

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

Pada bab III ini membahas bagaimana perancangan dan pembuatan simulasi pelindung hujan otomatis yang dikendalikan mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS) dimulai dengan pembuatan perangkat keras (*Hardware*) serta perancangan perangkat lunak (*Software*). Dalam pembuatan perangkat keras yaitu berupa perancangan catu daya, perancangan sistem minimum ATmega 8, perancangan sistem mekanik penggerak atap pelindung, perancangan sistem minimum *max 232* dan perancangan komunikasi *serial* sehingga *modem Wavecom* mengirim *report Short Message Service* (SMS) kepada *user* sebagai informasi kondisi.

Sementara pada perangkat lunak yaitu perancangan program bahasa C atau bahasa yang digunakan pada Mikrokontroler ATmega 8 sebagai pusat kendali dari semua perangkat keras dimana program bahasa C menentukan setiap jalan dari perangkat keras yang kemudian dapat diinformasikan kepada user melalui pesan *Short Message Service* (SMS).

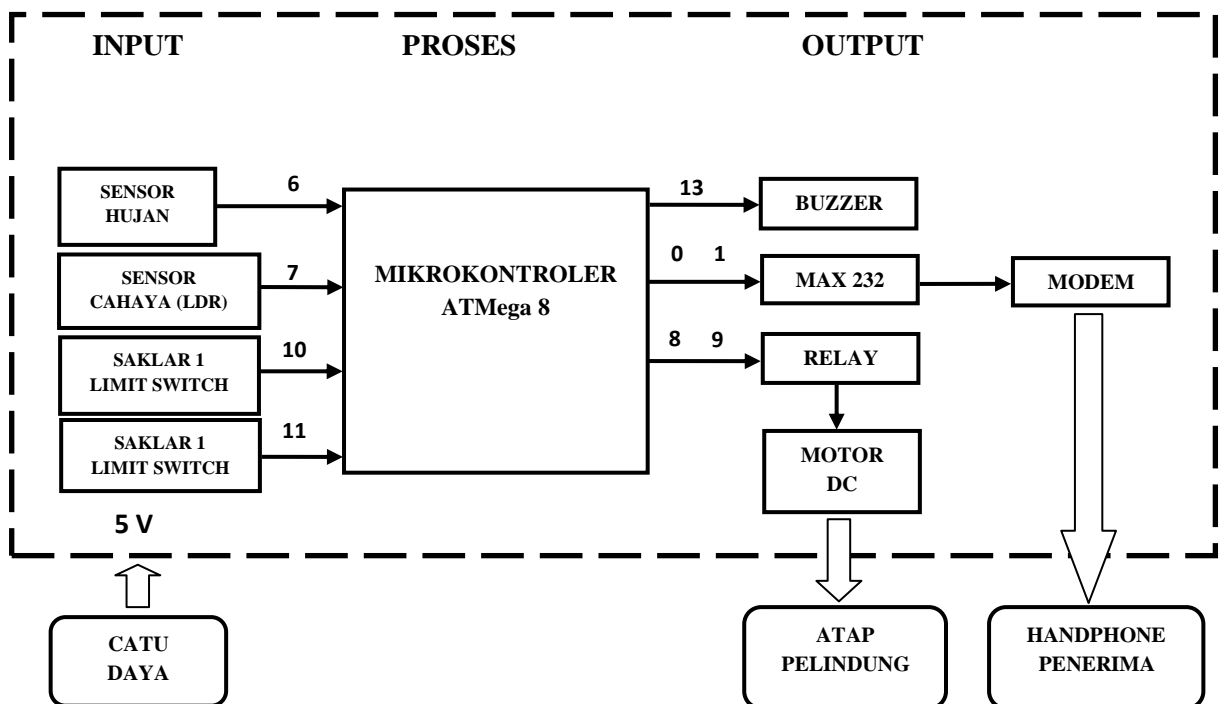
3.1 PERANCANGAN SISTEM

Perancangan simulasi alat pelindung hujan otomatis yang dikendalikan mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS) yang akan di fungsikan sebagai pelindung benda dari hujan yang kemudian di informasikan kepada user sebagai penerima pesan *Short Message Service* (SMS).

Pada saat sensor hujan mendeteksi adanya turun hujan ataupun sensor cahaya mendeteksi perubahan cahaya dari semula terang menjadi gelap maka sensor akan mengirimkan data kepada Mikrokontroler ATmega 8 sebagai pusat kendali dan mengolah data masukan yang kemudian memberikan perintah pada *output*. *Output* pada alat pelindung ini terdiri dari *buzzer*, *motor DC* sebagai penggerak mekanik atap pelindung, *IC Max 232* sebagai *interface* dengan *modem* dan *modem* sebagai pengirim pesan kepada *user*. Sehingga, pada saat seseorang dengan kegiatan yang padat setiap harinya diluar rumah menjadi tidak terlalu khawatir akan benda-benda yang sedang di jemur apabila turun hujan karena sedang bekerja dan tidak memungkinkan untuk pulang. Dan pada saat menjemur benda-benda tersebut

kemudian turun hujan, maka benda-benda tersebut dapat terlindungi oleh atap pelindung yang menutup secara otomatis. Demikian juga apabila cuaca mendung atau malam hari maka atap pelindung akan menutup dan menginformasikan kepada *user* dalam bentuk *Short Message Service* (SMS) yang sedang berada di luar rumah bahwa atap pelindung telah menutup.

Pada gambar 3.1 merupakan blok diagram dari Perancangan simulasi alat pelindung hujan otomatis yang dikendalikan Mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS)



Gambar 3.1 Blok Diagram Perancangan Simulasi Alat Pelindung Hujan Otomatis Yang Dikendalikan Mikrokontroler Atmega 8 Dengan *Report Short Message Service* (SMS)

Alat ini bekerja pada saat sensor hujan maupun sensor cahaya mendeteksi perubahan cuaca dan intensitas cahaya di sekitar sensor tersebut terpasang, yang kemudian diproses oleh Mikrokontroler ATmega 8 selanjutnya di terima oleh *buzzer*, *relay*, *motor DC* dan *modem* yang kemudian mengirimkan pemberitahuan berupa pesan kepada *user*. Berdasarkan blok diagram yang ditunjukkan pada

gambar 3.1 diketahui bahwa terdapat tiga rangkaian pada simulasi alat pelindung hujan otomatis tersebut, antara lain rangkaian *input*, rangkaian proses dan rangkaian *output*, dimana pada rangkaian input terdiri dari sensor hujan, sensor cahaya, saklar 1, saklar 2. Sedangkan pada rangkaian proses terdiri dari Mikrokontroler ATmega 8 dengan sistem minimum dan *reset*, serta pada rangkaian output terdiri dari *buzzer*, *relay*, *motor DC*, atap pelindung, *IC Max 232* dan *modem Wavecom* sebagai pengirim *report* kepada *user*.

