

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Hidrogen Sulfida ( H<sub>2</sub>S )**

Gas H<sub>2</sub>S terbentuk akibat adanya penguraian zat-zat organik oleh bakteri. Oleh karena itu gas ini dapat ditemukan di dalam operasi pengeboran minyak/ gas dan panas bumi, lokasi pembuangan limbah industri, peternakan atau pada lokasi pembuangan sampah. Gas H<sub>2</sub>S mempunyai sifat dan karakteristik antara lain :

- Tidak berwarna tetapi mempunyai bau khas seperti telur busuk pada konsentrasi rendah sehingga sering disebut sebagai gas telur busuk.
- Merupakan jenis gas beracun.
- Dapat terbakar dan meledak pada konsentrasi LEL (Lower Explosive Limit ) 4.3% ( 43000 PPM ) sampai UEL (Upper Explosive Limite ) 46% ( 460000 PPM ) dengan nyala api berwarna biru pada temperature 500 F( 260<sup>0</sup>C )
- Berat jenis gas H<sub>2</sub>S lebih berat dari udara sehingga gas H<sub>2</sub>S akan cenderung terkumpul di tempat / daerah yang rendah. Berat jenis gas H<sub>2</sub>S sekitar 20 % lebih berat dari udara dengan perbandingan berat jenis H<sub>2</sub>S :1.2 atm dan berat jenis udara : 1 atm.
- H<sub>2</sub>S dapat larut (bercampur) dengan air ( daya larut dalam air 437 ml/100ml air pada 0<sup>0</sup>C; 186 ml/100 ml air pada 40<sup>0</sup>C ).
- H<sub>2</sub>S bersifat korosif sehingga dapat mengakibatkan karat pada peralatan Logam.

**Tabel 2.1.** Tingkat konsentrasi H<sub>2</sub>S dan efek fisik gas H<sub>2</sub>S

Tingkat H <sub>2</sub> S (ppm)	Efek pada manusia
0.13	Bau minimal yang masih terasa
4.6	Mudah dideteksi, bau yang sedang
10	Permulaan iritasi mata dan mulai berair
27	Bau yang tidak enak dan tidak dapat ditoleransi lagi.
100	Batuk-batuk, iritasi mata dan indera penciuman sudah tidak berfungsi
200 – 300	Pembengkakan mata dan rasa kekeringan di tenggorokan
500 – 700	Kehilangan kesadaran dan bisa mematikan dalam waktu 30 - 1 jam
> 700	Kehilangan kesadaran dengan cepat dan berlanjut kematian

Sumber: MSDS H<sub>2</sub>S ,ELNUSA

Menurut **ACGIH** (American Conference Of Governmental Industrial Hygienists) :

- Nilai ambang batas (TLV-TWA /Threshold Limit Value-Time Weighted Average) H<sub>2</sub>S adalah 10 PPM, yang didefinisikan sebagai konsentrasi rata – rata yang diperkenankan untuk pemaparan selama 8 jam sehari atau 40 jam seminggu. Pekerja dapat terpapar secara berulang tanpa menimbulkan gangguan Kesehatan pada konsentrasi 10 PPM (Occupational Exposure Limit for Chemical Substances).
- Sedangkan nilai ambang batas yang merekomendasikan bahwa pekerja tidak boleh terpapar H<sub>2</sub>S untuk jangka waktu maksimal 15 menit adalah bila paparan melebihi 20 PPM atau yang disebut dengan TLV – STEL ( Treshold Limit

Value – Short Term Exposure Limit ). Efek fisik gas H<sub>2</sub>S pada tingkat rendah dapat menyebabkan terjadinya gejala-gejala sebagai berikut :

- Sakit kepala atau pusing
- Badan terasa lesu
- Hilangnya nafsu makan
- Rasa kering pada hidung, tenggorokan dan dada
- Batuk – batuk
- Kulit terasa perih

## 2.2. Gas Amonia (NH<sub>3</sub>)

Amonia adalah senyawa kimia dengan rumus NH<sub>3</sub> Massa molar = 17.0306 g/mol Penampilan = Gas tak berwarna dan berbau tajam.

Tabel 2.2. Sifat gas Ammonia

Sifat-sifat	
Massa jenis and fase	0.6942 g/L, gas.
Kelarutan dalam air	89.9 g/100 ml pada 0 °C.
Titik lebur	-77.73 °C (195.42 K)
Temperatur autosulutan	651 °C
Titik didih	-33.34 °C (239.81 K)
Keasaman (pK <sub>a</sub> )	9.25
Kebasaan (pK <sub>b</sub> )	4.75

Sumber: Chemtrade Logistics Inc.,2009

Tabel.2.3. Struktur Gas Ammonia

<b>Struktur</b>	
Bentuk molekul	piramida segitiga
Momen dipol	1.42 D
Sudut ikatan	107.5°

Sumber: Chemtrade Logistics Inc.,2009

Amonia sendiri adalah senyawa kaustik dan dapat merusak kesehatan. Stabil pada suhu kamar, tetapi dapat meledak oleh panas akibat kebakaran. Larut dalam air membentuk ammonium hidroksida. Administrasi Keselamatan dan Kesehatan Pekerjaan Amerika Serikat memberikan batas 15 menit bagi kontak dengan amonia dalam gas berkonsentrasi 35 ppm volum, atau 8 jam untuk 25 ppm volum. Kontak dengan gas amonia berkonsentrasi tinggi dapat menyebabkan kerusakan paru-paru dan bahkan kematian. Sekalipun ammonia di AS diatur sebagai gas tak mudah terbakar, amonia masih digolongkan sebagai bahan beracun jika terhirup, dan pengangkutan ammonia berjumlah lebih besar dari 3.500 galon (13,248 L) harus disertai surat izin.

