

BAB II

DASAR TEORI

2.1 PENGENALAN SISTEM KOMUNIKASI SELULAR

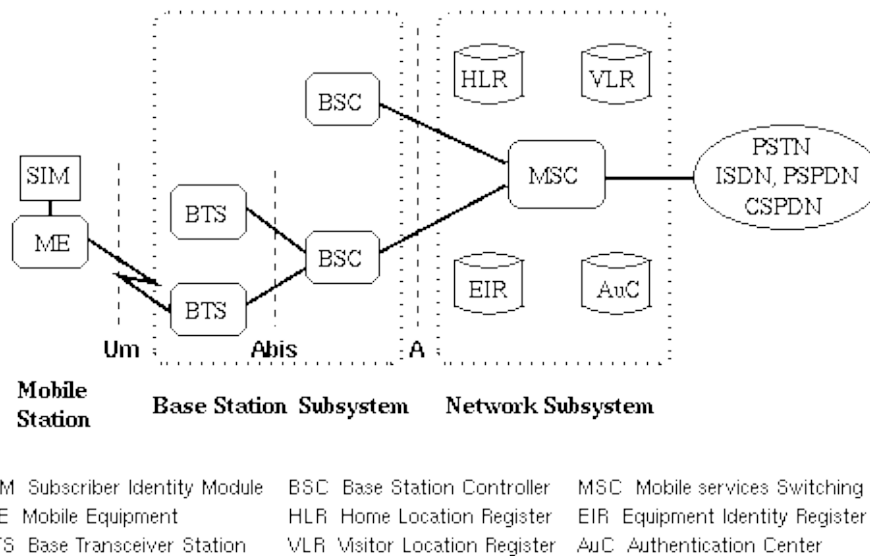
2.1.1 Mengenal Teknologi Selular

Teknologi telekomunikasi yang paling populer dan pesat perkembangannya pada saat ini adalah selular. Pada tahun 1978 teknologi selular masih dalam proses uji coba di Amerika Serikat, namun pada saat ini jutaan orang yang sudah menggunakan piranti telekomunikasi selular seperti *handphone*, PDA dan sebagainya. Selain untuk komunikasi suara, penggunaan jaringan selular telah berkembang ke bentuk komunikasi data seperti video, gambar, animasi dan teks. [1]

Pada dasarnya teknologi selular adalah hasil dari pengembangan dari teknologi pendahulunya yakni teknologi radio yang digabungkan dengan teknologi telepon bisaa (telepon rumah) sehingga hasil kombinasi ini menghasilkan teknologi baru yakni sebuah telepon selular tanpa kabel (bersifat *wireless*) yang penggunaannya sangat mudah dan dapat dibawa kemana-mana dengan syarat berada dalam cakupan *coverage area* sinyal pemancar dari *provider* yang digunakan pengguna. Pada jaringan komunikasi jaringan selular, teknologi ini memiliki sifat *full duplex* yang dimana pengguna teknologi dapat mengirim dan menerima informasi dalam waktu yang sama.

Seiring perkembangannya teknologi selular kian beragam mengikuti kebutuhan para konsumen. Salah satu jenis teknologinya adalah *Global System for Mobile communication* (GSM). *Global System for Mobile* (GSM) merupakan sebuah teknologi komunikasi selular yang bersifat digital. *Global System for Mobile* (GSM) adalah sebuah standar global untuk komunikasi bergerak digital. Teknologi GSM banyak diterapkan pada *mobile communication*, khususnya *handphone*. Teknologi ini memanfaatkan gelombang mikro dan pengirim sinyal yang dibagi berdasarkan waktu, sehingga sinyal informasi yang dikirim akan sampai pada tujuan. GSM adalah nama

sebuah grup standarisasi yang dibentuk di Eropa tahun 1982 untuk menciptakan sebuah standar bersama telepon bergerak selular di Eropa yang beroperasi pada daerah frekuensi 900 MHz. [2]



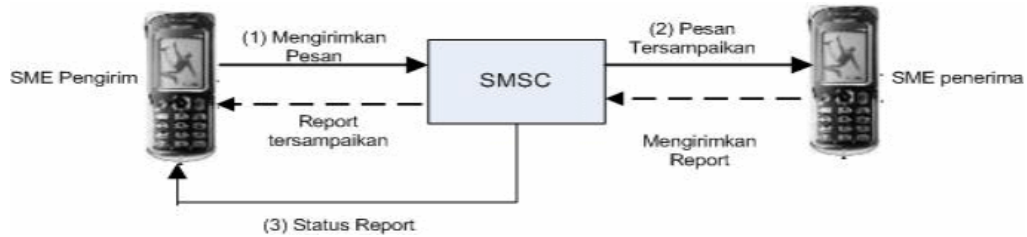
Gambar 2.1 Struktur Jaringan GSM [2]

2.1.2 Short Message Service (SMS)[2]

Short Message Service (SMS) merupakan fitur lain yang dimiliki teknologi GSM. SMS merupakan pesan instan yang dapat dikirimkan pengguna ke pengguna lain kapanpun, dimanapun asalkan pengguna tersebut berada dalam *coverage area provider* yang digunakannya.

Layanan SMS ini merupakan berbentuk seperti surat yang ditulis sangat singkat. Sebuah pesan singkat (SMS) bisaanya sangat terbatas, hanya dapat dikirimkan 160 karakter perhalamannya. Jika pengguna mengirimkan lebih dari 160 karakter maka pengguna akan dikenakan biaya dua kali lipat sesuai dengan tarif yang telah dibuat oleh *provider* tersebut. Namun, ada beberapa operator yang menggunakan pentarifan bukan perhalaman tapi per kilo *byte*.

Pesan-pesan SMS akan dikirim dari pengguna telepon selular ke pusat pesan *Short Message Service Center (SMSC)*, di SMSC ini pesan tersebut akan masuk ke SMSC terlebih dahulu. Seperti yang digambarkan pada gambar 2.2.



Gambar 2.2 Susunan Aliran Pengiriman Pesan

Pada gambar 2.2 dapat disimpulkan ketika *Mobile Station A* mengirimkan sebuah pesan singkat ke *Mobile Station B* maka pesan singkat tersebut akan masuk di SMSC dan SMSC akan memproses untuk melakukan pengiriman melalui HLR dan VLR yang ada di MSC sehingga pesan singkat tersebut sampai pada *Mobile Station B*. Namun, diantara *Mobile Station-Mobile Station* yang beraktivitas ini, ditengah-tengahnya terdapat jaringan yang menghubungkan keduanya sehingga kedua *Mobile Station* tersebut dapat bertukar informasi.

2.1.3 Short Message Entity (SME)

Short Message Entity (SME) adalah elemen yang dapat mengirim atau menerima pesan singkat. SME dapat berupa *software* aplikasi pada *mobile handset*, dapat juga berupa perangkat *facsimile*, perangkat *telex*, *remote internet server*, dll.

