja, akan tetapi jika semua inputan dalam keadaan normal maka sistem dapat dijalankan dengan mengatur suhu yang diharapkan dan nilai maksimum dan minimum dari suhu yang diharapkan. Ketika suhu yang diharapkan sudah di atur maka sistem akan bekerja sesuai dengan suhu yang diharapkan, dengan mengontrol kompresor, *fan*, dan kondensor suhu yang dihasilkan dapat sesuai dengan suhu yang diharapakan. Jika terjadi masalah pada inputan maka kita akan mengetahuinya pada menu informasi, didalam menu informasi terdapat sebuah pemberitahuan yang berupa permasalahan apa yang terjadi sehingga sistem tidak bekerja dan juga waktu kejadian ketika sistem itu bermasalah.

Ketika sistem keadaan dalam bekerja kemudian terjadi beberapa masalah atau kesalahan pada chiller, misalkan tekanan refrigeran terlalu tinggi maka smua ouput akan dimatikan secara otomatis dikarenakan jika salah satu sistem tetap bekerja maka akan terjadi kebocoran pada *evaporator*. Ataupun jika oli pada kompresor tidak ada atau kurang dari semestinya maka semua output akan berhenti bekerja juga, apabila kompresor yang berhenti bekerja sedangkan kondensor masih bekerja kemungkinan yang akan terjadi adalah tekanan refrigeran pada kondensor akan sangat tinggi sehingga akan menyebabkan kebocoran dan kerusakan pada *evaporator* yang terdapat pada kondensor dan akan terjadi kerusakan pula pada kondensornya.

Alasan mengapa sistem yang dibuat harus mengatur suhu secara manual pada LCD *touchscreen* dikarenakaan penggunaan *chiller* yang berbeda beda, seperti penggunaan *chiller* untuk mendinginkan ruangan biasanya hanya membutuhkan suhu pada kisaran 17 °C sampai dengan 30 °C, sedangkan untuk mendinginkan sayuran atau buah berada pada kisaran 10 °C, dan untuk mendinginkan daging agar tidak membusuk membutuhkan suhu pada kisaran 7 °C.



Gambar 3.2. Siklus Kerja *Chiller*

Sistem pengontrol yang dibuat memanfaatkan bebrapa instrumen atau indikator-indikator yang terdapat pada *chiller* seperti sensor suhu sebagai pendeteksi suhunya, *high pressure* sebagai pembca tekanan refrigeran dan lainnya sebagai inputan untuk mengendalikan kopresor, *fan*, dan kondensor. Kompresor akan dihidupkan dan dimatikan untuk mengatur suhu yang diharapkan, sementara *fan* dan kondesnsor akan dinyalakan secara terus menerus dikarenakan siklus perubahan yang terjadi pada evaporator dan kondensor harus terus menerus terjadi, akantetapi jika terjadi

tekanan refrigeran yang sangat tinggi barulah kompresor, *fan*, dan kondensor harus berhenti bekerja untuk menghindari terjadinya kebocoran pada *evaporator*, begitu juga jika terjadi masalah lainnya.

3.1.1 Fungsi Masing-Masing Sistem

1. Sensor suhu

   Sensor ini berfungsi untuk mengukur suhu pada *chiller* dan sebagai indikator pencapaian setpoint yang telah diatur, adapun jenis sensor suhu yang digunakan yaitu dengan jenis PTC.

2. Instrumen *High Pressure Switch*

   Instrumen ini berfungsi sebagai indikator jika terjadi tekanan *refrigerant* yang terlalu tinggi pada kondensor.

3. Instrumen *Low Pressure Switch*

   Instrumen ini berfungsi sebagai indikator jika terjadi tekanan *refrigerant* yang terlalu rendah pada kondensor.

4. Instrumen *Oil Pressure Switch*

   Instrumen ini berfungsi sebagai indikator jika terjadi perbedaan tekanan antara tekanan oli dan tekanan *refrigerant*.

5. Instrumen *Low Level Switch*

   Instrumen ini berfungsi sebagai indikator jika air yang terdapat pada *colling tower* terlalu rendah atau kurang dari seharusnya.

6. Instrumen *Over Load*

   Instrumen ini berfungsi sebagai indikator dan proteksi jika terjadi beban lebih pada motor DC.

7. Instrumen *Freeze Up Protection*

Instrumen ini berfungsi sebagai indikator dan proteksi jika terjadi pembekuan pada evaporator yang diakibatkan oleh beberapa hal yang memungkinkan adanya pembekuan tersebut.

8. Instrumen *Phase Failure Relay*

   Instrumen ini berfungsi sebagai indikator dan proteksi jika terjadi tegangan jatuh dan kebalikan gelombang pada motor induksi.

9. Instrumen *Flow Water Switch*

   Instrumen ini berfungsi sebagai indikator dan proteksi pengaman ketika pompa mati akan tetapi air di dalam tangki kosong maka akan mematikan sistem sampai air terisi kembali.

10. *Relay*

    Relay yang terletak setelah inputan atau sensor akan meproteksi mikrokontroller jika terjadi beban lebih sekaligus menurunkan tegangan yang dihasilkan oleh inputan atau sensor, dikarenakan tegangan yang dihasilkan oleh sensor atau inputan sebesar 25v sedangkan mokrokontroller hanya membutuhkan tegangan paling besar 5v saja. Jika lebih dari itu maka mikrokontroller akan rusak atau tidak dapat digunakan lagi.

    Sedangkan relay yang terletak sebelum kontaktor berfungsi untuk merubah tegangan dc menjadi tegangan ac sekaligus menaikan tegangan tersebut menjadi 220v, dikarenakan motor yang diguganakan membutuhkan tegangan sebesar 220v atau 1 fasa.