Pada rangkaian *input* dapat dilihat pada gambar blok diagram dimana terdapat dua sensor yaitu sensor hujan *Light Dependent Resistor* (LDR) dan sensor hujan yaitu tembaga yang dipasang pada papan PCB. Serta saklar *Limit Switch* sebanyak dua buah yang masing-masing sebagai saklar *close* dan saklar *open*. Kemudian pada rangkaian proses adalah Mikrokontroler ATmega 8 yang difungsikan sebagai penerima kondisi dari rangkaian *input*, dimana pada rangkaian proses, Mikrokontroler ATmega 8 harus memiliki sistem minimum agar dapat bekerja. Serta pada rangkaian *output* terdapat *buzzer* yang difungsikan sebagai tanda peringatan saat terjadi perubahan kondisi cuaca hujan, lalu ada *motor DC* yang berfungsi sebagai penggerak dari atap pelindung saat terjadi hujan, dan *modem* yang ter-*interface* dengan komunikasi serial *Max 232* sebagai pengirim pesan kepada *user* bahwa telah terjadi perubahan kondisi.

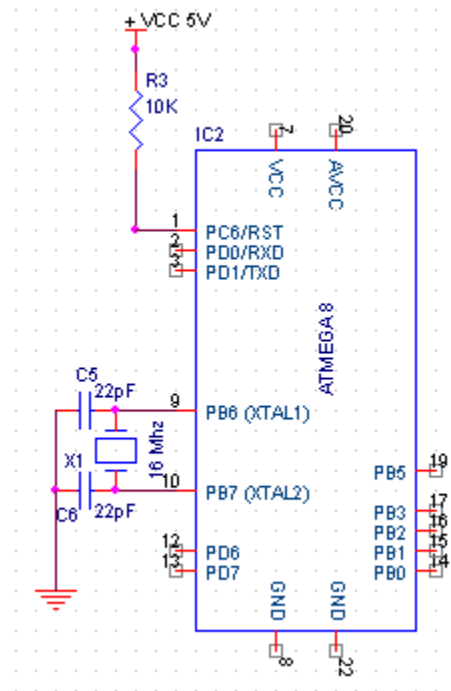
3.2 PERANCANGAN DAN PEMBUATAN *HARDWARE*

Perancangan *hardware* dalam simulasi alat pelindung hujan otomatis yang dikendalikan Mikrokontroler ATmega 8 dengan *repot Short Message Servie* (SMS) meliputi pembuatan rangkaian secara *schematic* dan perhitungan dari komponen yang akan digunakan.

3.2.1 Perancangan Sistem Minimum ATmega 8

Sistem minimum merupakan rangkaian minimal yang harus ada dalam sebuah mikrokontroler agar dapat bekerja, rangkaian minimum ATmega terdiri dari dua rangkaian yaitu rangkaian *oscillator* dan rangkaian *reset* sehingga apabila dalam sebuah mikrokontroler tidak terdapat sistem

minimum maka mikrokontroler tidak dapat bekerja. Rangkaian sistem minimum mikrokontroler ATmega 8 ditunjukkan pada gambar 3.2.



Gambar 3.2 Rangkaian Sistem Minimum Atmega 8

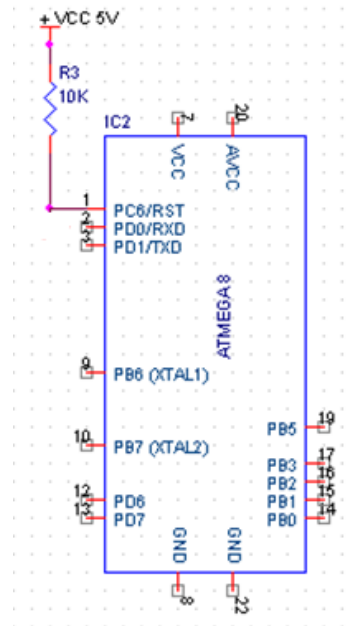
Dari jumlah pin yang ada pada mikrokontroler ATmega 8, tidak semua digunakan pada tugas akhir ini. Pin ATmega 8 yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3.1 Pin ATmega 8 yang digunakan

No	pin	fungsi
1	pin reset	sebagai <i>reset</i> ATmega 8
2	pin Xtal	sebagai pin Xtal (<i>rangkaian osilator</i>)
3	pin aref, avcc	sebagai tegangan kerja ADC
4	pin digital 0, 1	sebagai pin keluaran Rx dan Tx (<i>modem</i>)
5	pin digital 6	sebagai masukan dari sensor hujan
6	pin digital 7	sebagai masukan dari sensor cahaya LDR
7	pin digital 8, 9	sebagai pin keluaran relay dan penggerak motor DC
8	pin digital 10, 11	sebagai pin masukan saklar limit switch
9	pin digital 13	sebagai pin keluaran <i>buzzer</i>

3.2.2 Rangkaian Reset

Rangkaian *reset* pada mikrokontroler ATmega 8 berfungsi untuk mengembalikan ke kondisi awal apabila pada saat pengisian program terjadi kesalahan. Mikrokontroler melakukan *reset* jika mendapatkan ATmega 8 *logic low* (0) sebanyak dua kali siklus mesin. Rangkaian *reset* yang terhubung dengan mikrokontroler ditunjukkan pada gambar 3.3.



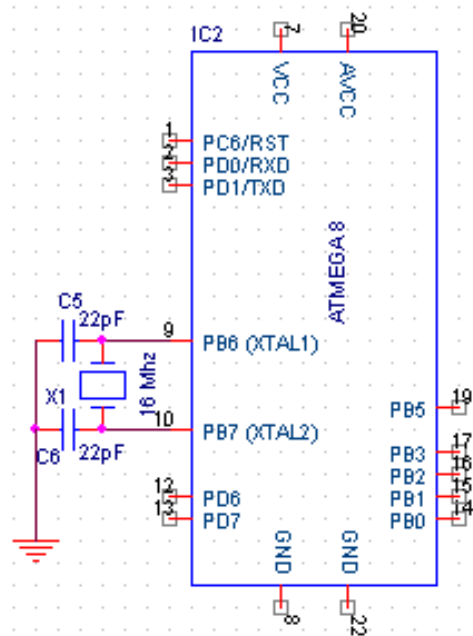
Gambar 3.3 Rangkaian Reset

Dalam pembuatan ini, frekuensi *oscillator* yang digunakan sebesar 16 MHz. Rangkaian reset pada mikrokontroler terdiri dari komponen push button, dimana rangkaian ini terhubung dengan PIN reset yang telah tersedia dalam mikrokontroler ATmega 8 dengan tegangan sekitar 4,5 – 5,5 volt.

