

SISTEM ALARM KEBOCORAN GAS LPG DILENGKAPI FASILITAS SMS GATEWAY

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Teknik Elektro



Disusun oleh :

Nama : Anjar Pribadi

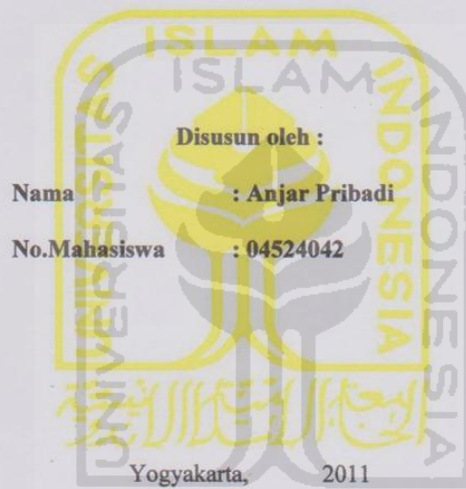
No.Mahasiswa : 04524042

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

2011

LEMBAR PENGESAHAN PEMBIMBING
SISTEM ALARM KEBOCORAN GAS LPG DILENGKAPI
FASILITAS SMS GATEWAY

TUGAS AKHIR



Pembimbing I

Pembimbing II

Ir.Hj Budi Astuti,MT

Wahyudi Budi Pramono, S.T, M.Eng

LEMBAR PENGESAHAN PENGUJI

SISTEM ALARM KEBOCORAN GAS LPG DILENGKAPI
FASILITAS SMS GATEWAY

Tugas Akhir

Oleh :

Nama : Anjar Pribadi

No.Mahasiswa : 04524042

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Indonesia

Yogyakarta, Juli 2011

Tim Penguji

Wahyudi Budi Pramono, ST., M.Eng.

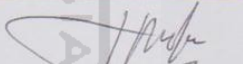
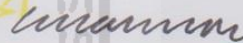
Ketua

Dr Eng Hendra Setiawan ST.,MT.

Anggota 1

Dwi Ana Ratnawati, ST., M.Eng.

Anggota II



Mengetahui:

Ketua Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia



Tito Yuwono, ST., M.Sc

HALAMAN PERSEMBAHAN



*Kupersembahkan karya ini untuk:
Orang tua dan keluargaku tercinta
yang telah mencurahkan kasih sayang, do'a dan dukungannya
yang telah mengajarkan keseimbangan dalam diriku
kesungguhan, ketekunan, kesabaran, welas asih dan cara mencintai makhluk
yang menjadikan aku merasa bersyukur diantara kelebihan dan kekuranganku
yang tetap membuatku ingat untuk selalu menunduk dihadapan – Nya
Semoga Allah SWT mencatatnya sebagai amal kebajikan.*

MOTTO

كُتِبَ عَلَيْكُمُ الْقِتَالُ وَهُوَ كُرْهُ لَكُمْ وَعَسَى أَنْ تَكْرَهُوا شَيْئًا
وَهُوَ خَيْرٌ لَّكُمْ وَعَسَى أَنْ تُحِبُّوا شَيْئًا وَهُوَ شَرٌّ لَّكُمْ وَاللَّهُ يَعْلَمُ
وَأَنْتُمْ لَا تَعْلَمُونَ ﴿٢١٦﴾

Diwajibkan atas kamu berperang, padahal berperang itu adalah sesuatu yang kamu benci. Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui.

(Q.S. Al Baqarah :216)

وَلَوْ أَنْتُمْ فِي الْأَرْضِ مِنْ شَجَرَةٍ أَقْلَمَ وَالْبَحْرُ يَمُدُّهُ مِنْ بَعْدِيهِ سَبْعَةُ
أَبْحُرٍ مَا نَفِدَتْ كَلِمَاتُ اللَّهِ إِنَّ اللَّهَ عَزِيزٌ حَكِيمٌ ﴿٢٧﴾

Dan seandainya pohon-pohon di bumi menjadi pena dan laut, ditambahkan kepadanya tujuh laut sesudahnya, niscaya tidak akan habis-habisnya kalimat Allah. Sesungguhnya Allah Maha Perkasa lagi Maha Bijaksana.

(Q.S. Al Lukman 27)

**SELANGKAH LEBIH MAJU DALAM ILMU AMALIAH
DAN AMAL ILMIAH**

KATA PENGANTAR



Assalamu'alaikum warahmatullahi wabarakatuh,

Alhamdulillah *rabbil'alamin*, segala puji syukur saya panjatkan kehadiran Allah SWT yang telah melimpahkan segala karunia-Nya, sehingga saya dapat menyelesaikan Tugas Akhir ini sebagai salah satu persyaratan program S-1 Jurusan Teknik Industri, Fakultas Teknik Elektro, Universitas Islam Indonesia. Shalawat dan salam semoga tercurah pada junjungan kita Rasulullah Muhammad SAW beserta keluarga, dan pengikutnya yang setia hingga akhir zaman.

Merupakan suatu kelegaan ketika akhirnya saya dapat menyelesaikan bagian akhir dari amanah yang panjang ini dengan begitu banyak kemudahan dan kekuatan yang Allah berikan melalui banyak pihak dengan bantuan, dukungan dan do'a. Untuk itulah, saya sangat ingin menghaturkan terima kasih kepada :

1. Ibu Ir.Hj Budi Astuti MT dan Bapak Wahyudi Budi Pramono, ST, M.Eng selaku dosen pembimbing yang telah memberikan banyak bimbingan dalam penyusunan Tugas Akhir ini.
2. Bapak Tito Yuwono, ST, M.Sc selaku ketua jurusan Teknik Elektro Universitas Islam Indonesia.
3. Seluruh Bapak Ibu dosen , staf dan karyawan jurusan Teknik Elektro.
4. Bapak Slamet Rianto, Ibu Sulistiowati, Mbak Nopa, Mbak Dwi dan kakak iparku serta adikku Setia Dani yang telah memberikan do'a serta dukungan, sehingga dapat menyelesaikan dan menyusun Tugas Akhir ini.

5. Citra Mei Suci wanita terindah yang telah menjadi energi tersendiri dalam diriku untuk menyempurnakan Tugas Akhir ini.
6. Sahabat-sahabatku(Wakidi kost) dan semua pihak yang telah membantu dalam penyusunan Tugas Akhir ini yang tidak dapat disebutkan satu-persatu.
7. Rahmad Perwira, Arif, Yudi terimakasih banyak buat semuanya, semoga selalu diberi kemudahan dan rizkinya lancar. Amien.
8. Puguh Jatmiko Krisnaji, Asep Juliyansyah, Yanuar Hendyanto, Ikhsan Maulana, Muhamad Hafizh, Handi Alaina, Muhamad Azra'i Bin Bahrum, Aulia Candra, Aan, Muhamad Wahyudi dan teman2 lainnya yang tidak bisa saya sebut satu persatu, terimakasih banyak kawan atas semua fasilitas-fasilitas yang telah kalian semua berikan. Semoga Allah SWT membalas budi baik kalian, dilapangkan rizkinya. Amin.

Saya menyadari sepenuhnya bahwa tugas akhir ini memiliki banyak kekurangan dan masih jauh dari idealisme seorang peneliti maupun karya ilmiah. Untuk itulah saya meminta maaf yang setulus-tulusnya dan menunggu masukan yang berguna untuk perbaikan selanjutnya.

Akhir kata, semoga tugas akhir ini dapat memberi manfaat bagi para pembacanya.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Jogjakarta, Juni 2011

Anjar Pribadi

ABSTRAK

Konversi dari minyak tanah ke tabung gas menjadi salah satu kebijakan pemerintah dalam menanggulangi krisis bahan bakar. Ledakan mengakibatkan kebakaran sehingga menimbulkan kerugian harta benda bahkan adanya korban nyawa sehingga membuat masyarakat menjadi merasa tidak aman untuk menggunakannya. Alat pendeteksi kebocoran gas LPG yang dilengkapi dengan alarm dan SMS ini dapat mendeteksi kebocoran dari tabung gas LPG dengan jarak maksimal 2 meter dalam waktu 1 menit baik di ruangan terbuka maupun tertutup dengan dimensi 2 X 3 meter. Apabila terdeteksi maka alat mampu membunyikan alarm dan mengirimkan SMS ke *user*.

Sistem ini menggunakan sensor LPG TGS 2610, mikrokontroler ATmega 8535, *Display LCD*, *HP*, *buzzer* dan beberapa komponen pendukung yang lain. Alat pendeteksi kebocoran gas LPG yang dilengkapi dengan alarm dan SMS ini diharapkan dapat menjadi alternatif bagi masyarakat untuk menjadi peringatan dini apabila terdapat kebocoran pada tabung gas sehingga dapat diantisipasi lebih dini.

Dalam peletakan sensor gas, jarak sangatlah berpengaruh terhadap kemampuan sensor gas untuk dapat mendeteksi kebocoran gas. Waktu yang diperlukan sensor agar dapat mendeteksi kebocoran gas akan lebih lama apabila jarak peletakan sensor dari sumber kebocoran gas yang diperkirakan semakin jauh. Sirkulasi udara dalam ruangan juga berpengaruh terhadap sensor gas dalam mendeteksi adanya kebocoran gas LPG. Semakin baik sirkulasi udara dalam ruangan maka semakin lama sensor gas untuk dapat mendeteksi gas apabila terjadi kebocoran gas LPG dalam ruangan tersebut.

Kata kunci (*keyword*): Sensor LPG TGS 2610. Mikrokontroler ATmega 8535

Display LCD, HP, buzzer

DAFTAR ISI

| | | |
|---|---|----|
| HALAMAN JUDUL | | |
| LEMBAR PENGESAHAN PEMBIMBING | ii | |
| HALAMAN PERSEMBAHAN | iii | |
| MOTTO | iv | |
| KATA PENGANTAR | v | |
| ABSTRAK | vii | |
| DAFTAR ISI | viii | |
| DAFTAR TABEL | xi | |
| DAFTAR GAMBAR | xii | |
| BAB I | PENDAHULUAN | |
| 1.1 | Latar Belakang Masalah | 1 |
| 1.2 | Maksud dan Tujuan | 1 |
| 1.3 | Batasan Masalah | 2 |
| 1.4 | Metodologi Pengerjaan | 2 |
| 1.5 | Sistematika Penulisan Laporan..... | 3 |
| BAB II | TINJAUAN PUSTAKA | |
| 2.1 | Sensor Gas LPG | 4 |
| 2.2 | Mikrokontroler Atmega 8535..... | 5 |
| 2.2.1 | Peta Memory | 8 |
| 2.2.2 | Status Register | 9 |
| 2.2.3 | General Purpose Register | 11 |
| 2.2.4 | Stack Pointer..... | 12 |
| 2.2.5 | Timer dan Counter..... | 12 |
| 2.3 | SMS (<i>Short Message Service</i>)..... | 22 |
| 2.3.1 | PDU Untuk Kirim SMS..... | 23 |
| 2.4 | Komunikasi Serial | 28 |

| | | |
|-------|---------------------------------------|----|
| 2.4.1 | Karakteristik Port Serial | 29 |
| 2.4.2 | Interface RS 232..... | 31 |
| 2.5 | Komunikasi Mikrokontroler dan HP..... | 31 |
| 2.6 | Driver Relay | 32 |
| 2.7 | Transistor..... | 33 |
| 2.8 | Penampil LCD 16 x 2..... | 33 |

BAB III PERANCANGAN SISTEM

| | | |
|-------|---|----|
| 3.1 | Perangkat Keras..... | 35 |
| 3.1.1 | Mikrokontroler | 36 |
| 3.1.2 | Sensor Gas LPG TGS 2610..... | 38 |
| 3.1.3 | Penampil LCD 16 x 2..... | 39 |
| 3.1.4 | Rangkaian Driver Relay..... | 40 |
| 3.1.5 | Komunikasi Mikrokontroler dan HP Terminal | 41 |
| 3.1.6 | Catu Daya..... | 43 |
| 3.2 | Perancangan Perangkat Lunak | 44 |
| 3.2.1 | Prosedur Penanganan Sistem | 44 |
| 3.2.2 | Perancangan Pesan SMS | 44 |
| 3.2.3 | Konversi pesan ke PDU | 45 |

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

| | | |
|-------|---|----|
| 4.1 | Pengujian Alat | 47 |
| 4.1.1 | Pengujian Terhadap Catu Daya..... | 47 |
| 4.1.2 | Pengujian Terhadap Sensor TGS 2610 | 49 |
| 4.1.3 | Pengujian Terhadap Driver | 50 |
| 4.1.4 | Pengujian Koneksi HP | 52 |
| 4.1.5 | Pengujian Kinerja Sistem Secara Keseluruhan | 54 |

BAB V PENUTUP

| | | |
|-----|-------------------|----|
| 5.1 | Kesimpulan | 55 |
| 5.2 | Saran-saran | 56 |

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR TABEL

| | | |
|------------|---|----|
| Tabel 2.1 | Konfigurasi Bit WGM01 dan WGM00 | 14 |
| Tabel 2.2 | Konfigurasi BitCOM 01 dan COM 00 non PWM | 14 |
| Tabel 2.4 | Konfigurasi BitCOM 01 dan COM 00 Fast PWM | 14 |
| Tabel 2.5 | Konfigurasi BitCOM 01 dan COM 00 Phase Correct PWM..... | 15 |
| Tabel 2.6 | Konfigurasi Bit Clock Select | 15 |
| Tabel 2.7 | Konfigurasi Bit compare Output Mode non PWM..... | 17 |
| Tabel 2.8 | Konfigurasi Bit compare Output Mode fast PWM..... | 17 |
| Tabel 2.9 | Konfigurasi Bit compare Output Mode Phase Correct | 17 |
| Tabel 2.10 | Konfigurasi Bit Clock Select | 18 |
| Tabel 2.11 | Konfigurasi Bit Clock Select | 19 |
| Tabel 2.12 | Konfigurasi Bit WMG 21 dan WGM 20..... | 20 |
| Tabel 2.13 | Konfigurasi BitCOM 21 dan COM 20 Non PWM..... | 21 |
| Tabel 2.14 | Konfigurasi BitCOM 21 dan COM 20 Fast PWM | 21 |
| Tabel 2.15 | Konfigurasi BitCOM 21 dan COM 20 Phase Correct PWM..... | 21 |
| Tabel 2.16 | SMS center operator GSM | 24 |
| Tabel 2.17 | Waktu Validitas | 26 |
| Tabel 2.18 | Fungsi Pin Konektor DB -9 | 30 |
| Tabel 3.1 | Fungsi PIN LCD 16 x 2 | 39 |
| Tabel 3.2 | Konversi pesan ke PDU | 45 |
| Tabel 4.1 | Hasil Pengamatan Catu Daya..... | 48 |
| Tabel 4.2 | Hasil pengamatan sensor gas TGS 2610 | 50 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Bentuk Fisik Sensor Gas LPG | 4 |
| Gambar 2.2 Skema System Elektronik Sensor Gas LPG | 4 |
| Gambar 2.3 Grafik Perbandingan Rs/Ro Sensor Gas TGS 2610..... | 5 |
| Gambar 2.4 Konfigurasi Pin Atmega 8535..... | 7 |
| Gambar 2.5 Konfigurasi Memori Data AVR ATMEGA 8535..... | 8 |
| Gambar 2.6 Memori Program AVR ATMEGA 8535..... | 9 |
| Gambar 2.7 Status Register ATMEGA 8535..... | 9 |
| Gambar 2.8 Ke – 32 Register General Purpose | 11 |
| Gambar 2.9 Register X Y Z | 12 |
| Gambar 2.10 Register TCCRIA..... | 16 |
| Gambar 2.11 Register TCCRIB..... | 18 |
| Gambar 2.12 Register TCCR2..... | 20 |
| Gambar 2.13 Konektor Serial DB-9..... | 30 |
| Gambar 2.14 Kabel Serial RS 232..... | 30 |
| Gambar 2.15 Kabel Data Siemens C35..... | 32 |
| Gambar 2.16 Simbol Relay | 33 |
| Gambar 2.18 LCD 16 x 2 | 34 |
| Gambar 3.1 Diagram Blok Perangkat Keras Sistem Utama | 35 |
| Gambar 3.2 Sistem Minimum Mikrokontroler ATMEGA 8535 | 36 |
| Gambar 3.3 Sensor TGS 2610..... | 38 |
| Gambar 3.4 Konfigurasi Mikrokontroler dan LCD | 40 |
| Gambar 3.5 Rangkain Driver Buzzer..... | 40 |
| Gambar 3.6 Rangkaian Driver Rellay | 41 |
| Gambar 3.7 Konsep Konverster RS 232..... | 42 |
| Gambar 3.8 Konfigurasi Max 232..... | 42 |
| Gambar 3.9 Rangkaian Power Supply..... | 43 |
| Gambar 3.10 Flow Chart Program..... | 46 |

| | |
|---|----|
| Gambar 4.1 Rencana Pengujian Catu Daya | 47 |
| Gambar 4.2 Rencana Pengujian Sensor TGS 2610..... | 49 |
| Gambar 4.3 Rencana Pengujian Driver..... | 51 |
| Gambar 4.4 Tampilan Hyperterminal..... | 52 |
| Gambar 4.5 Konfigurasi Hyper Terminal..... | 53 |



BAB I

PENDAHULUAN

1.1. Latar Belakang

Kebakaran yang disebabkan oleh meledaknya selang tabung gas LPG kini sering terjadi. Media masa banyak melansir informasi adanya ledakan tabung gas LPG. Ledakan mengakibatkan kebakaran sehingga menimbulkan kerugian harta benda bahkan adanya korban nyawa.

Dengan adanya kejadian ledakan-ledakan gas ini maka perlu adanya sebuah rekayasa teknologi yang mampu meminimalkan resiko adanya ledakan dengan mendeteksi lebih dini, jika terjadi kebocoran gas LPG. Rekayasa dilakukan dengan menggunakan pengupayaan teknologi di bidang elektronik, seperti teknologi sensor, mikroprosesor serta teknologi komunikasi.

Dengan latar belakang diperlukannya detektor gas sehingga mampu mereduksi bahaya ledakan gas LPG, maka dalam tugas akhir ini mengajukan gagasan sistem pendeteksi kebocoran gas LPG dengan dilengkapi alarm dan SMS peringatan.

1.2. Maksud dan Tujuan

Maksud dan tujuan dari pelaksanaan tugas akhir ini adalah merancang dan membuat sistem deteksi kebocoran gas LPG yang dilengkapi dengan fasilitas sms gateway.

1.3. Batasan Masalah

Agar cakupan dari tugas akhir ini tidak terlalu melebar, maka penulis membatasi dalam beberapa batasan masalah antara lain;

1. Merancang dan mengimplementasikan alarm detektor kebocoran gas LPG dengan sensor gas TGS 2610.
2. Merancang dan mengimplementasikan algoritma sistem alarm dengan menggunakan SMS *gateway*.
3. Menganalisa kinerja dari sistem alarm detektor kebocoran gas LPG.

1.4. Metodologi Pengerjaan

Metode pengerjaan yang digunakan dalam tugas akhir ini adalah:

1. Studi literatur
Mengumpulkan semua data dari berbagai sumber referensi seperti buku, jurnal, diklat maupun sumber-sumber yang lain yang berkaitan dalam pembuatan alat ini.
2. Perancangan dan Pembuatan Alat
Yaitu merencanakan dan membuat alat sesuai dengan yang direncanakan dan menentukan komponen-komponen yang diperlukan dalam penyusunan tuga akhir ini.
3. Pengujian dan Analisa
Yaitu melakukan pengujian dan analisa alat yang telah dibuat.

1.5. Sistematika Penulisan Laporan

Laporan tugas akhir ini, disusun dan dikelompokkan menjadi:

BAB I PENDAHULUAN

Berisi latar belakang masalah, permasalahan, maksud dan tujuan, ruang lingkup masalah, metodologi, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi dasar teori tentang perangkat keras dan perangkat lunak yang digunakan dalam pembuatan dan penyelesaian tugas akhir.

BAB III PERANCANGAN SISTEM

Berisi uraian mengenai perancangan perangkat keras dan perangkat lunak dari keseluruhan sistem.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

Berisi data-data pengamatan pengujian pada bagian-bagian tertentu dari keseluruhan rangkaian, serta pembahasan atau analisa data hasil pengujian, dengan melakukan perbandingan terhadap teori yang mendukung.

BAB V PENUTUP

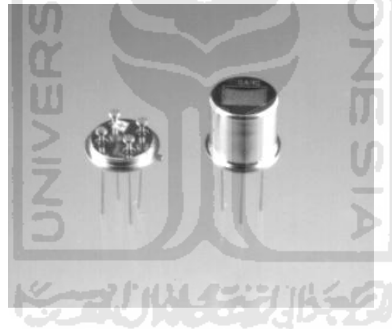
Berisi tentang kesimpulan secara teori maupun praktek, dan saran-saran yang dianggap perlu diperhatikan sehubungan dengan pembuatan alat tersebut.