#### **Efek Jangka Pendek (Akut)**

Iritasi terhadap saluran pernapasan, hidung, tenggorokan dan mata terjadi pada 400-700 ppm. Sedang pada 5000 ppm menimbulkan kematian. Kontak dengan mata dapat menimbulkan iritasi hingga kebutaan total. Kontak dengan kulit dapat menyebabkan luka bakar (frostbite).

### **Efek Jangka Panjang (Kronis)**

Menghirup uap asam pada jangka panjang mengakibatkan iritasi pada hidung, tenggorokan dan paru-paru. Termasuk bahan teratogenik. Nilai Ambang Batas : 25 ppm (18 mg/m<sup>3</sup>) (ACGIH 1987-88) STEL 35 ppm (27 mg/m<sup>3</sup>). Toksisitas : LD<sub>50</sub> = 3 mg/kg (oral, tikus). LC 50 = 200 ppm (tikus menghirup 4 jam). Amoniak dalam air amat beracun bagi ikan, udang dan binatang air lainnya. Dapat menimbulkan kesuburan tanaman air (eutropia). NH<sub>3</sub> dalam air dapat dibuang dengan proses *stripping* (pH optimum ± 12) atau dengan proses mikrobiologi. Limbah amoniak dapat dinetralkan dengan asam sulfat (pupuk ZA). Baku mutu ambien untuk pencemaran amoniak adalah 2 ppm. Asap tebal akibat kecelakaan dalam transportasi pengangkutan amoniak dapat disemprot dengan air.

Apalagi di daerah Gresik terdapat pabrik penghasil pupuk terbesar di Indonesia yang menggunakan amonia dalam proses produksinya, sehingga sangat besar kemungkinannya udara di Gresik terpolusi dengan gas amonia (NH<sub>3</sub>).

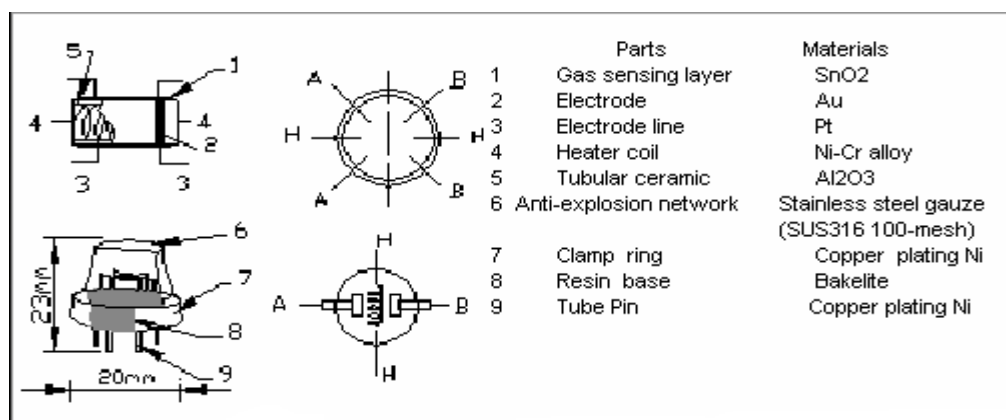
### **2.3. Sensor Gas Semikonduktor**

Sensor secara umum didefinisikan sebagai alat yang mampu menangkap fenomena kimia atau fisika, kemudian mengubahnya menjadi sinyal elektrik baik arus listrik ataupun tegangan. Fenomena kimia yang dimaksud dapat berupa konsentrasi dari bahan kimia baik cairan maupun gas. Dari definisi ini maka sensor merupakan alat elektronik yang begitu banyak dipakai dalam kehidupan manusia saat ini. Seperti sensor temperatur yang digunakan dalam mengatur temperatur ruangan pada AC, lantai lift yang kita tuju, menghasilkan perubahan pada layar komputer atau televisi, serta gerakan pada lift adalah contoh mudah

sensor secara luas. Sedangkan sensor semikonduktor adalah sejumlah komponen elektronik yang menggunakan sifat-sifat materi semikonduktor, yaitu Silikon, Germanium dan Gallium Arsenide. Alat-alat semikonduktor ini menggunakan konduksi elektronik dalam bentuk padat (*solid state*), bukannya bentuk hampa (*vacuum state*) atau bentuk gas (*gaseous state*). Elemen sensing yang digunakan adalah material Tin oksida ( $\text{SnO}_2$ ). Sensor ini tidak mahal, kecil, sudah tersedia luas dan memiliki sensitifitas tinggi. Mekanisme utama untuk reaksi gas dengan metal oksida terjadi pada temperatur tinggi yaitu  $200^\circ\text{C} - 600^\circ\text{C}$ .