3.2.3 Rangkaian Oscillator

Rangkaian *oscillator* adalah rangkaian yang pada simulasi alat pelindung hujan yang dikendalikan Mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS) ini digunakan untuk dapat menentukan frekuensi pada *clock* dalam mikrokontroler yang digunakan terhubung ke *Pin Digital* 9 dan *Pin Digital* 10. Nilai-nilai Kristal pada mikrokontroler berbeda-beda dan sesuai dengan pemakaiannya. Kristal yang digunakan

pada tugas akhir ini menggunakan kristal 16 MHz, dan untuk dapat mempermudah perhitungan. Rangkaian *oscillator* ditunjukkan pada gambar 3.4.



Gambar 3.4 Rangkaian *Oscillator*

Dengan melihat gambar 3.4 rangkaian *oscillator* dapat diketahui rangkaian *oscillator* terdiri dari dua kapasitor keramik dengan nilai sebesar 22 pF dan crystal 16 MHz.

Untuk dapat melakukan satu siklus dalam mengeksekusi program pada mikrokontroler dapat dihitung dengan rumus :

$$T = \frac{12}{\text{frekuensi crystal}}$$

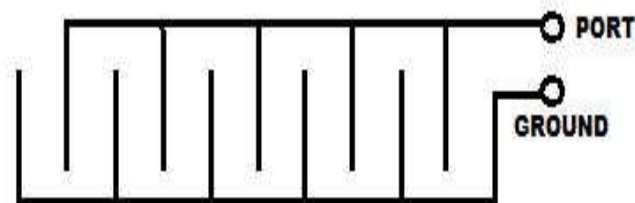
$$T = \frac{1 \times 12}{16 \text{ MHz}}$$

$$T = 0,75 \mu\text{s}$$

Semakin besar nilai Xtail maka kecepatan clock yang digunakan untuk keepatan data akan semakin cepat. Jadi waktu yang dibutuhkan untuk mengetahui kecepatan mengeksekusi program pada simulasi alat pelindung hujan otomatis yang dikendalikan mikrokontroler ATmega 8 dengan *report Short Message Service (SMS)*.

3.2.4 Perancangan Mikrokontroler ATmega 8 dengan Sensor Hujan

Sensor hujan pada tugas akhir ini digunakan untuk mendeteksi adanya hujan yang turun disekitar sensor. Sensor hujan yang digunakan dalam rangkaian ini terbuat dari potongan PCB yang dipotong sesuai dengan kebutuhan dan terdapat dua buah jalur tembaga yang terhubung ke *port Pin Digital 6* pada mikrokontroler ATmega 8, dan jalur yang satu lagi terhubung dengan *ground*. Pada saat hujan turun dan sensor hujan mendeteksi hujan maka kedua jalur akan basah sehingga antara jalur yang terhubung ke *Pin Digital 6* pada mikrokontroler ATmega 8 dengan yang terhubung ke *ground* akan terhubung singkat. Jadi pada saat sensor hujan dalam kondisi basah maka *port Pin Digital 6* mendapat logika 0. Gambar 3.5 bentuk sensor hujan.

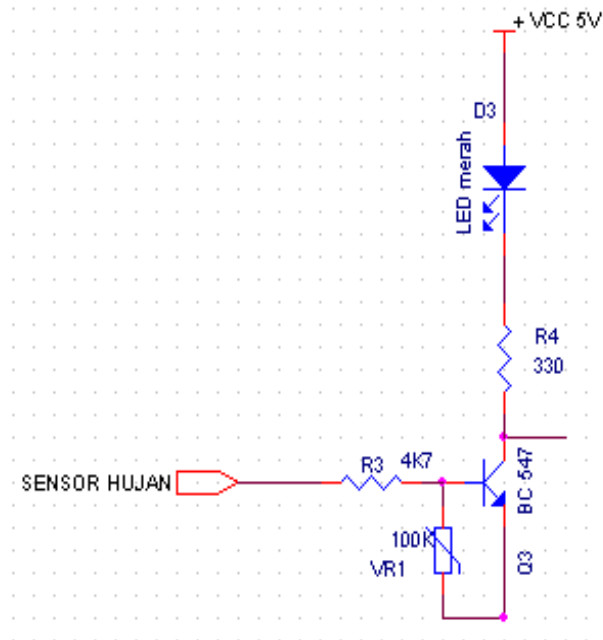


Gambar 3.5 Bentuk Sensor Hujan

Tabel 3.2 Tabel Pin Sensor Hujan

No	pin	fungsi
1	Port	sebagai jalur ke mikrokontroler ATmega 8
2	GND	sebagai ground dari sensor hujan

Berdasarkan dari data Tabel 3.2 dijelaskan bahwa sensor hujan terbentuk dari dua buah jalur yang masing-masing jalur terbuat dari lempeng tembaga yang disusun sesuai dengan keinginan. Pada gambar 3.6 ditunjukkan rangkaian sensor hujan.

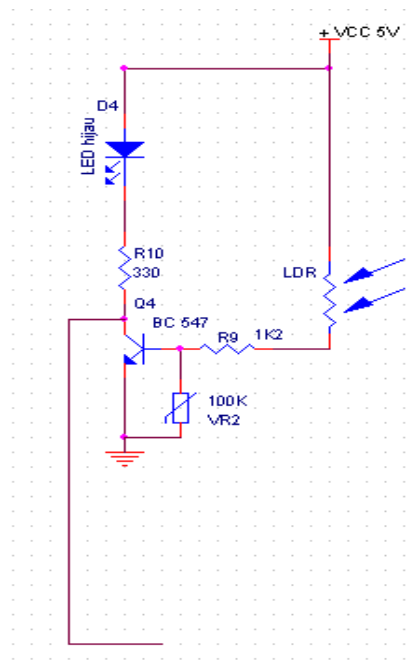


Gambar 3.6 Rangkaian Sensor Hujan

3.2.5 Perancangan Mikrokontroler ATmega 8 dengan Sensor Cahaya *Light Dependent Resistor (LDR)*

Sensor cahaya *Light Dependent Resistor (LDR)* pada tugas akhir ini berfungsi sebagai pendeteksi perubahan cahaya yang terjadi di sekitar pemasangan sensor. Sensor cahaya ini terhubung dengan *port pin digital 7* pada mikrokontroler ATmega 8. Sensor cahaya ini terbuat dari *cadmium sulfide* yang merupakan bahan semi konduktor dimana resistansinya dapat berubah-ubah sesuai dengan banyaknya cahaya (sinar) yang diperoleh oleh sensor cahaya tersebut. Bentuk dan simbol sensor cahaya *Light Dependent Resistor (LDR)* ditunjukkan pada gambar 2.14.