11. Kontaktor Magnet

Kontaktor magnet berfusngsi untuk menggerakan motor 1 fasa sekaligus sebagai sistem proteksi jika terjadi beban lebih pada motor tersebut, kontaktor magnet akan dikontrol oleh mikrokontroller sesuai dengan program yang telah dibuat untuk mengatur kompresor, *fan*, dan kondensor agar suhu yang dihasilkan sesuai dengan suhu yang diharapkan.

12. Kondensor

    Fungsi kondensor yakni untuk menjaga tekanan gas *superheat* refrigeran yang masuk ke kondensor kemudian suhu refrigeranya diturunkan dengan membuang sebagian kalornya ke medium pendingin yang di gunakan pada kondensor.

13. Kompresor

    fungsi kompresor yakni untuk menaikan tekanan refrigeran dengan mengkompresikannya dalam bentuk gas atau uap, akibatnya temperatur refrigeran juga ikut naik. Siklus kerja *chiller* dapat dilihat berikut ini.

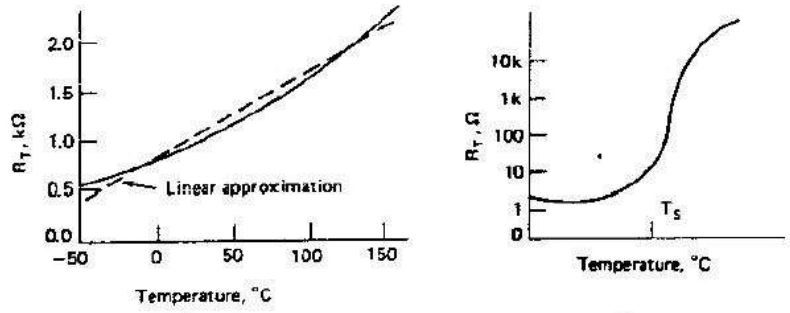
3.1.2 Pemilihan Komponen

Pemilihan komponen ini dilakukan karena jika salah dalam pemilihan komponen maka akan mempengaruhi kualitas, efektifitas, dan efisiensi perangkat yang akan dibuat.

1. Pemilihan Sensor Suhu

   Sensor suhu yang digunakan yaitu PT100 dikarenakan suhu yang terdapat pada *chiller* berubahrubah secara *real time* maka sensor yang digunakan harus dapat bertahan dengan perubahan suhu

tersebut agar dapat digunakan untuk jangka panjang, serta untuk memudahkan dalam proses pemrograman pada sistem kenali mikrokontroller.



Gambar 3.3 Karakteristik Sensor PTC

Pada gambar diatas dapat kita ketahui bahwa karakteristik dari sensor suhu dengan jenis PTC suhu yang terbaca akan sejajar dengan resistansi yang dihasilkan oleh sensor tersebut.

2. Pemilihan Instrumen-instrumen

   Instrumen-instrumen yang digunakan pada perangkat ini disesuaikan dengan isntrumen-instrumen yang sering digunakan pada semua jenis *chiller* pada umumnya.

3. Pemilihan Mikrokontroller

   Jenis mikrokontroller yang digunakan yaitu arduino mega dengan alasan mikrokontroller ini memiliki parameter-parameter yang mampu memenuhi kebutuhan dari fungsi *input* dan dapat menjalakan *output* yang digunakan, selain itu memudahkan pula dalam proses pemrograman karena banyaknya pustaka-pustaka yang dapat membantu dalam pembuatan program.

4. Pemilihan LCD

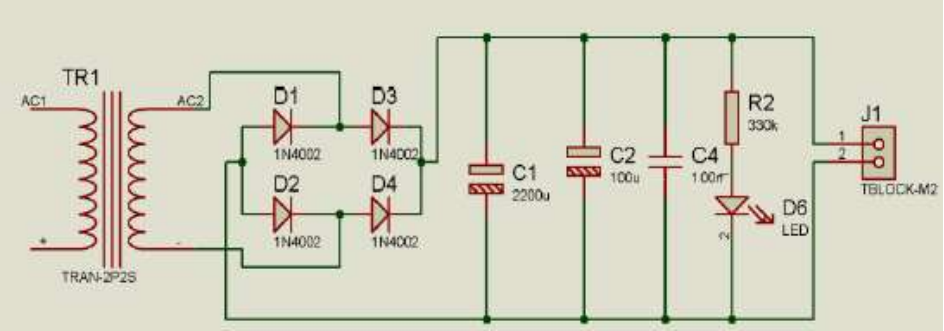
LCD yang digunakan yaitu LCD *touchscreen nextion* karena LCD ini tidak akan menghabiskan banyak pin yang terdapat pada arduino dan dapat dijadikan sebagai penginputan data untuk diproses pada arduino, tidak seperti LCD pada umumnya. Selain itu LCD ini juga memeliki pustaka-pustaka untuk membantu dalam proses pembuatan program.

3.2 Perancangan *Hardware*

Pada perancangan ini merancang sebuah sistem perangkat keras yang akan disesuaikan dengan perancangan pada program sehingga akan terjadinya kesinambungan antara *hardware* dan *software*, perancangan disini terdiri dari *input*, proses dan *output*.