Sebuah SME dapat berupa *server* yang terkoneksi dengan SMS center secara langsung atau melalui *gateway*. Dikenal juga *External SME* (ESME) yang memProsentasikan sebuah WAP proxy/server, *Email Gateway* atau *Voice Mail Server*.

2.1.4 Short Message Service Center (SMSC)

Pada dunia telekomunikasi khususnya dalam teknologi GSM, SMSC merupakan sistem yang bertugas untuk mengelola pesan singkat (SMS) dalam jaringan *wireless*. Ketika sebuah pesan singkat dikirim maka akan diproses pada SMSC terlebih dahulu untuk dikirimkan ke tujuan sesuai dengan yang diinginkan pengirim. Jadi ketika SMSC mendeteksi penerima pesan singkat tersebut dalam keadaan di dalam *coverage area* maka SMSC akan langsung mengirimkannya, namun apabila penerima berada di luar *coverage area* maka SMSC akan menyimpan pesan tersebut sementara hingga menunggu si penerima

pesan dalam jangkauan. Pesan yang disimpan oleh SMSC juga memiliki masa kadaluarsa, jika dalam waktu tertentu SMSC tidak mendeteksi bahwa nomor tujuan tidak berada dalam jangkauan maka pesan tersebut akan dihapus secara otomatis.

2.1.5 SMS Gateway dan SMS interworking mobile Switching Center.

SMS Gateway MSC merupakan aplikasi yang dimiliki oleh MSC yang akan menerima pesan singkat (SMS) dari SMSC dan melakukan pendeteksian *Home Location Register* (HLR) yang berfungsi sebagai informasi untuk melakukan perutean untuk pengiriman pesan singkat tersebut.

2.1.6 Mobile Switching Center (MSC)

MSC (Mobile Switching Center) merupakan perangkat yang menyediakan hubungan pelanggan mobile dengan mobile maupun dengan jaringan *Public Switched Telephone Network* (PSTN) dan *Integrated Service Digital Network* (ISDN). MSC memiliki seluruh fungsi penting dalam *Switching* komunikasi pada seluruh MS dalam MSC area atau MSC merupakan pusat penyambungan yang mengatur jalur hubungan antara BSS pada operator layanan telekomunikasi lain.

2.1.7 Home Location Register (HLR)

Home Location Register (HLR) merupakan media yang menyimpan seluruh data yang berhubungan dengan pelanggan, seperti data statis yang menerangkan kapabilitas akses pelanggan, jenis pelayanan dan pelayanan tambahan yang digunakan oleh pelanggan tersebut. HLR juga mempunyai data dinamis tentang pesawat pelanggan yang *roaming*. MSC menggunakan data dinamik untuk segera merutekan panggilan yang datang ke pesawat pelanggan yang dipanggil.

Data-data yang disimpan di HLR bersifat permanen yaitu :

- a. IMSI
- b. MS kategori
- c. Batasan *Roaming*
- d. *Supplementary service seperti call forwarding*

e. *Authentication key*

Data-data yang tersimpan di HLR bersifat sementara yaitu :

- a. *Local MS Identity (LMSI)*
- b. *VLR address*, yang menandakan current VLR yang menangani MS.
- c. *MSC address* yang menandakan MSC area dimana MS tersebut terdaftar.

2.1.8 Visitor Location Register (VLR)

Visitor Location Register (VLR) sama dengan HLR, VLR ini juga berfungsi sebagai media penyimpanan, akan tetapi data yang disimpan adalah data tentang pesawat pelanggan yang memasuki area pelayanan. VLR dapat dianggap sebagai data dinamik yang secara intensif bertukar data dengan HLR. Secara keseluruhan fungsi VLR adalah sebagai berikut:

- a. Bekerjasama dengan HLR dan AuC pada proses authentication
- b. Meneruskan chipering *key* dari HLR ke BSS untuk proses *encryption/decryption*.
- c. Mengontrol penempatan nomor TMSI yang baru
- d. *SupPort paging* dan mencatat kondisi semua MS yang berbeda di areanya.

2.2 ARDUINO

Arduino adalah *platform* pembuatan prototipe elektronik yang bersifat *open-source hardware* yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. Arduino ditujukan bagi para seniman, desainer dan siapapun yang tertarik dalam menciptakan objek atau lingkaran yang interaktif.

Arduino pada awal dikembangkan di Ivrea, Italia. Pada *platform* arduino terdiri dari *arduino board*, *shield*, bahasa pemrograman arduino dan *arduino development environment*. *Arduino board* bisaanya memiliki sebuah chip dasar mikrokontroler Atmel AVR Atmega8 serta turunannya. Blok diagram *arduino board* yang sudah disederhanakan dapat dilihat pada gambar 2.3.

Shield adalah sebuah papan yang dapat dipasang diatas arduino *board* yang berfungsi untuk meningkatkan kemampuan dari arduino itu sendiri.

Bahasa pemrograman arduino adalah bahasa pemrograman yang umum digunakan untuk membuat perangkat lunak yang ditanamkan pada arduino *board*. Bahasa pemrograman yang umum digunakan untuk membuat perangkat lunak yang ditanamkan pada arduino *board*. Bahasa pemrograman yang digunakan mirip dengan bahasa pemrograman C++. [3]

Arduino *Development Environment* adalah perangkat lunak yang digunakan untuk menulis dan *compile* program untuk arduino. Arduino *Development Environment* juga digunakan untuk *upload* program yang telah di-*compile* ke memori arduino *board*.

2.2.1 Arduino Uno

Arduino Uno adalah arduino *board* yang menggunakan mikrokontroler Atmega328. Arduino Uno memiliki 14 *pin* kaki digital, 6 *input* analog, osilator kristal 16 MHz, koneksi *Universal Serial Bus* (USB), konektor sumber tegangan, *header In-Circuit Serial Programming* (ICSP) dan tombol *reset*. Arduino Uno membuat segala hal yang dibutuhkan untuk mendukung sebuah mikrometer. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau memberikan tegangan *Direct Current* (DC) dari baterai atau adaptor *Alternating Current* (AC) to DC sebagai sumber tegangan untuk arduino. Arduino Uno menggunakan Atmega16U2 yang diprogram sebagai *USB to serial converter* untuk komunikasi *serial* ke komputer melalui *Port* USB. Arduino Uno dapat dilihat pada gambar 2.3



Gambar 2.3. Arduino Uno R3 [12]

Data *board* Arduino Uno R3 adalah sebagai berikut:

- a. Mikrokontroler : Atmega328
- b. Tegangan Operasi : 5V
- c. Tegangan *Input (recommended)* : 7 – 12 V
- d. Tegangan *Input (limit)*: 6-20 V
- e. *Pin* digital I/O : 14 (6 diantaranya *pin* PWM)
- f. *Pin* analog *input* :6
- g. Arus DC per *pin* I/O : 40 mA
- h. Arus DC untuk *pin* 3,3 V : 150 mA
- i. Flash Memory : 32 KB dengan 0,5 KB digunakan untuk *bootloader*
- j. SRAM : 2 KB
- k. EEPROM : 1 KB
- l. Kecepatan Perwaktuan : 16 Mhz

2.2.2 *Pin* Masukan Dan Keluaran Arduino Uno

Arduino Uno memiliki 14 *pin* digital yang digunakan sebagai masukan dan keluaran menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap *pin* beroperasi pada tegangan 5 Volt. Setiap *pin* mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up* internal sebesar 20-30 K Ω . Beberapa *pin* juga memiliki kemampuan khusus yaitu:

- a. Komunikasi *serial pin* 0 (RX) dan *pin* 1 (TX), digunakan untuk menerima dan mengirim data secara *serial*.
- b. External *Interrupt* pada *pin* 2 dan *pin* 3, *pin* ini dapat dikonfigurasi untuk memicu sebuah *interrupt* pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
- c. *Pulse width modulation* (PWM) pada *pin* 3,5,6,9,10 dan 11 menyediakan keluaran PWM 8-bit dengan menggunakan fungsi *analogWrite()*.
- d. *Serial Peripheral Interface* (SPI) pada *pin* 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), *pin* ini mendukung komunikasi SPI dengan menggunakan *SPI library*.

- e. *Light Emitting Diodes (LED)* pada *pin* 13, terdapat *built-in LED* yang terhubung ke *pin* digital 13. Ketika *pin* bernilai *HIGH* maka *LED* menyala, sebaliknya ketika *pin* bernilai *LOW* maka *LED* akan mati.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap *pin* menyediakan resolusi sebanyak 10 bit. Secara *default* pengukuran nilai tegangan dari *ground* (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan *pin* AREF dan fungsi analog *Reference*.

2.2.3 Sumber Tegangan *Pin* Tegangan Arduino

Arduino Uno dapat diberi daya melalui koneksi USB atau melalui *power supply* eksternal. Jika Arduino Uno dihubungkan ke kedua sumber tegangan tersebut secara bersamaan maka Arduino Uno akan memilih salah satu sumber daya secara otomatis untuk digunakan. *Power supply* eksternal (bukan melalui USB) dapat berasal dari adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan ke soket *power* pada Arduino Uno. Jika menggunakan baterai, ujung kabel yang dihubungkan ke baterai dimasukkan kedalam *pin* GND dan Vin yang berada pada konektor *POWER*.

Arduino Uno dapat beroperasi pada tegangan 6 sampai 20 *Volt*. Jika Arduino Uno diberi tegangan dibawah 7 *Volt*, maka *pin* 5V akan menyediakan tegangan di bawah 5 *Volt* dan Arduino Uno akan bekerja dengan tidak stabil. Jika diberikan tegangan melebihi 12 *Volt*, IC regulator akan menjadi terlalu panas dan merusak Arduino Uno. Tegangan rekomendasi yang diberikan ke Arduino Uno berkisar antara 7 sampai 12 *Volt*. *Pin-pin* pada bagian *power* yang terdapat pada Arduino Uno adalah sebagai berikut:

- a. Vin adalah *pin* untuk mengalirkan sumber tegangan ke Arduino Uno ketika menggunakan sumber daya eksternal. Sumber tegangan juga dapat disediakan melalui *pin* ini jika sumber daya yang digunakan untuk Arduino Uno dialirkan melalui soket *power*.
- b. 5V adalah *pin* yang menyediakan tegangan teregulasi sebesar 5 *Volt* berasal dari regulator tegangan pada Arduino Uno.

- c. 3V3 adalah *pin* yang menyediakan tegangan teregulasi sebesar 3,3 Volt berasal dari regulator tegangan pada Arduino Uno.
- d. GND adalah *pin ground*.

2.2.4 Instruksi-Instruksi Pada Arduino Uno [4]

Bahasa pemrograman Arduino memiliki struktur dasar yang terdiri dari dua bagian utama atau fungsi, blok tertutup (dengan pasangan kurung kurawal), pernyataan dan komentar seperti berikut ini :

```
//B// Baris komentar
```

```
/*
```

```
Blok komentar
```

```
*/
```

```
void setup()
```

```
{
```

```
Pertanyaan
```

```
}
```

```
void loop()
```

```
{
```

```
Pertanyaan
```

```
}aris komentar
```

Penjelasan untuk masing – masing bagian dari struktur program Arduino seperti diatas:

- a. setup()

Fungsi `setup()` dipanggil sekali pada waktu program pertama kali berjalan. Fungsi ini bisaanya digunakan sebagai inisialisasi atau tahap persiapan berupa pengaturan cara kerja atau nilai-nilai awal.

Contoh:

```
void setup()
{
// Mengatur pin sebagai keluaran
pinMode(pin, OUTPUT);
}
```

b. `loop()`

Fungsi `loop()` akan menjalankan pernyataan di dalamnya berulang kali tanpa henti. Fungsi ini berisi program utama yang akan menyebabkan perubahan keadaan pada program, menerima respon dari papan Arduino dan mengendalikannya.

Contoh:

```
void loop()
{
digitalWrite(pin, HIGH); /*Mengaktifkan Port */
delay(1000); // Jeda satu detik
digitalWrite(pin, LOW); /*Mematikan
Port*/
delay(1000); // Jeda satu detik
}
```

c. Komentar

Komentar adalah teks yang akan diabaikan oleh compiler, sehingga tidak akan dikompilasi. Komentar tidak wajib disertakan di

dalam suatu program, namun berguna untuk memberikan penjelasan mengenai suatu fungsi atau cara kerja dari program tersebut. Seperti C/C++, bahasa pemrograman Arduino mengenali dua jenis komentar yaitu dalam bentuk baris komentar dan blok komentar.

d. Blok kurung awal

Kurung kurawal menandai bagian awal dan akhir dari sebuah blok fungsi atau blok pernyataan. Contoh:

```
// Kurung kurawal sebagai penanda fungsi
```

```
void pesan()
```

```
{
```

```
/* Kurung kurawal sebagai penanda blok
```

```
pernyataan */
```

```
X=5;
```

```
while(x>0)
```

```
{
```

```
Serial,print('a');
```

```
X++
```

```
}
```

```
}
```

Kurung kurawal pembuka ({}) harus diikuti kurung kurawal tutup (}) dan Arduino IDE telah menyediakan fasilitas untuk memeriksa pasangan kurung kurawal tersebut. Pilih kurung kurawal pembuka atau spasi kosong sesudahnya dan Arduino akan menyorot kurung kurawal pasangannya.

e. Pernyataan

Pernyataan adalah instruksi pemrograman untuk melakukan sesuatu. Setiap pernyataan dalam bahasa pemrograman Arduino harus diakhiri dengan tanda titik koma (;). Tanda tersebut juga berfungsi untuk memisahkan satu elemen dengan lainnya dalam suatu pernyataan.

f. Fungsi

Fungsi adalah suatu blok kode program yang berisi satu atau sekumpulan pernyataan untuk melakukan sesuatu hal tertentu. Fungsi dapat memiliki argumen masukan dan menghasilkan suatu nilai keluaran dengan tipe data tertentu. Dua jenis fungsi yang wajib ada adalah *setup()* dan *loop()* dengan penjelasan seperti diatas.