Sensor cahaya *Light Dependent Resistor (LDR)* memiliki dua kaki, dimana masing-masing kaki fungsi sebagai penghubung jalur ke *port pin digital 7* pada mikrokontroler, dan kaki yang satu terhubung dengan *ground*. Gambar rangkaian sensor cahaya ditunjukkan pada gambar 3.7.

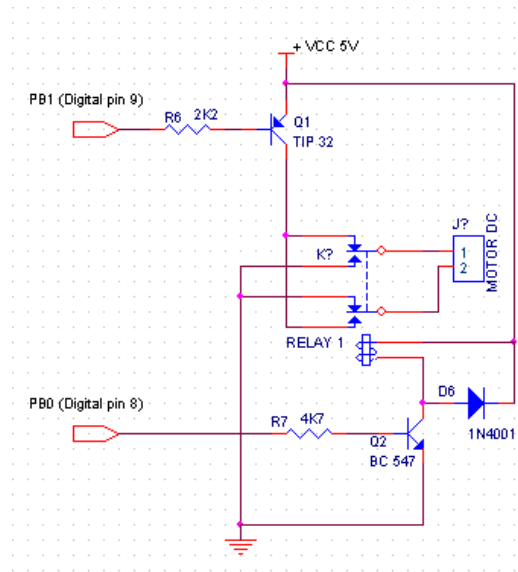


Gambar 3.7 Rangkaian Sensor Cahaya

Pada simulasi pelindung hujan otomatis yang di kendalikan mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS) ini *Light Dependent Resistor* (LDR) berfungsi sebagai *input* yang mendeteksi perubahan terang gelapnya cahaya. Apabila intensitas cahaya kecil atau dalam keadaan gelap, maka output yang akan di terima oleh mikrokontroler berlogika 1, dan tegangan keluaran *Light Dependent Resistor* (LDR) lebih dari 2,4 volt sampai dengan 5 volt. Sementara pada keadaan terang maka *output Light Dependent Resistor* (LDR) akan berlogika 0. Tegangan yang dihasilkan pada output *Light Dependent Resistor* (LDR) adalah kurang dari 2,4 volt.

3.2.6 Perancangan Mikrokontroler ATmega 8 dengan Motor DC Sebagai Penggerak Atap

Pada perancangan *motor DC* sebagai penggerak atap pelindung, digunakan sebuah *motor DC* yang berfungsi sebagai penggerak atap. Dalam mengontrol gerak *motor DC* di pasang sebuah *relay*. Rangkain *motor DC* sebagai penggerak atap ditunjukkan pada gambar 3.8.

Gambar 3.8 Rangkaian *Motor DC*

Pada gambar 3.8 bisa dilihat rangkaian *motor DC* terhubung ke *port Pin Digital* 8, dan 9 pada mikrokontroler ATmega 8. Terdapat dua buah *driver motor DC*, yaitu driver motor putar kanan dan kiri. Prinsip kerja dari *driver motor* cukup sederhana, dimana *input driver motor* diperoleh dari sistem minimum ATmega 8 pada *port Pin Digital* 8, dan 9. Pada saat *input 1* berlogika 1 (*Pin Digital 9* berlogika 1) dan *input 2* berlogika 0 (*Pin Digital 9* berlogika 0) maka *transistor Q1* akan ON, sehingga mengaktifkan kontak *relay*. Selanjutnya setelah kontak *relay* aktif maka motor akan mendapatkan tegangan sebesar 5 volt sehingga *motor DC* akan berputar. Arah putaran motor pada kondisi ini adalah ke kanan, dimana kondisi sensor hujan dalam keadaan basah atau sensor cahaya dalam keadaan gelap. Untuk dapat menghentikan putaran dari motor, maka digunakan *limit switch*. Apabila *limit switch* telah tertekan maka *motor DC* akan berhenti berputar.

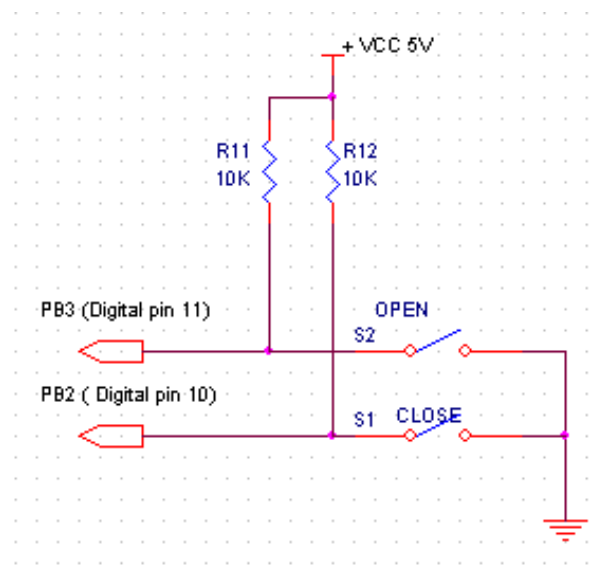
Untuk *motor DC* berputar ke kiri, maka sebaliknya kondisi sensor hujan harus dalam keadaan kering dan sensor cahaya dalam keadaan terang (*Pin Digital 8* dan 9 berlogika 0). Maka keluaran dari mikrokontroler adalah *Pin Digital 9* berlogika 0 dan *Pin Digital 8* berlogika 1, sehingga Q2 ON. Setelah Q2 aktif maka akan mengaktifkan *relay*. Dimana pada saat *relay ON* maka kontak *relay* akan saling terhubung yang semula terhubung ke

ground menjadi terhubung ke *VCC* sehingga *motor DC* mendapatkan tegangan catu sebesar 5 volt dan motor kan berputar ke arah kiri. Perputaran motor akan berhenti setelah atap menyentuh saklar *limit switch*.

Pada rangkaian *driver motor* terdapat sebuah dioda, dimana fungsi dari dioda ini untuk mengamankan *transistor* dari lonjakan tegangan yang besar yang berasal dari lilitan *relay* saat kondisi *relay off*.