BAB II

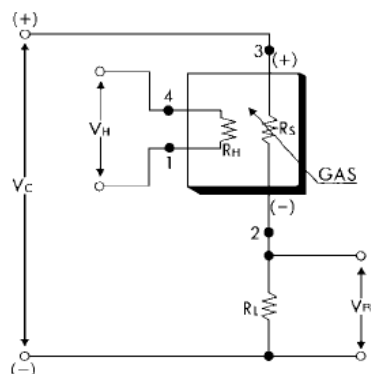
TINJAUAN PUSTAKA

2.1. Sensor gas LPG

Sensor gas LPG adalah sensor yang mampu mendeteksi keberadaan gas LPG di udara. Salah satu seri sensor yang dapat mendeteksi gas LPG adalah sensor TGS2610. Sensor ini memiliki prinsip kerja, terjadi perubahan resistansi internal sensor saat sensor mendeteksi adanya konsentrasi gas LPG di udara. Gambar 2.1 menunjukkan bentuk fisik dari sensor gas TGS2610. Gambar 2.2 menunjukkan skema perkawatan dari sensor TGS2610.

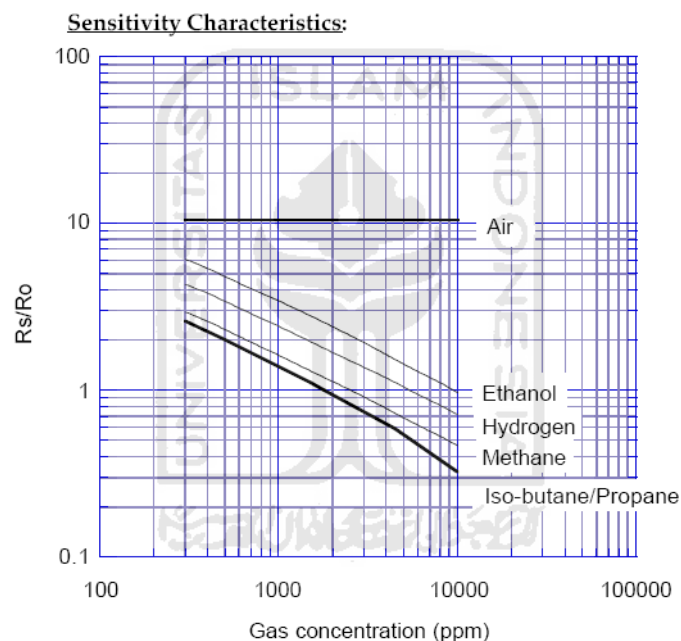


Gambar 2.1 Bentuk fisik sensor gas LPG



Gambar 2.2 Skema system elektronik sensor gas LPG

Sensor Gas TGS2610 ini sensitive terhadap gas propane dan butane dan sangat ideal untuk aplikasi monitoring gas LPG. Sensor TGS2610 ini memiliki konsumsi daya yang rendah dan sudah dilengkapi dengan *heater* sebagai pemanas saat proses pembacaan konsentrasi gas. Sensor TGS2610 ini memiliki *sensor resistance ratio* atau perbandingan dari resistansi R_s/R_o yang akan berubah berdasarkan konsentrasi gas yang masuk ke permukaan sensor. Gambar 2.3 adalah tabel perbandingan R_s/R_o pada gas butane



Gambar 2.3 Grafik perbandingan R_s/R_o sensor gas TGS2610

2.2. Mikrokontroler Atmega8535

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus klock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan *mode compare*, *interrupt internal dan eksternal*, serial UART, *programmable Watchdog Timer*, dan *mode power saving*. AVR

juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Atmega8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus clock, Atmega8535 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi komsumsi daya versus kecepatan proses. Beberapa keistimewaan dari AVR Atmega8535 antara lain:

1. *Advanced RISC Architecture*

- *131 Powerful Instructions, Most Single Clock Cycle Execution*
- *32 x 8 General Purpose Fully Static Operation*
- *Up to 16 MIPS Throughput at 16 MHz*
- *On-chip 2-cycle Multiplier*

2. *Nonvolatile Program and Data Memories*

- *8K Bytes of In-System Self-Programmable Flash*
- *Optional Boot Code Section with Independent Lock Bits*
- *512 Bytes EEPROM*
- *1KBytes Internal SRAM*
- *Programming Lock for Software Security*

3. *Peripheral Features*

- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes*
- *Two 16-bit Timer/Counters with Separate Prescalers, Compare Modes, and Capture Modes*
- *Real Time Counter with Separate Oscillator*

- *Six PWM Channels*
- *Master/Slave SPI Serial Interface*
- *Programmable Watchdog Timer with Separate On-chip Oscillator*
- *On-chip Analog Comparator*

4. Special Microcontroller Features

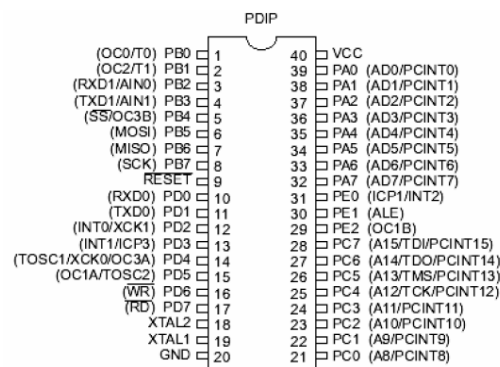
- *Power-on Reset and Programmable Brown-out Detection*
- *Internal Calibrated RC Oscillator*
- *External and Internal Interrupt Sources*
- *Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby*

5. I/O and Package

- *32 Programmable I/O Lines*
- *40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF*

6. Operating Voltages

- *2.7 - 5.5V for Atmega8535L*
- *4.5 - 5.5V for Atmega16*



Gambar 2.4 Konfigurasi pin Atmega8535

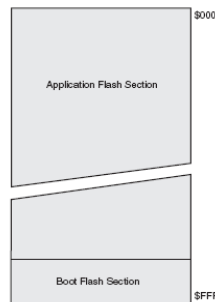
2.2.1. Peta Memory

Register keperluan umum (*general purpose*) menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti kontrol register, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan pada Gambar 2.5



Gambar 2.5 Konfigurasi memori data AVR ATmega8535

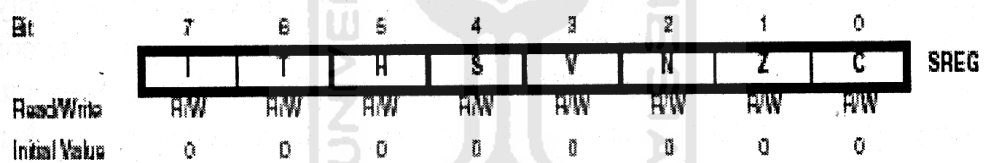
Memori program yang terletak dalam *Flash* PEROM tersusun dalam *word* atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR ATmega8535 memiliki 4Kbyte 16-bit *Flash* PEROM dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12-bit *Program Counter* (PC) sehingga mampu mengalami isi *Flash*. Konfigurasi memori program ditunjukkan pada Gambar 2.6.



Gambar 2.6 Memori program AVR ATmega8535

2.2.2. Status register

Register Status adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian inti CPU mikrokontroler. Gambar 2.7 menunjukkan register SREG.



Gambar 2.7 Status register ATmega8535

- Bit 7-I: *Global Interrupt Enable*

Bit harus diset untuk meng-*enable* interupsi. Setelah itu, dapat mengaktifkan interupsi yang akan digunakan dengan cara meng-*enable* bit kontrol register yang bersangkutan secara individu. Bit akan di-*clear* apabila terjadi suatu interupsi yang dipicu oleh *hardware*, dan bit tidak akan mengizinkan terjadinya interupsi, serta akan diset kembali oleh instruksi TERI.

- Bit 6-T: *Bit Copy Storage*

Instruksi BLD dan BST menggunakan bit-T sebagai sebuah sumber atau tujuan dalam operasi bit. Suatu bit dalam sebuah register GPR dapat disalin ke bit T menggunakan instruksi BST, dan sebaliknya bit T dapat disalin kembali ke suatu bit dalam register GPR menggunakan instruksi BLD.

- Bit 5-H: *Half Carry Flag*

- Bit 4-S: *Sign Bit*

Bit-S merupakan hasil operasi EOR antara *flag-N* (negatif) dan *flag V* (komplemen dua *overflow*).

- Bit 3-V: *Two's Complement Overflow Flag*

Bit ini berguna untuk mendukung operasi aritmetika.

- Bit 2-N: *Negative Flag*

Apabila suatu operasi menghasilkan bilangan negatif, maka *flag-N* akan diset.

- Bit 1-Z: *Zero Flag*

Bit ini akan diset bila hasil operasi yang diperoleh adalah nol.

- Bit 0-C : *Carry Flag*

Apabila suatu operasi menghasilkan carry, maka bit akan diset.

2.2.3 General purpose register

Register file ini dioptimalkan untuk AVR meningkatkan RISC *instruction set*, agar memperoleh *performance* dan *flexibility*. Gambar 2.8 menunjukkan register general purpose

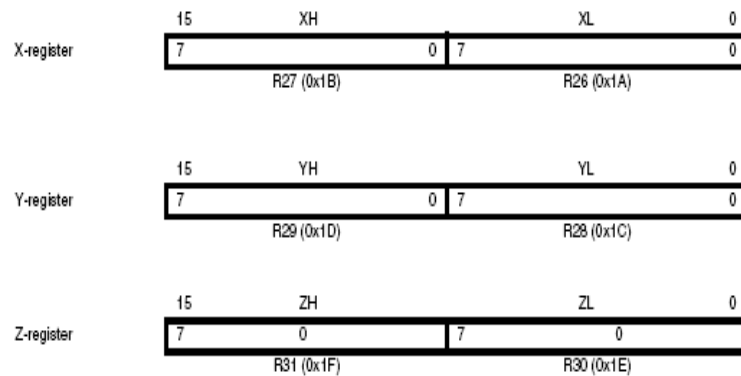
| | 7 | 0 | Addr. | |
|-----------------------------------|-----|---|-------|----------------------|
| General Purpose Working Registers | R0 | | 0x00 | |
| | R1 | | 0x01 | |
| | R2 | | 0x02 | |
| | ... | | | |
| | R13 | | 0x0D | |
| | R14 | | 0x0E | |
| | R15 | | 0x0F | |
| | R16 | | 0x10 | |
| | R17 | | 0x11 | |
| | ... | | | |
| | R26 | | 0x1A | X-register Low Byte |
| | R27 | | 0x1B | X-register High Byte |
| | R28 | | 0x1C | Y-register Low Byte |
| | R29 | | 0x1D | Y-register High Byte |
| | R30 | | 0x1E | Z-register Low Byte |
| | R31 | | 0x1F | Z-register High Byte |

Gambar 2.8 Ke-32 Register *general purpose*

Banyak operasi instruksi-instruksi pada *Register File* mempunyai akses langsung ke semua register. Dan banyak instruksi-instruksi yang merupakan instruksi *single cycle*.

Pada gambar diatas, setiap register di berikan alamat memori data dan memetakan alamat-alamat memori data tersebut secara langsung ke dalam 32 lokasi pada bagian User Data. Walaupun tidak secara fisis diimplementasikan sebagai lokasi SRAM, pengorganisasian memori ini memberikan tingkat *flexibility* yang tinggi dalam mengakses register-register tersebut seperti pada register *pointer X-, Y-, Z-* dapat diset untuk menunjuk satu atau beberapa register di *register file*.

Register 26 sampai dengan 31 mempunyai beberapa fungsi tambahan Register-register ini adalah register 16-bit sedang alamat register-register ini untuk pengalamatan tidak langsung pada *data space*. Tiga register *indirect adres*, yaitu X,Y, dan Z diperlihatkan gambar dibawah ini. Gambar 2.9 menunjukkan register X,Y,Z.



Gambar 2.9 Register X-, Y- dan Z-

2.2.4 Stack Pointer

Stack pointer merupakan suatu bagian dari AVR yang berguna untuk menyimpan data sementara, variabel lokal, dan alamat kembali dari suatu interupsi ataupun subrutin. *Stack pointer* diwujudkan sebagai dua unit register yaitu SPH dan SPL. Saat awal maka SPH dan SPL akan bernilai 0, sehingga perlu diinisialisasi terlebih dahulu jika diperlukan.

2.2.5 Timer dan Counter

AVR ATmega8535 memiliki tiga buah *timer*, yaitu *Timer/Counter 0* (8 bit), *Timer/Counter 1* (16 bit), dan *Timer/Counter 2* (8 bit).

a. Timer/Counter0

Timer/Counter 0 adalah 8-bit *Timer/Counter* yang multifungsi.

Deskripsi untuk *Timer/Counter 0* pada Atmega8535 adalah sebagai berikut:

- 1) Sebagai *Counter* 1 kanal.
- 2) *Timer* di-nol-kan saat *match compare (auto reload)*.
- 3) Dapat menghasilkan gelombang PWM dengan *glitch-free*.
- 4) *Frekuensi generator*.

- 5) *Prescaler* 10 bit untuk *timer*.
- 6) Intrupsi *timer* yang disebabkan *timer overflow* dan *match compare*.

Pengaturan *Timer/Counter* 0 diatur oleh TCCR0 (*Timer/Counter* control Register 0) yang dapat dilihat pada Gambar 2.9.



Gambar 2.9 Register TCCR0

Penjelasan untuk tiap bit-bitnya:

- Bit 7 – FOC0: *Force Output Compare*.
- Bit 6,3 –WGM0:WGM00: *Waveform generation Unit*.

Bit ini mengontrol kenaikan isi *counter*, sumber nilai maksimum *counter*, dan tipe jenis *timer/counter* yang dihasilkan, yaitu mode normal, *clear timer*, mode *compare match*, dan dua tipe dari PWM (*Pulse Width Modulation*). Tabel 2.2 berikut adalah *setting* pada bit ini untuk menghasilkan mode tertentu:

Tabel 2.2 Konfigurasi Bit WGM01 dan WGM00

| Mode | WGM01 (CTC0) | WGM00 (PWM0) | Timer/Counter Mode of Operation | TOP | Update of OCR0 | TOV0 Flag Set on |
|------|-----------------|-----------------|------------------------------------|------|-------------------|---------------------|
| 0 | 0 | 0 | Normal | 0xFF | Immediate | MAX |
| 1 | 0 | 1 | PWM, Phase Correct | 0xFF | TOP | BOTTOM |
| 2 | 1 | 0 | CTC | OCR0 | Immediate | MAX |
| 3 | 1 | 1 | Fast PWM | 0xFF | TOP | MAX |

- Bit 5, 4 – COM01:COM00: *Compare Match Output Mode*

Bit ini mengontrol pin OC0 (*Output Compare pin*). Apabila kedua bit ini nol atau *clear* maka pin OC0 berfungsi sebagai pin biasa tetapi bila

salah satu bit set. Maka fungsi pin ini tergantung pada setting bit pada WGM00 dan WGM01. Berikut Tabel 2.3 sampai dengan Tabel 2.6 adalah tabel *setting* bit ini sesuai *setting* bit pada WGM00 dan WGM01

Tabel 2.3 Konfigurasi Bit COM01 dan COM00 *Compare Output Mode non PWM*

| COM01 | COM00 | Description |
|-------|-------|--|
| 0 | 0 | Normal port operation, OC0 disconnected. |
| 0 | 1 | Toggle OC0 on Compare Match |
| 1 | 0 | Clear OC0 on Compare Match |
| 1 | 1 | Set OC0 on Compare Match |

Tabel 2.4 Konfigurasi Bit COM01 dan COM00 *Compare Output Mode Fast PWM*

| COM01 | COM00 | Description |
|-------|-------|--|
| 0 | 0 | Normal port operation, OC0 disconnected. |
| 0 | 1 | Reserved |
| 1 | 0 | Clear OC0 on Compare Match, set OC0 at TOP |
| 1 | 1 | Set OC0 on Compare Match, clear OC0 at TOP |

Tabel 2.5 Konfigurasi Bit COM01 dan COM00 *Compare Output Mode Phase Correct PWM*

| COM01 | COM00 | Description |
|-------|-------|---|
| 0 | 0 | Normal port operation, OC0 disconnected. |
| 0 | 1 | Reserved |
| 1 | 0 | Clear OC0 on Compare Match when up-counting. Set OC0 on Compare Match when down-counting. |
| 1 | 1 | Set OC0 on Compare Match when up-counting. Clear OC0 on Compare Match when down-counting. |

- Bit 2, 1, 0 – CS02; CS01, CS00: *Clock Selec*

- Ketiga bit ini untuk memilih sumber detak yang akan digunakan oleh *Timer/Counter*, Tabel 2.6 berikut menampilkan konfigurasi pemilihan sumber detak

Tabel 2.6 Konfigurasi Bit *Clock Select* untuk memilih sumber detak

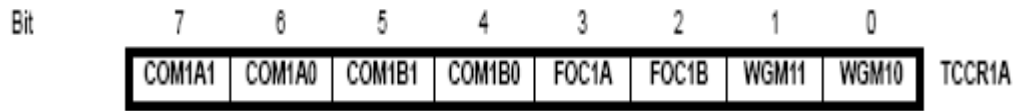
| CS02 | CS01 | CS00 | Description |
|------|------|------|---|
| 0 | 0 | 0 | No clock source (Timer/counter stopped). |
| 0 | 0 | 1 | $clk_{IO}/(No\ prescaling)$ |
| 0 | 1 | 0 | $clk_{IO}/8$ (From prescaler) |
| 0 | 1 | 1 | $clk_{IO}/64$ (From prescaler) |
| 1 | 0 | 0 | $clk_{IO}/256$ (From prescaler) |
| 1 | 0 | 1 | $clk_{IO}/1024$ (From prescaler) |
| 1 | 1 | 0 | External clock source on T0 pin. Clock on falling edge. |
| 1 | 1 | 1 | External clock source on T0 pin. Clock on rising edge. |

b. Timer Counter1

Timer/Counter1 adalah 16-bit *Timer/Counter* yang memungkinkan program pewaktuan lebih akurat. Gambar 8 menunjukkan Register TCCR1A Berbagai fitur dari *Timer/Counter1* sebagai berikut:

- 1) Desain 16 bit (juga memungkinkan 16 bit PWM).
- 2) Dua unit *compare* .
- 3) Dua unit register pembanding.
- 4) Satu unit *input capture unit*.
- 5) *Timer* dinolkan saat *match compare (autoreload)*.
- 6) Dapat menghasilkan gelombang PWM dengan *glitch-free*.
- 7) Periode PWM yang dapat diubah-ubah.
- 8) Pembangkit frekuensi.
- 9) Empat buah sumber interupsi (TOV1, OCF1A, OCF1B dan ICF1).

Pengaturan pada *Timer/Counter1* diatur melalui register TCCR1A



Gambar 2.10 Register TCCR1A

Penjelasan untuk tiap bit-bitnya:

- Bit 7:6 – COM1A1:0: *Compare Output Mode* untuk *channel A*.
- Bit 5:4 – COM1B1:0: *Compare Output Mode* untuk *channel B*.

Register COM1A1:0 dan COM1B1:0 mengontrol kondisi Pin Output *Compare* (OC1A dan OC1B). Jika salah satu atau kedua bit pada register COM1A1:0 ditulis menjadi satu maka kaki pin OC1A tidak berfungsi normal sebagai port I/O. Begitu juga pada register COM1B1:0 ditulis menjadi satu maka kaki pin OC1B juga tidak berfungsi normal sebagai port I/O. Fungsi pada pin OC1A dan OC1B tergantung pada setting bit pada register WGM13:0 diset sebagai mode PWM atau mode non-PWM.

- Bit 3 – FOC1A: *Force Output Compare* untuk *channel A*.
- Bit 2 – FOC1B: *Force Output Compare* untuk *channel B*.
- Bit 1:0 – WGM1 1:0: *Waveform Generation Mode*.

Dikombinasikan dengan bit WGM13:2 yang terdapat pada register TCCR1B, bit ini mengontrol urutan pencacah dari *counter*, sumber maksimum (TOP) nilai *counter*, dan tipe dari gelombang yang dibangkitkan. Mode yang dapat dilakukan antara lain: mode normal,

mode *Clear Timer on Compare Match* (CTC) dan tiga tipe mode PWM. Setingan mode dapat dilihat pada tabel berikut ini.