Struktur dari sebuah fungsi:

```
jenis_data nama_fungsi(argumen)

{

pernyataan;

}
```

Jika suatu fungsi tidak membutuhkan masukan, maka argumen dapat dikosongkan atau diisi dengan kata kunci *void*. Tipe data keluaran dari suatu fungsi didefinisikan dalam jenis data atau bila fungsi tidak memberikan nilai keluaran maka diisi dengan *void*.

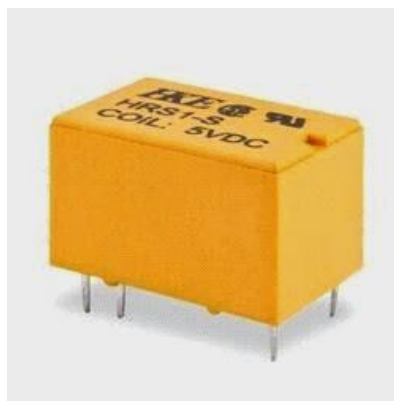
2.3 RELAY [11]

Relay merupakan suatu tombol elektrik yang bekerja berdasarkan prinsip elektro magnet untuk menutup atau membuka kontak saklar secara otomatis. *Relay* terdiri dari suatu lilitan dan *Switch* mekanik. *Switch* mekanik akan bergerak jika ada arus listrik yang mengalir melalui lilitan. Susunan kontak *Relay* sebagai berikut :

- a. *Normally Switch* (NC) : COM akan terhubung apabila kumparan *Relay* *off*.

- b. *Normally Open (NO)* : COM akan terhubung apabila kumparan *Relay* diberi arus listrik.
- c. *Change Over Relay* adalah kutub acuan (COM), memiliki kontak tengah yang akan melepaskan diri dan membuat kontak lainnya terhubung.

Pada *saat* lilitan disuplai tegangan, maka arus akan mengalir pada kumparan, sehingga pada inti besi yang dililit oleh kumparan akan timbul medan magnet. Karena inti besi bersifat magnetis, maka jangkar akan tertarik ke inti besi sehingga mengaktifkan kontak *Relay*. Bentuk fisik dari *Relay* ditunjukkan pada gambar 2.4

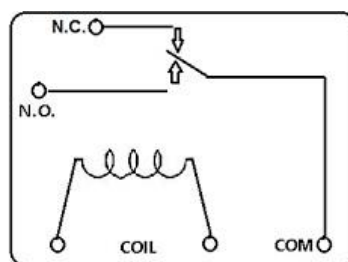


Gambar 2.4 Bentuk Fisik *Relay* [13]

Berdasarkan jumlah *kutub* pada *Relay*, maka *Relay* dibedakan menjadi 4 jenis, yaitu:

- a. SPST atau *single pole single throw*
- b. Spdt atau *single pole double throw*
- c. Dpst atau *double pole single throw*
- d. Dpdt atau *double pole double throw*

Pole adalah jumlah ommon dan throw adalah jumlah terminal *output* (NO dan NC) dari *Relay*. Skematik *Relay* ditunjukkan pada gambar 2.5



Gambar 2.5 Skematik Jenis *Relay* SPDT

2.4 MODEM WAVECOM FASTRACK M1306B [6]



Gambar 2.6 Modem Wavecom Fastrack M1306B

Modem *wavetrack* adalah modem alat produksi dari *Wavecom* yang berupa sebuah modem eksternal yang dijalankan dengan memasukkan *sim card* pada modem tersebut kemudian dihubungkan pada *Port serial* pada komputer *server* dan kemudian akan dijalankan dengan menggunakan perintah – perintah *AT-Command* yang khusus untuk menjalankan kerja dari *Wavecom* GSM modem ini.

Wavecome GSM modem ini mempunyai beberapa model fungsi yang dapat mengerjakan beberapa kerja tertentu diantaranya untuk *interface*, *standart*, *SMS*, *data,fax* dan *voice*. Bentuk fisik dari modem *wavecom* di tunjukan pada gambar 2.6. Perintah – perintah *AT-Command* merupakan susunan karakter yang membentuk suatu bahasa mesin yang dimengerti oleh GSM modem. Dimana setiap perintah telah dideklarasikan untuk menjalankan salah satu tugas yang diinginkan. Dengan kata lain *AT-Command* adalah satu-satunya perintah yang dapat dimengerti oleh GSM modem ini.

2.5 BAHASA C [7]

Bahasa pemrograman C dikenal di seluruh dunia sebagai bahasa pemrograman yang andal, cepat dan tergolong ke dalam medium level language. Bahasa C merupakan pengembangan dari bahasa (BCPL) Basic Combined Programming Language yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa C dikembangkan di Laboratorium Bell (USA) sekitar tahun 1972 oleh Dennis Ritchie, beliau adalah seorang pakar pemrograman.

Pada tahun 1978, Dennis Ritchie bersama dengan Brian Kernighan mempublikasikan buku yang kemudian menjadi legenda dalam sejarah *perkembangan* bahasa C, yang berjudul *The C Programming Language*. Buku ini diterbitkan oleh Prentice Hall. Seiring dengan berkembang pesatnya bahasa C, banyak vendor mengembangkan compiler C menurut versi masing-masing.

Untuk dapat mengerti dengan bahasa C, terlebih dahulu harus mengerti dengan struktur-struktur dari program. Apabila sudah mengerti dengan struktur-struktur yang sesuai maka bisa menjalankan program dengan tidak mendapatkan kesulitan. Setiap program C harus mengandung sedikitnya sebuah fungsi yang disebut dengan `main ()`. Suatu fungsi deprogram C dibuka dengan kurung kurawal “{“ dan ditutup dengan kurung kurawal tutup “}”. Di antara kurung-kurung kurawal dapat dituliskan statmen program C.