3.2.7 Perancangan Saklar *Limit Switch*

Perancangan saklar *limit switch* difungsikan sebagai pengontrol dari laju atap yang digerakan oleh *motor DC*. Saklar *limit switch* yang digunakan dalam simulasi pelindung hujan otomatis yang di kendalikan mikrokontroler ATmega 8 dengan report *Short Message Service (SMS)* ini sebanyak dua buah. Masing-masing fungsi dari saklar ini sebagai saklar *open* dan saklar *close*. Rangkaian saklar *limit switch* di tunjukan pada gambar 3.9.

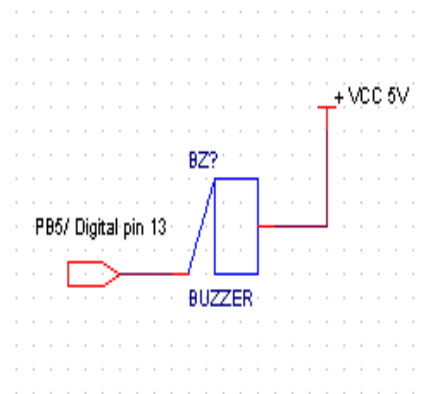


Gambar 3.9 Rangkaian Saklar *Limit Switch*

Dari gambar 3.9 dapat dilihat bahwa rangkaian saklar terhubung ke *port Pin Digital* 10 dan 11 pada mikrokontroler ATmega 8 dengan tegangan masukan sebesar 5 volt. Saklar *limit switch* bekerja pada saat atap pelindung bergerak menutup atau membuka sesuai dengan kondisi yang sedang terjadi.

3.2.8 Perancangan Mikrokontroler ATmega 8 dengan *Buzzer*

Pada simulasi alat pelindung hujan otomatis digunakan *buzzer*. *Buzzer* memiliki dua buah kaki dimana kaki yang pertama terhubung ke port *Pin Digital* 13 pada mikrokontroler ATmega 8, sementara kaki kedua terhubung ke catu daya dengan tegangan sebesar 5 volt. Rangkaian *buzzer* di tunjukan pada gambar 3.10.



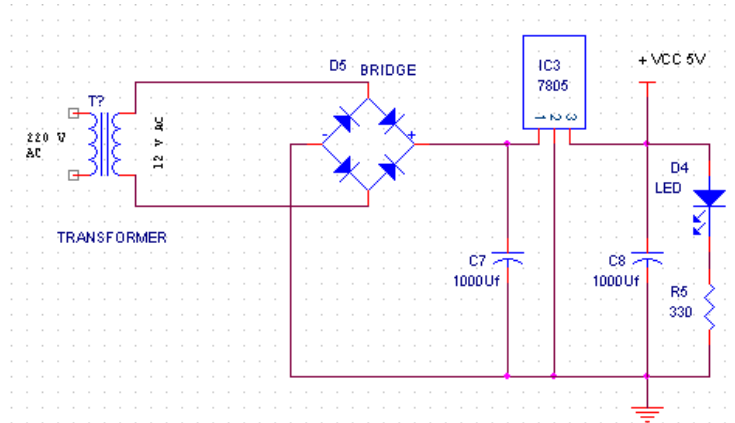
Gambar 3.10 Rangkaian *Buzzer*

Pada saat sensor hujan mendeteksi adanya hujan, maka *buzzer* akan berbunyi sebagai *alarm* penanda bahwa telah terjadi hujan.

3.2.9 Perancangan Mikrokontroler ATmega 8 dan RS232 dengan *Modem Serial Wavecom*

Pada perancangan mikrokontroler ATmega 8 dan RS 232 dengan *modem serial wavecom*. Pada rangkaian ini, mikrokontroler dan RS 232 berfungsi sebagai pengatur komunikasi *serial* antara mikrokontroler ATmega 8 dengan *modem serial wavecom*. Komunikasi serial ini nantinya untuk mengirimkan sebuah *report* ke *handphone*. Pada saat mikrokontroler ATmega 8 mendapat masukan dari sensor selanjutnya data terima akan dikirimkan ke *modem serial wavecom*. Untuk dapat mengirim data ke *modem serial wavecom* diperlukan sebuah *IC konverter* yaitu *IC MAX 232* yang berfungsi mengkonversi tegangan antara mikrokontroler ATmega 8 dengan *modem serial wavecom*. Tegangan dari mikrokontroler ATmega 8 adalah sebesar 4,5 volt - 5,5 volt yang sesuai dengan *datasheet* dan tegangan dari *modem serial wavecom* adalah sebesar 6 volt – 12 volt. Oleh sebab itu,

pada rangkaian agar alat dapat bekerja. Adapun tegangan yang dibutuhkan untuk men-catu keseluruhan rangkaian adalah sebesar 5 volt. Tegangan keluaran 5 volt ini diperoleh dari penggunaan *regulator 7805*. Perancangan dari catu daya ini terdiri dari beberapa bagian yaitu blok *transformator* yang mempunyai fungsi untuk mengubah dari tegangan *Alternating Current (AC)* menjadi tegangan *Dirrect Current (DC)*, kemudian blok *rectifier* yang berfungsi sebagai penyearah arus, serta blok *filter* dengan fungsi sebagai penyaring tegangan yang naik turun agar dengan adanya *filter* dapat mengurangi tegangan *ripple*, dan blok *regulator* yang berfungsi menghasilkan tegangan 5 volt yang dibutuhkan oleh sistem kerja mikrokontroler ATmega 8. Blok catu daya di tunjukan pada gambar 3.12.



Gambar 3.12 Blok Rangkaian Catu Daya

Berdasarkan gambar 3.12 catu daya memberikan tegangan keluaran sebesar 5 volt yang difungsikan untuk dapat men-sulpay seluruh rangkaian alat. Rangkaian catu daya terdiri dari beberapa bagian, yaitu :

3.2.10.1 *Transformator*

Transformator atau yang lebih dikenal dengan *trafo* merupakan komponen elektronika yang berfungsi untuk menaikkan dan menurunkan tegangan sesuai dengan kebutuhan dari rangkaian. *Trafo* yang digunakan pada rangkaian adalah *trafo step down* yang digunakan untuk menurunkan tegangan *Alternating Current (AC)* sebesar 220 volt

tegangan PLN menjadi tegangan keluaran *Dirrect Current* (DC) sebesar 12 volt.

3.2.10.2 *Rectifier*

Rectifier atau penyearah berfungsi sebagai komponen yang mengubah tegangan *Alternating Current* (AC) menjadi tegangan *Dirrect Current* (DC). Pada umumnya penyearah gelombang dibedakan menjadi dua, yaitu penyearah setengah gelombang dan penyearah gelombang penuh.