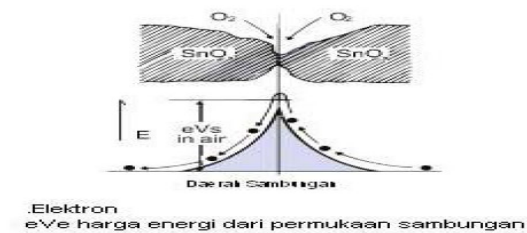
#### 2.4. Cara Kerja Sensor Gas Secara Umum

Pada sensor MQ137 pendeteksi gas Amoniak menggunakan logam oksida Tin oksida ( $\text{SnO}_2$ ) dipanaskan pada temperatur tinggi di udara, oksigen akan diserap pada permukaan kristal menghasilkan muatan listrik negatif. Kemudian elektron donor pada permukaan kristal akan ditransfer ke oksigen penyerapan sehingga dihasilkan listrik bermuatan negatif. Didalam sensor, arus listrik mengalir melewati daerah sambungan (*grain boundary*) dari kristal  $\text{SnO}_2$ . Pada daerah sambungan penyerapan oksigen mencegah muatan untuk bergerak bebas.



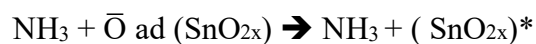
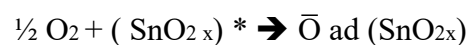
Gambar 2.1. Struktur dan konfigurasi sensor MQ137

Jika konsentrasi gas menurun, proses deoksidasi akan terjadi, kerapatan permukaan dan muatan negatif oksigen akan berkurang dan mengakibatkan menurunnya ketinggian penghalang dari daerah sambungan, misalnya terdapat adanya gas  $\text{NH}_3$  yang terdeteksi. Ilustrasi gambar ketika terjadi penyerapan gas  $\text{O}_2$  oleh sensor, dapat dilihat pada gambar berikut ini :



Gambar 2.2. Ilustrasi gambar penyerapan gas  $\text{O}_2$  oleh sensor

Persamaan reaksi kimia yang terjadi adalah sebagai berikut:

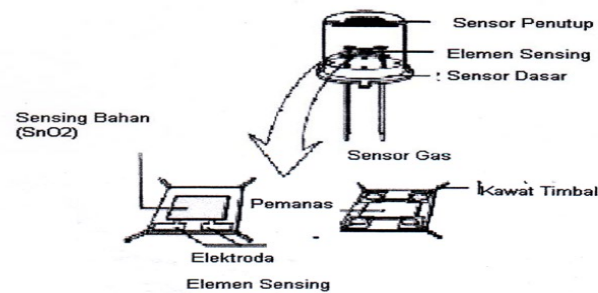


\* = elektron bebas

Dengan menurunnya penghalang maka resistansi sensor juga akan turut menurun. Sensor gas MQ136 & MQ137 merupakan salah satu sensor yang dipakai dalam penelitian ini. Sensor ini adalah sebuah sensor kimia atau sensor gas yang mempunyai nilai resistansi ( $R_s$ ) yang akan berubah bila terkena gas Ammonia ( $\text{NH}_3$ ) dan Hidrogen Sulfida ( $\text{H}_2\text{S}$ ). Jika sensor tersebut mendeteksi keberadaan salah satu komponen gas diudara dengan tingkat konsentrasi tertentu maka resistansi elektrik sensor tersebut akan menurun. Sehingga akan menyebabkan tegangan yang dihasilkan oleh output sensor akan semakin besar. Selain itu, sensor juga mempunyai sebuah pemanas (*heater*) yang digunakan

untuk membersihkan ruangan sensor dari kontaminasi udara luar, agar sensor dapat bekerja kembali secara efektif (Margaretha, S, 2010).

Secara umum bentuk dari sensor gas Gas dapat dilihat pada gambar 2.3 berikut:



Gambar 2.3. Ilustrasi gambar komponen sensor MQ.

Untuk mengukur karakteristik sensitivitas dari polutan gas buang industri, semua data diuji pada kondisi standar. Sumbu Y axis menunjukkan rasio resistansi sensor ( $R_s/R_0$ ), dengan ketentuan :

\*  $R_s$  = sensor perlawanan dari gas dengan berbagai konsentrasi.

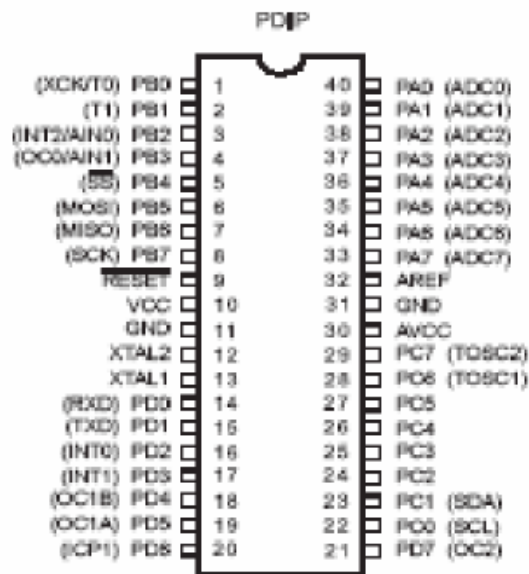
\*  $R_0$  = resistansi sensor dalam udara bersih