3.2.1 Perancangan Rangkaian Catu Daya

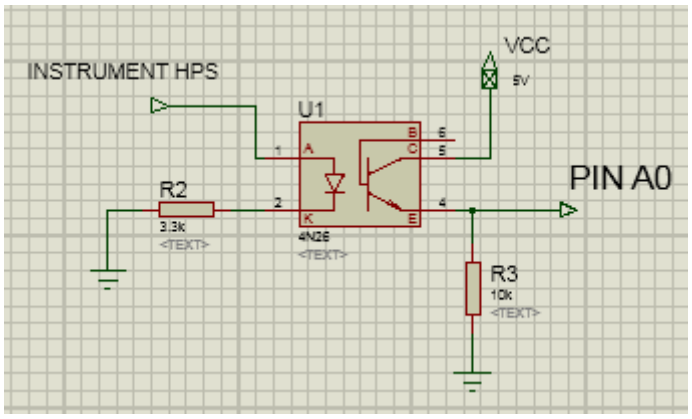
Rangkaian catu daya memberikan suplai tegangan pada alat pengendali rangkaian ini mendapatkan sumber tegangan sebesar 220 VAC, kemudian tegangan tersebut diturunkan menjadi 5 VAC dan akan dirubah menjadi tegangan 5 VDC oleh dioda bridge. Keluaran daridioda ini kemudian masuk ke kapasitor yang fungsinya adalah untuk menstabilkan tegangan, adapun kapasitor 100nF berfungsi untuk membuang nois pada tegangan.



Gambar 3.4 Rangkaian Catu Daya

3.2.2 Perancangan Rangkaian *input* HPS

Rangkaian ini berfungsi untuk menerima sinyal dari instrumen *high pressure switch* berupa tegangan sebesar 24 vdc, pada rangkaian ini terdapat sebuah *optocoupler* untuk menahan tegangan masuk berlebih yang akan merusak mikrokontroller yang kemudian dihubungkan dengan pin input pada arduino untuk diproses dan dijadikan pengaman sekaligus sebagai sensor jika terjadi tegangan *refrigerant* terlalu tinggi.

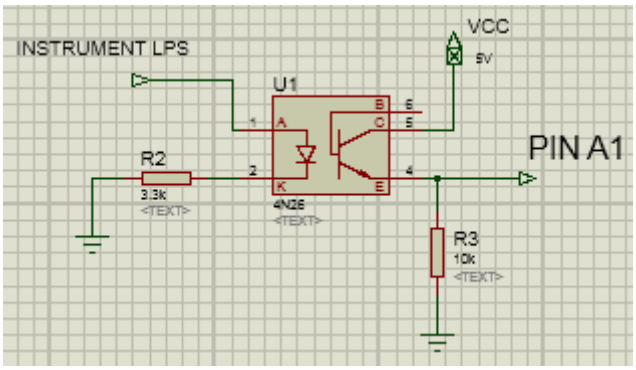


Gambar 3.5 Rangkaian *Input* HPS

3.2.3 Perancangan Rangkaian *input* LPS

Sama seperti rangkaian HPS rangkaian ini memiliki *optocoupler* sebagai penganman dari tegangan berlebih yang akan merusak

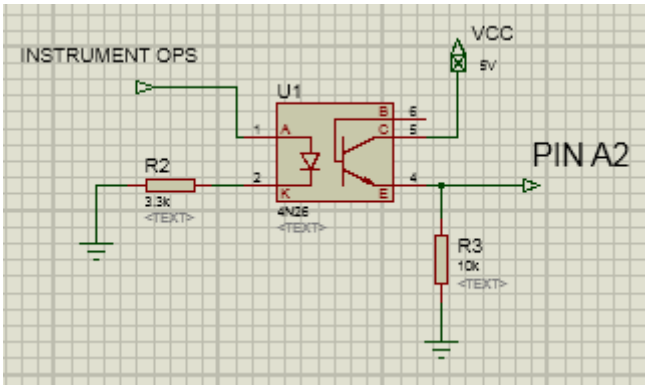
mikrokontroller, akan tetapi penggunaan atau fungsinya saja yang berbeda pada rangkaian ini fungsinya digunakan sebagai pendeteksi untuk menentukan nilai tekanan terendah pada *refrigerant*.



Gambar 3.6 Rangkaian *Input* LPS

3.2.4 Perancangan Rangkaian *input* OPS

Pada perancangan ini masukan tegangan pada instrumen OPS dihubungkan dengan pin input arduino yang bertujuan untuk menentukan sekaligus mendeteksi jika perbandingan antara tekanan oli dan tekanan pada refrigerant tidak sama atau berbanding lurus.

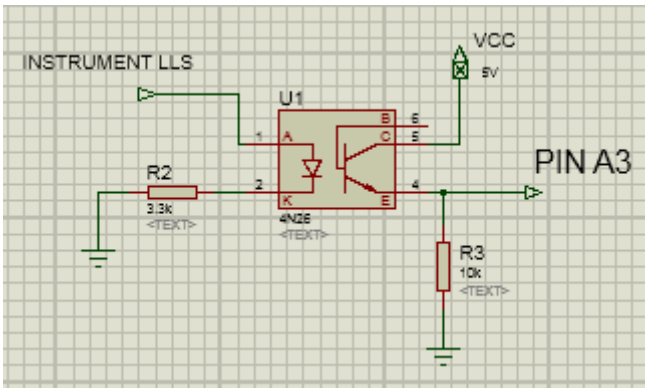


Gambar 3.7 Rangkaian OPS

Pada rangkaian diatas instrumen OPS dihubungkan dengan *optocoupler* dan dihubungkan dengan arduino untuk selanjutnya diproses dan dan difungsikan.

3.2.5 Perancangan Rangkaian *input* LLS

Pada perancangan ini pemanfaatan input digunakan untuk menentukan sekaligus pendeteksi jika *volume* air tidak sesuai dengan kebutuhan banyaknya air pada *chiller*.