❖ Struktur Pemrograman Bahasa C pada Arduino

Struktur dasar bahasa pemrograman Arduino terdiri dari dua bagian yaitu :

a. `Setup()`

Fungsi `setup()` hanya dipanggil satu kali ketika program pertama kali di jalankan. Ini digunakan untuk mendefinisikan *mode pin* itu memulai komunikasi *serial*. Fungsi `setup()` harus di ikut sertakan dalam program walaupun tidak ada *statement* yang di jalankan.

b. `Loop()`

Setelah melakukan fungsi `setup()` maka secara langsung akan melakukan fungsi `loop()` secara berurutan dan melakukan instruksi- instruksi yang ada dalam fungsi `loop()`.

2.6 BAHASA AT COMMAND [5]

AT Command adalah perintah yang digunakan dalam komunikasi dengan *serialPort*. Dengan *AT Command* kita dapat mengetahui vendor dari handphone, mengetahui kekuatan sinyal, membaca pesan SMS, mengirim

SMS, menghapus pesan dan masih banyak lagi. Beberapa hal yang perlu diperhatikan dalam menggunakan *AT Command* adalah, *Command* apa yang harus dimasukkan ke terminal, langkah apa yang harus dilakukan setelah *Command* dimasukkan dan respon yang didapat setelah *Command* dimasukkan. Tidak semua *AT Command* memiliki tindakan dan respon yang sama. Handphone GSM dan modem dapat dioperasikan melalui remote control menggunakan *Portserial* (kabel Data atau koneksi infrared), bluetooth, atau usb, akan tetapi semua antarmuka tersebut akan dikenali oleh komputer sebagai *Serial Port*. Remote Control di implementasikan melalui bahasa *AT+C Command* menurut spesifikasi dari ETSI GSM 07.07 dan GSM 07.05. *AT Command* merupakan kepanjangan dari *attention Command*, dan selalu digunakan untuk memulai pengiriman baris perintah dari Terminal Equipment (TE) kepada Terminal Adaptor (TA). Contoh TE adalah komputer, sedangkan contoh TA adalah GSM Data Card. Baris perintah terdiri dari karakter string (alphanumeric) yang dikirimkan kepada modem untuk melakukan perintah tertentu. *AT Command* digunakan untuk mengoperasikan modem, dengan fungsi secara umum adalah sebagai berikut :

- Konfigurasi dan mengontrol dari dan ke jaringan GSM.
- Konfigurasi koneksi Modem melalui antarmuka *Serial RS-232*.
- Memperoleh status informasi dari jaringan GSM.

AT Command dapat dipergunakan untuk menginstruksikan perintah mengirim SMS dan menerima SMS seperti yang akan dilakukan alat agar dapat menerima pesan sesuai yang diinginkan. Untuk dapat menerima SMS dari *Handphone* pengirim ke modem penerima dalam pembuatan alat ini menggunakan dua perintah yang telah ditentukan sebelumnya yaitu sebuah perintah *AT+CMGR* dan *AT+CMGD*. Ini merupakan perintah untuk membaca SMS yang masuk dan untuk *AT+CMGD* yaitu perintah untuk menghapus pesan. Adapun format pengiriman SMS dengan menggunakan *AT+CMGR* (“Nomer Tujuan”) dilanjutkan dengan isi SMS yang akan dikirimkan. Selain itu *AT Command* juga dapat dipergunakan untuk melakukan uji coba terhadap modem yang dipergunakan dalam keadain baik atau tidak. Untuk dapat melakukan ini dapat dipergunakan sebuah

aplikasi yaitu *Simple therm*. Untuk dapat melakukan ini hanya diperlukan untuk membuka *Simple therm* dan pasang modem pada komputer atau laptop yang dipergunakan untuk uji coba modem. Ketikkan "AT" pada *Simple therm*, jika muncul "OK" ini menandakan modem yang akan dipergunakan dalam keadaan bagus.

Jadi *AT Command* inilah bahasa yang dipahami oleh modem. Untuk memulai suatu perintah *AT Command*, diperlukan prefiks "AT" atau "AT" dalam setiap perintah *AT Command*, dan diakhiri dengan "<CR>" (= 0x0D). Beberapa perintah *AT Command* dapat dituliskan pada baris yang sama dengan hanya menggunakan satu prefiks "AT" atau "AT", kemudian antar perintah dibatasi oleh karakter ";" . Beberapa perintah *AT Command* yang digunakan untuk keperluan SMS (pengiriman dan penerimaan) adalah sebagai berikut :

2.6.1 At+Cmgs

Perintah *AT Command* ini digunakan untuk mengirimkan SMS. Format yang digunakan adalah "AT+CMGS = <length><CR><PDU is given>". Apabila pengiriman sukses dilakukan, format respon yang diterima adalah "+CMGS : <mr>", dengan "<mr>" adalah *message reference* dari SMSC. Sedangkan jika pengiriman gagal dilakukan, respon yang diterima adalah "+CMS error".

2.6.2 At+Cmgr

Perintah ini digunakan untuk membaca sebuah pesan pada *indeks* tertentu. Format yang digunakan adalah "AT+CMGR = <index>". Apabila perintah ini berhasil dieksekusi, format respon yang diterima adalah "+CMGR:<stat>,<length><CR><LF><pdu>". "<stat>" berarti status, parameter status pesan adalah sebagai berikut :

Perintah ini digunakan untuk membaca sebuah SMS pada program *AT Command* adalah :

- a. Pesan yang diterima dan belum dibaca, merupakan parameter standar adalah 0.
- b. Pesan yang diterima dan sudah dibaca adalah 1.
- c. Pesan tersimpan pada *memory* SMS yang tidak terkirim adalah 2.

- d. Pesan tersimpan pada *memory* SMS yang berhasil dikirimkan adalah 3.
- e. Semua pesan pada *memory* SMS adalah 4.

2.6.3 At+Cmgd

Perintah ini digunakan menghapus sebuah SMS pada *memory* SMS. Format yang digunakan adalah “AT=CMGD=<*index*>”, respon yang diterima adalah “OK/ERROR /+CMS ERROR ”

2.6.4 At+Cmgl

Perintah At+Cmgl ini digunakan untuk membaca daftar SMS sesuai parameter tertentu. Format AT yang digunakan adalah “AT+CMGL[=<*stat*>]”. Parameter status pesan adalah sebagai berikut :

- a. Pesan yang diterima dan belum dibaca, merupakan parameter standar adalah 0.
- b. Pesan yang diterima dan sudah dibaca adalah 1.
- c. Pesan tersimpan pada memory SMS yang tidak terkirim adalah 2.
- d. Pesan tersimpan pada memory SMS yang berhasil dikirimkan adalah 3.
- e. Semua pesan pada memory SMS. Respon yang diterima adalah 4.

