BAB II
LANDASAN TEORI

Bab ini membahas mengenai teori-teori dasar penunjang perancangan termometer badan yang akan dibuat. Landasan teori ini mencakup teori dasar termometer, teori tentang antar muka yang dipakai yaitu mikrokontroler AT89S52, sensor suhu LM 35D, konversi sinyal ADC0804, IC suara ISD 25120, LCD (*Liquid Crystal Display*) sebagai display, dan bahasa pemograman *Assembler*.

2.1. Teori Dasar Termometer

Termometer adalah suatu alat yang digunakan untuk mengukur suhu (temperatur), dan perubahan suhu. Istilah termometer berasal dari bahasa Latin *thermo* yang berarti suhu dan *meter* yang berarti mengukur.

Suhu adalah sifat yang menentukan apakah sistem setimbang termal dengan sistem lain atau tidak, bila dua sistem atau lebih dalam setimbang termal maka sistem ini dikatakan mempunyai suhu yang sama. Suhu menunjukkan derajat panas suatu benda. Mudahnya, semakin tinggi suhu suatu benda, semakin panas benda tersebut, sebaliknya semakin rendah suhu suatu benda semakin dingin benda tersebut.

Secara mikroskopis, suhu menunjukkan energi yang dimiliki oleh suatu benda. Setiap atom dalam suatu benda masing-masing bergerak, baik itu dalam bentuk perpindahan maupun gerakan di tempat berupa getaran.

Termometer memang bukan barang antik yang jarang diketahui orang. Walau demikian, sayangnya benda yang sangat penting dan murah ini jarang dijumpai di rumah-rumah. Padahal, demam merupakan salah satu gejala penyakit yang paling sering dijumpai di dunia.

Termometer ditemukan beberapa abad lalu, tepatnya di abad XVI oleh ahli Fisika sekaligus astronom Italia bernama *Galileo*. Tetapi, termometer air raksa yang akurat baru berhasil dibuat pada dua abad sesudahnya oleh *Daniel Gabriel Fahrenheit*. Karena itu, sampai saat ini skala Fahrenheit masih umum dipakai di Amerika. Penemuan termometer berlanjut dengan dikembangkannya termometer yang memakai skala 100 oleh astronom Swedia bernama *Anders Celcius*. Nama belakangnya tentu tidak asing terdengar bagi orang Indonesia, karena umumnya orang Indonesia memakai termometer dengan skala Celcius.

Termometer *konvensional* terdiri atas tabung gelas tertutup yang berisi cairan. Cairan yang umum dipakai dalam termometer kita adalah air raksa (merkuri). Di tepi tabung terlihat garis-garis yang menunjukkan skala temperatur. Bila suhu meningkat, air raksa dalam tabung yang sempit itu akan naik. Titik dimana air raksa tersebut berhenti naik menunjukkan berapa suhu tubuh yang tertera pada skala temperatur.

Yang harus diingat juga adalah selalu mengibas-kibaskan termometer sebelum dipakai. Hal ini disebabkan tabung termometer yang sempit itu akan mencegah air raksa yang sudah terlanjur naik untuk turun dengan sendirinya. Satu-satunya cara untuk menurunkan air raksa tersebut adalah dengan menggerakan termometer dengan tangan.

Penjalanan termometer tidak berhenti sampai disana. Alat ini terus dikembangkan untuk memberikan kemudahan dan ketepatan yang lebih baik bagi penggunanya. Beragam cara penggunaan termometer yang kini ada misalnya:

2.1.1 Cara Penggunaan Termometer di Ketiak.

Ini merupakan cara penggunaan termometer yang tampaknya paling sering dilakukan. Caranya terlihat sangat sederhana sehingga hal ini mungkin yang menjadikannya pilihan pertama dan utama di kalangan masyarakat. Pengukuran suhu ketiak sesungguhnya tidak seakurat pengukuran mulut atau rektal. Temperatur yang terukur akan menghasilkan nilai $10^{o}C$ lebih rendah dibandingkan dengan hasil pengukuran mulut.

2.1.2 Cara Penggunaan Termometer di Mulut.

Prinsip utama yang harus diingat adalah jangan menggunakan cara ini pada bayi dan anak yang masih kecil, terlebih bila menggunakan termometer air raksa. Pengukuran suhu melalui mulut lebih akurat bila dibandingkan dengan pengukuran melalui ketiak. Tetapi untuk mendapatkan hasil yang akurat, biarkan termometer di dalam mulut selama 3-4 menit sebelum di baca. Selain itu, jangan lupa mengibas-kibaskan termometer sebelum digunakan.

Saat meletakkan termometer ke dalam mulut, pastikan ujung termometer ditempatkan di bawah lidah sejauh mungkin. Sekali lagi, di bawah lidah. Hal ini penting mengingat kebanyakan orang melakukan dengan salah dan sekedar memasukkan termometer ke dalam mulut, dikulum di atas lidah dan cuma selama 1-2 menit saja. Cara pengukuran yang salah tentu menghasilkan informasi yang tidak akurat.