3.2.10.3 *Filter*

Filter merupakan komponen elektronika yang berfungsi untuk mengurangi tegangan *ripple* atau tegangan naik turun akibat arus yang tidak stabil pada bagian output di penyearah gelombang penuh, dimana semakin besar nilai *filter* yang digunakan maka nilai tegangan *ripple* akan semakin menurun. Dalam rangkaian catu daya ini menggunakan *filter* kapasitor. Dalam hal ini besarnya *presentase* (%) dari tegangan *ripple* yang terdapat pada suatu sistem merupakan perbandingan antara tegangan *ripple* dan tegangan *Dirrect Current* (D) dari suatu sistem.

3.2.10.4 *IC Regulator*

Pada catu daya ini digunakan *IC Regulator* yang berfungsi memberikan tegangan keluaran yang sesuai dan stabil. *IC Regulator* yang digunakan adalah LM 7805 yang berfungsi mengubah tegangan 12 volt menjadi 5 volt.

3.2.11 Rangkaian Sistem Keseluruhan

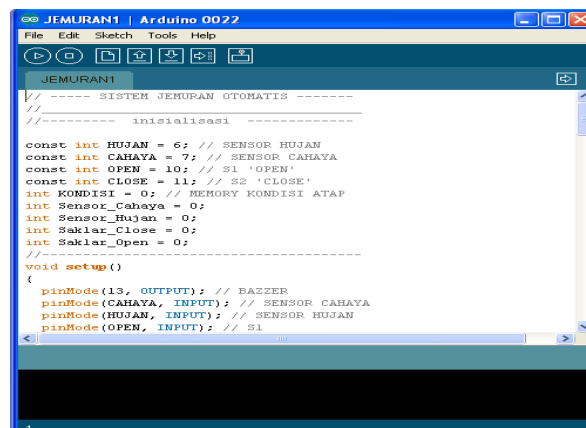
Berikut adalah perancangan rangkaian sistem keseluruhan pada pembuatan tugas akhir simulasi alat pelindung hujan otomatis yang di kendalikan mikrokontroler ATmega 8 dengan *report Short Message Service* (SMS) yang ditunjukkan pada gambar 3.13.

3.3 PERANCANGAN DAN PEMBUATAN PROGRAM

Dalam simulasi alat pelindung hujan otomatis yang di kendalikan mikrokontroler AT Mega 8 dengan *report Short Message Service* (SMS) meliputi pembuatan *flowchart* program, perancangan program sensor hujan, perancangan program sensor cahaya *Light Dependent Resistor* (LDR), perancangan program *buzzer*, perancangan program saklar *limit Switch*, perancangan program *driver motor DC*, dan perancangan program *AT Command* sebagai perintah untuk mengirimkan report berupa *Short Message Service* (SMS). Dalam simulasi alat pelindung hujan ini, *software* yang digunakan adalah *software arduino 0022* dengan bahasa pemrograman yang digunakan adalah bahasa pemrograman C serta *software In System Programming* (ISP) yang difungsikan untuk men-*download* program ke dalam mikrokontroler.

3.3.1 Software Arduino 0022 sebagai Program Mikrokontroler Atmega 8

Arduino 0022 merupakan alat bantu yang dipergunakan untuk memprogram bahasa C. adanya software arduino ini mempermudah kerja untuk dapat mengetahui sistem kerja pendeteksi hujan. Tampilan dari arduino di tunjukan pada gambar 3.14.



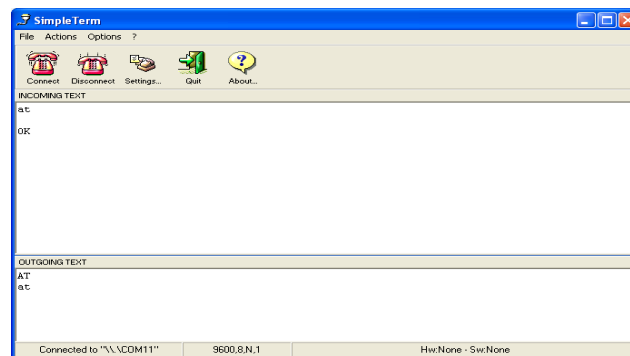
```
JEMURAN1 | Arduino 0022
File Edit Sketch Tools Help
JEMURAN1
// ---- SISTEM JEMURAN OTOMATIS ----
//
//----- inisialisasi -----
const int HUJAN = 6; // SENSOR HUJAN
const int CAHAYA = 7; // SENSOR CAHAYA
const int OPEN = 10; // S1 'OPEN'
const int CLOSE = 11; // S2 'CLOSE'
int KONDISI = 0; // MEMORY KONDISI ATAP
int Sensor_Cahaya = 0;
int Sensor_Hujan = 0;
int Saklar_Close = 0;
int Saklar_Open = 0;
//-----
void setup()
{
  pinMode(13, OUTPUT); // BAZZER
  pinMode(CAHAYA, INPUT); // SENSOR CAHAYA
  pinMode(HUJAN, INPUT); // SENSOR HUJAN
  pinMode(OPEN, INPUT); // S1
```

Gambar 3.14 Tampilan Arduino 0022 *Compiler*

Hasil dari pembuatan program pada *software* arduino yang sudah di *compiler* yaitu **.hex*. Karena tidak ada pengaturan-pengaturan untuk mengganti mikrokontroler yang akan digunakan, maka dapat langsung dituliskan program untuk dapat menjalankannya.

3.3.2 AT Command

Program *AT Command* digunakan untuk menginstruksikan atau memberi perintah pengiriman *Short Message Service* (SMS) dari sebuah *modem wavecom* ke sebuah handphone penerima untuk dapat mengetahui perubahan kondisi yang terjadi. Mikrokontroler memberikan perintah *AT Command* melalui hubungan *serial*. Contoh perintah untuk mengirimkan sebuah *Short Message Service* (SMS) dengan menggunakan perintah *AT+CMGS* (“nomor tujuan”). *AT Command* dapat melakukan pengujian dengan menggunakan aplikasi *hyperterminal*. Tampilan *hyperterminal* ditunjukkan pada gambar 3.15.