## 2.5. Mikrokontroler At Mega-8535.

Mikrokontroler adalah suatu chip yang dapat digunakan sebagai pengontrol utama sistem elektronika, misalnya sistem pengukur suhu digital (thermometer digital), sistem keamanan rumah (home remote system), sistem kendali mesin industri, robot penjajak bom, dan lain-lain. Mikrokontroler AVR (*Alf and Vegar's Rics prosesor*) memiliki arsitektur RICS 8 bit, dimana semua instruksi dikemas dalam kode 16 bit (16-bits word) dan sebagian besar instruksi dieksekusi dalam 1(satu) siklus clock, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus clock. Secara umum, AVR dapat ATtiny, keluarga

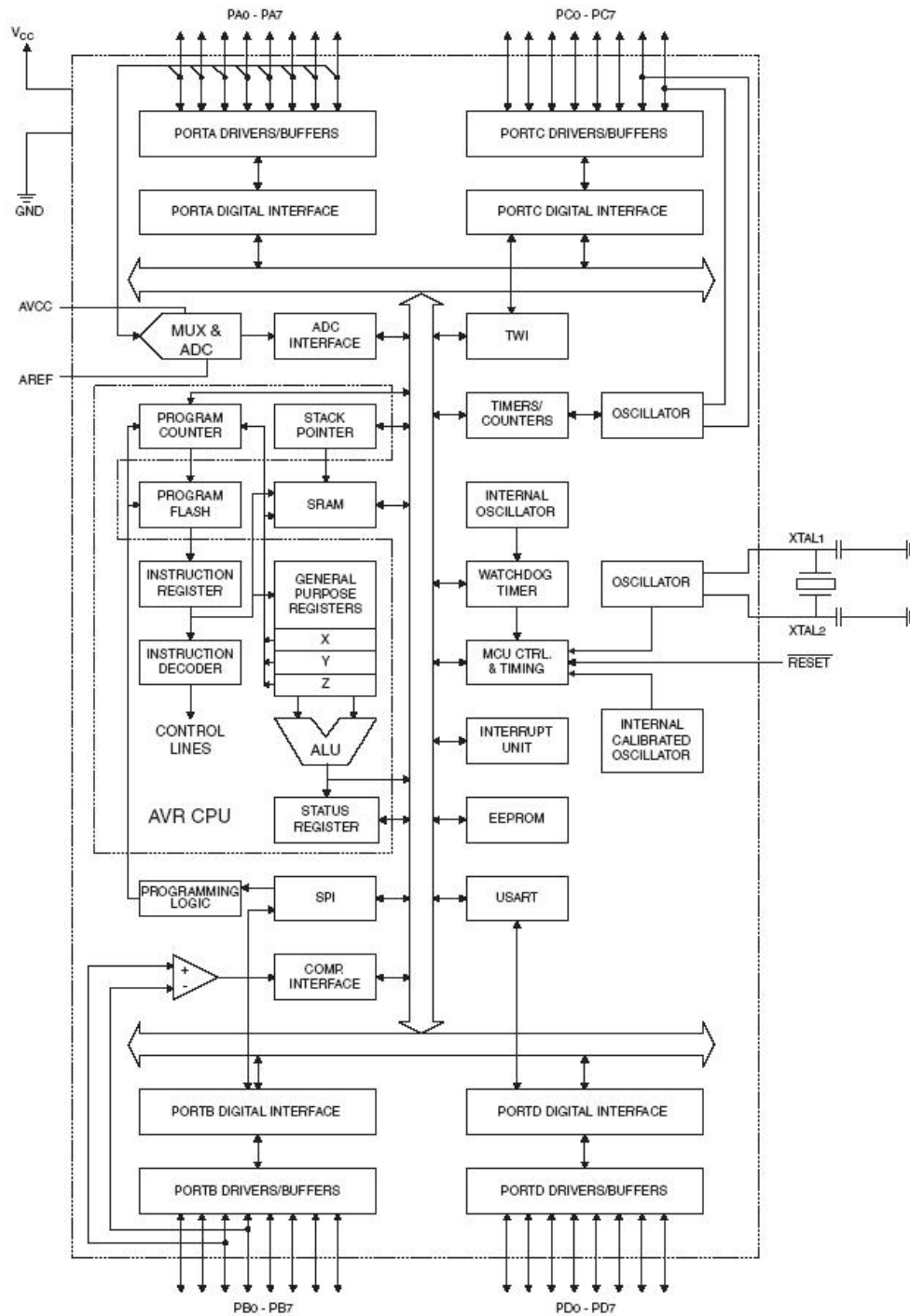


AT90Sxx, keluarga ATMega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan intruksi yang digunakan, mereka bisa dikatakan sama. dikelompokkan menjadi 4 kelompok, yaitu keluarga MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.