Apabila semua pesan telah terkirim maka sintak yang muncul adalah sebagai berikut :

```
+CMGL:<index>,<stat>,<oa/da>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<CR><LF><Data>[<CR><LF>
+CMGL:
<index>,<stat>,<da/oa>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<
CR><LF>
<Data>[...]] OK ”
```

Atau “+CMSERROR : <err>”. “[<alpha>]”, adalah deretan alfanumerik yang merepresentasikan nomor pengirim atau penerima. Yang terpenting adalah mengetahui perintah *AT Command* untuk terima, kirim, dan delete SMS. Kemudian perintah tersebut dimasukkan ke dalam coding program yang telah di buat.

2.7 KOMUNIKASI SERIAL RS 232

Komputer pada umumnya mempunyai *Port serial* dan *Port paralel*, *Port serial* dibagi menjadi dua bagian, yaitu komunikasi *serial* RS 232 yang menggunakan *Port* atau terminal DB-9 dan komunikasi *serial* dengan menggunakan terminal *Universal Serial Bus* (USB). Walaupun seluruh terminal tersebut dapat digunakan untuk berkomunikasi dengan perangkat yang berada diluar kompute, tetapi semuanya mempunyai cara kerja yang berbeda-beda satu dengan yang lain.

Paralel *Port* dapat menerima data 8 *bit* secara bersamaan melalui 8 jalur kabel melalui terminal paralel *Port* DB 25. Bila menggunakan paralel *Port* jenis ini maka data dapat ditransfer dengan cepat, akan tetapi kabel data yang dibutuhkan sangat banyak dan juga jarak atau panjang kabel yang digunakan tidak dapat terlalu jauh ± 2 meter.



Gambar 2.7 *Port* DB9 [14]

Tabel 2.1 merupakan tabel yang menjelaskan fungsi dari *PIN* yang terdapat pada DB9. Bagian yang terpenting dalam komunikasi *serial* adalah konektor DB9 seperti yang tampak pada gambar 2.7 sendiri merupakan konektor yang dipergunakan untuk dapat menghubungkan *hardware* dengan komputer atau perangkat lain yang mendukung sistem komunikasi *serial*.

Tabel 2.1 Fungsi *PIN* DB9

Nama <i>PIN</i>	KETERANGAN	FUNGSI
TD	Transmisi Data	Untuk pengiriman Data <i>serial</i>
RD	<i>Receive Data</i>	Untuk penerimaan Data <i>serial</i>
RTS	<i>Request To Send</i>	Sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran Data.
CTS	<i>Clear To Send</i>	Memberitahukan bahwa modem siap melakukan pertukaran Data
DSR	<i>Data Set Ready</i>	Untuk memberitahukan UART bahwa modem siap melakukan pertukaran Data
CD	<i>Carrier Detect</i>	Saat modem mendeteksi suatu “ <i>carrier</i> ” dari modem lain, maka sinyal ini akan diaktifkan
DTR	<i>Data Terminal Ready</i>	Kebalikan dari DSR untuk memberitahu bahwa UART siap melakukan komunikasi.
RI	<i>Ring Indicator</i>	Aktif saat mendeteksi sinyal dering dari <i>telephone</i> .

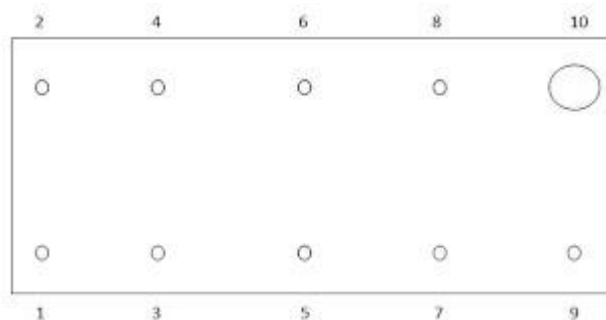
RS 232 merupakan alat yang berfungsi sebagai penghubung dalam proses transfer data antar komputer, atau komputer dengan perangkat lain dalam bentuk komunikasi *serial*. RS 232 merupakan singkatan dari *Recommended Standart Number 232*. Standar ini dibuat oleh *Eletronic Industry Association*, dengan tujuan untuk membuat standarisasi antarmuka atau *interface* antar peralatan terminal data dan peralatan komunikasi data *serial* dengan menggunakan data biner sebagai data yang akan dikirim maupun yang diterima. [8]

Serial Port digunakan untuk mengirim dan menerima data satu bit pada waktu dan membutuhkan waktu yang jauh lebih lama jika dibandingkan dengan paralel *Port* jika datanya berjumlah 8 bit, akan tetapi jika menggunakan *serial Port* maka kabel yang digunakan lebih sedikit dan mampu untuk melakukan komunikasi secara *full duplex* atau dua arah. Kabel yang dibutuhkan untuk komunikasi *serial* ini cukup tiga jalur, yang pertama untuk mengirim, yang kedua untuk menerima dan yang ketiga untuk *ground*.

Port COM pada komputer tidak dapat secara langsung dihubungkan pada mikrokontroler maka dibutuhkan IC max232 yang berfungsi untuk mengubah tegangan sinyal RS232 kelevel tegangan TTL agar dapat diterima oleh mikrokontroler AVR.

Salah satu chip yang dapat digunakan adalah MAX232. Keuntungan tambahan dari MAX232 karena chip ini hanya membutuhkan tegangan *power* suplay +5V, yang sama dengan sumber tegangan mikrokontroler AVR, sehingga tidak perlu menggunakan dua jenis tegangan *power* supplay yang berbeda. MAX232 mempunyai dua perangkat IC max232 yang masing-masing dapat mengirim dan menerima data. Pengemudi jalur yang digunakan untuk mengirimkan data (TXD) disebut T1 dan T2, dan pengemudi jalur untuk menerima data (RXD) ditandai sebagai R1 dan R2.

2.8 PIN GROUP [9]



Gambar 2.8 Susunan *PIN GROUP*

PIN GROUP adalah sekumpulan konektor yang terhubung dengan kabel *reset*, *HDD LED*, *power LED*, *speaker*, dan *power Switch*. Pada *PIN GROUP* terdapat 10 *pin*, tetapi ada satu *pin* yang dikosongkan. *Pin* yang kosong tersebut adalah *pin* nomor 10, sedangkan ujung yang berseberangan adalah *pin* nomor 1. Sehingga lima *pin* dibawah semua bernomor ganjil, dari kiri ke kanan bernomor 1-3-5-7-9 sedangkan *pin-pin* pada barisan atas bernomor genap 2-4-6-8-10. *Pin* pada *PIN GROUP* mempunyai pasangan-pasangan tersendiri. Pasangan *pin* tersebut adalah :

- a. 1 - 3 Ke kabel *HDD LED*
- b. 2 - 4 Ke kabel *POWER LED*
- c. 5 - 7 Ke kabel *RESET SW*

d. 6 - 8 Ke Kabel *POWER SW*

Sedangkan *pin* nomor 9 biarkan kosong. Perhatikan kolom sebelah kiri, *Pin* 1, 2 dan 6 adalah *pin* positif, hanya *pin* nomor 5 yang *pin* negatif. Sebaliknya, pada kolom sebelah kanan hanya *pin* nomor 7 yang *pin* positif sedangkan *pin* nomor 3, 4 dan 8 adalah *pin* negatif.