2.1.3 Cara Penggunaan Termometer di Rektal.

Cara ini merupakan cara yang sangat baik hanya saja mungkin kurang menyenangkan. Bayi sebaiknya diukur dengan cara ini mengingat keakuratannya yang lebih tinggi. Suhu yang tercatat umumnya lebih tinggi 10 dibanding pengukuran suhu melalui mulut.

Bila memakai termometer air raksa, pastikan menggunakanya dengan hati-hati. Gerakan mendadak dari bayi dapat membuat termometer pecah dan menimbulkan bahaya. Termometer dimasukkan melalui lubang pantat selama paling sedikit 3 menit. Jangan lupa untuk mengibas-kibaskan dulu termometer tersebut sebelum pemakaian agar air raksa yang sudah naik dapat turun kembali.

2.1.4 Cara Penggunaan Termometer di Telinga.

Ini merupakan cara yang cukup canggih di mana yang diukur adalah temperatur gendang telinga. Cara kerjanya dalam menentukan suhu tubuh adalah membaca radiasi infrared yang berasal dari jaringan gendang telinga. Pengukuran dengan cara ini memang memiliki beberapa kelebihan seperti kemampuan untuk mengukur secara lebih tepat temperatur dalam otak. Untuk diketahui, temperatur otak sebenarnya adalah ukuran yang paling tepat dalam hal pengukuran temperatur tubuh. Kelebihan lain adalah penggunaan waktu yang sangat singkat, sekitar 2 sampai 3 detik saja.

Hanya saja, keakuratan termometer model ini memang masih dipertanyakan oleh sebagian kalangan dokter. Selain itu, kelemahan lainnya terletak pada harganya yang masih cukup menguras kantong kita.

2.1.5 Cara Penggunaan Termometer Basal.

Temperatur basal adalah temperatur tubuh saat bangun pagi. Temperatur ini biasanya juga merupakan temperatur tubuh yang terendah. Termometer basal merupakan temometer yang sangat sensitif yang digunakan untuk menilai perubahan temperatur yang sangat sedikit. Cara penggunaannya dapat melalui mulut.

Karena suhu yang ingin diukur adalah suhu basal, tentu saja pengukuran suhu dilakukan segera setelah bangun tidur. Jadi, jangan letakkan termometer ini di sembarang tempat. Letakanlah pada lokasi yang sangat mudah dicapai dari tempat tidur. Bila sudah terlanjur bangun dan berjalan, suhu tubuh tentu telah berubah dan pengukuran menjadi tidak akurat.

2.1.6 Cara Penggunaan Termometer Digital.

Ini adalah bagian dari kemajuan teknologi dimana termometer air raksa mulai digantikan oleh cara lain yang relatif lebih aman. Termometer digital biasanya

dilengkapi dengan bunyi (misalnya bip) yang akan memberitahukan bahwa pengukuran suhu telah selesai dilakukan. Cara pengukuran umumnya sama dengan cara pengukuran dengan memakai termometer air raksa, hanya tidak perlu melihat jam untuk mengetahui kapan pengukuran suhu selesai.

Keragaman termometer ini tentu tidak akan berguna bila hanya untuk dibaca tapi tidak dimiliki dan dimanfaatkan di rumah. Termometer air raksa sangat akurat dan murah harganya. Bila jatuh akan langsung pecah sehingga tidak terlalu dianjurkan penggunaannya pada bayi dan balita. Termometer digital lebih mahal harganya tetapi relatif lebih aman penggunaannya.

2.2 Mikrokontroler AT89S52

Mikrokontroler AT89S52 merupakan pengembangan dari mikrokontroler standar MCS-51. Hanya mikrokontroler AT89S52 memiliki kelebihan dan mempunyai fitur tambahan yang tidak terdapat dalam MCS-51.

Mikrokontroler Atmel AT89S52 memiliki banyak fitur yang menguntungkan. Dipakainya *Downloadable flash memory* memungkinkan mikrokontroler ini bekerja sendiri tanpa diperlukan tambahan chip lain. Sementara *Flash memory* mampu diprogram hingga seribu kali. Hal lain yang menguntungkan adalah sistem pemograman menjadi lebih sederhana dan tidak memerlukan rangkaian yang rumit.