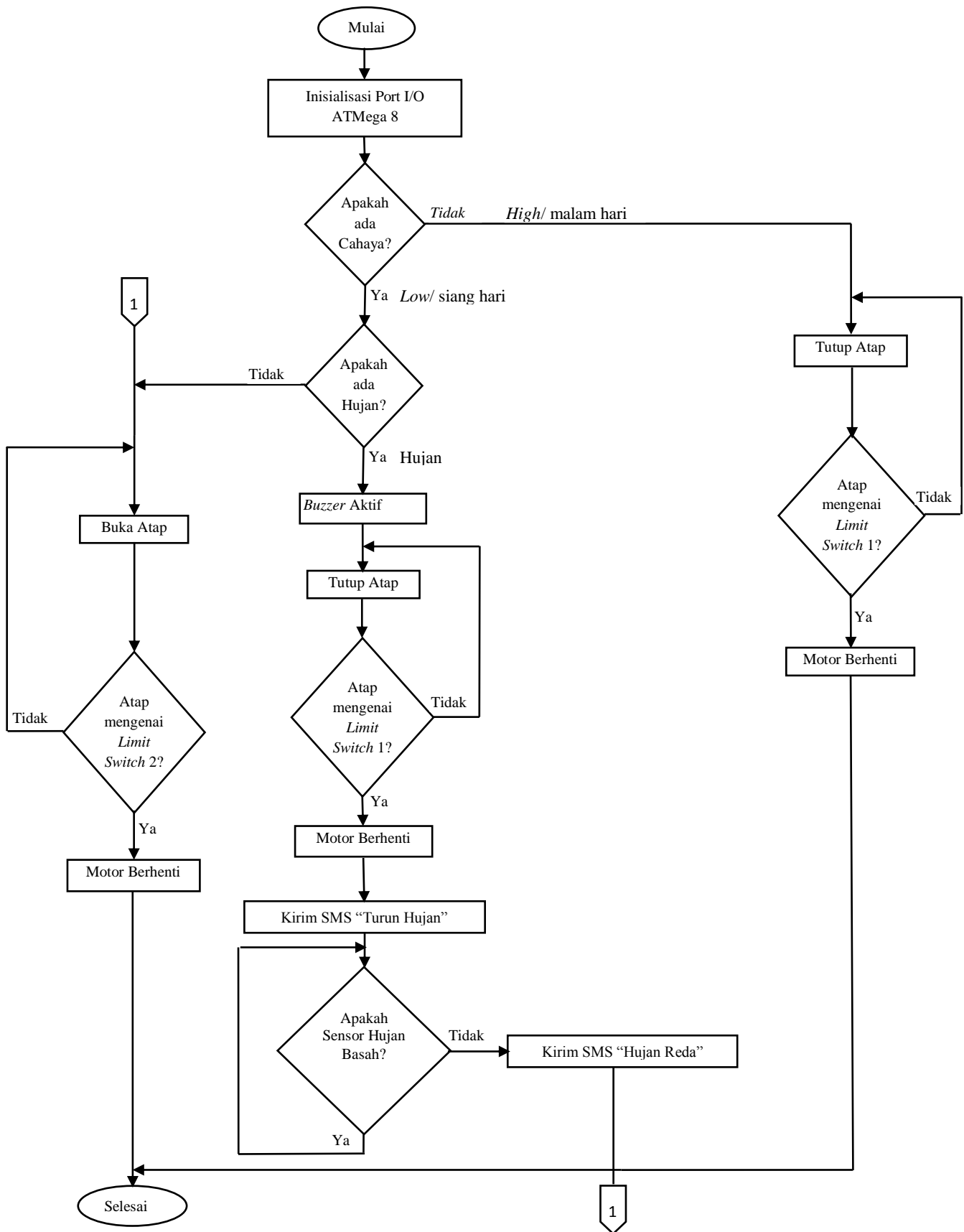


Gambar 3.15 Tampilan *Simple Therm*

untuk dapat menjalankan *hyperterminal*, buka *hyperterminal* kemudian pasangkan *modem* pada perangkat *computer* atau laptop yang akan digunakan untuk melihat hasil pengujian. Ketikkan kata “AT” pada *Simple Therm*, apabila keluar kata “OK”, maka pengujian berhasil dan *modem* dapat digunakan.

3.3.3 Alur dan Penjelasan Program

Program yang terdapat pada simulasi alat pelindung hujan otomatis yang di kendalikan mikrokontroler AT Mega 8 dengan *report Short Message Servie* (SMS), dibagi menjadi beberapa bagian. Terdapat *flowchart* pembuatan program yang menjelaskan *step by step* dari kerja program dari awal sampai akhir. Dengan melihat *flowchart* langkah kerja dapat diketahui instruksi kerja dari sebuah sistem yang nantinya akan dijalankan. *Flowchart* rancangan pembuatan program di tunjukkan pada gambar 3.16.



Gambar 3.16 Flowchart Program

Pada gambar 3.16 *flowchart* program bisa dijelaskan dimana proses-proses atau instruksi kerja dari sebuah program yang akan dijalankan. Pembuatan program yang ditunjukkan pada gambar 3.16 dalam sebuah *flowchart* bekerja ketika program memulai dengan melakukan inisialisasi *port input* dan *port output* yang terhubung pada mikrokontroler ATmega 8. Dalam inisialisasi *port port* yang terhubung dengan mikrokontroler ATmega 8 dijelaskan apakah *port port* tersebut telah terhubung dan siap untuk digunakan. Pada program sensor cahaya *Light Dependent Resistor* (LDR) dan sensor hujan terdapat dua kondisi yang dipahami oleh sensor yaitu logika 0 dan logika 1. Sensor cahaya *Light Dependent Resistor* (LDR) dalam logika 0 menjelaskan sensor cahaya dalam kondisi *standby*, dan kondisi logika 1 sensor cahaya dalam kondisi bekerja. Sementara pada sensor hujan, kondisi logika 0 menjelaskan bahwa sensor dalam kondisi bekerja sementara pada kondisi 1 sensor hujan dalam kondisi *standby*.

Pada sensor cahaya, logika 0 menjelaskan bahwa sensor dalam kondisi *standby* atau dalam kondisi malam hari sehingga atap pelindung dalam kondisi tertutup, sementara untuk logika 1 pada sensor cahaya menjelaskan bahwa sensor dalam kondisi bekerja atau kondisi siang hari, itu mengindikasikan proses selanjutnya dalam pengamanan benda masuk pada proses sensor hujan. Apabila kondisi sensor hujan pada logika 0 atau dalam kondisi hujan turun dan diikuti oleh bunyi *alarm* dari *buzzer*, dan proses berikutnya adalah menutup atap pelindung sampai pada proses atap pelindung berhenti menutup atap saat menyentuh saklar *limit switch* dan motor berhenti. Sampai pada proses motor berhenti kemudian *modem wavecom* secara otomatis akan mengirimkan pesan berupa *Short Message Service* (SMS) pada *handphone*.