Gambar 2.4. Mikrokontroler ATMega8535

Guna memaksimalkan performa dan paralelisme, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Arsitektur CPU dari AVR ditunjukkan oleh gambar 3.7. Instruksi pada memori program dieksekusi dengan pipelining single level. Selagi sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program (Suyantoro, Fl. Sigit, 2007)



Gambar 2.5. Blok Diagram Atmega 8535  
 Sumber: ATmega 8535, ATMEL datasheet, [www.atmel.com](http://www.atmel.com).

AVR menggabungkan banyak instruksi dengan 32 register, dan langsung terhubung ke Logic Unit aritmatika (ALU), sehingga memungkinkan dua register independen untuk diakses dalam satu instruksi dieksekusi dalam satu jam siklus. Arsitektur yang dihasilkan lebih efisien mencapai throughputs sampai sepuluh kali lebih cepat daripada mikrokontroler CISC konvensional.

ATMega8535 menyediakan fitur sebagai berikut: Read-While-Write 8K byte In-System Programmable Flash, 512 byte EEPROM, 512 byte SRAM, 32 I / O, 32 register, tiga Timer / Counter dengan compare mode, internal dan eksternal interrupt, serial programmable USART, dengan byte berorientasi pada Two-wire Serial Interface, 8-channel 10-bit ADC dengan diferensial masukan opsional dengan gain diprogram dalam paket TQFP, Watchdog Timer Oscillator diprogram dengan internal, port SPI serial, dan enam perangkat lunak mode penghematan daya dipilih. Modus idle berhenti CPU sementara memungkinkan SRAM, Timer / Counter, SPI port, dan interrupt system untuk meneruskan fungsi. Modus Power-down menyimpan isi mendaftarkan namun membekukan Osilator, menonaktifkan semua lainnya keping fungsi sampai yang selanjutnya mengganggu atau hardware Ulang. Dalam Power-simpan modus, yang asynchronous waktu terus untuk menjalankan, memungkinkan pengguna untuk mempertahankan timer dasar sementara yang istirahat dari perangkat tidur. Para ADC Noise Reduction modus berhenti yang CPU dan semua I / O modul kecuali asynchronous Timer dan ADC, untuk memperkecil beralih kebisingan selama ADC konversi. Dalam modus siaga, kristal / resonator osilator berjalan sementara istirahat dari perangkat tidur. Hal ini memungkinkan sangat cepat start-up gabungan dengan daya rendah konsumsi.

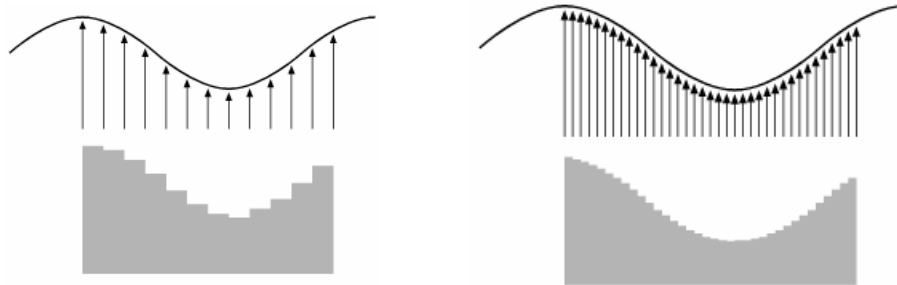
Dalam Diperpanjang siaga modus, baik yang utama osilator dan yang asynchronous Timer melanjutkan untuk dijalankan. Perangkat ini diproduksi dengan menggunakan kepadatan tinggi Atmel teknologi nonvolatile memori. On-chip ISP Flash memungkinkan memori program untuk diprogram In-System melalui antarmuka SPI serial, oleh seorang programmer memori nonvolatile konvensional, atau oleh program Boot On-chip yang berjalan pada inti AVR. Program boot dapat menggunakan antarmuka untuk men-download program aplikasi dalam memori Flash Aplikasi. Perangkat Lunak di Flash Boot bagian akan terus berjalan sementara yang aplikasi Bagian flash diperbarui, menyediakan benar Baca-Sementara-Tulis operasi.

Dengan menggabungkan sebuah 8-bit RISC CPU dengan In-System Programmable Diri-Flash pada sebuah chip monolitik, Atmel ATMEGA8535 merupakan mikrokontroler yang kuat yang memberikan solusi yang efektif sangat fleksibel dan biaya untuk banyak tertanam aplikasi kontrol. AVR ATMEGA8535 didukung dengan rangkaian lengkap program dan pengembangan system alat termasuk: C compiler, makro perakitan, program debugger / simulator, terhubung emulator, dan evaluasi kit.

## **2.6. ADC (*Analog Digital Converter*)**

Mikrokontroler ATMEGA8535 memiliki 8 channel ADC10-bit, ADC (Analog to Digital Converter) berfungsi untuk mengubah data analog menjadi data digital. ADC memiliki 2 karakter prinsip, yaitu *kecepatan sampling* dan *resolusi*. Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal

analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling dinyatakan dalam *sample per second (SPS)*.



Gambar 2.6. ADC kecepatan sampling rendah dan kecepatan sampling tinggi.