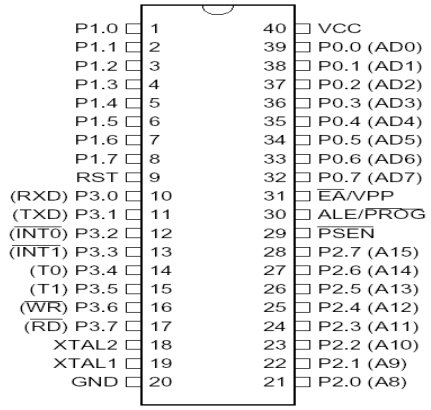
Timer atau counter juga bertambah satu dari standar 2 buah pada MCS-51. selain itu frekuensi kerja yang lebar dan rancangan statis sangat membantu untuk proses *debuggin*g. Dengan adanya fitur tambahan itu, maka akan bertambahnya SFR (*Special Function Register*). SFR tambahan ini adalah SFR untuk mengontrol alat tambahan pada mikrokontroler Atmel 89S52.

SFR ini meliputi : T2CON (Timer 2 Register dengan alamat 0C8h), T2MOD (Timer 2 mode dengan alamat 0C9h), WMCON (*Wactdog and memory control Register* dengan alamat 96h), SPCR (*SPI Control Register* dengan alamat D5h), SPSR (SPI Status Register dengan alamat AAh), SPDR (SPI data Register dengan alamat 86h).

2.2.1 Deskripsi Pin

AT89S52 mempunyai 40 kaki, 32 kaki diantaranya adalah kaki untuk keperluan *Port* paralel. Satu *Port* paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah *Port* paralel, yang masing-masing dikenal sebagai *Port*-0, *Port*-1,

Port-2 dan *Port*-3. Nomor dari masing-masing kaki dari *Port* paralel mulai dari 0 sampai 7. Jalur atau kaki pertama *Port*-0 disebut sebagai P0.0 dan jalur terakhir untuk *Port*-0 adalah P0.7. Letak dari masing-masing *Port* diperlihatkan pada gambar 2.1.



Gambar 2.1 Konfigurasi pin ATMEL AT89S52

Berikut ini nama dan fungsi dari kaki-kaki pin pada mikrokontroler AT89S52 adalah sebagai berikut:

1. **VCC** (pin 40) : *Power supply*
2. **GND** (pin 20) : *Ground*
3. ***Port* 0** (pin 32 – 39)

   Port 0 dapat berfungsi sebagai I/O biasa, *low order multiplex address/data* ataupun menerima kode byte pada saat *Flash Programming*. Pada fungsi sebagai I/O biasa *Port* ini dapat memberikan *output sink* ke delapan buah TTL Input atau dapat diubah sebagai input dengan memberikan logika 1 pada *Port* tersebut. Pada fungsi sebagai *low order multiplex address/data Port* ini akan mempunyai *internal pull up*. Pada saat *Flash Programming* diperlukan *external pull up* dengan menggunakan Resistor Pack.
4. ***Port* 1** (pin 1 – 8)

   Port 1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* pada saat *Flash Programming*. *Port* ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output *Port* ini dapat memberikan *output sink* keempat buah input TTL.
5. ***Port* 2** (pin 21 – 28)

   Port 2 berfungsi sebagai I/O biasa atau *high order address,* pada saat mengakses memori secara 16 bit (Movx @Dptr). Pada saat mengakses memori secara 8 bit,

(Mov @Rn) *Port* ini akan mengeluarkan isi dari *P2 Special Function Register*. *Port* ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output, *Port* ini dapat memberikan *output sink* keempat buah input TTL.

6. *Port* 3 (pin 10 – 17)

Sebagai I/O biasa *Port* 3 mempunyai sifat yang sama dengan *Port* 1 maupun *Port* 2. *Port* 3 menyediakan beberapa fungsi khusus sebagaimana diperlihatkan pada tabel dibawah.

Tabel 2.1 Fungsi-fungsi alternatif pada Port 3

Kaki *Port*	**Fungsi Alternatif**
P3.0	RXD (*Port* input serial)
P3.1	TXD (*Port* output serial)
P3.2	INT0 (interupsi eksternal 0)
P3.3	INT1 (interupsi eksternal 1)
P3.4	T0 (input eksternal timer 0)
P3.5	T1 (input eksternal timer 1)
P3.6	WR (sinyal *write* pada data memori eksternal)
P3.7	RD (sinyal *read* pada data memori eksternal)

Fungsi-fungsi alternatif pada tabel diatas hanya dapat diaktifkan jika bit-bit pengancing (*latch*) *Port* yang bersangkutan berisi '1'.

7. RST (pin 9)

Masukan reset kondisi 1 selama 2 siklus mesin selama osilator bekerja akan me-*reset* mikrokontroler yang bersangkutan ke alamat awal.

8. ALE/PROG (pin 30)

Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang me-*latch low byte address* pada saat mengakses memori eksternal. Sedangkan pada saat *Flash Programming* (PROG) berfungsi sebagai pulse input. Pada operasi normal ALE akan mengeluarkan sinyal *clock* sebesar 1/16 frekuensi oscillator kecuali pada saat mengakses memori eksternal. Sinyal *clock* pada pin ini dapat pula di-*disable* dengan men-set bit 0 dari *special function register* di alamat 8EH. ALE hanya akan aktif pada saat mengakses memori eksternal (MOVX & MOVC).