a. Program inisialisasi *port I/O* ATmega 8

Program untuk melakukan inisialisasi sistem port I/O ATmega 8 yang digunakan untuk dapat dikendalikan oleh mikrokontroler. Berikut *listing* program inisialisasi I/O yang digunakan :

```
// ----- SISTEM JEMURAN OTOMATIS -----  
// _____  
//----- inisialisasi -----  
  
const int HUJAN = 6; // SENSOR HUJAN  
const int CAHAYA = 7; // SENSOR CAHAYA  
const int OPEN = 10; // S1 'OPEN'  
const int CLOSE = 11; // S2 'CLOSE'  
int KONDISI = 0; // MEMORY KONDISI ATAP  
int Sensor_Cahaya = 0;  
int Sensor_Hujan = 0;  
int Saklar_Close = 0;  
int Saklar_Open = 0;  
//-----  
void setup()  
{  
  pinMode(13, OUTPUT); // BUZZER  
  pinMode(CAHAYA, INPUT); // SENSOR CAHAYA  
  pinMode(HUJAN, INPUT); // SENSOR HUJAN  
  pinMode(OPEN, INPUT); // S1  
  pinMode(CLOSE, INPUT); // S2  
  pinMode(HUJAN, INPUT); // SENSOR HUJAN  
  pinMode(8, OUTPUT); // OUTPUT RELAY  
  pinMode(9, OUTPUT); // OUTPUT SUPPLY TEGANGAN  
  digitalWrite(8, LOW); // RELAY  
  digitalWrite(9, HIGH); // SUPPLY MOTOR  
  //digitalWrite(OPEN, HIGH); // OPEN  
  //digitalWrite(CLOSE, HIGH); // CLOSE  
  digitalWrite(13, HIGH); // BUZZER  
  Serial.begin(9600);  
}
```

Pada *listing* program diatas merupakan program inisialisasi I/O yang digunakan pada simulasi alat pelindung hujan. Pada tugas akhir ini komponen-komponen yang digunakan pada I/O adalah sensor cahaya *Light Dependent Resistor* (LDR), sensor cahaya, *buzzer*, *limit switch*, *motor DC* dan komunikasi *serial*. Program untuk mengaktifkan sensor hujan yaitu `cons int HUJAN = 6;` dan untuk mengaktifkan sensor cahaya yaitu `cont int CAHAYA = 7;` yang mengindikasikan bahwa pada sensor hujan dan sensor cahaya menggunakan *pin digital* 6 dan *pin digital* 7. Program untuk mengaktifkan saklar *limit switch 1* yaitu `cons int OPEN = 10;` dan untuk mengaktifkan saklar *limit switch 2* yaitu `cons int CLOSE = 11;` dimana fungsi dari saklar *limit switch* ini adalah sebagai pengontrol gerak dari atap pelindung. Untuk mengaktifkan kondisi *Short Message Service* (SMS) yaitu `int KONDISI = 0` ;. Perintah program untuk komunikasi serial yaitu `serial begin (9600)` ;. Yang dimaksud 9600 adalah kecepatan *transfer* data dengan kecepatan 9600 *bit/sec*.

b. Penjelasan program sensor

```
void setup()
{
  pinMode(CAHAYA, INPUT); // SENSOR CAHAYA
  pinMode(HUJAN, INPUT); // SENSOR HUJAN
```

Pada program diatas merupakan program dari sensor cahaya *Light Dependent Resistor* (LDR) dan sensor hujan. Dalam alat pelindung ini menggunakan dua buah sensor yaitu sensor cahaya *Light Dependent Resistor* (LDR) dan sensor hujan, dimana data yang diperoleh akan diproses oleh mikrokontroler.

```
Sensor_Cahaya = digitalRead(CAHAYA);
```

```
Saklar_Close = digitalRead(CLOSE);  
Saklar_Open = digitalRead(OPEN);  
Sensor_Hujan = digitalRead(HUJAN);
```

Program diatas merupakan kondisi dari masing-masing pin yang difungsikan sebagai *input*, data hasil dari *input* berupa *High* (mendapat tegangan) atau *Low* (menjadi *ground*).

c. Penjelasan program *output*

```
pinMode(13, OUTPUT); // BUZZER  
pinMode(8, OUTPUT); // OUTPUT RELAY  
pinMode(9, OUTPUT); // OUTPUT SUPPLY TEGANGAN
```

Pengertian dari program diatas adalah fungsi dari masing-masing *port* sebagai *output* dari keluaran mikrokontroler ATmega 8. Pada *output pinMode* 13 adalah *buzzer* difungsikan sebagai *alarm* dan untuk *output pinMode* 8 difungsikan sebagai *relay* dan 9 difungsikan sebagai keluaran *supply* tegangan.

d. Penjelasan program *buzzer, relay* dan *motor DC*

```
{  
    digitalWrite(13, LOW); // BUZZER NYALA  
    delay (1000);  
    digitalWrite(13, HIGH); // BUZZER MATI  
    while (digitalRead(OPEN) == HIGH)  
    {  
        digitalWrite(8, HIGH); // PERINTAH  
        MENYALAKAN RELAY  
        digitalWrite(9, LOW); // MEMERINTAHKAN  
        MOTOR UNTUK MENUTUP  
    }  
}
```

```
        digitalWrite(8, LOW); // PERINTAH MEMATIKAN
RELAY
        digitalWrite(9, HIGH); // MEMERINTAHKAN
MOTOR UNTUK BERHENTI
```

Pengertian dari program diatas yaitu pada saat kondisi siang hari dan hujan dan *buzzer* akan berbunyi yang kemudian *relay* mendapat tegangan dan memerintahkan *motor DC* untuk bergerak menuju atap pelindung.

e. Penjelasan program komunikasi *serial Short Message Service (SMS)*

```
if (KONDISI == 0)
{
  delay(100);
  digitalWrite(13, LOW);
  delay(100);
  Serial.flush();
  Serial.print("AT+CMGS=");
  Serial.print("085743271913");
  Serial.write(13); // kode ascii <CR>
  Serial.print("PERHATIAN! TURUN HUJAN");
  Serial.write(26); // kode ASCII untuk CTRL-Z
  delay(5000);
  KONDISI = 1;
}
```

Pada program diatas menjelaskan program komunikasi *serial* untuk mengirim *Short Message Service (SMS)* dengan menggunakan perintah *AT Command*. Perintah program *AT Command* untuk mengirimkan *Short Message Service (SMS)* `Serial.print ("AT+CMGS=")` ; dimana untuk mengirimkan *Short Message Service (SMS)* ke nomor tujuan "085743271913" adalah `Serial.print`

("085743271913") ;. Pesan yang akan diterima oleh *handphone* adalah `Serial.print ("PERHATIAN ! TURUN HUJAN") ;` dengan waktu tunda 5000 *ms* atau 5 detik.