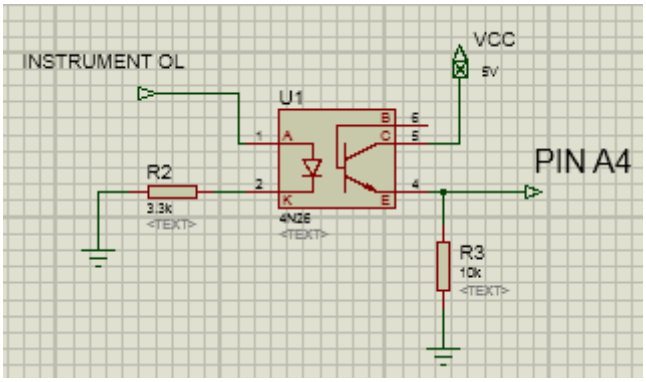


Gambar 3.8 Rangkaian LLS

Instrumen LLS dihubungkan dengan pin A3 yang telah disesuaikan dengan program yang telah dibuat pada arduino sesuai dengan fungsi dari tujuan utama pembuatan rangkaian ini.

3.2.6 Perancangan Rangkaian *input* OL

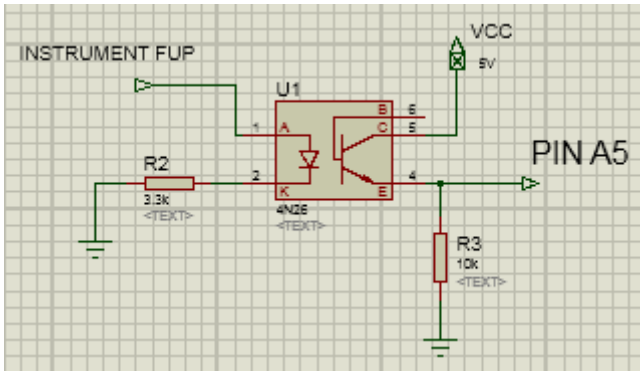
secara keseluruhan rangkaian ini sama dengan semua rangkaian *input* lainnya akan tetapi yang mebedakan adalah pin *input* yang digunakan tidak sama dengan pin input lainnya, dan rangkaian ini berfungsi untuk mendeteksi sekaligus pengaman jika terjadi *over load* pada setiap motor dc yang digunakan.



Gambar 3.9 Rangkaian OL

3.2.7 Perancangan Rangkaian *input* FUP

Pada rangkaian input ini berfungsi untuk mendeteksi jika terjadi pembekuan pada *refrigerant* yang akan menyebabkan penyumbatan pada *evaporator* yang dilalui oleh *refrigerant* tersebut yang jika dibiarkan akan menyebabkan kebocoran pada *evaporator*, komponen yang digunakan pada rangkaian ini tampak seperti gambar berikut.



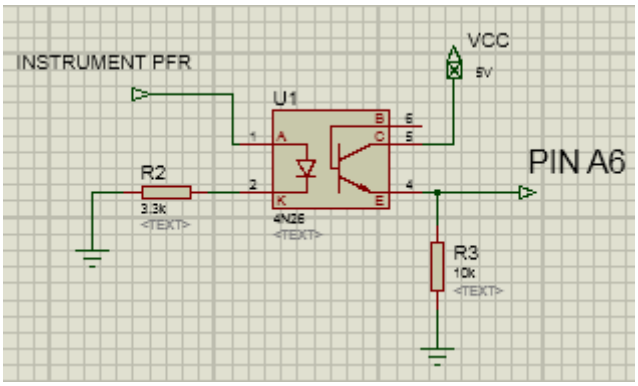
Gambar 3.10 Rangkaian FUP

Instrumen dihubungkan dengan pin *input* A5 dengan menggunakan pengaman optocoupler agar arduino tidak mendapatkan tegangan berlebih yang akan merusak arduino tersebut, pin *input* tersebut akan membaca nilai

yang berupa tegangan yang dihasilkan dari instrumen tersebut untuk dapat diproses oleh arduino.

3.2.8 Perancangan Rangkaian *input* PFR

Tegangan sebesar 24vdc akan melewati *instrument* PFR yang akan dihubungkan ke pin input arduino untuk proteksi jika terjadi tegangan jatuh pada motor yang digunakan sehingga dapat mengamankan semua motor DC yang digunakan.



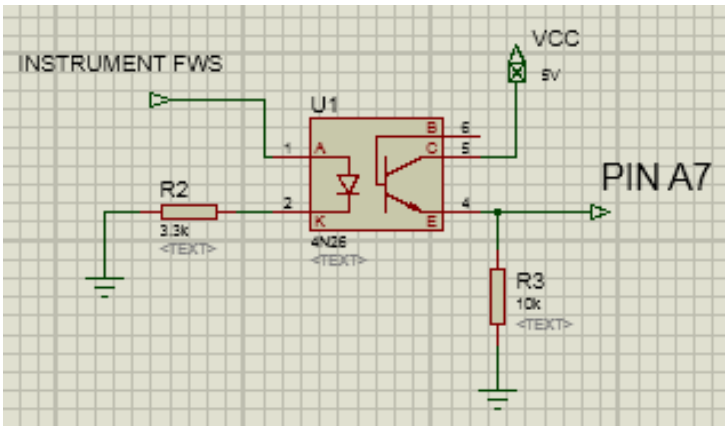
Gambar 3.11 Rangkaian PFR

Jika terjadi tegangan jatuh maka arduino akan memproses agar tegangan jatuh yang masuk pada motor tersebut dapat dihentikan sehingga tegangan tersebut tidak akan memberikan pengaruh *negative* pada motor DC yang digunakan.