Tabel 2.7 Konfigurasi Bit *Compare Output Mode non PWM*

| COM1A1/ COM1B1/ | COM1A0/ COM1B0/ | Keterangan |
|--------------------|--------------------|--|
| 0 | 0 | Operasi port normal, OC1A/OC1B tidak dihubungkan. |
| 0 | 1 | <i>Toggle</i> OC1A/OC1B saat proses perbandingan tercapai. |
| 1 | 0 | <i>Clear</i> OC1A/OC1B saat proses perbandingan tercapai (set keluaran ke level rendah). |
| 1 | 1 | Set OC1A/OC1B saat proses perbandingan tercapai (set keluaran ke level tinggi). |

Tabel 2.8 Konfigurasi Bit *Compare Output Mode Fast PWM*

| COM1A1/ COM1B1/ | COM1A0/ COM1B0/ | Keterangan |
|--------------------|--------------------|--|
| 0 | 0 | Operasi port normal, OC1A/OC1B tidak dihubungkan. |
| 0 | 1 | WGM13:0= 15: <i>toggle</i> OC1A saat proses perbandingan tercapai dan OC1B tidak terhubung. Untuk semua pengaturan WGM1 yang lain, operasi port normal, OC1A/OC1B tidak terhubung. |
| 1 | 0 | <i>Clear</i> OC1A/OC1B saat proses perbandingan tercapai, set OC1A/OC1B pada nilai <i>TOP</i> . |
| 1 | 1 | Set OC1A/OC1B saat proses perbandingan tercapai, <i>clear</i> OC1A/OC1B pada nilai <i>TOP</i> . |

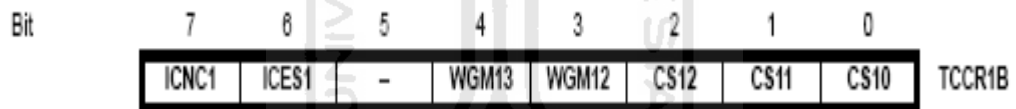
Tabel 2.9 Konfigurasi Bit *Compare Output Mode Phase Correct dan Frequency Correct PWM*

| COM1A1/ COM1B1/ | COM1A0/ COM1B0/ | Keterangan |
|--------------------|--------------------|---|
| 0 | 0 | Operasi port normal, OC1A/OC1B tidak dihubungkan. |
| 0 | 1 | WGM13:0= 9 atau 14: <i>toggle</i> OC1A saat proses perbandingan tercapai dan OC1B tidak terhubung. |
| 1 | 0 | <i>Clear</i> OC1A/OC1B saat proses perbandingan tercapai ketika mencacah naik. Set OC1A/OC1B saat proses perbandingan tercapai ketika mencacah turun. |
| 1 | 1 | Set OC1A/OC1B saat proses perbandingan tercapai ketika mencacah naik. <i>Clear</i> OC1A/OC1B saat proses perbandingan tercapai ketika mencacah turun. |

Tabel 2.10 Konfigurasi bit *Clock Select* untuk memilih sumber detak

| Mode | WGM13 | WGM12 (CTC1) | WGM11 (PWM11) | WGM10 (PWM10) | Mode Operasi Timer/Counter | TOP | Update OCR1x saat | TOV1 flag diset saat |
|------|-------|--------------|---------------|---------------|----------------------------------|--------|-------------------|----------------------|
| 0 | 0 | 0 | 0 | 0 | Normal | 0xFFFF | Immediate | MAX |
| 1 | 0 | 0 | 0 | 1 | PWM, Phase Correct, 8 | 0x00FF | TOP | BOTTOM |
| 2 | 0 | 0 | 1 | 0 | PWM, Phase Correct, 9 | 0x01FF | TOP | BOTTOM |
| 3 | 0 | 0 | 1 | 1 | PWM, phase correct, 10 | 0x03FF | TOP | BOTTOM |
| 4 | 0 | 1 | 0 | 0 | CTC | OCR1A | Immediate | MAX |
| 5 | 0 | 1 | 0 | 1 | Fast PWM, 8 bit | 0x00FF | TOP | TOP |
| 6 | 0 | 1 | 1 | 0 | Fast PWM, 9 bit | 0x01FF | TOP | TOP |
| 7 | 0 | 1 | 1 | 1 | Fast PWM, 10 bit | 0x03FF | TOP | TOP |
| 8 | 1 | 0 | 0 | 0 | PWM, Phase and Frequency Correct | ICR1 | BOTTOM | BOTTOM |
| 9 | 1 | 0 | 0 | 1 | PWM, Phase and | OCR1A | BOTTOM | BOTTOM |
| 10 | 1 | 0 | 1 | 0 | PWM, Phase Correct | ICR1 | TOP | BOTTOM |
| 11 | 1 | 0 | 1 | 1 | PWM, Phase Correct | OCR1A | TOP | BOTTOM |
| 12 | 1 | 1 | 0 | 0 | CTC | ICR1 | Immediate | MAX |
| 13 | 1 | 1 | 0 | 1 | Reserved | - | - | - |
| 14 | 1 | 1 | 1 | 0 | Fast PWM | ICR1 | TOP | TOP |
| 15 | 1 | 1 | 1 | 1 | Fast PWM | OCR1A | TOP | TOP |

Pengaturan Timer 1 juga diatur oleh register TCCR1B



Gambar 2.11 Register TCCR1B

- Bit 7 – ICNC1: *Input Capture Noise Canceller*.
- Bit 6 – ICES1: *Input Capture Edge Select*.
- *Reserved Bit*.
- Bit 4:3 – WGM1 1:3: *Waveform Generation Mode*.
- Bit 2:0 – CS12:0: *Clock Select*.

Ketiga bit ini mengatur sumber detak yang digunakan untuk *Timer/Counter1*. Untuk settingnya dapat dilihat pada tabel dibawah ini.

Tabel 2.11 Konfigurasi bit *Clock Select* untuk memilih sumber detak

| CS12 | CS11 | CS10 | Keterangan |
|------|------|------|---|
| 0 | 0 | 0 | Tanpa sumber detak (<i>timer/counter</i> tidak difungsikan) |
| 0 | 0 | 1 | $clk_{IO}/1$ (tanpa <i>prescaling</i>) |
| 0 | 1 | 0 | $clk_{IO}/8$ (dari <i>prescaling</i>) |
| 0 | 1 | 1 | $clk_{IO}/64$ (dari <i>prescaling</i>) |
| 1 | 0 | 0 | $clk_{IO}/256$ (dari <i>prescaling</i>) |
| 1 | 0 | 1 | $clk_{IO}/1024$ (dari <i>prescaling</i>) |
| 1 | 1 | 0 | Sumber detak eksternal pin T1. Detak pada <i>falling edge</i> . |
| 1 | 1 | 1 | Sumber detak eksternal pin T1. Detak pada <i>rising edge</i> . |
| | | | |

c. Timer Counter2

Timer/Counter 2 adalah 8-bit *Timer/Counter* yang multifungsi.

Deskripsi untuk *Timer/Counter 0* pada ATMega8535 adalah sebagai berikut:

- 1) Sebagai *Counter* 1 kanal.
- 2) Pewaktu di-nol-kan saat *match compare (autoreload)*.
- 3) Dapat menghasilkan gelombang PWM dengan *glitch-free*.
- 4) Frekuensi generator.
- 5) *Prescaler* 10 bit untuk pewaktu.
- 6) Intrupsi *timer* yang disebabkan *timer overflow* dan *match compare*.

Pengaturan *Timer/Counter 2* diatur oleh TCCR2 (*Timer/Counter control Register 0*) yang dapat dilihat pada Gambar 2.12.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|------|-------|-------|-------|-------|------|------|------|-------|
| FOC2 | WGM20 | COM21 | COM20 | WGM21 | CS22 | CS21 | CS20 | TCCR2 |
| W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Gambar 2.12 Register TCCR2

Penjelasan untuk tiap bit-bitnya:

- Bit 7 – FOC2: *Force Output Compare*.
- Bit 6,3 –WGM21:WGM20: *Waveform generation Unit*.

Bit ini mengontrol kenaikan dari *counter*, sumber dari nilai maksimum *counter*, dan tipe dari jenis timer/conter yang dihasilkan yaitu mode normal, *clear timer*, mode *compare match*, dan dua tipe dari PWM (*Pulse Width Modulation*). Berikut tabel seting pada bit ini untuk menghasilkan mode tertentu:

Tabel 2.12 Konfigurasi Bit WGM21 dan WGM20

| Mode | WGM21 (CTC2) | WGM20 (PWM2) | Timer/Counter Mode of Operation | TOP | Update of OCR2 | TOV2 Flag Set on |
|------|--------------|--------------|---------------------------------|------|----------------|------------------|
| 0 | 0 | 0 | Normal | 0xFF | Immediate | MAX |
| 1 | 0 | 1 | PWM, Phase Correct | 0xFF | TOP | BOTTOM |
| 2 | 1 | 0 | CTC | OCR2 | Immediate | MAX |
| 3 | 1 | 1 | Fast PWM | 0xFF | TOP | MAX |

- Bit 5, 4 – COM01:COM00: *Compare Match Output Mode*.

Bit ini mengontrol pin OC0 (*Output Compare pin*). Apabila kedua bit ini nol atau clear maka pin OC0 berfungsi sebagai pin biasa tetapi bila salah satu bit set. Maka fungsi pin ini tergantung dari seting bit pada WGM00 dan WGM01. Berikut daftar tabel setting bit ini sesuai seting bit pada WGM00 dan WGM01

- Bit 2, 1, 0 – CS22; CS21, CS20: *Clock Select*.

Ketiga bit ini untuk memilih sumber detak yang akan digunakan oleh *Timer/Counter*.

Tabel 2.13 Konfigurasi Bit COM21 dan COM20 *Compare Output Mode non PWM*

| COM21 | COM20 | Description |
|-------|-------|--|
| 0 | 0 | Normal port operation, OC2 disconnected. |
| 0 | 1 | Toggle OC2 on Compare Match. |
| 1 | 0 | Clear OC2 on Compare Match. |
| 1 | 1 | Set OC2 on Compare Match. |

Tabel 2.14 Konfigurasi Bit COM21 dan COM20 *Compare Output Mode Fast PWM*

| COM21 | COM20 | Description |
|-------|-------|---|
| 0 | 0 | Normal port operation, OC2 disconnected. |
| 0 | 1 | Reserved |
| 1 | 0 | Clear OC2 on Compare Match, set OC2 at TOP. |
| 1 | 1 | Set OC2 on Compare Match, clear OC2 at TOP. |

Tabel 2.15 Konfigurasi Bit COM21 dan COM20 *Compare Output Mode Phase Correct PWM*

| COM21 | COM20 | Description |
|-------|-------|---|
| 0 | 0 | Normal port operation, OC2 disconnected. |
| 0 | 1 | Reserved |
| 1 | 0 | Clear OC2 on Compare Match when up-counting. Set OC2 on Compare Match when down-counting. |
| 1 | 1 | Set OC2 on Compare Match when up-counting. Clear OC2 on Compare Match when down-counting. |

2.3. SMS (Short Message Service)

Semua perintah untuk mengendalikan ponsel menggunakan perintah *AT command*. Beberapa *AT Command* yang penting untuk mengakses SMS antara lain sebagai berikut:

1. AT+CMGR

Perintah ini digunakan untuk membaca isi SMS. Untuk mengetahui kesiapan sistem, bisa digunakan perintah ‘AT+CMGS=?’, jika respon dari ponsel

adalah 'OK', maka ponsel siap dibaca. Untuk menjalankan perintah ini, sintaks penulisan yang dipakai adalah sebagai berikut:

```
AT+CMGR=<index>
```

Index adalah lokasi memori yang akan dibaca isi SMS nya., misal AT+CMGR=1, hal ini menandakan SMS yang dibaca adalah lokasi nomor 1.

Jika perintah ini berhasil, maka ponsel akan merespon dengan parameter-parameter sebagai berikut:

```
+CMGR: <stat>, [<alpha>], <length><CR><LF><pdu>
```

2. AT+CMGS

Perintah ini digunakan untuk mengirim SMS. Untuk mengetahui kesiapan ponsel, bisa digunakan *test command* AT+CMGS=?. Jika ponsel siap maka akan merespon dengan 'OK'. Sintaks penulisan perintah yang digunakan adalah sebagai berikut:

```
AT+CMGS=<length>
```

<length> diisi dengan jumlah karakter yang akan dikirimkan. Jika perintah berhasil, maka akan ponsel akan merespon dengan tanda '>', kemudian kode PDU untuk SMS kirim siap diisikan.

3. AT+CMGD

Perintah ini digunakan untuk menghapus isi SMS. Sintaks penulisan yang digunakan untuk menjalankan perintah ini, adalah

```
AT+CMGD=<index>
```

Indeks adalah lokasi memori dari SMS yang akan dihapus. Jika perintah ini berhasil, maka ponsel akan merespon dengan 'OK'.

AT Command untuk SMS, biasanya diwakili oleh data dalam format PDU (*Protocol Data Unit*). Data dalam format PDU mempunyai beberapa *header*. *Header-header* untuk pengiriman SMS dan penerimaan SMS berbeda. Dalam format PDU, data yang dikirim tidak hanya data yang berisi pesan saja, namun ada beberapa informasi mengenai pengirimnya, seperti nomor pengirim, *service center* SMS, waktu pengiriman, dsb. Adapun format PDU untuk tersebut terbagi dalam format 2 bagian, yaitu:

2.3.1 PDU untuk kirim SMS

PDU untuk SMS terima terdiri dari 8 *header*, dan masing-masing adalah:

1) Nomor SMS Center

Header yang pertama ini dibagi dalam tiga sub *header*, yaitu jumlah pasangan hexadesimal SMS center dalam format hexadesimal. *Header* yang kedua adalah *header* untuk tanda pengiriman wilayah nasional dan internasional. Untuk kode nasional kode *header*-nya 81 dan untuk kode internasional kode *header*-nya 91. yang ketiga adalah pasangan nomor SMS center yang dibalik. Apabila salah satu angka tidak memiliki pasangan maka dipasangkan dengan bilangan F.

Dibawah ini disajikan contoh *header SMS center*

SMS center dengan nomor : 62811000000 dalam format PDU menjadi:

1. 07 : Jumlah pasangan hexadesimal SMS center
2. 91 : Kode internasional
3. 2618010000F0 : Pasangan nomor SMS center yang dibolak-balik.

Jadi jika ketiga *header* tersebut digabung menjadi 071261010000F0. Tabel 2.17 berikut adalah nomor *SMS center* beserta kode PDU untuk operator GSM di indonesia.

Tabel 2.16 *SMS center* operator GSM

| Operator | <i>SMS center</i> | Kode PDU |
|------------|-------------------|------------------|
| Telkomsel | 62811000000 | 07912618010000F0 |
| Satelindo | 62816125 | 059126181652 |
| Exelcom | 62818445009 | 07912618485400F9 |
| Indosat M3 | 62855000000 | 07912658050000F0 |

2) Tipe SMS

Untuk mengirim SMS tipe yang digunakan adalah 1, sehingga bilangan hexa desimal-nya adalah 01.

3) Nomor Referensi

Nomor ini secara otomatis diberikan oleh ponsel, sehingga harus diberikan kode 0 terlebih dahulu, sehingga pasangan bilangan hexa-nya menjadi 00.

4) Nomor Ponsel Penerima

Nomor ponsel penerima dibagi dalam tiga *header* seperti pada nomor *SMS center*. *Header* yang pertama adalah jumlah angka nomor ponsel penerima. *Header* yang kedua adalah kode nasional dengan angka 81 dan internasional dengan angka 91. *Header* yang ketiga adalah nomor ponsel penerima dengan pasangan-pasangan yang dibolak-balik. Sebagai contoh untuk nomor ponsel yang dituju 6285228391602 diubah menjadi:

1. 0D : Ada 13 angka
2. 91 : Kode internasional

3. 265822381906F2 : Nomor ponsel yang dituju dengan pasangan-pasangan yang dibalik. Jika ketiga *header* diatas digabung, menjadi 0D91265822381906F2.

5) Bentuk SMS

Bentuk SMS yang dikirim ke *SMS center* dikenal ada tiga bentuk pengiriman, yaitu:

1. 00 : dikirim sebagai SMS
2. 01 : dikirim sebagai telex
3. 02 : dikirim sebagai fax

6) Skema *Encoding*

Skema yang digunakan dalam proses pengiriman ke *SMS center* melalui ponsel GSM menggunakan skema 7 bit. Format ini ditandai dengan kode 0, dan jika diubah dalam format PDU *header* ini menjadi 00.

7) Jangka Waktu SMS

Jika bagian ini di lompai berarti jangka waktu pengirimannya tak terbatas. Jika diisi bilangan integer dan kemudian diubah dalam pasangan hexa maka bilangan tersebut akan mewakili jangka waktu pengiriman. Tabel 2.18 adalah daftar perhitungan waktu validitas.

Tabel 2.17 Waktu validitas

| Integer (INT) | Jangka waktu validitas |
|---------------|--|
| 0-143 | $(INT+1) \times 5$ menit |
| 144-167 | 12 jam + $(INT - 143) \times 30$ menit |
| 168-196 | $(INT-166) \times 1$ hari |
| 197-255 | $(INT-192) \times 1$ minggu |

8) Isi SMS

Header yang terakhir ini terdiri dari dua sub *header*, yaitu panjang isi dan isi SMS. Untuk ponsel GSM menggunakan skema encoding 7 bit. Ada dua langkah yang harus dilakukan untuk mengkonversi isi SMS, yaitu:

1. Langkah pertama, nilai biner dari kode ASCII.
2. Langkah kedua, penggeseran nilai biner.

2.3.2 PDU untuk terima SMS

PDU untuk terima SMS dari *SMS center* dibagi menjadi delapan *header* yang hampir sama dengan *PDU* untuk mengirim SMS. Kedelapan *header* tersebut adalah:

1) Nomor *SMS center*

Header yang pertama ini dibagi dalam tiga sub *header*, yaitu jumlah pasangan hexadesimal *SMS center* dalam format hexa desimal. *Header* yang kedua adalah *header* untuk tanda pengiriman wilayah nasional dan internasional. Untuk kode nasional kode *header*-nya 81 dan untuk kode internasional kode *header*-nya 91. *Header* yang ketiga adalah pasangan nomor *SMS center* yang dibalik. Apabila salah satu angka tidak memiliki pasangan maka dipasangkan dengan bilangan F.

Dibawah ini disajikan contoh *header SMS center*:

SMS center dengan nomor : 62811000000 dalam format *PDU* menjadi:

- 07 = Jumlah pasangan hexadesimal *SMS center*
- 91 = Kode internasional

- 2618010000F0= Pasangan nomor *SMS center* yang balik.

2) Tipe SMS

Untuk mengirim SMS tipe yang digunakan adalah 4, sehingga bilangan hexa desimalnya adalah 04.

3) Nomor Ponsel Penerima

Nomor ponsel penerima dibagi dalam tiga *header* seperti pada nomor *SMS center*. *Header* yang pertama adalah jumlah nomor ponsel penerima. *Header* yang kedua adalah kode nasional dengan angka 81 dan internasional dengan angka 91. *Header* yang ketiga adalah nomor ponsel penerima dengan pasangan-pasangan yang dibolak-balik. Sebagai contoh untuk nomor posel yang dituju 6285228391602 diubah menjadi

1. 0D : ada 13 angka
2. 91 : kode internasional
3. 265822381906F2 : nomor ponsel yang dituju dengan pasangan-pasangan yang dibolak-balik.

Jika ketiga *header* diatas digabung, menjadi 0D91265822381906F2.

4) Bentuk SMS

SMS terima memiliki kode 00.

5) Skema encoding

Skema yang digunakan dalam proses pengiriman ke *SMS center* melalui ponsel GSM menggunakan skema 7 bit. Format ini ditandai dengan kode 0, dan jika diubah dalam format PDU *header* ini menjadi 00.

6) Tanggal dan waktu SMS

Header tanggal dan waktu diwakili oleh 12 bilangan hexa atau 6 pasang yang berarti: yy/mm/dd hh:mm:ss, sebagai contoh 207022512380 yang berarti SMS tersebut sampai pada *SMS center* pada tanggal 22 juli 2002, pukul 15:32:08 WIB.

7) Jangka waktu

Header jangka waktu, jika tidak dibatasi dilambangkan dengan 00.

8) Isi SMS

Header ini terdiri dari dua sub *header*, yaitu panjang isi dan isi SMS.

2.4. Komunikasi Serial

Dalam dunia digital dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data secara asinkron. Pada komunikasi data serial sinkron, sinyal *clock* dikirim bersama-sama dengan data serial, sedangkan komunikasi data serial asinkron, *clock* tidak dikirim bersama-sama data serial tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim maupun pada sisi penerima. Komunikasi data serial dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial yang kemudian diubah kembali menjadi data paralel. Sebagai contoh IC UART 8250 dari intel merupakan salah satunya. Selain berbentuk IC mandiri, berbagai macam mikrokontroler ada yang dilengkapi UART, misalnya keluarga mikrokontroler MCS51.

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400 dan 9600 (bit/detik). Dalam komunikasi data serial, *baud rate* dari kedua

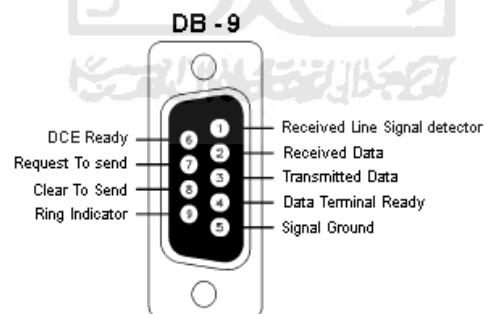
alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6, 7 atau 8 bit), paritas (genap ganjil atau tanpa paritas), dan jumlah bit 'Stop' (1, 1,5 atau 2 bit).