9. PSEN (pin 29)

Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori eksternal. PSEN akan aktif dua kali setiap *cycle*.

10. EA/VPP (pin 31)

Pada kondisi *low*, pin ini akan berfungsi sebagai *External Acces Enable* (EA), yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal setelah sistem di-*reset*. Jika berkondisi *high*, pin ini akan berfungsi untuk menjalankan program yang ada di memori internal.

11. XTAL1 (pin 19) : *input oscillator*

12. XTAL2 (pin 18) : *output oscillator.*

2.2.2 Struktur Memori

1. Flash PEROM

Memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51 dialamati oleh *Program Address Register* (Register Alamat Program). AT89S52 memiliki 8 Kb *Flash* PEROM yang menggunakan *Atmel's High-Density Non Volatile Technology*.

Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem di-*reset*, pin EA/VP berlogika satu maka mikrokontroler aktif berdasarkan program yang ada pada *Flash* PEROM-nya. jika pin EA/VP berlogika nol, mikrokontroler aktif berdasarkan program yang ada pada memori eksternal.

2. EEPROM

Mikrokontroler Atmel AT89S52 dilengkapi dengan data memori yang berupa EEPROM (*electrically Eraseble Programmable Read Only Memory*). EEPROM ini besarnya 2K byte dan dipakai untuk menyimpan data

3. RAM

RAM yang ada dalam AT89S52 berkapasitas 256 byte. RAM (*Random Access Memory*) merupakan memori yang bersifat mudah dihapus. Karena itu RAM tidak digunakan sebagai penyimpanan program tetapi untuk menyimpan data sementara.

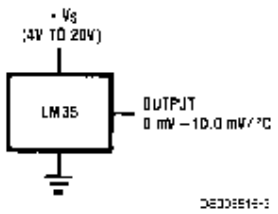
2.3 Sensor Suhu LM 35D

LM 35D adalah jenis sensor suhu dengan rangkaian ketepatan yang terintegrasi dan tegangan outputnya secara linier proporsional terhadap ^{0}C. Keuntunganya melebihi

sensor suhu yang dikalibrasi oleh satuan *Kelvin* serta output impedansi dengan catu daya Vin sebesar 5V.

Keuntungan menggunakan sensor suhu LM 35D :

- Dikalibrasi secara langsung pada satuan derajat celcius.
- Skala factor linier ± 10,0 mV/^{0}C.
- Bekerja antara suhu 0^{0}C sampai dengan +100^{0}C.
- Arus kurang dari 2 mA.
- Pemanasan rendah 0,08^{0}C.

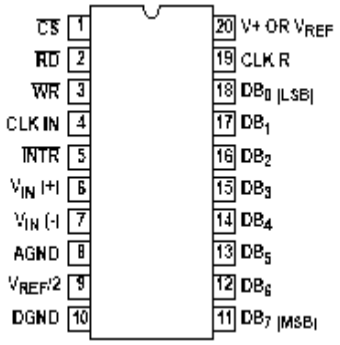


Gambar 2.2 LM 35D

2.4 Analog to Digital Corventer ADC 0804

Salah satu komponen penting dalam sistem akuisisi data adalah pengubah besaran *analog* ke *digital* atau disebut juga ADC (*Analog to Digital Converter*). Pengubah ini akan mengubah besaran-besaran analog menjadi bilangan-bilangan *digital* sehingga dapat diproses oleh mikrokontroler maupun oleh PC. Peranan pengubah ini menjadi semakin penting sebagai dampak dari perkembangan sistem perangkat keras digital.

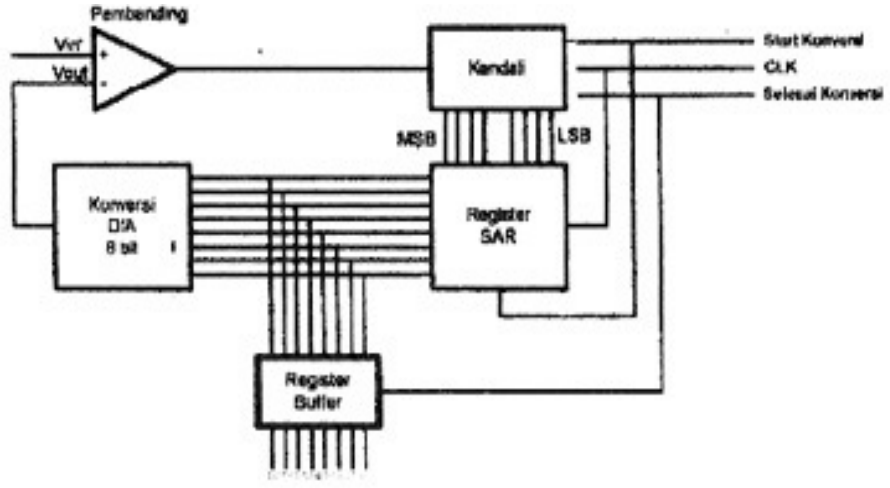
Contoh aplikasi ADC ini bisa dilihat misalnya pada *voltmeter digital, sampling* suara dengan komputer sehingga suara dapat disimpan secara *digital* dalam disket dan lain sebagainya. Konsep pengubah *analog* ke *digital* ini adalah sampling (mengambil contoh dalam waktu tertentu) kemudian mewakilinya dengan bilangan *digital* dengan batas yang sudah diberikan.