3.2.9 Perancangan Rangkaian *input* FWS

Rangkaian ini berfungsi untuk mendeteksi air pada *colling tower* jika air tersebut berada pada titik terendah yag sudah di atur sebelumnya yangkemudian akan di baca oleh arduino sehingga dapat memproteksi dan menampilkannya pada LCD. Nilai yang diberikan instrumen ini akan dibaca

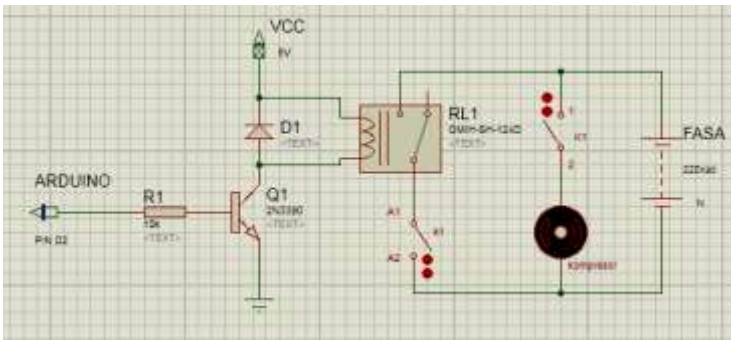
pada pin A7 sehingga arduino dapat memproses hasil dari nilai tersebut sebagai pamarameter untuk mengamankan *chiller* agar tidak terjadi gangguan yang dapat merusak *chiller* tersebut.



Gambar 3.12 Rangkaian FWS

3.2.10 Perancangan Rangkaian *Output* Kompresor

Keluaran dari pin D2 arduino masuk ke resistor yang telah terhubung dengan *base* pada transistor, kemudian akan membuat relay bekerja untuk menghubungkan kontaktor magnet dengan kompresor. Untuk lebih jelasnya dapat dilihat pada gambar betikut.

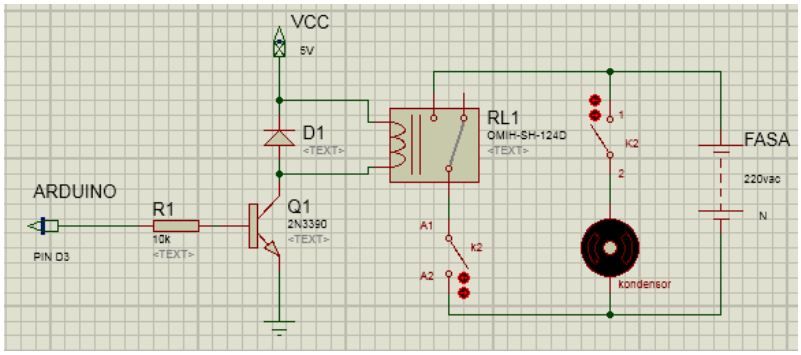


Gambar 3.13 Rangkaian *Output* Kompresor

Relay akan mendapat tegangan dari pin arduino yang selanjutnya tegangan tersebut akan dihubungkan dengan kontaktor magnet yang telah terhubung dengan kompresor, pengontrolan dengan menggunakan kontaktor magnet akan sangat membantu untuk menunjang keberhasilan proses pengontrolan yang akan dilakukan atau dijalankan.

3.2.11 Perancangan Rangkaian *Output* Kondensor

Cara kerja rangkaian *output* kondensor yaitu ketika arduino memberikan nilai *high* pin D3 maka tegangan akan masuk ke resistor yang telah terhubung dengan *base* pada transistor, kemudian akan membuat relay bekerja untuk menghubungkan kontaktor magnet dengan konensor. Untuk lebih jelasnya dapat dilihat pada gambar betikut.

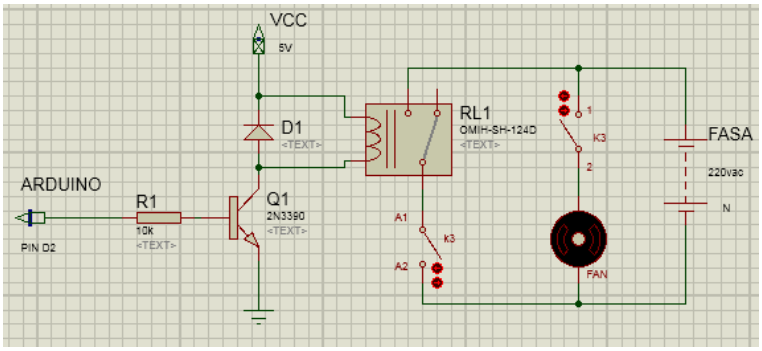


Gambar 3.14 Rangkaian *Output* Kondensor

Output yang dihasilkan arduino berupa tegangan 5v tegangan tersebut akan menghubungkan koil pada relay sehingga relay tersebut dapat membuat kontaktor magnet bekerja untuk menghidupkan kondensor yang telah dihubungkan dengan pin kontaktor magnet

3.2.12 Perancangan Rangkaian *FAN*

Cara kerja rangkaian yaitu ketika arduino memberikan nilai *high* pin D4 maka tegangan akan masuk ke resistor yang telah terhubung dengan *base* pada transistor, kemudian akan membuat relay bekerja untuk menghubungkan kontaktor magnet dengan *FAN*. Untuk lebih jelasnya dapat dilihat pada gambar betikut.



Gambar 3.15 Rangkaian *Output FAN*

Fan akan bekerja sesuai dengan perintah yang dibuat pada arduino dan rangkaian diatas dirancang untuk dapat menyesuaikan dengan perintah tersebut.