2.4.1. Karakteristik Port Serial.

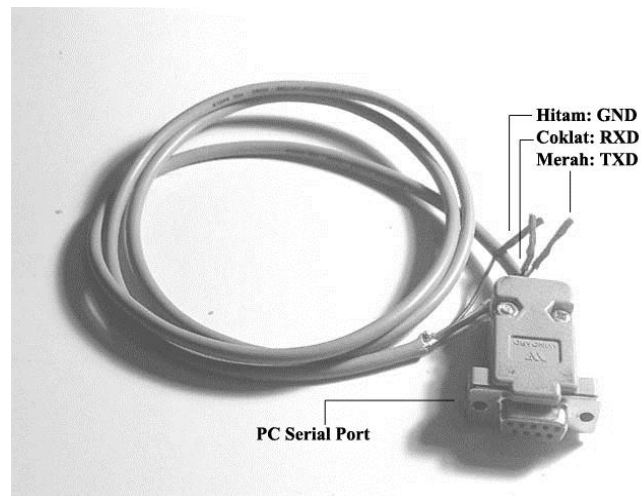
Standar sinyal serial RS232 memiliki ketentuan level tegangan sebagai berikut:

- Logika '1' disebut '*mark*' terletak antara -3 V hingga -25 V
- Logika '0' disebut '*space*' terletak antara +3 V hingga +25 V
- Daerah tegangan -3 Volt hingga +3 Volt adalah *invalid level*.

Pada komputer IBM PC kompatibel biasanya dapat ditemukan dua *conecktor port serial DB-9* yang biasa dinamai COM1 dan COM2. Pada Gambar 2.13 ditunjukkan gambar konektor DB serial DB 9 standar. Sementara itu Gambar 2.14 adalah gambar kabel serial RS232.



Gambar 2.13 Konektor serial DB-9



Gambar 2.14 Kabel serial RS232

Pada Tabel 2.19 berikut adalah fungsi dan nama dari pin-pin yang ada pada sistem komunikasi serial RS232.

Tabel 2.18 Fungsi pin konektor DB-9

| Nomor pin | Nama sinyal | Direction | Keterangan |
|-----------|-------------|-----------|---|
| 1 | DCD | In | <i>Data Carrier Detect/ Received Line Signal Detect</i> |
| 2 | RxD | In | <i>Receive Data</i> |
| 3 | TxD | Out | <i>Transmit Data</i> |
| 4 | DTR | Out | <i>Data Terminal Ready</i> |
| 5 | GND | - | <i>Ground</i> |
| 6 | DSR | In | <i>Data Set Ready</i> |
| 7 | RST | Out | <i>Request to Send</i> |
| 8 | CTS | In | <i>Clear to Send</i> |
| 9 | RI | In | <i>Ring Indicator</i> |

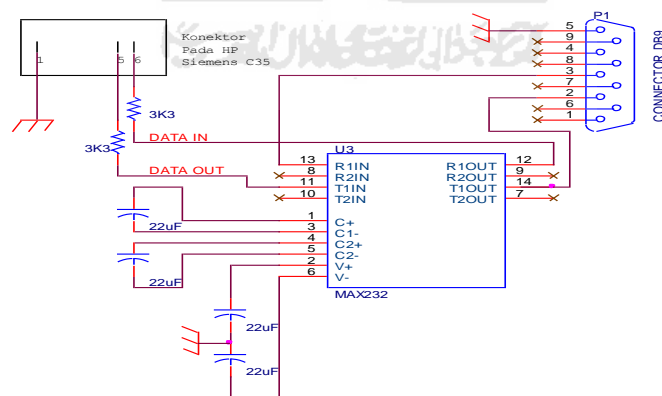
Pada Tabel 2.19 disajikan tabel fungsi masing-masing pin dari konektor DB9. Fungsi utama yang digunakan adalah pin RxD, TxD, serta pin GND. Pin tersebut merupakan syarat minimal untuk komunikasi serial.

2.4.2. Interface RS232

Pada komunikasi serial UART selain dengan sistem RS232, terkadang terdapat sistem komunikasi UART yang menggunakan sistem TTL ataupun CMOS. Dengan sistem seperti ini level tegangan yang digunakan akan berbeda dengan level tegangan RS232, sehingga umumnya perlu sebuah piranti *interface* RS232. Salah satu contoh IC yang digunakan sebagai *interface* piranti TTL/CMOS dengan piranti RS232 adalah IC MAX

2.5. Komunikasi Mikrokontroler dan HP

Untuk berkomunikasi dengan HP menggunakan mikrokontroler ataupun PC diperlukan sebuah kabel data. Jenis dari kabel data tergantung dari merk dan seri HP itu sendiri. Dalam perancangan ini, jenis HP yang digunakan adalah *Siemens C35*. Gambar 2.15 adalah model kabel data yang digunakan untuk berkomunikasi dengan PC atau mikrokontroler.



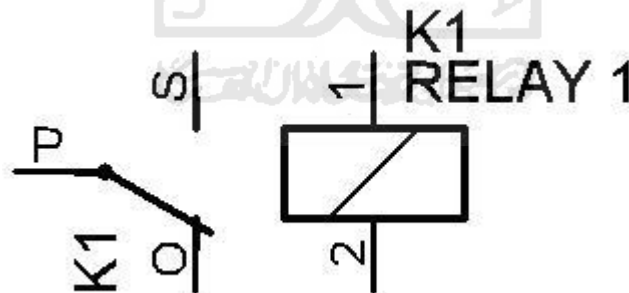
Gambar 2.15. Kabel data *Siemens C35*.

Kabel data *Siemens C35* menggunakan komponen dasar MAX232. IC ini digunakan sebagai konverter agar mampu berkomunikasi dengan PC. Pemasangan kapasitor 22uF digunakan untuk menghasilkan tegangan yang sesuai dengan level

RS232, atau biasa disebut *voltage doubler*. Komunikasi yang disediakan adalah komunikasi serial UART dengan *baud rate* 19200 yang kompatibel dengan mikrokontroler. Pemasangan Resistor $3K3\Omega$ untuk menjaga agar sinyal tidak hilang.

2.6. Driver relay

Relay merupakan salah satu komponen output yang paling sering digunakan baik pada industri, otomotif, ataupun peralatan elektronika lainnya. Relay berfungsi untuk menghubungkan atau memutus aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada koilnya. Ada 2 macam relay berdasarkan tegangan untuk menggerakkan koilnya yaitu AC dan DC. Pada relay board ini digunakan relay DC dengan tegangan koil 12V DC, arus yang diperlukan sekitar 20 - 30mA.



Gambar 2.16 Simbol relay

Pada gambar diatas adalah simbol relay. Cara kerjanya, jika kumparan di titik 1 dan dua diberi tegangan DC 12V maka kumparan ini, akan menjadi magnet. Kemudian titik P akan ditarik ke titik S, Sehingga P akan terhubung ke S dan P akan putus dengan titik O.

2.7. Transistor

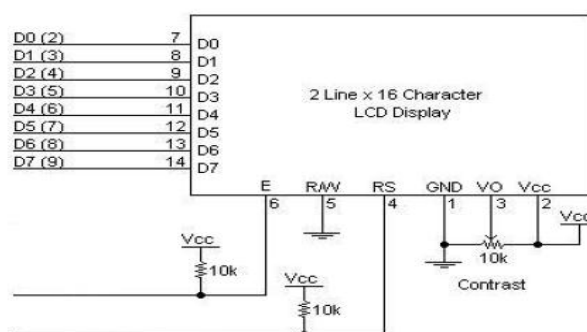
Transistor merupakan komponen aktif dimana arus, tegangan atau daya keluarannya dikendalikan oleh arus masukan. Didalam sistem komunikasi, transistor digunakan untuk menguatkan sinyal. Di dalam untaian elektronis komputer transistor digunakan untuk saklar elektronis laju tinggi.

2.8. Penampil LCD 16x2

Penampil informasi yang lazim digunakan adalah LCD (*Liquid Crystal Display*). LCD yang digunakan adalah LCD *Dot Matrik* dengan jumlah karakter 16x2. LCD ini nantinya akan digunakan untuk menampilkan informasi nilai suhu, asap dan status dari proses.

Adapun fitur yang disajikan dalam LCD ini adalah:

- Terdiri dari 16 karakter dan 2 baris
- Mempunyai 192 karakter tersimpan
- Terdapat karakter generator terprogram
- Dapat dialamati dengan mode 8-bit dan 4-bit
- Dilengkapi dengan *backlight*.



Gambar 2.12 LCD 16x2

Pada LCD ini terdiri dari 8 jalur data, 3 jalur kendali dan fasilitas pengaturan kontras serta *backlight*. LCD ini dapat dikendalikan dengan mikrokontroler atau mikroprosesor.

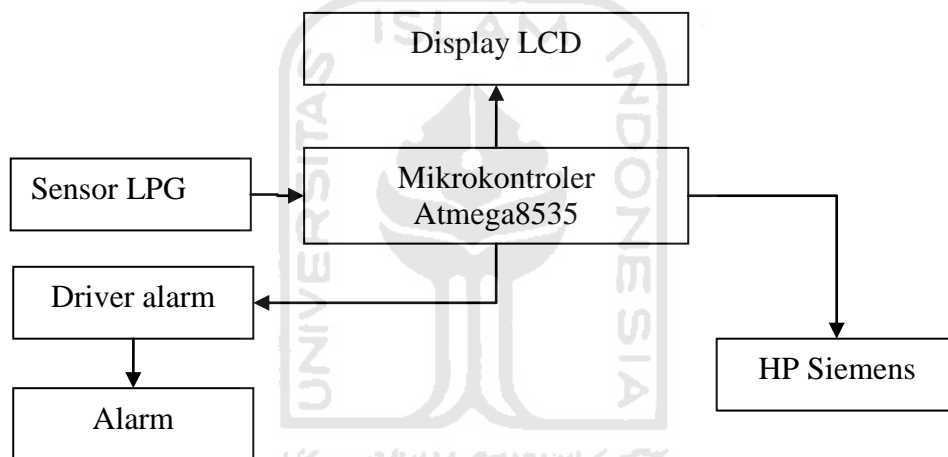


BAB III

PERANCANGAN SISTEM

3.1. Perangkat Keras

Didalam perancangan sistem monitoring kebocoran gas LPG ini dibutuhkan beberapa perangkat keras atau lebih dikenal dengan *hardware* . Gambar 3.1 berikut menjelaskan perangkat keras yang digunakan pada sistem ini.



Gambar 3.1. Diagram blok perangkat keras sistem utama

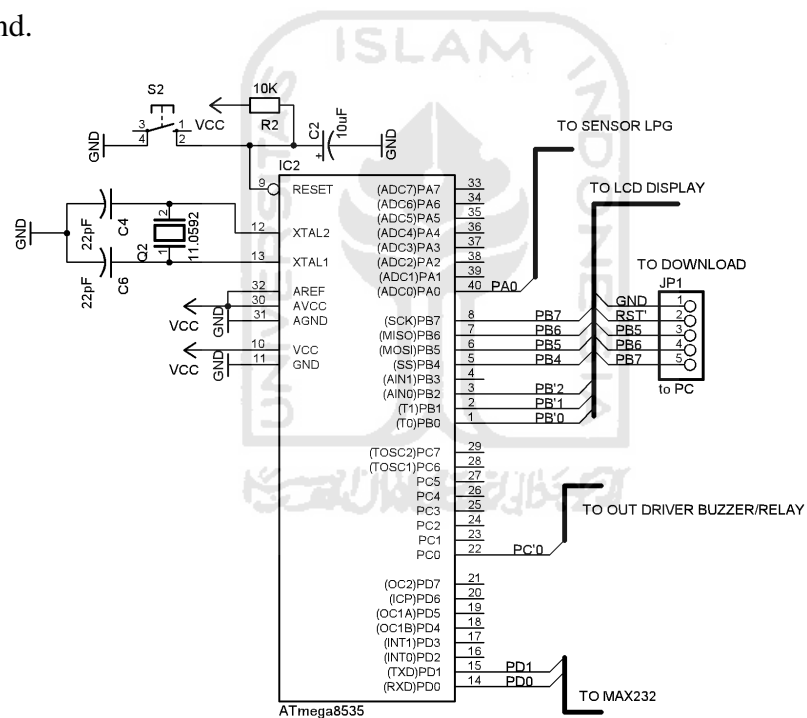
Pada Gambar 3.1 ditunjukkan diagram kotak sistem yang digunakan dalam penelitian ini. Prinsip kerjanya adalah sensor gas akan mendeteksi adanya konsentrasi gas LPG di udara. Jika terdapat gas LPG maka akan terjadi perubahan resistensi dalam sensor, sehingga akan menyebabkan adanya perubahan tegangan. Perubahan tegangan ini dibaca oleh mikrokontroler Atmega8535.

Jika batas ambang kadar LPG di terdeteksi maka sistem akan menyalakan alarm dan mengirimkan SMS ke user yang telah ditentukan. Untuk menyalakan

alarm mikrokontroler dibantu oleh driver alarm dengan memanfaatkan relay. Display LCD digunakan untuk menampilkan informasi konsentrasi gas.

3.1.1. Mikrokontroler

Tugas dari mikrokontroler ini adalah mengendalikan seluruh sistem. Membaca sensor menampilkan informasinya dalam penampil LCD dan memberikan eksekusi atau perintah ke output atau aktuator. Mikrokontroler juga mengakses HP menggunakan komunikasi serial dan perintah-perintah AT-Command.



Gambar 3.2. Sistem minimum mikrokontroler ATmega8535

Mikrokontroler Atmega8535 merupakan sebuah mikrokontroler CMOS 8-bit yang diproduksi oleh Atmel, mikrokontroler ini bekerja pada tegangan rendah dan memiliki 8kb *flash* memori didalamnya yang bersifat ISP (*In System Programmable Flash Memory*). Mikrokontroler ini sudah dilengkapi dengan

RAM, ROM, EEPROM pembangkit detak internal, *timer*, interupsi serta piranti untuk komunikasi serial. Sehingga dengan komponen-komponen ini menjadikan mikrokontroler sebagai komputer dalam satu chip.

Pada Gambar 3.2. ditunjukkan rangkaian minimal yang diperlukan agar mikrokontroler mampu bekerja. Sistem tersebut terdiri dari kristal Q2 senilai 11,0593MHz, 2 buah kapasitor senilai 33pF. Komponen ini berfungsi sebagai osilator atau pembangkit detak untuk mikrokontroler. Nilai x-tal 11,0592MHz diatur dengan pertimbangan *baud rate* yang akan digunakan pada komunikasi serial dengan HP. Jika menggunakan nilai ini maka tidak akan terjadi *error* pada saat komunikasi dengan HP. Jika menggunakan x-tal dengan nilai lain, maka perhitungan nilai *baud rate* tidak akan sesuai dengan kata lain akan menghasilkan *error*.

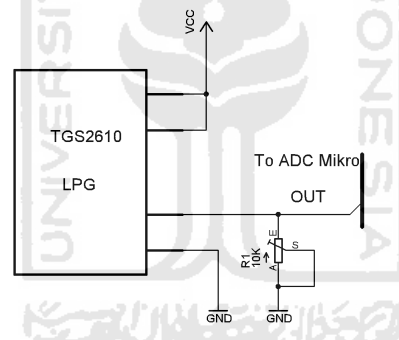
Sistem reset menggunakan kapasitor bernilai 10uF/16V dan sebuah resistor senilai 10kΩ. Dengan pemasangan kapasitor dan resistor ini maka pada saat *power supply* dinyalakan maka mikrokontroler akan *reset* secara otomatis, kemudian bekerja secara normal. Hal ini dikarenakan adanya proses pengisian dan pengosongan pada komponen kapasitor. Saklar SW2 berguna untuk *me-reset* mikrokontroler secara manual. Prinsip kerjanya adalah memberikan logika redah ke kaki RST maka mikrokontroler akan *reset*.

Mikrokontroler Atmega8535 dirancang untuk mengendalikan seluruh sistem. Piranti masukan dan keluaran diatur sesuai dengan Gambar 3.2 diatas. PB digunakan sebagai jalur data ke LCD, jalur ini merupakan jalur keluaran mikrokontroler. PA.0 adalah jalur ADC (*Analog to Digital Converter*) yang

digunakan untuk membaca sensor LPG TGS2610. Jalur komunikasi dengan HP menggunakan jalur khusus yaitu PD.0 dan PD.1 atau jalur Tx dan Rx. Sementara itu untuk jalur PC.0 digunakan untuk kontrol kontrol alarm.

3.1.2. Sensor Gas LPG TGS2610

Sensor yang digunakan untuk mendeteksi keberadaan Gas LPG di udara adalah sensor TGS2610. Sensor ini memiliki sifat terjadi perubahan R_s/R_o saat terdapat perubahan konsentrasi Gas LPG di permukaan sensor. Besarnya perubahan R_s/R_o tergantung dari banyaknya konsentrasi LPG di udara. Gambar 3.3 menunjukkan Gambar skema sensor LPG TGS2610.



Gambar 3.3 Sensor TGS2610

Keluaran sensor TGS2610 dihubungkan dengan potensiometer RL R1 senilai $10K\Omega$ yang digunakan sebagai resistor beban atau resistor pembagi tegangan output. Keluaran dari sensor ini berupa tegangan analog yang langsung dapat dimasukkan ke pin ADC dari mikrokontroler. Dalam penelitian ini dipilih pin PA.0 yang merupakan pin ADC channel 0.

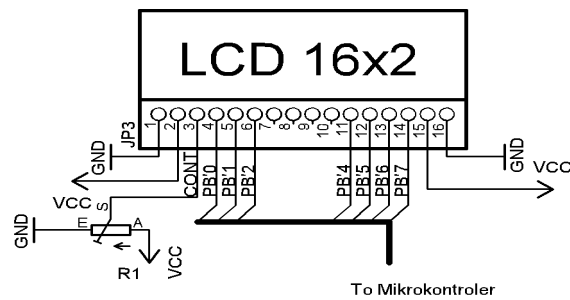
3.1.3. Penampil LCD 16x2

Untuk menampilkan data konsentrasi gas LPG di udara diperlukan media berupa display. Penampil yang digunakan adalah LCD dengan tipe dot matrix 16x2. Dibawah ini adalah fungsi dari masing-masing pin LCD.

Tabel 3.1. Fungsi Pin LCD 16x2

| No | Simbol | Level | Fungsi |
|----|-----------------|-------|---|
| 1 | V _{SS} | -- | 0V |
| 2 | V _{DD} | -- | +5V |
| 3 | V ₀ | -- | Kontras |
| 4 | RS | H/L | Register Select : H--Data, L-- Intruksi |
| 5 | R/W | H/L | H--Baca, L--Tulis |
| 6 | E | H.H-L | Sinyal Enable |
| 7 | DB0 | H/L | Data Bus untuk pengalamatan 8-bit |
| 8 | DB 1 | H/L | |
| 9 | DB2 | H/L | |
| 10 | DB3 | H/L | |
| 11 | DB4 | H/L | Data Bus untuk pengalamatan 4-bit dan 8-bit |
| 12 | DB5 | H/L | |
| 13 | DB6 | H/L | |
| 14 | DB7 | H/L | |
| 15 | A | -- | Back Light (+) |
| 16 | K | -- | Back Light (-) |

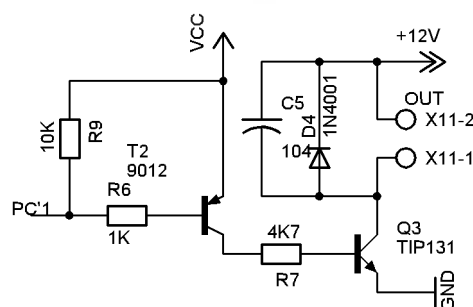
Dalam pembuatan *interface* antara LCD dan mikrokontroler digunakan metode pengalamatan 4-bit data. Alasan utama menggunakan metode ini adalah kesederhanaan proses dalam program. Adapun peran utama LCD hanya difungsikan dalam kondisi tulis dan baca. Konfigurasi LCD dan mikrokontroler ditunjukkan pada gambar 3.4.



Gambar 3.4 Konfigurasi mikrokontroler dan LCD

3.1.4. Rangkaian *Driver Relay*

Dalam aplikasi ini digunakan dua buah driver, yaitu driver untuk menyalakan alarm serta driver untuk mengaktifkan relay, yang dapat dihubungkan ke piranti lain dengan tegangan yang berbeda. Mikrokontroler ATmega8535 memiliki tegangan kerja level TTL, yaitu 0 dan 5V. Untuk dapat mengendalikan piranti lain yang memiliki tegangan kerja yang berbeda seperti relai, *buzzer* mikrokontroler membutuhkan *driver*. Gambar 3.5 adalah rangkaian driver dengan transistor untuk mengaktifkan buzzer atau alarm kecil.