2.9 CATU DAYA [10]

2.9.1 Dioda

Dioda merupakan komponen elektronika *non-linier* yang sederhana. Struktur dasar dioda berupa bahan semikonduktor tipe P yang disambung dengan bahan tipe N. Pada ujung bahan tipe P dijadikan terminal *Anoda* (A) dan ujung lainnya *katoda* (K), sehingga dua terminal inilah yang menyiratkan nama dioda. Operasi dioda ditentukan oleh polaritas relative kaki *Anoda* terhadap kaki *Katoda*. Penerapan dioda semikonduktor dalam bidang elektronika sangatlah luas. Hal ini karena sifat dioda yang sangat mendasar yaitu hanya dapat melewatkan arus dalam satu arah saja. Rangkaian penyearah merupakan salah penerapan dioda. Sesuai dengan bentuk gelombang *output*-nya, maka penyearah terdapat dua macam yaitu setengah gelombang dan gelombang penuh.

Rangkaian pemotong dan penggeser merupakan penerapan lain yang juga banyak digunakan dalam teknik pulsa. Jenis dioda semikonduktor yang khusus dioperasikan pada bias mundur yang pada titik *break-down*-nya sering disebut dengan dioda *Zener*. *Zener* ini merupakan inti dari rangkaian penyetabil tegangan.

2.9.2 Penyearah Gelombang Penuh Jembatan Dioda

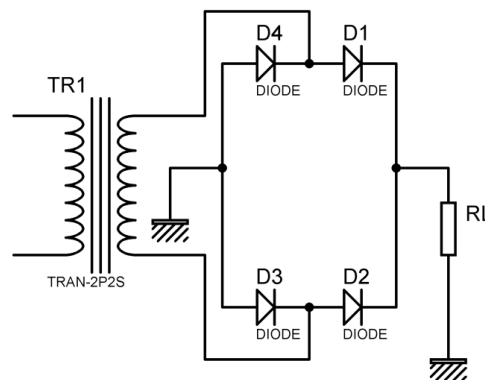
Penyearah gelombang penuh dengan menggunakan sistem jembatan bisa menggunakan trafo CT maupun yang biasa, atau bahkan bisa juga tanpa menggunakan trafo. Rangkaian dasar untuk penyearah gelombang penuh jembatan dapat dilihat pda gambar 2.9. Prinsip kerja penyearah gelombang penuh sistem jembatan dapat dijelaskan melalui gambar 2.9.

Pada saat rangkaian mendapat bagian positif dari siklus sinyal *alternating current* (AC) maka :

- D1 dan D3 *on* karena dibias maju
- D2 dan D4 *off* karena dibias mundur
- Sehingga arus i_1 mengalir melalui D1, RL dan D3

Sedangkan jika jembatan memperoleh siklus negatif, maka:

- D2 dan D4 *on* karena dibias maju
- D1 dan D3 *off* karena dibias mundur
- Sehingga arus i_2 mengalir melalui D2, RL dan D4



Gambar 2.9 Rangkaian penyearah gelombang penuh jembatan

Arus i_1 dan i_2 yang melewati beban RL mempunyai nilai yang sama, yaitu dari ujung atas RL menuju *ground*. Dengan demikian arus yang mengalir ke beban (i_L) merupakan penjumlahan dari dua arus i_1 dan i_2 , dengan menempati waktu masing-masing. Untuk besarnya arus rata-rata pada beban adalah sama seperti penyearah gelombang penuh trafo CT. Sedangkan untuk nilai VDC adalah:

$$V_{dc} = 0,636 (V_m - 2V_\gamma) \dots \dots \dots (II, 2.1)$$

Keterangan :

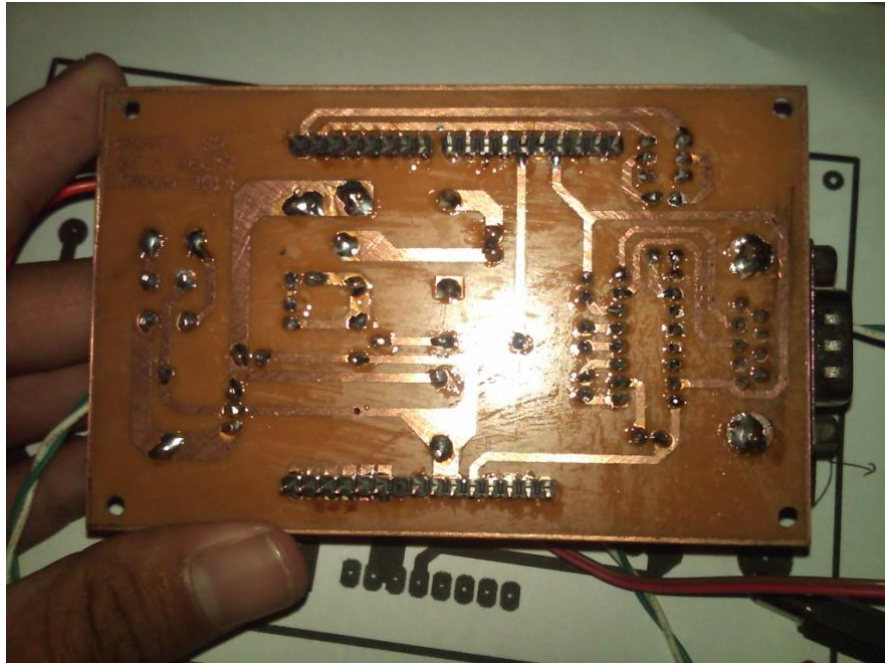
V_{dc} adalah Nilai hasil penyearahan rangkaian penyearah.

Peak Inverse Voltage (PIV) atau V_m adalah nilai tegangan tertinggi yang tidak boleh dilewatipada dioda saat dioda mengalami bias mundur.

V_γ atau V_{ripple} adalah nilai selisih tertinggi antarategangan maksimum dan tegangan minimumhasil penyearahan.