3.2.13 Perancangan Rangkaian LCD

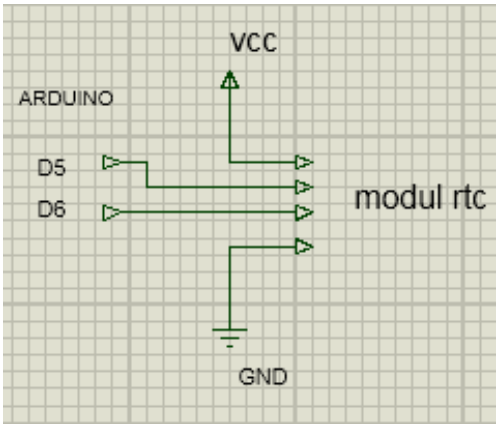
Fungsi rangkaian ini adalah sebagai indikator, *interface*, dan menginput semua data yang dibutuhkan untuk mengontrol *chiller*, pada rangkaian ini LCD dihubungkan dengan TX dan RX pada arduino sebagai pengirim dan penerima data yang menghubungkan komunikasi antara arduino dan LCD, tegangan dan *ground* yang diperlukan oleh LCD berasal dari tegangan dan *ground* yang dimiliki oleh arduino.



Gambar 3.16 Rangkaian LCD

3.2.14 Perancangan Rangkaian RTC

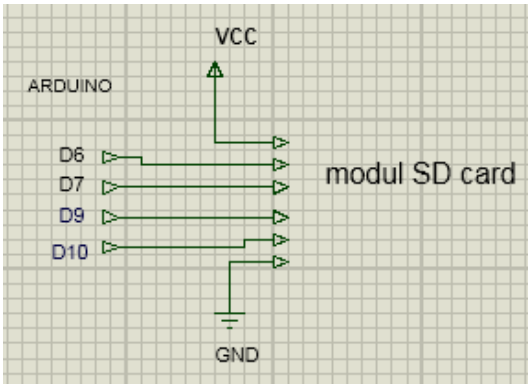
Perancang ini bertujuan untuk menyimpan data berdasarkan waktu dan tanggal ketika terjadi maslah pada *chiller*. Modul RTC dihubungkan dengan arduino untuk menentukan waktu dan tanggal serta mengolah data untuk dilihat pada waktu yag berbeda. Pin pada RTC yang dihubungkan dengan arduino tersebut menyesuaikan dengan program yang telah dibuat untuk dapat memanfaatkan fungsi dari modul RTC tersebut.



Gambar 3.17 Rangkaian Modul RTC

3.2.15 Perancangan Rangkaian SD *Card*

Tujuan dari perancangan alat ini adalah untuk menyimpan data setiap kali terjadi masalah pada *chiller* untuk dapat dilihat dikemudian hari tanpa ada data yang hilang sehingga pengguna tau kapan dan kenapa msalah tersebut terjadi.