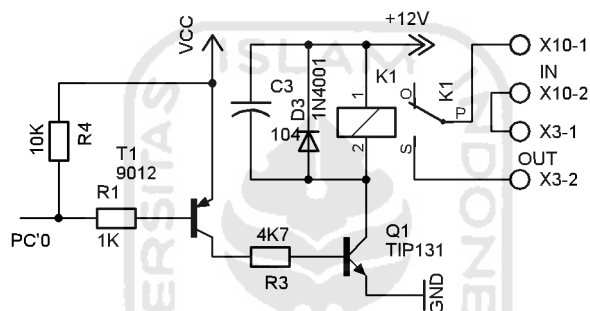


Gambar 3.5 Rangkaian *driver buzzer*

Pada rangkaian *driver* Gambar 3.5 mikrokontroler mengendalikan transistor 9012 dengan memberikan bias basis. Keadaan transistor akan berada pada posisi saturasi ketika logika rendah diberikan pada kaki basis, sedangkan

kerika diberi logika tinggi maka transistor akan berada pada kondisi *cut off* atau tidak menghantar. Transistor ini akan mengendalikan transistor TIP31, dan transistor TIP31 akan mengendalikan buzzer.

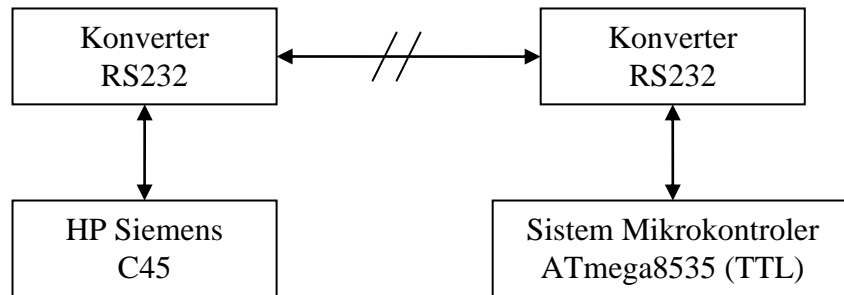
Pada konfigurasi untuk mengendalikan relay, rangkaian yang digunakan sama seperti diatas namun ada sedikit perbedaan, yaitu outputnya adalah relay, keluaran dari relay ini kontak-kontaknya dapat dihubungkan dengan level tegangan yang berbeda. Gambar driver relay ditunjukkan pada Gambar 3.6.



Gambar 3.6. Rangkaian *driver relay*

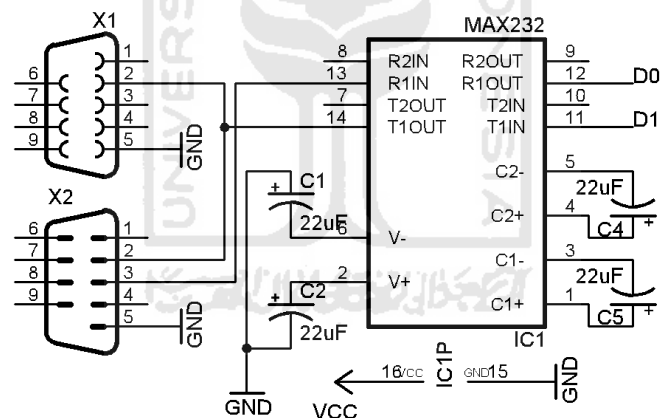
3.1.5 Komunikasi mikrokontroler dan HP Terminal

Kabel data difungsikan sebagai konverter atau penyesuai logika mikrokontroler dan logika pada HP. Konverter yang digunakan pada kabel data siemens C55 atau M35 adalah IC konverter Max232. Konsep penggunaan sistem konverter ditunjukkan pada Gambar 3.7.



Gambar 3.7 Konsep konverter RS232

Gambar 3.7 menyajikan konsep sistem konverter RS232. Sistem ini mengkonversi level logika TTL ke RS 232, kemudian ditransmisikan, untuk selanjutnya diterima oleh konverter RS 232 yang akan merubah menjadi logika CMOS.

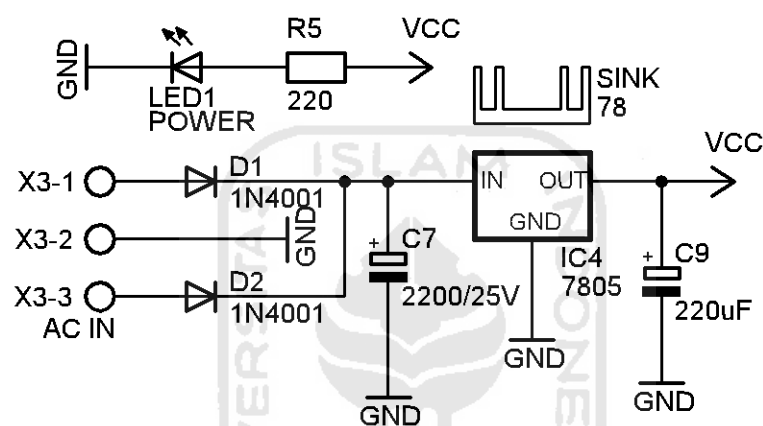


Gambar 3.8 Konfigurasi Max 232

Pada Gambar 3.8 ditunjukkan konfigurasi IC Max 232 dengan beberapa komponen tambahan. IC Max 232 perlu dipasang beberapa komponen tambahan antara lain, kapasitor senilai 22uF/16V. Dengan menghubungkan PD.0 dan PD.1 dari mikrokontroler dengan IC ini, maka komunikasi serial yang berlevel TTL sudah mampu diubah ke level RS232.

3.1.6 Catu Daya

Perancangan sistem catu daya dalam penelitian ini disesuaikan dengan kebutuhan catu daya mikrokontroler. Mikrokontroler AT89S51 membutuhkan *supply* +5V dan *Ground*. Pada perancangan sistem catu daya ini menggunakan sistem penyearah dan regulator. Gambar 3.9 menunjukkan skema catu daya yang digunakan.



Gambar 3.9 Rangkaian *power supply*

Pada Gambar 3.9 menampilkan rangkaian catu daya. AC 220V diturunkan menggunakan transformator step down. Tegangan output trafo ini sebesar 12V. Trafo yang digunakan adalah trafo yang dilengkapi dengan *center tap* (CT). Pemasangan dua buah dioda 1N4001 akan menghasilkan penyearah gelombang penuh. Pemasangan kapasitor 2200uF/25V bertujuan untuk meratakan arus dan menghilangkan *riple*. IC regulator 7805 akan menghasilkan tegangan output atau V_{CC} sebesar 5V. Pemberian kapasitor 220uF/16V bertujuan agar tidak terjadi jatuh tegangan ketika catu daya diberi beban.

3.2. Perancangan Perangkat Lunak

Pada perangkat lunak ini digunakan bahasa *C* dengan *compiler code vision AVR*. Pada perancangan perangkat lunak, *software* di-develop dengan *compiler* kemudian kode *hexa* hasil dari proses kompilasi di-download ke mikrokontroler dengan *software AVR Studio*.

Langkah-langkah perancangan lunak dapat ditarik garis besarnya terlebih dahulu setelah memahami cara kerja peralatan dan proses pengendalian yang akan dilaksanakan, antara lain :

- 1) Pembuatan *flow chart* urutan pengendalian program
- 2) Pembuatan *listing* program dalam bentuk file berekstensi *.c*
- 3) Pengubahan file berekstensi *obj* menjadi *hex*
- 4) Pengisian program dengan *software download AVR Studio*

3.2.1. Prosedur Penanganan Sistem

Pada program penanganan sistem, ada beberapa tahapan yang harus dilakukan. Adapun tahapannya adalah membaca sensor membandingkan dengan konstanta limit bahaya, menyalakan alarm saat keadaan bahaya terdeteksi, serta mengirim SMS ke *user*.

3.2.2 Perancangan Pesan SMS

Ketika sistem mendeteksi bahaya, maka sistem akan menyalakan alarm dan relai yang keluarannya dihubungkan dapat dihubungkan ke sistem yang lain. Untuk itu diperlukan perancangan mengenai pesan yang akan dikirim. Untuk mendapatkan data pesan SMS harus dilakukan dengan cara merubah isi pesan ke dalam format PDU. Dalam penelitian ini diinginkan pesan yang terkirim adalah

'Terdeteksi Kebocoran Gas LPG' dengan menggunakan *software PDUsPY*, pesan tersebut akan diubah menjadi data PDU.

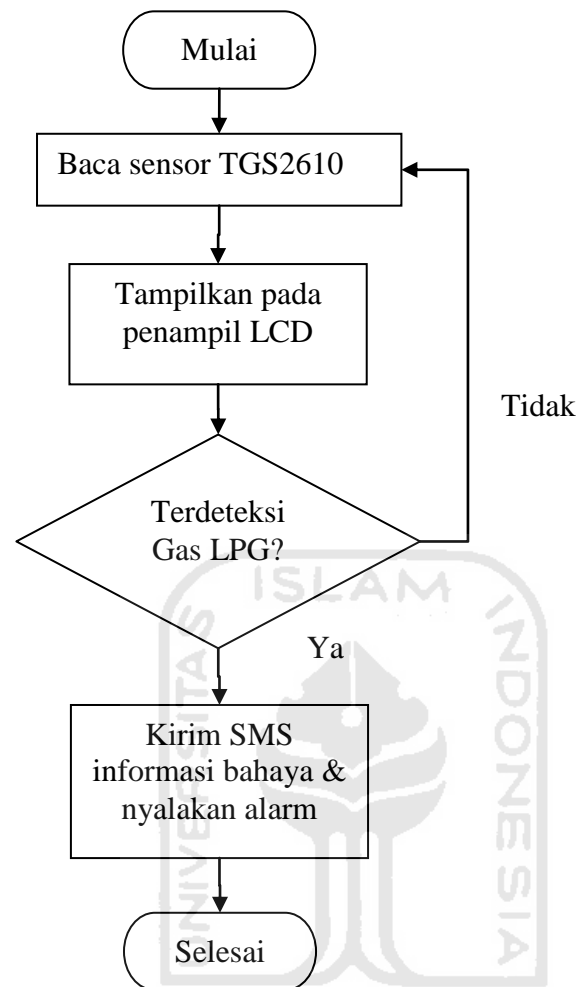
Proses pengubahan kode ASCII menjadi kode PDU yang dilakukan dengan *software PDUsPY*, dapat dilihat pada tabel. Berikut adalah detail dari hasil proses pengubahan data disajikan pada Tabel 3.2.

Tabel 3.2. Konversi pesan ke PDU

| Kondisi | Data Asli | Data PDU |
|-----------------|------------------------------|--|
| SMS center | <i>Default</i> | 000100 |
| Nomor HP tujuan | +6285292248644 | 0D91265892228446F4 |
| SMS referensi | <i>Default</i> | 000000 |
| Isi pesan | Terdeteksi Kebocoran Gas LPG | 1CD4B29C5CA697D7F334685916BFC76F79D80D3A86E72026F408 |

3.2.3 Perancangan *flow chart*

Untuk menangani seluruh tugas dalam skenario diatas, maka mikrokontroler diberi program dengan algoritma yang mampu menangani seluruh sistem. Gambar 3.10 adalah *flow chart* yang digunakan sebagai acuan program untuk menangani sistem.



Gambar 3.10. *Flow chart* program

Pada algoritma program gambar 3.10, yang pertama dilakukan adalah, sistem membaca TGS2610. Informasi ini kemudian ditampilkan dalam penampil LCD. Jika sistem mendeteksi adanya kebocoran gas LPG maka sistem akan menyalakan alarm dan mengirimkan informasi SMS peringatan ke user.

BAB IV

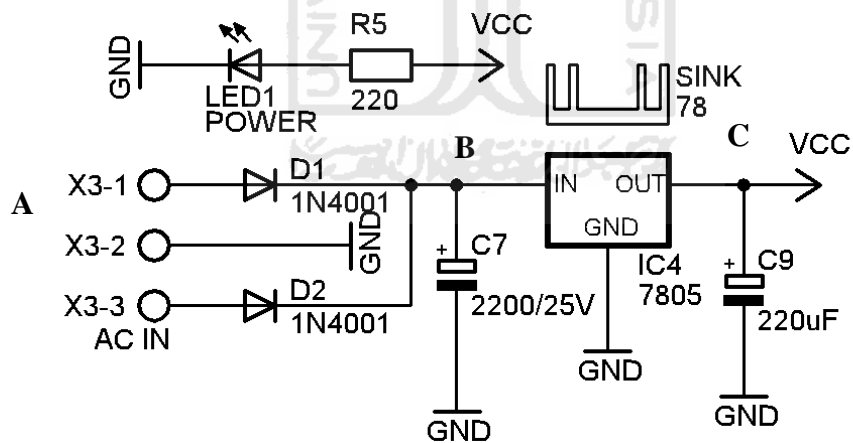
PENGUJIAN, ANALISIS DAN PEMBAHASAN

4.1. Pengujian Alat

Hasil pengujian diperoleh dengan menggunakan alat ukur. Pengujian dilakukan pada sensor TGS2610, pengujian terhadap catu daya dan pengujian koneksi HP. Beberapa tahapan pengujian adalah sebagai berikut:

4.1.1. Pengujian terhadap catu daya

Pengujian terhadap catu daya dilakukan dengan mengukur nilai tegangan pada beberapa titik ukur. Gambar 4.1. adalah rencana titik uji pada rangkaian catu daya.



Gambar 4.1. Rencana pengujian catu daya

Sumber tegangan yang digunakan dalam sistem ini adalah sumber tegangan DC sebesar 12V dan 5V. Gambar rangkaian yang diuji tunjukkan pada Gambar 4.1. Power supply diuji pertama kali, karena digunakan untuk menjalankan sistem secara keseluruhan. Tegangan 5V digunakan untuk

menjalankan mikrokontroler sedangkan tegangan 12V digunakan untuk menjalankan relay atau driver.

Setelah dilakukan percobaan dan pengujian seperti pada Gambar 4.1 diperoleh hasil pengamatan yang disajikan pada Tabel 4.1.

Tabel 4.1. Hasil Pengamatan Catu Daya

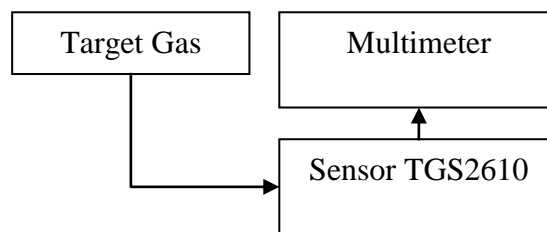
| No | Titik Uji | Hasil ukur (V) |
|----|-----------|----------------|
| 1 | Titik A | 11,97 |
| 2 | Titik B | 13,8 |
| 3 | Titik C | 4.98V |

Berdasarkan data pengamatan pada Tabel 4.1 diketahui bahwa nilai *output* dari trafo *step down* adalah 11,97V. Nilai tegangan ini berasal dari trafo *step down*, yang menurunkan tegangan dari AC 220V ke AC 12V. Nilai tegangan ini kemudian disearahkan dengan dua buah dioda dengan sistem penyearah gelombang penuh. Keluaran dioda ini setelah diberi kapasitor nilainya terukur sebesar 13,8V. Tegangan 13.8V ini digunakan untuk men-supply relay dan alarm, yang memiliki tegangan kerja sekitar 12V.

Tegangan 13.8V ini terlalu besar untuk men-suplai mikrokontroler sehingga digunakan regulator 7805 seperti pada Gambar 4.1. Regulator 7805 menghasilkan tegangan keluaran sebesar output terukur 4.98V. Nilai keluaran ini digunakan untuk men-suplai mikrokontroler, LCD dan IC-IC digital yang membutuhkan tegangan kerja 5V.

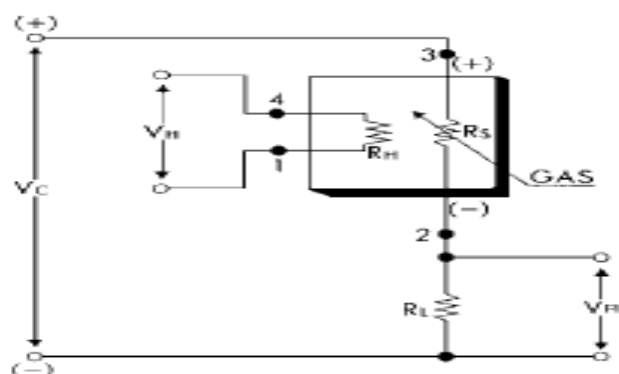
4.1.2. Pengujian Terhadap Sensor TGS2610

Pengujian terhadap sensor gas TGS2610 dilakukan dengan mengukur nilai tegangan output saat terdapat LPG dan saat kondisi normal. Gambar 4.2 adalah metode pengujian sensor LPG.



Gambar 4.2. Rencana pengujian sensor TGS2610

Hasil dari pengujian sensor TGS2610, digunakan untuk mengetahui kinerja dari sensor. Proses pengujian dilakukan dengan menggunakan multimeter, dengan mengukur tegangan output saat terdapat LPG dan saat tidak ada gas berdasarkan jarak antara sumber gas dan sensor gas. Untuk mendapatkan nilai resistansi pada sensor TGS 2610 dan kadar gas di udara dapat digunakan rumus sebagai berikut :



Gambar 4.3. Rangkaian sensor TGS 2610

$$R_s = \left(\frac{R_{beban} \times V_{input}}{V_{output}} \right) - R_{beban} \quad (4.1)$$

Dimana :

R_s = Resistansi sensor

R_{beban} = Resistansi beban

V_{input} = Tegangan input (5 V)

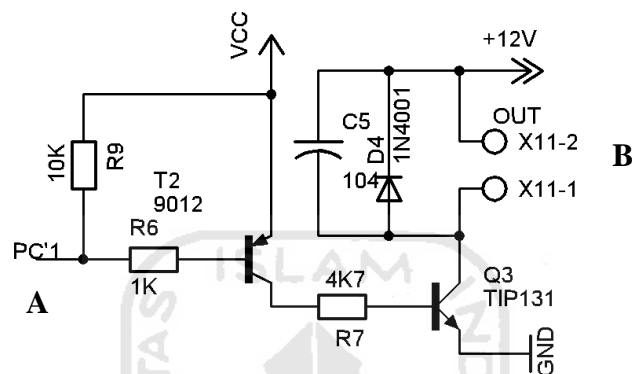
V_{output} = Tegangan output

Tabel 4.2 Pengamatan sensor gas TGS2610 berdasarkan percobaan dan perhitungan

| No | Tegangan (V) | Perhitungan Resistansi (Ohm) | Kadar Gas (PPM) | Jarak (Cm) |
|----|--------------|------------------------------|-----------------|------------|
| 1 | 3.2 | 562.5 | 9215.5 | 2 |
| 2 | 3 | 666.6 | 9136.3 | 3 |
| 3 | 2.5 | 1000 | 8550.8 | 3.5 |
| 4 | 2 | 1500 | 7921.8 | 4 |
| 5 | 1.7 | 1941.2 | 7285.71 | 4.3 |
| 6 | 1.5 | 2333.3 | 6833.3 | 4.5 |
| 7 | 1 | 4000 | 4409.5 | 5 |
| 8 | 0.8 | 6250 | 1268.6 | 5.1 |
| 9 | 0.6 | 6700 | 639.8 | 5.2 |
| 10 | 0.4 | 11500 | 0 | 5.5 |

4.1.3. Pengujian Terhadap *Driver*

Pengujian terhadap *driver* digunakan untuk mengetahui kinerja mikrokontroler dalam menyalakan dan mematikan alarm. Gambar 4.3 adalah titik uji rangkaian *driver*.



Gambar 4.4 Rencana pengujian *driver*

Pengujian terhadap *driver* dilakukan dengan memberikan logika dari mikrokontroler pada sisi input, kemudian mengukur tegangan keluaran driver. Dengan konfigurasi seperti pada Gambar 4.3 percobaan dilakukan dengan memberikan *input* logika yang berasal dari mikrokontroler. Tabel 4.3 adalah data dari hasil percobaan.