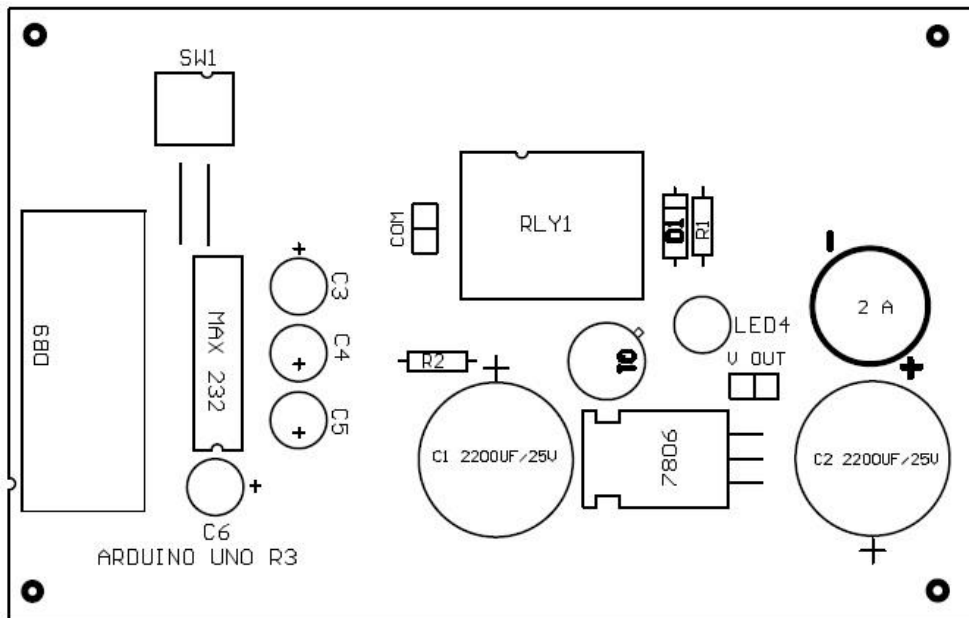
2.10 PCB, SKEMATIK, DAN ALAT TUGAS AKHIR

Gambar 2.10 dibawah ini adalah gambar *PCB* yang telah dibaut dalam tugas akhir ini.



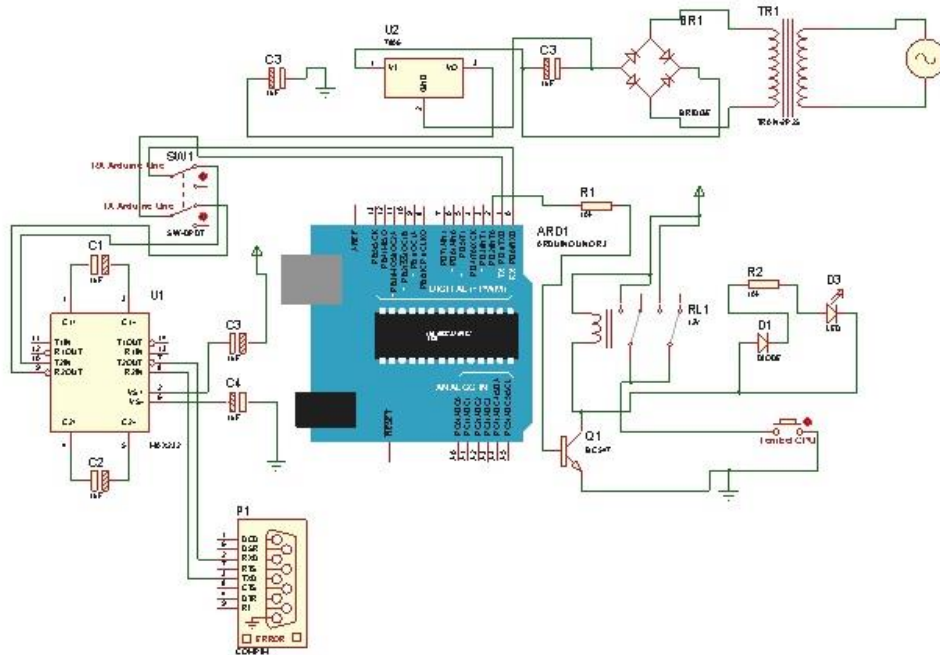
Gambar 2.10 *PCB*

Sedangkan gambar 2.11 adalah gambar *PCB* menurut komponen-komponen yang dirangkai didalamnya beserta keterangan.



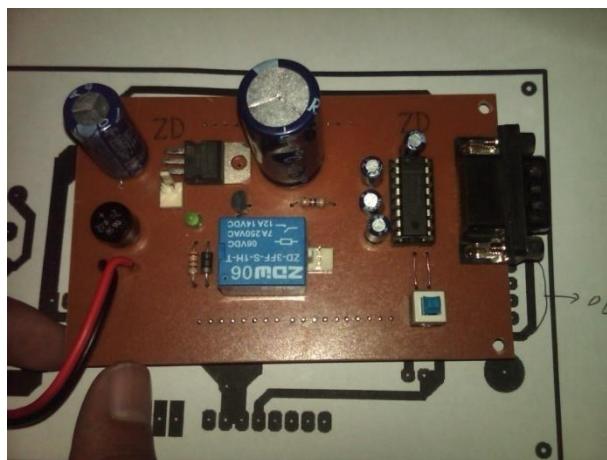
Gambar 2.11 *PCB* Perkomponen

Untuk skematik dari tugas akhir ini dibuat dengan menggunakan *software proteus*. *Software proteus* adalah *software* yang digunakan untuk membuat skematik sebuah design rangkaian dan berguna untuk membuat *layout PCB* dari skematik yang kita buat. Dengan *software proteus* bisa mengetahui letak kesalahn sebelum menyetak *layout PCB*. Gambar 2.12 adalah skematik keseluruhan dari alat yang dibuat pada tugas akhir ini.



Gambar 2.12 Skematik Dari Tugas Akhir

Kemudian gambar 2.13 adalah gambar dari alat yang sudah dibuat untuk tugas akhir dengan belum terpasangnya modem *wavecom*, arduino uno, dan *CPU*.



Gambar 2.14 Alat Tugas Akhir

2.1 SWITCH

Saklar adalah sebuah perangkat yang berfungsi untuk memutuskan listrik atau menghubungkan listrik. Dalam saklar terdapat kondisi on dan off, bila saklar dalam kondisi off (*open circuit*) maka listrik yang mengalir pada saklar akan terputus, namun bila saklar dalam kondisi on (*close circuit*) maka listrik pada saklar akan mengalir kembali. Saklar yang biasanya terpasang pada rumah-rumah disebut dengan saklar listrik yang mempunyai batas maksimal besar sedangkan saklar yang terpasang pada *PCB* adalah saklar elektronik yang mempunyai batasan maksimal kecil. Saklar listrik digunakan untuk arus yang kuat dan saklar elektronik digunakan untuk arus yang lemah. Besar kecilnya ukuran saklar tergantung pada arus yang akan dilewatkan, artinya semakin besar ukuran saklar maka batas arus listrik yang dipunya ajab semakin besar atau sebaliknya. Gambar 2.15 akan menunjukkan jenis, symbol, dan contoh bentuk fisik dari saklar.