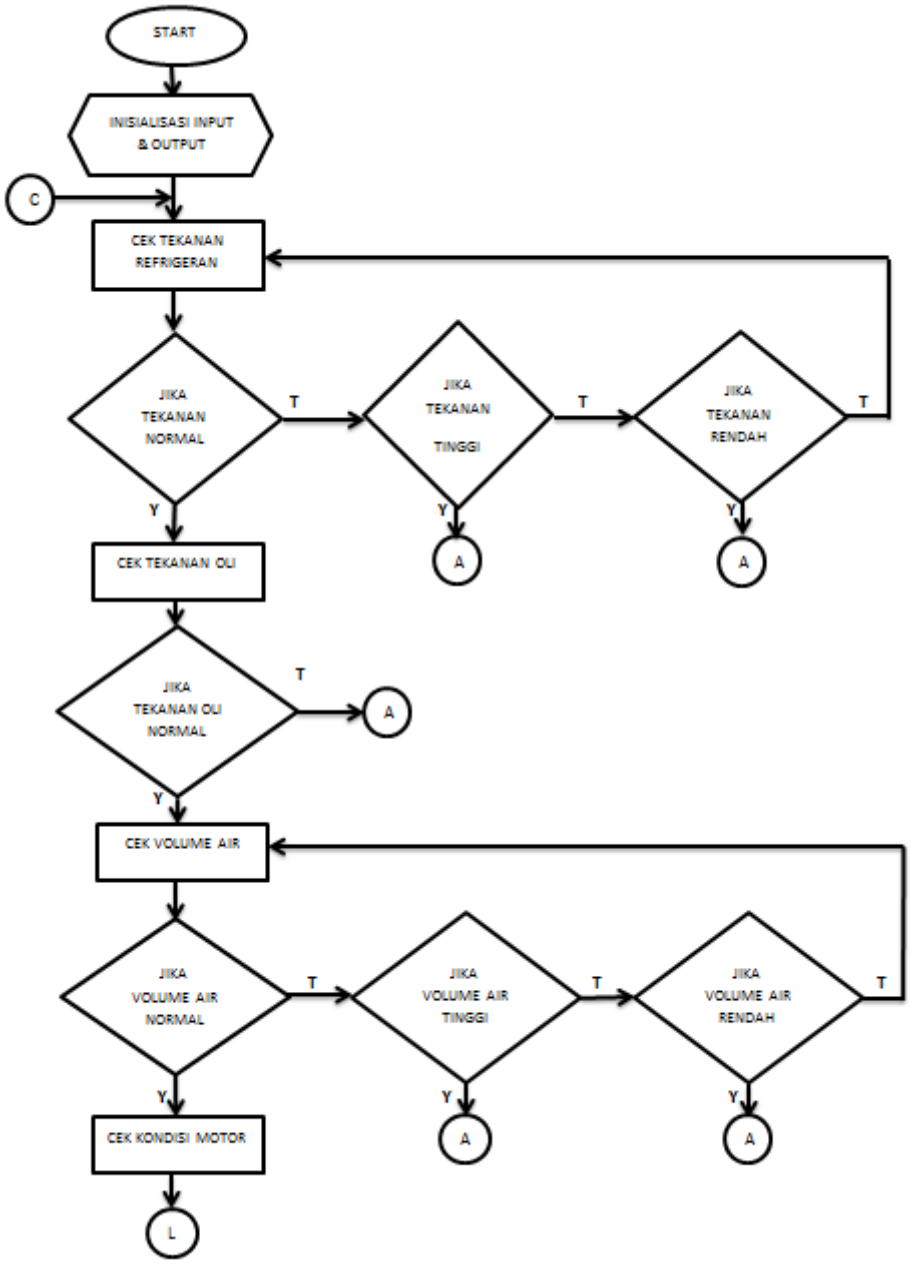
Gambar 3.18 Rangkaian Modul SD *Card*

Perancangan diatas menyesuaikan dengan program yang dibuat sehingga proses penyimpanan data akan berjalan dengan semstinya sesuai dengan perancangan dan manfaat dari modul tersebut.

3.3 Perancangan *Software*

Perancangan perangkat lunak merupakan sebuah proses yang berkelanjutan dari riset yang telah dilakukan sebelumnya dengan melakukan identifikasi serta menghasilkan konsep program pengendalian dan monitoring perangkat keras. Konsep program yang dibuat meliputi program pada mikrokontroller dan program pada LCD *nextion*. Perancangan dari seluruh sistem yang dibuat dapat diterapkan pada diagram alir berikut ini.

3.3.1 Diagram Alir Sistem Pengaman *Chiller*

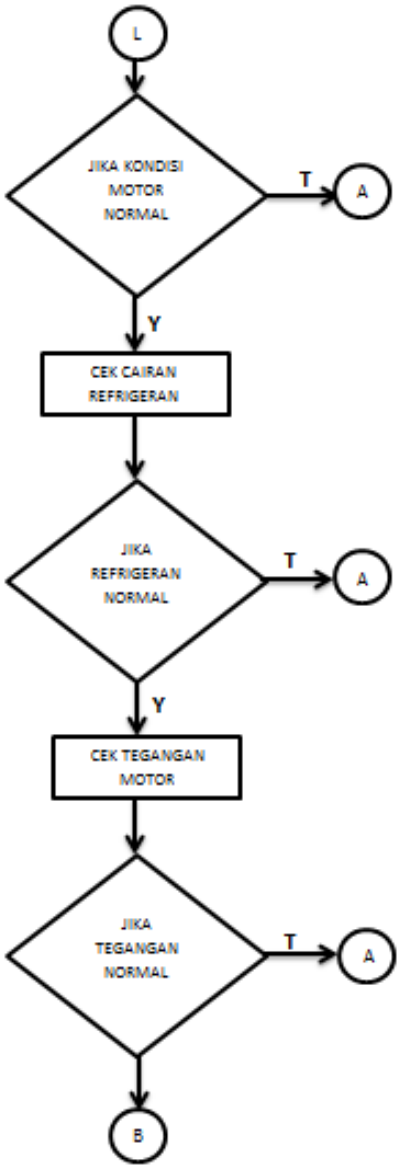


Keterangan :

- C diatas adalah proses dari sistem secara keseluruhan yang di ulang kembali atau sistem looping sehingga sistem kan berjalan secara terus menerus.

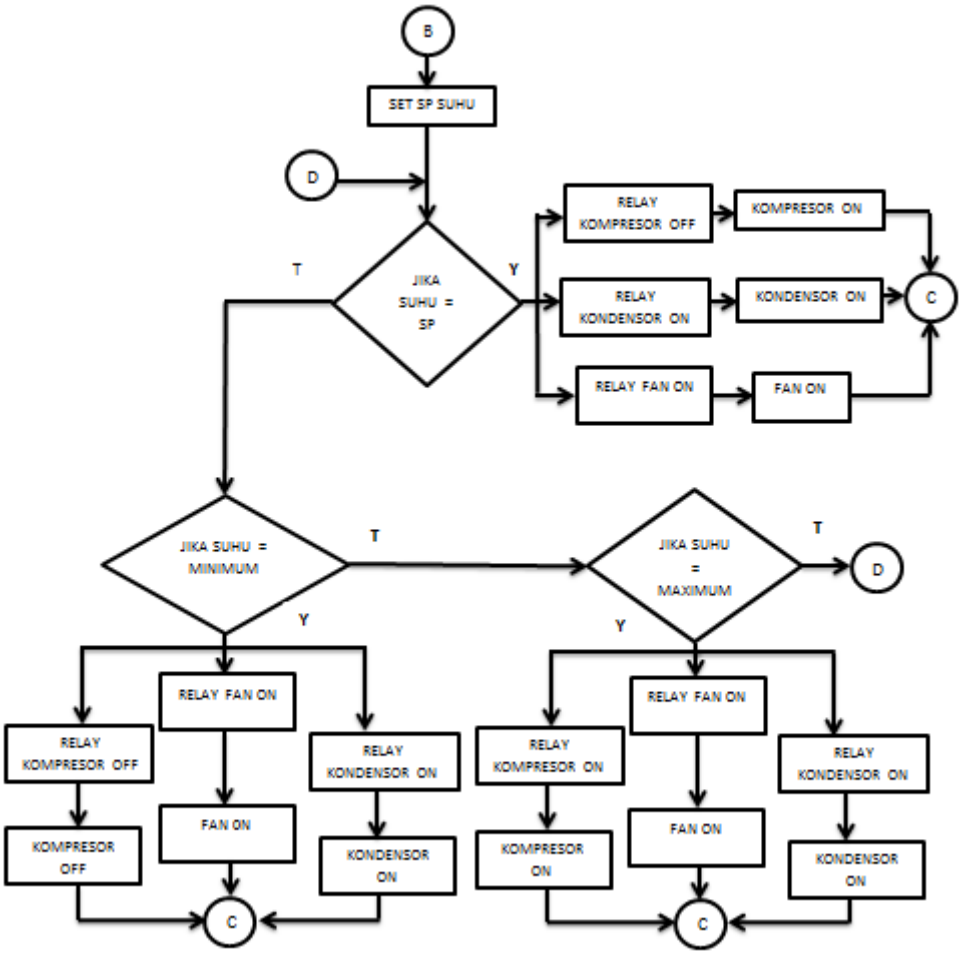
- A adalah indikator *error* dan LCD yang akan menampilkan *error* tersebut.
- L adalah lanjutan dari sistem sebelumnya