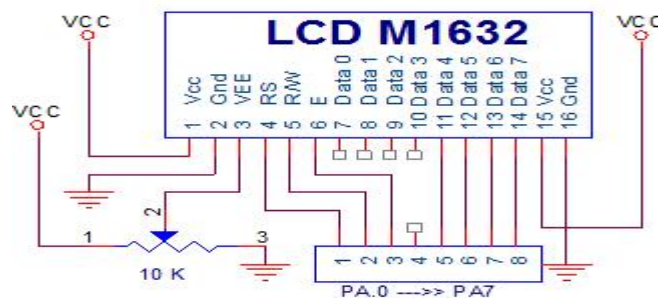
Resolusi ADC menentukan *ketelitian* nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ( $2^n - 1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit. Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar  $60\% \times 255 = 153$  (bentuk decimal) atau 10011001 (bentuk biner).

$$\begin{aligned} \text{Signal} &= (\text{sample} / \text{max\_value}) * \text{reference voltage} \\ &= (153 / 255) * 5 \\ &= 3 \text{ Volts} \end{aligned}$$

ADC pada ATMEGA8535 adalah jenis *10 bit successive approximation* dengan tegangan referensi maksimum 5 volt. Pada universal board M.B.3.2 tegangan referensi dibuat *fix tidak dapat diubah yaitu 5 volt* yang diambil dari *tegangan sumber (Vcc)*.

## 2.7. LCD (*Liquid Crystal Display*)

Banyak sekali kegunaan LCD dalam perancangan suatu system yang menggunakan mikrokontroler. LCD berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler. LCD yang digunakan adalah jenis LCD M1632 merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Dengan alasan di atas maka dalam pembuatan alat tugas akhir ini penulis menggunakan LCD M1632 dengan tampilan 16 x 2 baris.



Gambar 2.7. Modul dari *Liquid Crystal Display*

## 2.8. Codevision AVR

CodeVisionAVR merupakan sebuah cross-compiler C, Integrated Development Environment (IDE), dan Automatic Program Generator yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat

dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, dan XP serta Win7. Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem embedded. Codevision AVR merupakan software compiler dengan menggunakan bahasa basic yang dibuat untuk melakukan pemrograman chip-chip mikrokontroler tertentu, salah satunya ATmega 8535. File object COFF hasil kompilasi dapat digunakan untuk keperluan debugging pada tingkatan C, dengan pengamatan variabel, menggunakan debugger Atmel AVR Studio. IDE mempunyai fasilitas internal berupa software AVR Chip In-System Programmer yang memungkinkan Anda untuk melakukan transfer program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis.

Software In-System Programmer didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, FuturlecJRAVR dan MicroTronics ATCPU/Mega2000 programmers/development boards. Untuk keperluan debugging sistem embedded, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal.

CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengujikan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- Set-up akses memori eksternal
- Identifikasi sumber reset untuk chip

- Inisialisasi port input/output
- Inisialisasi interupsi eksternal
- Inisialisasi Timer/Counter
- Inisialisasi Watchdog-Timer
- Inisialisasi UART (USART) dan komunikasi serial berbasis buffer yang digerakkan oleh interupsi
- Inisialisasi Pembanding Analog
- Inisialisasi ADC
- Inisialisasi Antarmuka SPI
- Inisialisasi Antarmuka Two-Wire
- Inisialisasi Antarmuka CAN
- Inisialisasi Bus I2C, Sensor Suhu LM75, Thermometer/Thermostat DS1621 dan Real-Time Clock PCF8563, PCF8583, DS1302, dan DS1307
- Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20
- Inisialisasi modul LCD

## **2.9. Bahasa C**

### **2.9.1.sejarah dan standar C**

Akar dari bahasa c adalah dari bahasa bcpl yang dikembangkan oleh martin richards pada tahun 1967. bahasa ini memberikan ide kepada ken thomson yang kemudian mengembangkan bahasa yang disebut dengan B pada tahun 1970. Perkembangan selanjutnya Dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT



dan T Bell laboratories). Bahasa C pertama kali digunakan di computer Digital Equipment Corporation PDP-11 yang menggunakan system operasi UNIX. C adalah bahasa yang standar, artinya suatu program yang ditulis dengan bahasa C tertentu akan dapat dikonversi dengan bahasa C yang lain dengan sedikit modifikasi. Standar bahasa C yang asli adalah standar UNIX. Patokan dari standar UNIX ini di ambil dari buku yang ditulis oleh Brian Kerninghan dan Dennis Ritchie berjudul "The C programming Language", diterbitkan oleh Prentice-Hall tahun 198. Deskripsi C dari Kerninghan dan Ritchie ini kemudian dikenal secara umum sebagai "K dan R C".

### **2.9.2. Tipe Data**

Di dalam bahasa pemrograman computer, data yang digunakan umumnya dibedakan menjadi data nilai numerik dan nilai karakter. Nilai numeric dapat dibedakan lagi menjadi nilai numeric pecahan. Nilai numeric pecahan dapat dibedakan lagi menjadi nilai numeric pecahan ketetapan tunggal dan nilai numeric pecahan ketetapan ganda. Bahasa-bahasa pemrograman computer membedakan data ke dalam beberapa tipe dengan tujuan supaya data menjadi efisien dan efektif. C menyediakan lima macam tipe data integer ( nilai numeric bulat yang dideklarasikan dengan `int`), floatingpoint (nilai numeric pecahan ketetapan tunggal yang dideklarasikan dengan `float`), double-precision (nilai numeric pecahan ketetapan ganda yang dideklarasikan dengan `double`).