Tabel 4.3. Hasil percobaan terhadap driver

| PC.1 (Logika) | Titik A (V) | Titik B (V) | Alarm |
|---------------|-------------|-------------|-------|
| Low | 0 | 13 | On |
| High | 4.98 | 0 | Off |

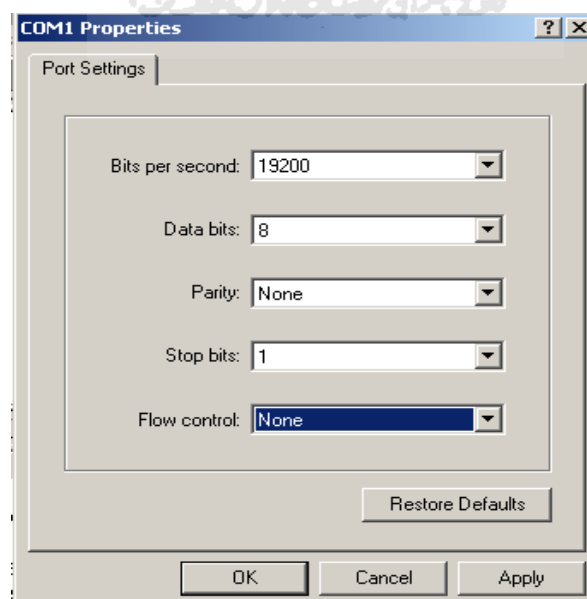
Dari hasil percobaan yang telah dilakukan, dapat diketahui bahwa jika diinginkan alarm dalam kondisi aktif atau *on*, maka mikrokontroler harus

mengirimkan logika *low* ke masukan *driver*. Demikian juga sebaliknya untuk menjadikan alarm *off*, maka logika *high* harus diberikan pada masukan *driver*.

Saat logika *low* diberikan pada input basis transistor 9012 maka transistor PNP ini akan aktif, sehingga kondisinya saturasi dan kaki kolektor akan bernilai V_{cc} . Tegangan V_{cc} ini akan memberikan bias pada basis transistor NPN TIP31, sehingga transistor ini dalam kondisi saturasi dan arus akan mengalir ke buzzer atau output melalui kolektor transistor.

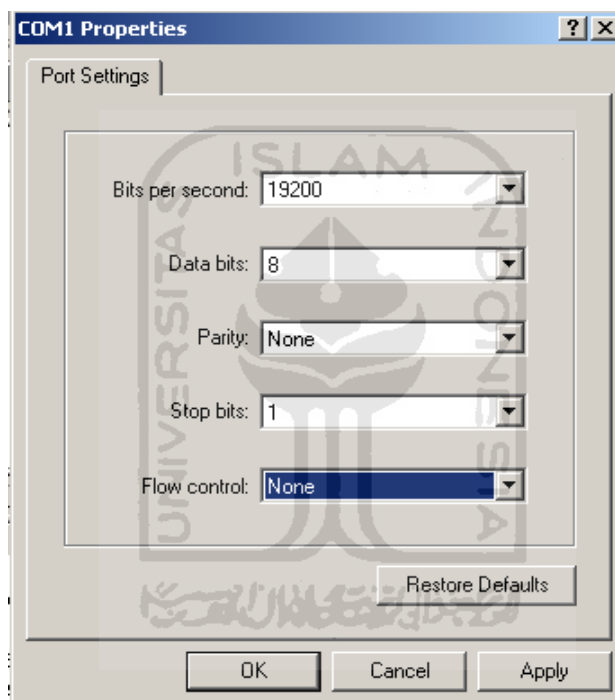
4.1.4. Pengujian Koneksi HP

Pengujian perintah-perintah dan koneksi HP dilakukan dengan menggunakan bantuan *software hyperterminal*. Komunikasi serial antara mikrokontroler dan HP menggunakan kecepatan *baut rate* 19200 bps, untuk itu perlu dilakukan pengaturan nilai kecepatan pada *software hyperterminal*. Gambar 4.4 adalah tampilan dan *setting hyperterminal*.



Gambar 4.4 Tampilan *hyperterminal*

Pada penelitian ini, HP yang digunakan adalah HP Siemens C55. HP ini memiliki koneksi dengan peralatan eksternal dengan menggunakan komunikasi serial UART (*Universal Asynchronous Receiver Transmitter*) dengan *baud rate* 19200bps. Untuk itu agar mampu melakukan pengujian terhadap HP maka harus menggunakan fasilitas *hyper terminal* pada windows. Pada Gambar 4.4. ditunjukkan konfigurasi pada *hyper terminal* untuk berkomunikasi dengan HP.



Gambar 4.4 Konfigurasi *hyper terminal*

Dengan menghubungkan ponsel ke COM1 pada komputer maka dapat diperoleh terminal untuk berkomunikasi dengan HP. Ketika sistem mengirim SMS diperoleh data pengujian kode PDU sebagai berikut

```
AT+CMGS=33
```

```
0001000D91265892228446F4000014D7A0D499741E5DAE10E8189C8
```

```
284CFE1530A
```

Data-data *PDU* tersebut merupakan paket yang dikirim oleh sistem menuju ponsel. Data ‘AT+CMGS=33’ adalah data *AT command* yang digunakan untuk mengirim SMS. Data ‘000100’ adalah data pengaturan *SMS center* secara *default*. Data 0D91265892228446F4 adalah data nomor HP tujuan, jika diterjemahkan menjadi +628292248644. Data *PDU* berikutnya adalah ‘0000’ merupakan SMS referensi *default*. Kode *PDU* yang terakhir adalah isi sms, datanya adalah ‘14D7A0D499741E5DAE10E8189C8284CFE1530A ’, data tersebut dapat diterjemahkan menjadi ‘WARNING..! GAS BOCOR’.

4.1.5. Pengujian kinerja sistem secara keseluruhan

Pengujian sistem secara menyeluruh dilakukan dengan menggunakan simulasi terhadap perubahan konsentrasi gas, kemudian mengamati kinerja sistem mengenai alarm dan sms yang dihasilkan. Pengujian sistem dilakukan untuk mengetahui unjuk kerja dari alat yang dibangun. Pengujian dilakukan dengan minimal 10 (sepuluh) kali percobaan . Percobaan ini dilakukan dalam ruangan tertutup dengan memberikan jarak yang berbeda-beda antara sumber gas dengan alat pendeteksi kebocoran gas LPG. Ruangan untuk uji coba ini harus tertutup untuk meminimalisir adanya angin bertiup dari luar ruangan yang akan mempengaruhi kadar gas dalam ruangan. Berikut adalah hasil pengujian yang telah dilakukan.

Sistem diuji dengan memberikan simulasi gas bocor dari tabung gas LPG 3KG. Setelah diuji, sistem memberikan respon, alarm bunyi dan terdapat kiriman SMS kepada user yang isinya ‘WARNING..! GAS BOCOR’.

BAB V

KESIMPULAN DAN SARAN

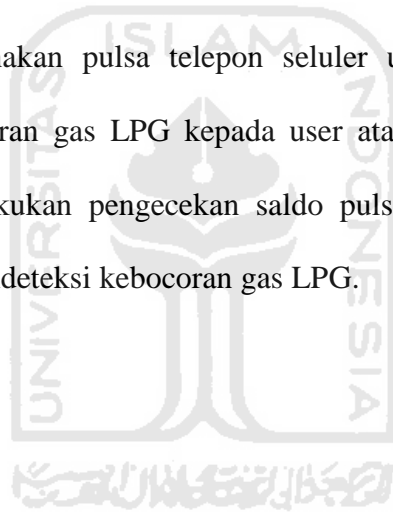
5.1. Kesimpulan

1. Resistansi pada sensor gas TGS 2610 berbanding terbalik terhadap tegangan output sensor gas tersebut. Hal ini dapat dilihat dari semakin besar tegangan yang diperoleh dari pengukuran maka semakin kecil nilai resistansinya.
2. Peletakan sensor gas TGS 2610 tidak boleh terlalu jauh dari sumber kebocoran gas. Hal ini akan sangat mempengaruhi sensor gas untuk dapat mendeteksi adanya kebocoran gas LPG, semakin jauh peletakan sensor gas maka kemampuan sensor gas untuk mendeteksi kebocoran gas LPG sangatlah kecil sehingga akan membutuhkan waktu yang lama agar sensor gas dapat mendeteksi adanya kebocoran gas LPG. Hal ini akan sangat membahayakan apabila adanya percikan – percikan api disekitar sumber kebocoran gas yang timbul sebelum gas terdeteksi oleh sensor gas.
3. Kepekatan gas LPG sangat dipengaruhi oleh kondisi ruangan dimana terdapat kebocoran gas LPG tersebut. Semakin baik sirkulasi udara dari luar ke dalam ruangan atau sebaliknya, maka semakin lama kadar gas LPG untuk menjadi pekat di dalam ruangan tersebut. Sehingga tingkat keamanan untuk terjadinya ledakan semakin tinggi.
4. Posisi pendeteksian gas pada titik aman yaitu sekitar 1000 Ppm campuran gas di udara.

5.2. Saran

Dalam penelitian ini banyak terjadi kekurangan, untuk itu penulis memberikan beberapa saran, antara lain:

1. Alat ini hanya mampu mendeteksi kebocoran gas LPG pada ruangan. Bukan untuk mengatasi kebocoran gas LPG, oleh karena itu diperlukan kewaspadaan apabila terjadi kebocoran gas LPG yang terdeteksi oleh alat ini maka janganlah melakukan sesuatu yang bisa menimbulkan percikan - percikan api.
2. Alat ini menggunakan pulsa telepon seluler untuk mengirimkan sms peringatan kebocoran gas LPG kepada user atau pengguna. Oleh karena itu, seringlah melakukan pengecekan saldo pulsa pada handphone yang terdapat di alat pendeteksi kebocoran gas LPG.



DAFTAR PUSTAKA

Budiharjo , W ; *Elektronika Digital dan Mikroprosesor* , Andi, Yogyakarta , 2005.

Budiharjo, W ; *Interfacing Komputer dan Mikrokontroler* , Elex Media Komputindo , Jakarta , 2004.

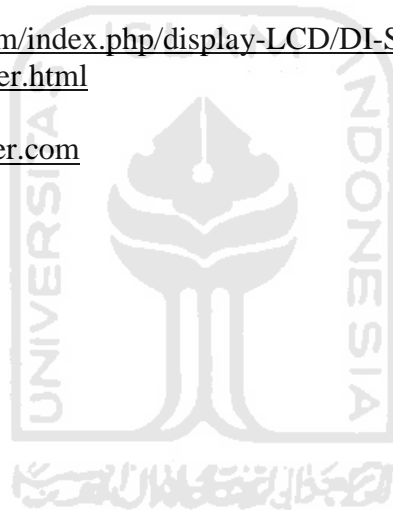
Wardhana , L ; *Belajar Sendiri : Mikrokontroler AVR Seri ATmega 8535*. Yogyakarta : ANDI. 2006

Simanjuntak , T LH ; *Dasar - dasar Telekomunikasi* , Penerbit Alumni , Bandung , 2003.

<http://www.mikron123.com/index.php/display-LCD/DI-Smart-LCD-16x2/Detailed-product-flyer.html>

<http://www.mikrokontroller.com>

<http://www.atmel.com>



LAMPIRAN

```
#include <mega8535.h>
#include <stdio.h>
#include <delay.h>

#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

#define relay PORTC.0
#define buzzer PORTC.1

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;

    // Start the AD conversion
    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);

    ADCSRA|=0x10;
```




```
return ADCW;
```

```
}
```

```
unsigned int humi1,humi2;
```

```
unsigned char temp,layar_1[16],layar_2[16];
```

```
unsigned long int gas;
```

```
unsigned char count,c,i,x,pdu,jmlh_nope,derajat=0xDF;;
```

```
unsigned char data[100];
```

```
unsigned char message[60];
```

```
unsigned char header[]={\"001100\"};
```

```
unsigned char time_limit[]={\"000000\"};
```

```
unsigned char no_hp[20],no_hp_alert[20]={\"0D91261893727371F1\"};  
//{\"0D91265892228446F4\"};//{\"0D91265827796192F1\"}; {\"0D91265827376527F1\"};//
```

```
        //{\"0D91261875785715F6\"}; {\"0D91265827796192F1\"};
```

```
unsigned char no_hp_alert2[20]={\"0D91265847530030F0\"};
```

```
bit alarm,flag_no;
```

```
const long int osilator=11059200;
```

```
void InialisasiUART(unsigned long int baud_rate);
```

```
unsigned long int UBRR;
```

```
void InialisasiUART (unsigned long int baud_rate)
```

```
{
```

```
UBRR = (osilator/(16*baud_rate))-1;
```

```
UBRR = UBRR;  
UBRRH = UBRR>>8;  
UCSRB = 0x18;//18 = RX AND TX; 10== RX  
UCSRC = 0x86;  
}
```

```
unsigned char terima(void)  
{  
while ( !(UCSRA & 0x80) );  
return UDR;  
}
```

```
void kirim_z()  
{  
while ( !(UCSRA & 0x20) );  
UDR=0x1A;  
}
```

```
void kirim_char()  
{  
while ( !(UCSRA & 0x20) );  
UDR=pdu;  
}
```

```
void init_io()  
{  
PORTA=0xFF;  
DDRA=0x00;  
PORTB=0x00;
```



```

DDRB=0xFF;
PORTC=0x0F;
DDRC=0xFF;
PORTD=0x00;
DDRD=0xFF;

// ADC initialization
// ADC Clock frequency: 172,800 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x86;
SFIOR&=0xEF;
}

void opening()
{
  lcd_clear();
  sprintf(layar_1," LPG Detector ");
  sprintf(layar_2,"Via SMS Gateway ");
  lcd_gotoxy(0,0);lcd_puts(layar_1);
  lcd_gotoxy(0,1);lcd_puts(layar_2);
  delay_ms(3);
}

void konfirmasi_hp()

```



```

{
lcd_clear();
sprintf(layar_1,"Connection : OK ");
sprintf(layar_2,"Init Sensor:");
lcd_gotoxy(0,0);lcd_puts(layar_1);
lcd_gotoxy(0,1);lcd_puts(layar_2);
}

```

```

void init_sensor_ok()
{
sprintf(layar_1,"OK");
lcd_gotoxy(13,1);lcd_puts(layar_1);
}

```

```

void view_gas()
{
lcd_clear();
sprintf(layar_1," LPG Detector ");
//sprintf(layar_2," [Status Aman] ");
sprintf(layar_2,"Gas:%imV [Aman]",gas);
lcd_gotoxy(0,0);lcd_puts(layar_1);
lcd_gotoxy(0,1);lcd_puts(layar_2);
delay_ms(3);
}

```

```

void view_gas_bocor()
{
lcd_clear();

```



```

sprintf(layar_1," LPG Detector ");
//sprintf(layar_2," [Status Aman] ");
sprintf(layar_2,"Gas:%imV [Bocor]",gas);
lcd_gotoxy(0,0);lcd_puts(layar_1);
lcd_gotoxy(0,1);lcd_puts(layar_2);
delay_ms(3);
}

```

```

void gas_alert()
{
lcd_clear();
sprintf(layar_1," LPG Detector ");
sprintf(layar_2," [Gas Bocor]");
lcd_gotoxy(0,0);lcd_puts(layar_1);
lcd_gotoxy(0,1);lcd_puts(layar_2);
delay_ms(3);
}

```

```

void baca_sms1()
{
putsf("at+cmgr=1\r"); /\r
}

```

```

void baca_sms2()
{
putsf("at+cmgr=2\r"); /\r
}

```



```
void inisial_hp()
{
    putsf("at\r"); //\r
}
```

```
void hapus_sms1(void)
{
    putsf("at+cmgd=1\r");
    while(1)
    {
        temp=terima();
        if (temp=='K')break;
    }
    delay_ms(1000);
}
```

```
void hapus_sms2(void)
{
    putsf("at+cmgd=2\r");

    while(1)
    {
        temp=terima();
        if (temp=='K')break;
    }
    delay_ms(5000);
}
```



```
void indikator_sms(void)
{
putsf("at+cnmi=1,1\r");
}
```

```
void cek_header_sms()
```

```
{
while(1)
{
data[0]=terima();

if (data[0]=='+') break;
}
while(1)
{
temp=terima();
if (temp=='+') break;
}
}
```



```
// unsigned char ff[]="001100";
```

```
// unsigned no_hp[13]; //0D91265827376527F1
```

```
// unsigned char time_limit[]="000000";
```

```
// unsigned char message_true[]="14D9775D0E9296E3F5F29C0E4ACE415479BD0C";
```

```
void ubah_PDU()
```

```
{
```

```
//0001000081000014D7A0D499741E5DAE10E8189C8284CFE1530A
```

```
// HUMY 56,70% TEMP 30C
```

alert:

```
message[0]=20; // JUMLAH KARAKTER;
```

```
message[1]='W';
```

```
message[2]='A';
```

```
message[3]='R';
```

```
message[4]='N';
```

```
message[5]='I';
```

```
message[6]='N';
```

```
message[7]='G';
```

```
message[8]='.';
```

```
message[9]='.';
```

```
message[10]='!';
```

```
message[11]=' ';
```

```
message[12]='G';
```

```
message[13]='A';
```

```
message[14]='S';
```

```
message[15]=' ';
```

```
message[16]='B';
```

```
message[17]='O';
```

```
message[18]='C';
```

```
message[19]='O';
```

```
message[20]='R';
```



konversi:

```
//-----
```



```
// RUMUS KONVERSI KE PDU
```

```
//-----
```

```
message[1]=(message[1])+((message[2]&0b00000001)<<7);  
message[2]=(message[2]>>1)+((message[3]&0b00000011)<<6);  
message[3]=(message[3]>>2)+((message[4]&0b00000111)<<5);  
message[4]=(message[4]>>3)+((message[5]&0b00001111)<<4);  
message[5]=(message[5]>>4)+((message[6]&0b00011111)<<3);  
message[6]=(message[6]>>5)+((message[7]&0b00111111)<<2);  
message[7]=(message[7]>>6)+((message[8]&0b01111111)<<1);
```

```
message[8]=(message[9])+((message[10]&0b00000001)<<7);  
message[9]=(message[10]>>1)+((message[11]&0b00000011)<<6);  
message[10]=(message[11]>>2)+((message[12]&0b00000111)<<5);  
message[11]=(message[12]>>3)+((message[13]&0b00001111)<<4);  
message[12]=(message[13]>>4)+((message[14]&0b00011111)<<3);  
message[13]=(message[14]>>5)+((message[15]&0b00111111)<<2);  
message[14]=(message[15]>>6)+((message[16]&0b01111111)<<1);
```

```
message[15]=(message[17])+((message[18]&0b00000001)<<7);  
message[16]=(message[18]>>1)+((message[19]&0b00000011)<<6);  
message[17]=(message[19]>>2)+((message[20]&0b00000111)<<5);  
message[18]=(message[20]>>3);
```

```
message[19]=((message[0]&0xF0)>>4); //JUMLAH KARAKTER D UBAH KE ASCII  
message[20]=(message[0]&0x0F);
```

```
//-----
```

```
// RUMUS KONVERSI KE ASCII
```

//-----

```
message[21]=((message[1]&0xF0)>>4);
message[22]=(message[1]&0x0F);
message[23]=((message[2]&0xF0)>>4);
message[24]=(message[2]&0x0F);
message[25]=((message[3]&0xF0)>>4);
message[26]=(message[3]&0x0F);
message[27]=((message[4]&0xF0)>>4);
message[28]=(message[4]&0x0F);
message[29]=((message[5]&0xF0)>>4);
message[30]=(message[5]&0x0F);
message[31]=((message[6]&0xF0)>>4);
message[32]=(message[6]&0x0F);
message[33]=((message[7]&0xF0)>>4);
message[34]=(message[7]&0x0F);
message[35]=((message[8]&0xF0)>>4);
message[36]=(message[8]&0x0F);
message[37]=((message[9]&0xF0)>>4);
message[38]=(message[9]&0x0F);
message[39]=((message[10]&0xF0)>>4);
message[40]=(message[10]&0x0F);
message[41]=((message[11]&0xF0)>>4);
message[42]=(message[11]&0x0F);
message[43]=((message[12]&0xF0)>>4);
message[44]=(message[12]&0x0F);
message[45]=((message[13]&0xF0)>>4);
message[46]=(message[13]&0x0F);
```