3.3.2 Diagram Alir Sistem Pengaman Motor DC



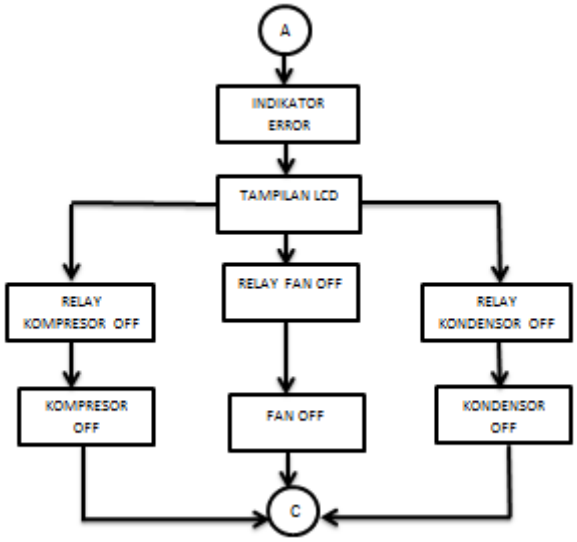
- B melanjutkan proses yang telah terjadi pada sistem diatas.

3.3.3 Diagram Alir Proses Pengontrolan



- D adalah proses yang diulang kembali dari hasil proses pembacaan diatas sehingga sistem akan berjalan secara terus menerus
- C adalah hasil dari proses secara keseluruhan sistem yang telah dibuat

3.3.4 Diagram Alir proses Input



Gambar 3.19 Diagram Alir Keseluruhan

3.3.1 Perancangan Menu Awal

Perancangan ini menampilkan menu awal pada saat alat dijalankan pertama kali, pada menu awal ini terdapat beberapa tampilan seperti suhu dari hasil pembacaan sensor, tampilan *set value* yang sudah di atur atau yang akan diatur, menampilkan smua input yang terhubung pada alat yang dibuat, menampilkan *output* yang terhubung, menampilkan waktu beserta tanggal, dan menampilkan berbagai macam pilihan. Pada tampilan menu awal dapat dilihat pada gambar berikut.



Gambar 3.20 Menu Awal

3.3.2 Perancangan Menu Pilihan

Menu pilihan ini akan muncul ketika pengguna menekan menu pada tampilan awal, pada menu pilihan ini terdapat beberapa pilihan seperti *run info*, *operatio*, dan *trouble history*. Menu pilihan dapat dilihat pada gambar berikut.



Gambar 3.21 Menu Pilihan

3.3.3 Perancangan *Run Info*

Pada *run info* menampilkan semua kondisi *input* yang terhubung dengan *mikrokontroller*, untuk dapat melanjutkan pada proses berikutnya maka semua inputan harus pada kondisi siap dan tidak memiliki masalah atau *error* pada *chiller*. Pada *run info* dapat dilihat pada gambar berikut.



Gambar 3.22 *Run Info*

Data diatas akan berubah sewaktu-waktu sesuai dengan hasil pembacaan pada input yang terhubung dengan arduino, maka data tersebut akan menampilkan hasil yang sesuai dengan apa yang terbaca pada arduino.

3.3.4 Perancangan Menu Awal

Pada menu *operation* terdapat pengaturan untuk memasukan nilai *max*, *min*, dan *set value*. Nilai-nilai ini akan menjadi target pencapaian suhu pada perangkat yang dibuat, ketika nilai sudah di atur maka pengguna dapat menekan tombol *start* pada menu awal untuk menjalankan perangkat yang telah dibuat. Untuk masuk pada menu ini pengguna harus menekan *operation* pada pilihan menu. Adapun tampilan pada menu *operation* dapat dilihat pada gambar berikut.



Gambar 3.23 *Operation*

3.3.5 Perancangan *Trouble History*

Pada perancangan ini terdapat informasi tentang masalah atau *error* yang terjadi sehingga menyebabkan perangkat berhenti berfungsi, keterangan yang muncul akan sesuai dengan kondisi *chiller* yang bermasalah berikut dengan tanggal dan kejadian pada saat *chiller* tersebut

bermasalah. Pada gambar di bawah menunjukan contoh masalah yang terjadi pada *chiller*.



Gambar 3.24 *Trouble History*

Jika tidak terjadi permasalahan pada proses kerja chiller maka didalam menu ini tidak akan menunjukan apa-apa atau kosong pada kotak yang terdapat pada menu tersebut.