Tabel 2.4. Tipe-tipe data dasar

Type	Lenght	Range
Unsigned chart	8 bits	0 to 255
Chart	8 bits	-128 to 127
Enum	16 bits	-32,768 to 32,767
Unsigned int	16 bits	0 to 65,535
Short int	16 bits	-32,768 to 32,767
Int	16 bits	-32,768 to 32,767
Unsigned long	32 bits	0 to 4,294,967,295
Long	32 bits	-2,147,483,648 to 2,147,483,647
Float	32 bits	$3.4 \times 10^{-38}$ to $3.4 \times 10^{+38}$
Double	64 bits	$1.7 \times 10^{-308}$ to $1.7 \times 10^{+308}$
Long double	80 bits	$3.4 \times 10^{-4932}$ to $1.1 \times 10^{+4932}$

### 2.9.3. Pengenalan Fungsi-Fungsi Dasar

Fungsi main harus ada program, sebab fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda { di awal fungsi menyatakan awal tubuh fungsi dan sekaligus awal eksekusi program, sedangkan tanda } di akhir fungsi merupakan akhir tubuh fungsi dan sekaligus adalah akhir program. Jika program terdiri atas lebih dari satu fungsi, fungsi main biasa di tempatkan pada posisi yang paling atas dalam pendefinisikan fungsi. Hal ini hanya merupakan kebiasaan. Tujuannya untuk memudahkan pencarian terhadap program utama bagi pemrogram. Jadi bukanlah merupakan suatu keharusan.

## Pengarah Preprosesor

Digunakan untuk mendefinisikan prosesor yang digunakan, dalam hal ini adalah untuk mendefinisikan jenis mikrokontroler yang digunakan. Dengan pengarah preprosesor ini, pendeklarasian register-register dan penamaannya dilakukan pada file lain yang disisipkan dalam program utama dengan sintaks sebagai berikut:

```
# include <nama_prosesor>
```

Contoh:

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

Konstanta dan Variabel

Konstanta berisi data yang nilainya tetap, tidak dapat diubah selama program dijalankan.

Contoh: **const** [*tipe\_data*] [*nama\_konstanta*]=[*nilai*]

```
const char kontantaku = 0x10;
```

Variabel berisi data yang bisa berubah nilainya pada saat program dijalankan.

Contoh: [*tipe\_data*] [*nama\_variabel*]=[*nilai\_awal*]

```
char variabelku; (atau)
```

```
char variabelku = 0x20;
```

## Pengenal

Pengenal atau *identifier* merupakan sebuah nama yang didefinisikan oleh pemrogram untuk menunjukkan identitas dari sebuah konstanta, variabel, fungsi, label atau type data khusus. Pemberian nama sebuah pengenal dapat ditentukan bebas sesuai keinginan pemrogram tetapi tetap mengikuti aturan berikut ini:

- Karakter pertama tidak boleh menggunakan angka,
- Karakter kedua dapat berupa huruf, angka atau garis bawah,
- Tidak boleh menggunakan spasi,
- Bersifat *case Sensitive*, artinya huruf kapital dan huruf kecil dianggap berbeda,
- Tidak boleh menggunakan kata-kata yang merupakan sintaks maupun operator dalam pemrograman C, misalnya: void, bit, long, char dan lain-lain.

### Operator Aritmatika

Merupakan beberapa operator yang digunakan untuk melakukan perhitungan aritmatika.

Tabel 2.5. Operator aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Pembagian

Operator \*, / dan % mempunyai prioritas paling tinggi dibandingkan + dan -.

Contoh:

$$2 + 3 * 4$$

Hasilnya bukan 20, tetapi 14.

$$\text{Prosesnya: } 2 + (3 * 4) = 14$$

## Operator Perbandingan

Digunakan untuk membandingkan 2 buah data. Hasil operator perbandingan bukan berupa nilai data tetapi hanya bernilai benar („1“) atau salah („0“) saja.

Tabel 2.5. Operator perbandingan

Operator	Contoh	Keterangan
==	x==y	Benar : keduanya sama
!=	x!=y	Benar
>	x>y	Benar : x lebih besar dari y
<	x<y	Benar : x lebih kecil dari y
>=	x>=y	Benar : x lebih besar atau sama dengan y
<=	x<=y	Benar : x lebih kecil atau sama dengan y

## Operator Logika

Digunakan untuk membentuk suatu logika atas dua buah kondisi atau lebih.

Tabel 2.6. Operator logika

Operator	Keterangan
&&	Logika AND
	Logika OR
!	Logika NOT

## Operator Bit

Merupakan operator logika yang bekerja pada level bit. Berbeda dengan operator logika, dimana pada operator logika menghasilkan benar atau salah (boolean), pada operator ini akan dihasilkan data dalam bentuk biner.