Gambar 2.3 konfigurasi pin ADC0804

Berikut ini Deskripsi pin ADC 0804

1. Pin 1 (CS-Chip Select), kanal pemilihan ADC0804.
2. Pin 2 (RD-Read), merupakan penanda bahwa pembacaan data dari ADC sudah siap untuk dilakukan.
3. Pin 3(WR-Write), diberikan untuk menandakan awal konversi. Pin ini berfungsi sebagai penanda input analog sudah siap untuk dikonversi menjadi data digital atau belum. Jika pin ini diberi nilai 1 maka konversi sudah bisa dimulai, sebaliknya jika diberi nilai 0 maka menunggu sampai diberi nilai 1.
4. Pin 4(CLK IN), berfungsi sebagai jalur pengaturan ADC oleh *schmitt trigger.*
5. Pin 5 (INTR), merupakan pin interrupt internal dari ADC0804.
6. Pin 6 (Vin+), berfungsi sebagai kanal input positif dari data analog yang akan dikonversi.
7. Pin 7 (Vin-), berfungsi sebagai kanal input negatif dari data analog yang akan dikonversi.
8. Pin 8 (A GND), pin ini merupakan ground bagi data analog.
9. Pin 9 (Vref/2), merupakan tegangan referensi dari ADC.
10. Pin 10 (D GND), pin ini merupakan ground bagi data digital.
11. Pin 11-18 (DB7-DB0), berfungsi sebagai digital output 8 bit.
12. Pin 19 (CLKR), berfungsi untuk mereset ADC.
13. Pin 20 (VCC), merupakan input catu daya untuk ADC0804 dan diberi tegangan sebesar 5 volt.



Gambar 2.4 Diagram Blok ADC0804

Secara singkat prinsip kerja dari konverter A/D adalah semua bit-bit diset konversi akan diselesaikan sesudah 8 clock, dan keluaran D/A merupakan nilai analog yang ekivalen dengan nilai register SAR. Apabila konversi telah dilaksanakan, rangkaian kembali mengirim sinyal selesai konversi yang berlogika rendah. Sisi turun sinyal ini akan menghasilkan data digital yang ekivalen ke dalam register buffer. Dengan demikian, keluaran digital akan tetap tersimpan sekalipun akan di mulai siklus konversi yang baru.

2.5 IC Suara ISD 25120

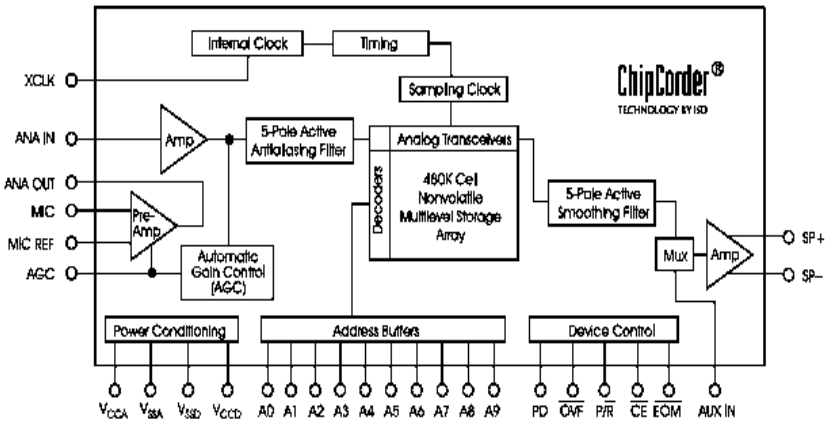
ISD 25120 adalah single chip dengan kualitas tinggi dengan durasi rekam atau putar ulang (*Record or Playback*) antara 60 sampai 120 detik. Merupakan komponen CMOS yang terdiri atas *on-chip oscilator*, *microphone amplifier*, *automatic gain control, antialiasing filter, smoothing filter, speaker preamplifier, dan high density multi-level storage array*.