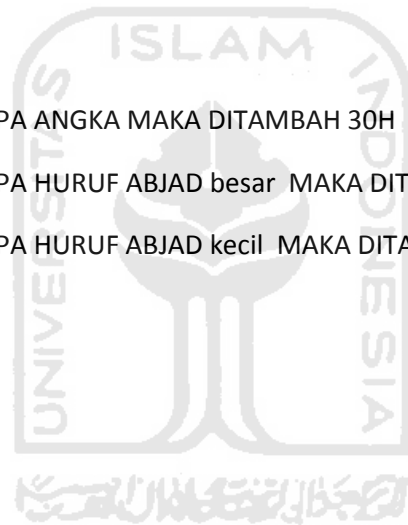
```
message[47]=((message[14]&0xF0)>>4);
message[48]=(message[14]&0x0F);
message[49]=((message[15]&0xF0)>>4);
message[50]=(message[15]&0x0F);
message[51]=((message[16]&0xF0)>>4);
message[52]=(message[16]&0x0F);
message[53]=((message[17]&0xF0)>>4);
message[54]=(message[17]&0x0F);
message[55]=((message[18]&0xF0)>>4);
message[56]=(message[18]&0x0F);
```

```
//KONVERSI HEXA TO ASCII,JIKA BERUPA ANGKA MAKA DITAMBAH 30H
```

```
//KONVERSI HEXA TO ASCII,JIKA BERUPA HURUF ABJAD besar MAKA DITAMBAH 37H
```

```
//KONVERSI HEXA TO ASCII,JIKA BERUPA HURUF ABJAD kecil MAKA DITAMBAH 57H
```

```
i=19;
while(1)
{
temp=message[i];
if (temp<10) message[i]=message[i]+0x30;
if (temp>=10) message[i]=message[i]+0x37;
i++;
if (i==57) break;
}
i=1;
//tampil_pdu();
}
```



```
void write_sms() //21 karakter----- // Your Request Is False
```

```
{  
  putsf("at+cmgs=33\r");  
  while(1)  
  {  
    temp=terima();  
    if (temp=='>') break;  
  }  
  delay_ms(100);
```

```
//-----  
//KIRIM HEADER
```

```
//-----  
i=0;  
while(1)  
{  
  pdu=header[i];  
  kirim_char();  
  delay_ms(50);  
  i++;  
  if(i==6) break;  
}  
delay_ms(100);
```

```
//-----  
//KIRIM NO HP  
//-----
```



```
if(flag_no==0)
{
for(i=0;i<18;i++)
{
pdu=no_hp_alert[i];
 kirim_char();
 delay_ms(50);
}
}
```

```
if(flag_no==1)
{
for(i=0;i<18;i++)
{
pdu=no_hp_alert2[i];
 kirim_char();
 delay_ms(50);
}
}
```

```
//-----
//KIRIM TIME LIMIT
//-----

limit:
i=0;
while(1)
{
pdu=time_limit[i];
```



```
    kirim_char();  
    delay_ms(50);  
    i++;  
    if(i==6) break;  
}  
delay_ms(100);
```

```
//-----
```

```
//KIRIM MESSAGE
```

```
//-----
```

```
i=19;  
while(1)  
{  
    if(i==57) break;  
    pdu=message[i];  
    kirim_char();  
    delay_ms(50);  
    i++;  
}  
i=1;
```

```
delay_ms(5000);  
kirim_z();  
delay_ms(1000);  
//putsf("\r");  
}
```

```
void read_gas()
```



```
{  
  gas=read_adc(7);  
  gas=((gas*1000)/1023)*5;  
}
```

```
void main (void)
```

```
{  
  init_io();  
  lcd_init(16);  
  InisialisasiUART(19200);  
  count=0;
```

```
  opening();  
  i=1;  
  delay_ms(2500);  
  buzzer=relay=1;
```

```
cek_HP:
```

```
  inisial_hp();  
  while(1){temp=terima();if (temp=='K')break;}  
  konfirmasi_hp();delay_ms(500);  
  for(;;){read_gas();if(gas<=700) break;} init_sensor_ok();delay_ms(1000);
```

```
start:
```

```
  for(;;)  
  {  
    read_gas();  
    delay_ms(100);
```



```
if(alarm==0&&gas>=900) break;  
else if(gas>900) view_gas_bocor();  
else view_gas();  
}
```

alert:

```
gas_alert();  
delay_ms(1500);  
ubah_PDU();  
write_sms();flag_no=1;delay_ms(5000);  
write_sms();flag_no=0;delay_ms(5000);  
hapus_sms1();  
relay=buzzer=0;  
delay_ms(10000);  
relay=buzzer=1;  
alarm=1;  
goto start;  
}
```



SISTEM ALARM KEBOCORAN GAS LPG DILENGKAPI FASILITAS SMS GATEWAY

Tugas Akhir Anjar Pribadi

Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia

Jalan Kaliurang km. 14,5 Sleman, Jogjakarta 55501

Telp. (0274) 895007, 895287 Faks. (0274) 895007 Ext. 131

E-mail : anjar.pribadi@yahoo.co.id

ABSTRAK

Konversi dari minyak tanah ke tabung gas menjadi salah satu kebijakan pemerintah dalam menanggulangi krisis bahan bakar. Ledakan mengakibatkan kebakaran sehingga menimbulkan kerugian harta benda bahkan adanya korban nyawa sehingga membuat masyarakat menjadi merasa tidak aman untuk menggunakannya. Alat pendeteksi kebocoran gas LPG yang dilengkapi dengan alarm dan SMS ini dapat mendeteksi kebocoran dari tabung gas LPG dengan jarak maksimal 2 meter dalam waktu 1 menit baik di ruangan terbuka maupun tertutup dengan dimensi 2 X 3 meter. Apabila terdeteksi maka alat mampu membunyikan alarm dan mengirimkan SMS ke *user*.

Sistem ini menggunakan sensor LPG TGS 2610, mikrokontroler ATmega 8535, *Display LCD*, *HP*, *buzzer* dan beberapa komponen pendukung yang lain. Alat pendeteksi kebocoran gas LPG yang dilengkapi dengan alarm dan SMS ini diharapkan dapat menjadi alternatif bagi masyarakat untuk menjadi peringatan dini apabila terdapat kebocoran pada tabung gas sehingga dapat diantisipasi lebih dini.

Dalam peletakan sensor gas, jarak sangatlah berpengaruh terhadap kemampuan sensor gas untuk dapat mendeteksi kebocoran gas. Waktu yang diperlukan sensor agar dapat mendeteksi kebocoran gas akan lebih lama apabila jarak peletakan sensor dari sumber kebocoran gas yang diperkirakan semakin jauh. Sirkulasi udara dalam ruangan juga berpengaruh terhadap sensor gas dalam mendeteksi adanya kebocoran gas LPG. Semakin baik sirkulasi udara dalam ruangan maka semakin lama sensor gas untuk dapat mendeteksi gas apabila terjadi kebocoran gas LPG dalam ruangan tersebut.

Kata kunci (*keyword*): Sensor LPG TGS 2610. Mikrokontroler ATmega 8535 *Display LCD*, *HP*, *buzzer*.

1. Pendahuluan

1.1. Latar Belakang Masalah

Kebakaran yang disebabkan oleh meledaknya selang tabung gas LPG kini sering terjadi. Media masa banyak melansir informasi adanya ledakan tabung gas LPG. Ledakan mengakibatkan kebakaran sehingga menimbulkan kerugian harta benda bahkan adanya korbannya.

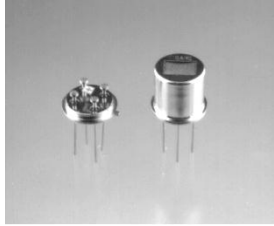
Dengan adanya kejadian ledakan-ledakan gas ini maka perlu adanya sebuah rekayasa teknologi yang mampu meminimalkan resiko adanya ledakan dengan mendeteksi lebih dini, jika terjadi kebocoran gas LPG. Rekayasa dilakukan dengan menggunakan pengupayaan teknologi di bidang elektronik, seperti teknologi sensor, mikroprosesor serta teknologi komunikasi. Dengan latar belakang diperlukannya detektor gas sehingga mampu mereduksi bahaya ledakan gas LPG, maka dalam tugas akhir ini

mengajukan gagasan sistem pendeteksi kebocoran gas LPG dengan dilengkapi alarm dan SMS peringatan.

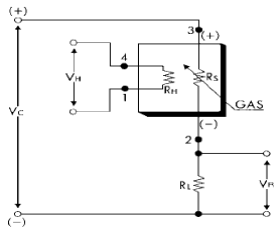
2. Studi Pustaka

2.1. Sensor gas LPG

Sensor gas LPG adalah sensor yang mampu mendeteksi keberadaan gas LPG di udara. Salah satu seri sensor yang dapat mendeteksi gas LPG adalah sensor TGS2610. Sensor ini memiliki prinsip kerja, terjadi perubahan resistansi internal sensor saat sensor mendeteksi adanya konsentrasi gas LPG di udara. Gambar 2.1 menunjukkan bentuk fisik dari sensor gas TGS2610. Gambar 2.2 menunjukkan skema perkawatan dari sensor TGS2610.



Gambar 2.1 Bentuk fisik sensor gas LPG



Gambar 2.2 Skema system elektronik sensor gas LPG

2.2. Mikrokontroler Atmega8535

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus klock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan *mode compare*, *interrupt internal dan eksternal*, serial UART, *programmable Watchdog Timer*, dan *mode power saving*. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

2.3. SMS (Short Message Service)

Semua perintah untuk mengendalikan ponsel menggunakan perintah *AT command*. Beberapa *AT Command* yang penting untuk mengakses SMS antara lain sebagai berikut:

1. AT+CMGR

Perintah ini digunakan untuk membaca isi SMS. Untuk mengetahui kesiapan sistem, bisa digunakan perintah 'AT+CMGS=?', jika respon dari ponsel adalah 'OK', maka ponsel siap dibaca. Untuk menjalankan perintah ini, sintaks penulisan yang dipakai adalah sebagai berikut:

AT+CMGR=<index>

Index adalah lokasi memori yang akan dibaca isi SMS nya., misal AT+CMGR=1, hal ini menandakan SMS yang dibaca adalah lokasi nomor 1.

Jika perintah ini berhasil, maka ponsel akan merespon dengan parameter-parameter sebagai berikut:

+CMGR:
<stat>, [<alpha>], <length><CR><LF><p
du>

2. AT+CMGS

Perintah ini digunakan untuk mengirim SMS. Untuk mengetahui kesiapan ponsel, bisa digunakan *test command* AT+CMGS=?. Jika ponsel siap maka akan merespon dengan 'OK'. Sintaks penulisan perintah yang digunakan adalah sebagai berikut:

AT+CMGS=<length>

<length> diisi dengan jumlah karakter yang akan dikirimkan. Jika perintah berhasil, maka akan merespon dengan tanda '>', kemudian kode PDU untuk SMS kirim siap diisikan.

3. AT+CMGD

Perintah ini digunakan untuk menghapus isi SMS. Sintaks penulisan yang digunakan untuk menjalankan perintah ini, adalah

AT+CMGD=<index>

Index adalah lokasi memori dari SMS yang akan dihapus. Jika perintah ini berhasil, maka ponsel akan merespon dengan 'OK'.

AT Command untuk SMS, biasanya diwakili oleh data dalam format PDU (*Protocol Data Unit*). Data dalam format PDU mempunyai beberapa *header*. *Header-header* untuk pengiriman SMS dan penerimaan SMS berbeda. Dalam format *PDU*, data yang dikirim tidak hanya data yang berisi pesan saja, namun ada beberapa informasi mengenai pengirimnya, seperti nomor pengirim, *service center* SMS, waktu pengiriman, dsb.

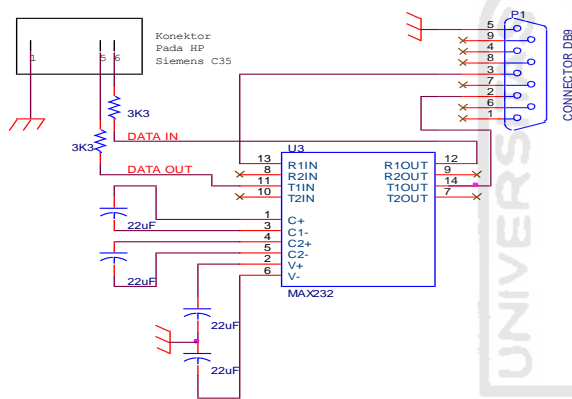
2.4. Komunikasi Serial

Dalam dunia digital dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data secara asinkron. Pada komunikasi data serial sinkron, sinyal *clock* dikirim bersama-sama dengan data serial, sedangkan komunikasi data serial asinkron, *clock* tidak dikirim bersama-sama data serial tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim maupun pada sisi penerima. Komunikasi

data serial dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial yang kemudian diubah kembali menjadi data paralel. Sebagai contoh IC UART 8250 dari intel merupakan salah satunya. Selain berbentuk IC mandiri, berbagai macam mikrokontroler ada yang dilengkapi UART, misalnya keluarga mikrokontroler MCS51.

2.5. Komunikasi Mikrokontroler dan HP

Untuk berkomunikasi dengan HP menggunakan mikrokontroler ataupun PC diperlukan sebuah kabel data. Jenis dari kabel data tergantung dari merk dan seri HP itu sendiri. Dalam perancangan ini, jenis HP yang digunakan adalah *Siemens C35*. Gambar 2.15 adalah model kabel data yang digunakan untuk berkomunikasi dengan PC atau mikrokontroler.



Gambar 2.15. Kabel data *Siemens C35*.

Kabel data *Siemens C35* menggunakan komponen dasar MAX232. IC ini digunakan sebagai konverter agar mampu berkomunikasi dengan PC. Pemasangan kapasitor 22uF digunakan untuk menghasilkan tegangan yang sesuai dengan level RS232, atau biasa disebut *vltage doubler*. Komunikasi yang disediakan adalah komunikasi serial UART dengan *baud rate* 19200 yang kompatibel dengan mikrokontroler. Pemasangan Resistor 3K3Ω untuk menjaga agar sinyal tidak hilang.

2.6. Driver relay

Relay merupakan salah satu komponen output yang paling sering digunakan baik pada industri, otomotif, ataupun peralatan elektronika lainnya. Relay berfungsi untuk menghubungkan atau memutuskan aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada

koilnya. Ada 2 macam relay berdasarkan tegangan untuk menggerakkan koilnya yaitu AC dan DC. Pada relay board ini digunakan relay DC dengan tegangan koil 12V DC, arus yang diperlukan sekitar 20 - 30mA.

2.7. Transistor

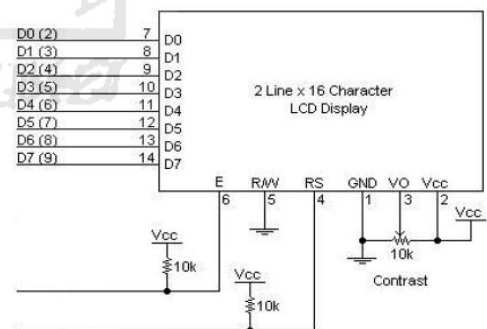
Transistor merupakan komponen aktif dimana arus, tegangan atau daya keluarannya dikendalikan oleh arus masukan. Didalam sistem komunikasi, transistor digunakan untuk menguatkan sinyal. Di dalam untai elektronis komputer transistor digunakan untuk saklar elektronis laju tinggi.

2.8. Penampil LCD 16x2

Penampil informasi yang lazim digunakan adalah LCD (*Liquid Crystal Display*). LCD yang digunakan adalah LCD *Dot Matrik* dengan jumlah karakter 16x2. LCD ini nantinya akan digunakan untuk menampilkan informasi nilai suhu, asap dan status dari proses.

Adapun fitur yang disajikan dalam LCD ini adalah:

- Terdiri dari 16 karakter dan 2 baris
- Mempunyai 192 karakter tersimpan
- Terdapat karakter generator terprogram
- Dapat dialamati dengan mode 8-bit dan 4-bit
- Dilengkapi dengan *backlight*.



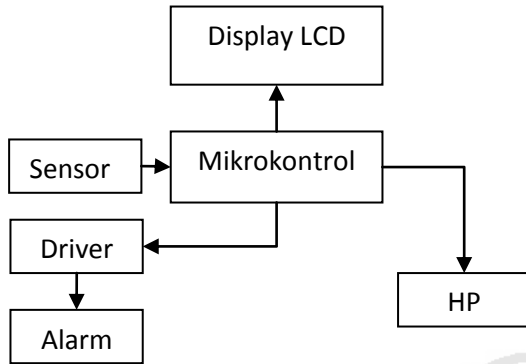
Gambar 2.12 LCD 16x2

Pada LCD ini terdiri dari 8 jalur data, 3 jalur kendali dan fasilitas pengaturan kontras serta *backlight*. LCD ini dapat dikendalikan dengan mikrokontroler atau mikroprosesor.

3.1 Perancangan Sistem

3.1. Perangkat Keras

Didalam perancangan sistem monitoring kebocoran gas LPG inidibutuhkan beberapa perangkat keras atau lebih dikenal dengan *hardware* Gambar 3.1 berikut menjelaskan perangkat keras yang digunakan pada sistem ini.



Gambar 3.1. Diagram blok perangkat keras sistem utama

Pada Gambar 3.1 ditunjukkan diagram kotak sistem yang digunakan dalam penelitian ini. Prinsip kerjanya adalah sensor gas akan mendeteksi adanya konsentrasi gas LPG di udara. Jika terdapat gas LPG maka akan terjadi perubahan resistensi dalam sensor, sehingga akan menyebabkan adanya perubahan tegangan. Perubahan tegangan ini dibaca oleh mikrokontroler Atmega8535.

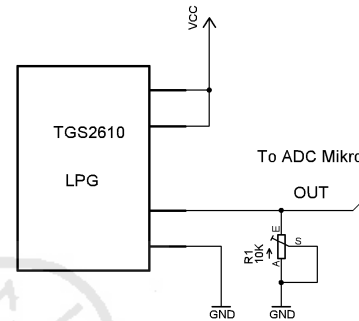
3.1.1. Mikrokontroler

Tugas dari mikrokontroler ini adalah mengendalikan seluruh sistem. Membaca sensor menampilkan informasinya dalam penampil LCD dan memberikan eksekusi atau perintah ke output atau aktuator. Mikrokontroler juga mengakses HP menggunakan komunikasi serial dan perintah-perintah AT-Command.

Mikrokontroler Atmega8535 merupakan sebuah mikrokontroler CMOS 8-bit yang diproduksi oleh Atmel, mikrokontroler ini bekerja pada tegangan rendah dan memiliki 8kb *flash* memori didalamnya yang bersifat ISP (*In System Programmable Flash Memory*). Mikrokontroler ini sudah dilengkapi dengan RAM, ROM, EEPROM pembangkit detak internal, *timer*, interupsi serta piranti untuk komunikasi serial. Sehingga dengan komponen-komponen ini menjadikan mikrokontroler sebagai komputer dalam satu chip.

3.1.2. Sensor Gas LPG TGS2610

Sensor yang digunakan untuk mendeteksi keberadaan Gas LPG diudara adalah sensor TGS2610. Sensor ini memiliki sifat terjadi perubahan R_s/R_o saat terdapat perubahan konsentrasi Gas LPG di permukaan sensor. Besarnya perubahan R_s/R_o tergantung dari banyaknya konsentrasi LPG di udara. Gambar 3.3 menunjukkan Gambar skema sensor LPG TGS2610.



Gambar 3.3 Sensor TGS2610

3.1.3. Penampil LCD 16x2

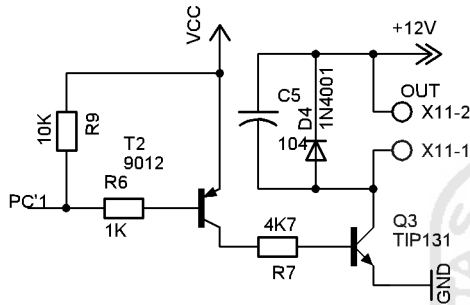
Untuk menampilkan data konsentrasi gas LPG di udara diperlukan media berupa display. Penampil yang digunakan adalah LCD dengan tipe dot matrix 16x2. Dibawah ini adalah fungsi dari masing-masing pin LCD.

Tabel 3.1. Fungsi Pin LCD 16x2

| No | Simbol | Level | Fungsi |
|----|-----------------|-------|---|
| 1 | V _{SS} | - | 0V |
| 2 | V _{DD} | - | +5V |
| 3 | V ₀ | -- | Kontras |
| 4 | RS | H/L | Register Select : H--Data, L-- Intruksi |
| 5 | R/W | H/L | H--Baca, L--Tulis |
| 6 | E | H,H-L | Sinyal Enable |
| 7 | DB0 | H/L | Data Bus untuk pengalamanatan 8-bit |
| 8 | DB 1 | H/L | |
| 9 | DB2 | H/L | |
| 10 | DB3 | H/L | Data Bus untuk pengalamanatan 4-bit dan 8-bit |
| 11 | DB4 | H/L | |
| 12 | DB5 | H/L | |
| 13 | DB6 | H/L | |
| 14 | DB7 | H/L | |
| 15 | A | -- | Back Light (+) |
| 16 | K | -- | Back Light (-) |

3.1.4. Rangkaian Driver Relay

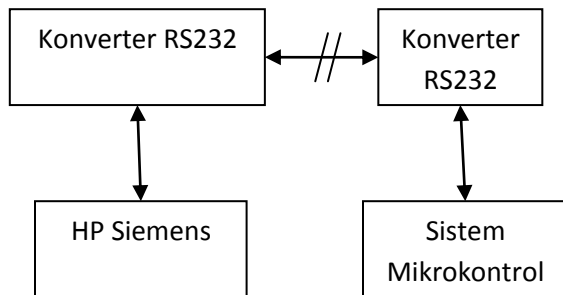
Dalam aplikasi ini digunakan dua buah driver, yaitu driver untuk menyalakan alarm serta driver untuk mengaktifkan relay, yang dapat dihubungkan ke piranti lain dengan tegangan yang berbeda. Mikrokontroler ATmega8535 memiliki tegangan kerja level TTL, yaitu 0 dan 5V. Untuk dapat mengendalikan piranti lain yang memiliki tegangan kerja yang berbeda seperti relai, *buzzer* mikrokontroler membutuhkan *driver*. Gambar 3.5 adalah rangkaian driver dengan transistor untuk mengaktifkan buzzer atau alarm kecil.