Tabel 2.7. Operator bit

Operator	Keterangan
&	Operasi AND level bit (biner)
	Operasi OR
^	Operasi XOR
~	Operasi NOT
<<	Operasi untuk menggeser ke kiri
>	Operasi untuk menggeser ke kanan

### Fungsi Pustaka

Bahasa C memiliki sejumlah fungsi pustaka yang berada pada file-file tertentu dan sengaja disediakan untuk menangani berbagai hal dengan cara memanggil fungsi-fungsi yang telah dideklarasikan di dalam file tersebut. Dalam banyak hal, pustaka yang tersedia tidak berbentuk kode sumber, melainkan dalam bentuk yang telah dikompilasi. Pada saat proses *linking*, kode-kode dari fungsi ini akan dikaitkan dengan kode-kode yang dituliskan oleh pemrogram. Sintaks untuk menggunakan fungsi pustaka adalah sebagai berikut:

**#include** <nama\_file\_pustaka> Contoh:

**#include** <lcd.h>

**#include** <string.h>

**#include** <mem.h>

dll

### While (kondisi)

```
{
// sebuah pernyataan atau blok pernyataan
}
```

Jika pernyataan yang akan diulang hanya berupa sebuah pernyataan saja

maka tanda {dan} bisa dihilangkan.

Contoh:

**Unsigned char** a=0;

...

**While** (a<10)

{ PORTC=a; a++;

}

Pernyataan di atas akan mengeluarkan data a ke port c secara berulang-ulang. Setiap kali pengulangan, nilai a akan bertambah 1 dan setelah nilai a tidak lagi lebih kecil dari 10 maka pengulangan akan berhenti.

### **Pernyataan do...while**

Pernyataan do...while hampir sama dengan pernyataan while, yaitu pernyataan yang digunakan untuk pengulangan sebuah pernyataan atau blok pernyataan secara terus menerus selama kondisi tertentu masih terpenuhi.

Bentuk pernyataan while adalah sebagai berikut:

**do** {

//sebuah pernyataan atau blok pernyataan

**}while** (kondisi);

### **Pernyataan While**

Digunakan untuk pengulangan sebuah pernyataan atau blok pernyataan secara terus menerus selama kondisi tertentu masih terpenuhi. Bentuk pernyataan while adalah sebagai berikut: Yang membedakan antara pernyataan while dengan do..while adalah bahwa pada pernyataan while pengetesan kondisi dilakukan

terlebih dahulu, jika kondisi terpenuhi, maka barulah blok pernyataan dikerjakan. Sebaliknya, pada pernyataan `do..while`, blok pernyataan dikerjakan terlebih dahulu setelah itu baru dilakukan pengecekan kondisi, jika kondisi terpenuhi maka dilakukan pengulangan pernyataan atau blok pernyataan lagi. Sehingga, dengan demikian pada pernyataan `do...while` blok pernyataan pasti akan dikerjakan minimal satu kali, sedangkan pada pernyataan `while` blok pernyataan belum tentu dikerjakan.

### **Pernyataan For**

Digunakan untuk melakukan pengulangan sebuah pernyataan atau blok pernyataan, tetapi berapa kali jumlah pengulangannya dapat ditentukan terlebih spesifik. Bentuk pernyataan `for` adalah sebagai berikut:

```
for (nilai_awal , kondisi , perubahan)
{
//sebuah pernyataan atau blok pernyataan
}
```

```
Unsigned int a;
```

```
For (a=1 , a<10, a++)
```

```
{ PORTC=a;
```

Pertama kali nilai `a` adalah 1, kemudian data `a` dikeluarkan ke port `C`. Selanjutnya, data `a` dinaikkan (`a++`) jika kondisi `a < 10` masih terpenuhi, maka data `a` akan terus dikeluarkan ke port `C`



## ARRAY

Merupakan sekumpulan data dengan type yang sama yang dideklarasikan dalam satu nama variabel. Array dapat memiliki dimensi satu, dua atau lebih, tetapi umumnya hanya sampai 3 dimensi.

Contoh:

```
Char data[7];
```

Artinya, variabel array dengan nama *data* terdiri dari 8 elemen data yang bertype sama, yaitu: char

Deklarasi variabel array di atas dapat juga diberi inisial awal seperti contoh berikut:

```
Char dataku[7] = {0, 1, 2, 3, 4, 5, 6};
```

Kita juga dapat mendeklarasikan variabel array dengan jumlah elemen yang tidak ditentukan, tetapi untuk hal ini inisial awal datanya harus diberikan.

Contoh:

```
Char data[]; //tidak boleh dilakukan
```

```
Char data[]={0, 1, 2, 3, 4, 5, 6};
```

Deklarasi variabel array dengan jumlah elemen yang tidak ditentukan biasanya digunakan untuk menyimpan data string karena kita sering malas untuk menghitung berapa jumlah elemennya seperti contoh berikut:

```
Char data[]={“Mikrokontroler ATmega8535”};
```

**atau**

```
Char data[]=”Mikrokontroler ATmega8535”;
```

Untuk mengakses data yang berada di dalam variabel array dapat

dilakukan dengan cara berikut:

```
Nama_array [indeks];
```

Indeks adalah penunjuk data elemen tertentu pada variabel array yang nilainya selalu dimulai dari 0