2.5.1 Ciri – ciri ISD 25120

ISD 25120 mempunyai ciri – ciri sebagai berikut :

1. Single-chip mudah digunakan untuk merekam suara atau memtarnya ulang.
2. Kualitas suara atau audio yang dihasilkan tinggi dan tampak alami.
3. Single-chip dengan durasi 120 detik.
4. Dapat digunakan dengan atau tanpa mikrokontroler.
5. Secara langsung merekam dengan durasi yang sangat panjang.
6. Power down otomatis (mode *push button*).

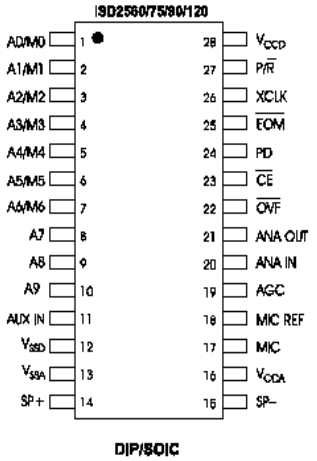
7. Dapat dilamatkan secara langsung untuk mengatasi pesan panjang.
8. Penyimpanan pesan selam 100 tahun.
9. Siklus perekaman 100.000 kali.
10. Sumber clock *0n-chip*.
11. Dapat diprogram untuk aplikasi putar ulang semata.
12. Catu daya +5 volt.



Gambar 2.5 Diagram blok ISD 25120

2.5.2 Deskripsi Pin ISD 25120

ISD 25120 mempunyai 28 kaki, 10 kaki diantaranya adalah kaki untuk keperluan alamat penyimpanan pesan, pin CE untuk meng-*enable* semua operasi perekaman atau mendengarkan pesan, pin P/R digunakan untuk proses perkaman dan putar ulang pesan. Untuk lebih jelasnya konfigurasi pin ISD 25120 dapat dilihat pada gambar 2.6.



Gambar 2.6 Konfigurasi pin ISD 25120

Nama dan fungsi dari kaki-kaki pin pada ISD 25120 adalah sebagai berikut

1. **AX** (pin 1-10)
   Address / mode input : *Address mode input* memiliki dua fungsi bergantung pada level dari dua buah MSB dari pin – pin alamat (A8 dan A9). Jika salah satu atau kedua MSB dalam kondisi *low* semua masukan diinterprestasikan sebagai bit alamat dan digunakan sebagai alamat awal untuk siklus perekaman atau putar ulang. Jika kedua MSB dalam kondisi *high, address mode input* diinterprestasikan sebagai mode operasional (*operational mod*e). Ada enam mode operasional dan memungkinkan menggunakan banyak mode operasi secara simultan.
2. **AUX IN** (Vin 11)
   Auxiliary Input : AUX IN digunakan untuk menghubungkan sinyal *playback* dengan speaker.
3. **Vssa, Vssd** (pin 12, 13)
   Ground
4. **SP+ / SP-** (pin 14, 15)
   Speaker output : bagian yang mengeluarkan suara atau audio. Semua komponen termasuk *on-chip differential speaker driver*, terbatas pada 15 mW sampai 16 ohm
5. **Vcca, Vccd** (pin 16, 18)
   Power Suply
6. **MIC** (pin 17)
   Microphone : mikrophone mentranfer sinyal menuju on-chip *Preamplifier*, Rangkaian *Automatic Gain Control* (*AGC*) mengambil gain preamplifier mulai dari -15 sampai 24 dB
7. **MIC REF** (pin 18)
   Microphone Reference : masukan MIC REF merupakan masukan inverting untuk *microphone preamplifier*
8. **AGC** (pin 19)
   Automatic Gain Control : dengan adanya AGC dapat meminimalisi *distorsi* suara yang direkam.
9. **ANA IN** (pin 20)
   Analog Input : *analog input* mentransfer sinyal masukan *mikrophone*. Pin ANA IN harus digabungkan dengan pin ANA OUT melalui kapasitor eksternal.
10. **ANA OUT** (pin 21)
    Analog output : *Preamplifier output.*

11. OUF (pin 22)

OverFlow : untuk mengindikasikan bahwa piranti rekam terisi penuh dengan pesan.

12. CE (pin 23)

Chip Enable : pin CE diberi kondisi *low* untuk meng-*enable* semua operasi perekaman atau putar ulang.

13. PD (pin 24)

Power Down : ketika sedang merekam atau memutar ulang hasil rekaman pin PD harus kondisi *high* untuk menempatkan perangkat dalam *mode standby*.

14. EOM (pin 25)

End Of Message : untuk mengindikasikan bahwa perangkat sedang beroperasi.

15. XCLK (pin 26)

External Clock

16. P / R (pin 27)

Playback / Record : pin ini digunakan untuk proses rekam atau putar ulang.

2.5.3 Mode Operasional

ISD 25120 dirancang dengan beberapa built-in mode operasional yang mentediakan kemampuan maksimum denagan komponen eksternal minimum. Mode operasional diakses melalui pin – pin alamat dan dipetakan diluar cakupan alamat pesan normal. Ketika kedia MSB, A8 dan A9 *high*, isyarat alamat ditafsirkan sebagai mode bit dan bukan sebagai bit alamat.