Gambar 3.5 Rangkaian driver buzzer

3.1.5 Komunikasi mikrokontroler dan HP Terminal

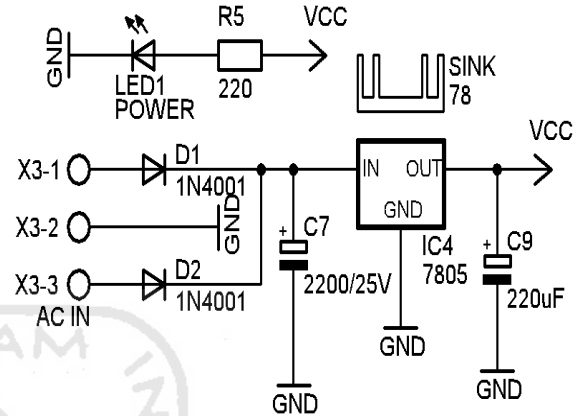
Kabel data difungsikan sebagai konverter atau penyesuai logika mikrokontroler dan logika pada HP. Konverter yang digunakan pada kabel data siemens C55 atau M35 adalah IC konverter Max232. Konsep penggunaan sistem konverter ditunjukkan pada Gambar 3.7.



Gambar 3.7 Konsep konverter RS232

3.1.6 Catu Daya

Perancangan sistem catu daya dalam penelitian ini disesuaikan dengan kebutuhan catu daya mikrokontroler. Mikrokontroler AT89S51 membutuhkan *supply* +5V dan *Ground*. Pada perancangan sistem catu daya ini menggunakan sistem penyearah dan regulator. Gambar 3.9 menunjukkan skema catu daya yang digunakan.



Gambar 3.9 Rangkaian power supply

3.2. Perancangan Perangkat Lunak

Pada perangkat lunak ini digunakan bahasa C dengan *compiler code vision AVR*. Pada perancangan perangkat lunak, *software* di-develop dengan *compiler* kemudian kode *hexa* hasil dari proses kompilasi di-download ke mikrokontroler dengan *software AVR Studio*.

Langkah-langkah perancangan lunak dapat ditarik garis besarnya terlebih dahulu setelah memahami cara kerja peralatan dan proses pengendalian yang akan dilaksanakan, antara lain :

- 1) Pembuatan *flow chart* urutan pengendalian program
- 2) Pembuatan *listing* program dalam bentuk file berekstensi .c
- 3) Pengubahan file berekstensi *obj* menjadi *hex*
- 4) Pengisian program dengan *software download AVR Studio*

3.2.1. Prosedur Penanganan Sistem

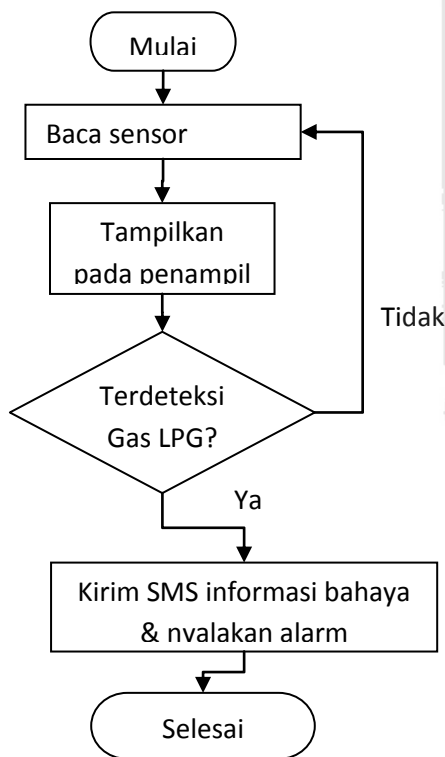
Pada program penanganan sistem, ada beberapa tahapan yang harus dilakukan. Adapun tahapannya adalah membaca sensor membandingkan dengan konstanta limit bahaya, menyalakan alarm saat keadaan bahaya terdeteksi, serta mengirim SMS ke *user*.

3.2.2 Perancangan Pesan SMS

Ketika sistem mendeteksi bahaya, maka sistem akan menyalakan alarm dan relai yang keluarannya dihubungkan dapat dihubungkan ke sistem yang lain. Untuk itu diperlukan perancangan mengenai pesan yang akan dikirim. Untuk mendapatkan data pesan SMS harus dilakukan dengan cara merubah isi pesan ke dalam format PDU. Dalam penelitian ini diinginkan pesan yang terkirim adalah 'Terdeteksi Kebocoran Gas LPG' dengan menggunakan *software PDUspy*, pesan tersebut akan diubah menjadi data PDU.

3.2.3 Perancangan flow chart

Untuk menangani seluruh tugas dalam skenario diatas, maka mikrokontroler diberi program dengan algoritma yang mampu menangani seluruh sistem. Gambar 3.10 adalah *flow chart* yang digunakan sebagai acuan program untuk menangani sistem.



Gambar 3.10. *Flow chart* program

Pada algoritma program gambar 3.10, yang pertama dilakukan adalah, sistem membaca TGS2610. Informasi ini kemudian ditampilkan dalam penampil LCD. Jika sistem mendeteksi adanya kebocoran gas LPG maka sistem akan menyalakan

alarm dan mengirimkan informasi SMS peringatan ke user.

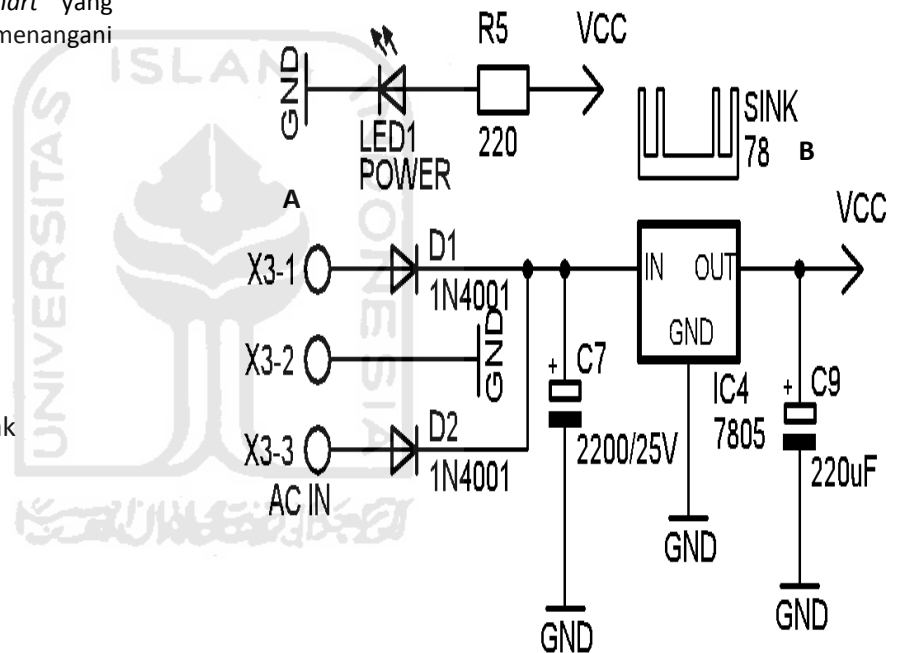
4. Pengujian system

4.1. Pengujian Alat

Hasil pengujian diperoleh dengan menggunakan alat ukur. Pengujian dilakukan pada sensor TGS2610, pengujian terhadap catu daya dan pengujian koneksi HP. Beberapa tahapan pengujian adalah sebagai berikut:

4.1.1. Pengujian terhadap catu daya

Pengujian terhadap catu daya dilakukan dengan mengukur nilai tegangan pada beberapa titik ukur. Gambar 4.1. adalah rencana titik uji pada rangkaian catu daya.



Gambar 4.1. Rencana pengujian catu daya

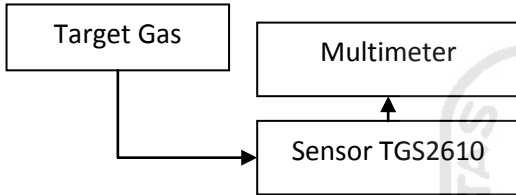
Sumber tegangan yang digunakan dalam sistem ini adalah sumber tegangan DC sebesar 12V dan 5V. Gambar rangkaian yang diuji tunjukkan pada Gambar 4.1. Power supply diuji pertama kali, karena digunakan untuk menjalankan sistem secara keseluruhan. Tegangan 5V digunakan untuk menjalankan mikrokontroler sedangkan tegangan 12V digunakan untuk menjalankan relay atau driver. Setelah dilakukan percobaan dan pengujian seperti pada Gambar 4.1 diperoleh hasil pengamatan yang disajikan pada Tabel 4.1.

Tabel 4.1. Hasil Pengamatan Catu Daya

| No | Titik Uji | Hasil ukur (V) |
|----|-----------|----------------|
| 1 | Titik A | 11,97 |
| 2 | Titik B | 13,8 |
| 3 | Titik C | 4.98V |

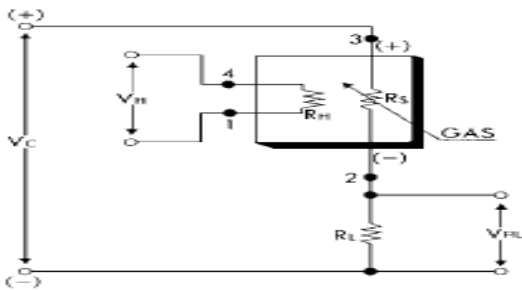
4.1.2. Pengujian Terhadap Sensor TGS2610

Pengujian terhadap sensor gas TGS2610 dilakukan dengan mengukur nilai tegangan output saat terdapat LPG dan saat kondisi normal. Gambar 4.2 adalah metode pengujian sensor LPG.



Gambar 4.2. Rencana pengujian sensor TGS2610

Hasil dari pengujian sensor TGS2610, digunakan untuk mengetahui kinerja dari sensor. Proses pengujian dilakukan dengan menggunakan multimeter, dengan mengukur tegangan output saat terdapat LPG dan saat tidak ada gas berdasarkan jarak antara sumber gas dan sensor gas. Untuk mendapatkan nilai resistansi pada sensor TGS 2610 dan kadar gas di udara dapat digunakan rumus sebagai berikut :



Gambar 4.3. Rangkaian sensor TGS 2610

$$R_s = \left(\frac{R_{beban} \times V_{input}}{V_{output}} \right) - R_{beban} \quad (4.1)$$

Dimana :

R_s = Resistansi sensor

R_{beban} = Resistansi beban

V_{input} = Tegangan input (5 V)

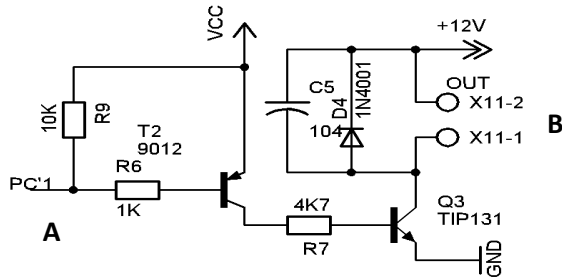
V_{output} = Tegangan output

Tabel 4.2 Pengamatan sensor gas TGS2610 berdasarkan percobaan dan perhitungan

| No | Tegangan (V) | Perhitungan Resistansi (Ohm) | Kadar Gas (PPM) | Jarak (Cm) |
|----|----------------|--------------------------------|-------------------|--------------|
| 1 | 3.2 | 562.5 | 9215.5 | 2 |
| 2 | 3 | 666.6 | 9136.3 | 3 |
| 3 | 2.5 | 1000 | 8550.8 | 3.5 |
| 4 | 2 | 1500 | 7921.8 | 4 |
| 5 | 1.7 | 1941.2 | 7285.71 | 4.3 |
| 6 | 1.5 | 2333.3 | 6833.3 | 4.5 |
| 7 | 1 | 4000 | 4409.5 | 5 |
| 8 | 0.8 | 6250 | 1268.6 | 5.1 |
| 9 | 0.6 | 6700 | 639.8 | 5.2 |
| 10 | 0.4 | 11500 | 0 | 5.5 |

4.1.3. Pengujian Terhadap Driver

Pengujian terhadap *driver* digunakan untuk mengetahui kinerja mikrokontroler dalam menyalakan dan mematikan alarm. Gambar 4.3 adalah titik uji rangkaian *driver*.



Gambar 4.4 Rencana pengujian *driver*

Pengujian terhadap *driver* dilakukan dengan memberikan logika dari mikrokontroler pada sisi input, kemudian mengukur tegangan keluaran driver. Dengan konfigurasi seperti pada Gambar 4.3 percobaan dilakukan dengan memberikan *input* logika yang berasal dari mikrokontroler. Tabel 4.3 adalah data dari hasil percobaan.

Tabel 4.3. Hasil percobaan terhadap driver

| PC.1 (Logika) | Titik A (V) | Titik B (V) | Alarm |
|---------------|-------------|-------------|-------|
| Low | 0 | 13 | On |
| High | 4.98 | 0 | Off |

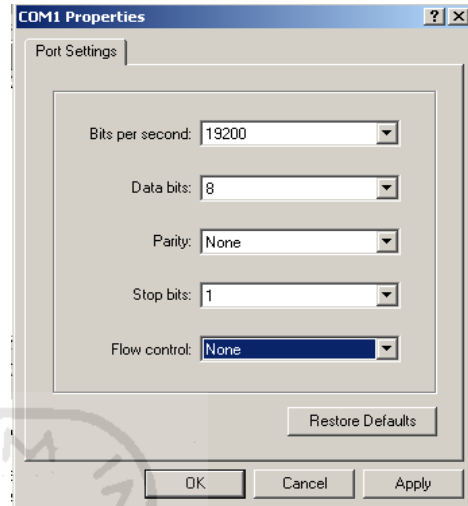
Dari hasil percobaan yang telah dilakukan, dapat diketahui bahwa jika diinginkan alarm dalam kondisi aktif atau *on*, maka mikrokontroler harus mengirimkan logika *low* ke masukan *driver*. Demikian juga sebaliknya untuk menjadikan alarm *off*, maka logika *high* harus diberikan pada masukan *driver*.

Saat logika *low* diberikan pada input basis transistor 9012 maka transistor PNP ini akan aktif, sehingga kondisinya saturasi dan kaki kolektor akan bernilai Vcc. Tegangan Vcc ini akan memberikan bias pada basis transistor NPN TIP31, sehingga transistor ini dalam kondisi saturasi dan arus akan mengalir ke buzzer atau output melalui kolektor transistor.

4.1.4. Pengujian Koneksi HP

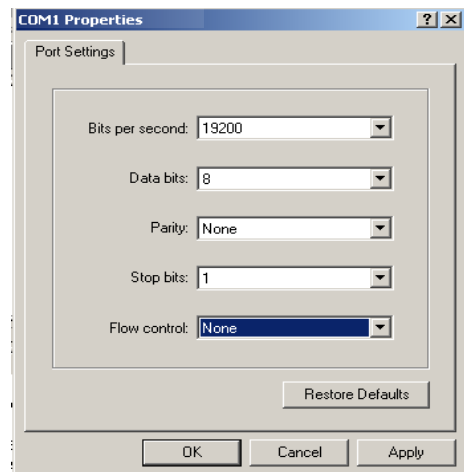
Pengujian perintah-perintah dan koneksi HP dilakukan dengan menggunakan bantuan *software hyperterminal*. Komunikasi serial antara

mikrokontroler dan HP menggunakan kecepatan *baud rate* 19200 bps, untuk itu perlu dilakukan pengaturan nilai kecepatan pada *software hyperterminal*. Gambar 4.4 adalah tampilan dan setting *hyperterminal*.



Gambar 4.4 Tampilan *hyperterminal*

Pada penelitian ini, HP yang digunakan adalah HP Siemens C55. HP ini memiliki koneksi dengan peralatan eksternal dengan menggunakan komunikasi serial UART (*Universal Asynchronous Receiver Transmitter*) dengan *baud rate* 19200bps. Untuk itu agar mampu melakukan pengujian terhadap HP maka harus menggunakan fasilitas *hyper terminal* pada windows. Pada Gambar 4.4. ditunjukkan konfigurasi pada *hyper terminal* untuk berkomunikasi dengan HP.



Gambar 4.4 Konfigurasi *hyper terminal*

Dengan menghubungkan ponsel ke COM1 pada komputer maka dapat diperoleh terminal untuk berkomunikasi dengan HP. Ketika sistem mengirim SMS diperoleh data pengujian kode PDU sebagai berikut

```
AT+CMGS=33
```

```
0001000D91265892228446F4000014D7A0D  
499741E5DAE10E8189C8284CFE1530A
```

Data-data PDU tersebut merupakan paket yang dikirim oleh sistem menuju ponsel. Data 'AT+CMGS=33' adalah data AT command yang digunakan untuk mengirim SMS. Data '000100' adalah data pengaturan SMS center secara default. Data 0D91265892228446F4 adalah data nomor HP tujuan, jika diterjemahkan menjadi +628292248644. Data PDU berikutnya adalah '0000' merupakan SMS referensi default. Kode PDU yang terakhir adalah isi sms, datanya adalah '14D7A0D499741E5DAE10E8189C8284CFE1530A ', data tersebut dapat diterjemahkan menjadi 'WARNING..! GAS BOCOR'.

4.1.5. Pengujian kinerja sistem secara keseluruhan

Pengujian sistem secara menyeluruh dilakukan dengan menggunakan simulasi terhadap perubahan konsentrasi gas, kemudian mengamati kinerja sistem mengenai alarm dan sms yang dihasilkan. Pengujian sistem dilakukan untuk mengetahui unjuk kerja dari alat yang dibangun. Pengujian dilakukan dengan minimal 10 (sepuluh) kali percobaan . Percobaan ini dilakukan dalam ruangan tertutup dengan memberikan jarak yang berbeda-beda antara sumber gas dengan alat pendeteksi kebocoran gas LPG. Ruangan untuk uji coba ini harus tertutup untuk meminimalisir adanya angin bertiup dari luar ruangan yang akan mempengaruhi kadar gas dalam ruangan. Berikut adalah hasil pengujian yang telah dilakukan.

Sistem diuji dengan memberikan simulasi gas bocor dari tabung gas LPG 3KG. Setelah diuji, sistem memberikan respon, alarm bunyi dan terdapat kiriman SMS kepada user yang isinya 'WARNING..! GAS BOCOR'.

5.1. Kesimpulan

1. Resistansi pada sensor gas TGS 2610 berbanding terbalik terhadap tegangan output sensor gas tersebut. Hal ini dapat dilihat dari semakin besar tegangan yang

diperoleh dari pengukuran maka semakin kecil nilai resistansinya.

2. Peletakan sensor gas TGS 2610 tidak boleh terlalu jauh dari sumber kebocoran gas. Hal ini akan sangat mempengaruhi sensor gas untuk dapat mendeteksi adanya kebocoran gas LPG, semakin jauh peletakan sensor gas maka kemampuan sensor gas untuk mendeteksi kebocoran gas LPG sangatlah kecil sehingga akan membutuhkan waktu yang lama agar sensor gas dapat mendeteksi adanya kebocoran gas LPG. Hal ini akan sangat membahayakan apabila adanya percikan – percikan api disekitar sumber kebocoran gas yang timbul sebelum gas terdeteksi oleh sensor gas.
3. Kepekatan gas LPG sangat dipengaruhi oleh kondisi ruangan dimana terdapat kebocoran gas LPG tersebut. Semakin baik sirkulasi udara dari luar ke dalam ruangan atau sebaliknya, maka semakin lama kadar gas LPG untuk menjadi pekat di dalam ruangan tersebut. Sehingga tingkat keamanan untuk terjadinya ledakan semakin tinggi.
4. Posisi pendeteksian gas pada titik aman yaitu sekitar 1000 Ppm campuran gas di udara.

DAFTAR PUSTAKA

Wardhana , L ; *Belajar Sendiri : Mikrokontroler AVR Seri ATmega 8535*. Yogyakarta : ANDI. 2006

Budiharjo , W ; *Elektronika Digital dan Mikroprosesor* , Andi, Yogyakarta , 2005.

Budiharjo, W ; *Interfacing Komputer dan Mikrokontroler* , Elex Media Komputindo , Jakarta , 2004.

Simanjuntak , T LH ; *Dasar - dasar Telekomunikasi* , Penerbit Alumni , Bandung , 2003.

<http://www.mikron123.com/index.php/display-LCD/DI-Smart-LCD-16x2/Detailed-product-flyer.html>

<http://www.mikrokontroller.com>

<http://www.atmel.com>