Ada dua hal yang perlu diperhatikan ketika menggunakan mode operasional. Pertama, semua operasi dimulai dengan alamat nol operasi selanjutnya dapat dimulai dari alamt lain tergantung mode opersi yang dipilih. Kedua, mode operasi dieksekusi ketika CE kondisi *low*.

2.5.4 Deskripsi Mode Operasional

Mode operasi dapat digunakan bersama dengan suatu mikrokontroler atau dapat di hardwired untuk menyediakan pengoperasian yang diinginkan. Ada beberapa macam mede opersi antara lain

1. **M0 – Pesan Isyarat (*Message Cueing*).**

Pesan isyarat mengijinkan pemakai untuk melompat beberapa pesan, tanpa mengetahua fisik yang nyata mengenai alamat dari tiap pesan. Setiap CE kondisi *low*

menyebabkan alamat internal ditujukan pada pesan berikutnya. Mode ini hanya digunakan untuk *Playback* saja. Dan sering digunakan bersama M4 mode operasi.

2. **M1 – Hapus penanda EOM (*Delete EOM markers*).**

Mode operasi ini mengijikan secara sekuen perekam pesan untuk dikombinasikan kedalam satu pesan, dengan satu penanda EOM ditetapkan pada pesan terakhir.

3. **M2 – Tidak dipakai (*Unused*).**

4. **M3 - Pengulangan Pesan (*message looping*).**

Mode ini menghasilkan playback secara otomatis. Dilanjutkan enagan playback pesan yang dialokasikan pada permulaan alamat. Suatu pesan dapat sepenuhnya mengisi ISD 25120 dan akan berulang daria awal sampai akhir tanpa *overflow low*.

5. **M4 – Alamat berurutan (*Consecutive addressing*).**

Selama operasi normal, penunjuk alamat akan mereset ketika suatu pesan dimasukan melalui suatu penanda EOM. Mode operasi ini menghalangi penunjuk alamat merest lagi pada EOM. Menyebabkan pesan untuk diputar kemebali dengan teratur.

6. **M5 – Tingkat aktikan CE (*CE level activated*).**

Mode ini menyebabkan pin CE ditafsierkan sebagai level activated selama proses playback. Ini bermanfaat untuk mengakhiri proses playback yang menggunakan sinyal CE.

7. **M6 – Mode tombol (*Push-button mode*).**

ISD 25120 berisi mode mode operasional tombol. Tombol tekan digunakan adalam mode operasi ini karena murah dan dirancang untuk memperekecil rangkaian dan komponen eksternal. Dalam mengatur mode operasional ini kedua MSB harus kondisi *high* dan pin mode M6 harus *high* pula.

Dalam menggunakan mode operasional ini ada tiga pin yang mempunyai fungsi khusus seperti dijelaskan dalam tabel dibawah ini.

Tabel 2.2 Fungsi alternatif mode operasional

Nama pin	Fungsi alternatif mode push-button
CE	Start / pause tombol tekan (LOW pulse – activated)
PD	Start / reset tombol tekan (HIGH pulse – activared)
EOM	Active HIGH run indikator

CE (Start / pause)

Dalam mode operasional push-button, CE berperan sebagai pulsa aktif *low* untuk sinyal start / pause.

PD (stop / reset).

PD berperan sebagai pulsa aktif *high* untuk sinyal stop / reset.

EOM (Run).

Dalam mode operasi push-button, EOM menjadi sinyal aktif *high* yang bisa digunakan untuk menghidupkan Led atau perangkat eksternal lainnya.

2.5.5 Merekam dengan Mode *Push-button*.

Langkah – langkah merekam menggunakan mode ini adalah sebagai berikut :

1) Pin PD harus kondisi *low*, biasanya digunakan pull-down resistor.
2) Pin P/R kondisi *low*.
3) Pin CE kondisi *low* proses perekaman dimulai. EOM kondisi *high* , mengindikasikan operasi sedang berlangsung.
4) Ketika CE diberi pulsa *low* perkaman berhenti. EOM kembali ke kondis *low*, penunjuk alamat internal dibersihkan tetapi disimpan dalam memori sebagai indikasi akhir pesan.
5) Pin CE diberi kondisi *low* perekaman dimulai kembali dengan alamat selanjutnya, EOM kembali posisi *high*.

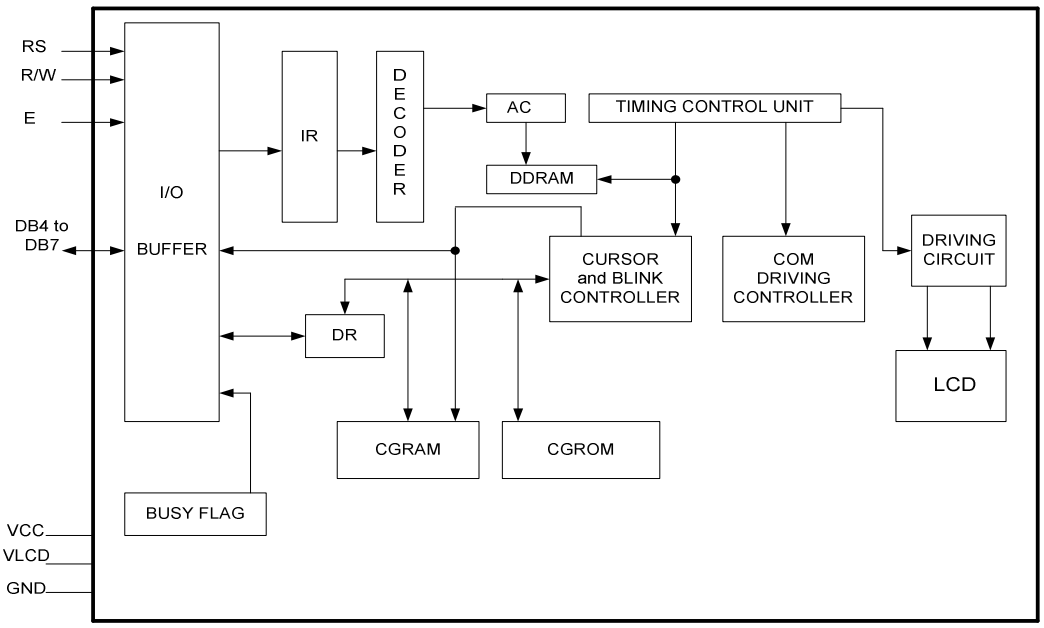
2.5.6 Playback Dengan Menggunakan Mode *Push-button*.

1) Pin PD dikondisikan *low*.
2) Pin P/R dieri pulsa *high*.
3) Pin CE diberi pulsa *low* playback dimulai, EOM kondidsi *high* mengindikasikan proses sedang berlangsung.
4) Jika pin CE diberi pulsa *low*, Playback berhenti dan EOM kondisi *high*.

5) CE kembali diberi pulsa *high* playback dimaulai dari pemberhentian sebelumnya dengan kondisi EOM kembali *high*.
6) Playback berlangsung 4 – 5 kali sampai PD berkondisi *high* atau *Overflow*.
7) Dalam kondisi *overflow*, Ce kondisi *low* akan mereset penunjuk alamat dan memulai proses playback dari alamat nol.

2.6 LCD (*Liquid Crystal Display*)

Liquid Crystal Display (LCD) merupakan perangkat yang digunakan untuk menampilkan data agar informasi dapat dibaca dan dimengerti. LCD sangat sesuai untuk peraga yang lebih kompleks, seperti memperagakan angka, huruf dan symbol. LCD sangat ideal untuk dikombinasikan dengan keluarga mikrokontroler AT89C51. Sebelumnya, kesulitan utama dalam menggunakan LCD adalah diperlukannya sebuah rangkaian pengatur *scanning* dan pembangkit tegangan sinus.



Gambar 2.7 Diagram Blok LCD M1602 – 04

2.6.1 Deskripsi Fungsi Pin LCD M1602 – 04

Deskripsi fungsi pin-pin LCD M1602 - 04, sebagai berikut :

1. Pin 1

   GND atau pentanahan.

2. Pin 2

   Merupakan pin Vcc untuk sumber tegangan sebesar +5 Volt.

3. Pin 3

   VLCD untuk mengatur kontras pada layar LCD.

4. Pin 4

   RS berfungsi untuk memilih register yang akan diakes. Diberi logika 1 untuk memilih data register dan diberi logika 0 untuk memilih control register.

5. Pin 5

   R/W berfungsi untuk control bagi transfer data dari dan ke LCD, pin ini diberi logika 1 untuk membaca data dari LCD ke bus data CPU, dan logika 0 untuk menulis data dari bus CPU ke LCD.

6. Pin 6

   Enable LCD berfungsi untuk memberikan perintah dari CPU kepada LCD untuk memproses data yang masuk atau keluar, perintah ini berupa pulsa (Logika L – H – L).

7. Pin 7 – Pin 14

   DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7 selebar 8-bit adalah saluran untuk melakukan transfer data dan memberikan instruksi kepada LCD

8. Pin 15

   Sebagai kaki Anoda dari backlight.

9. Pin 16

   Sebagai kaki Katoda dari backlight.

2.6.2 Modul LCD M1602 – 04

Sebelumnya, kesulitan utama dalam menggunakan LCD adalah diperlukannya sebuah rangkaian pengatur *scanning*. Pada perancangan ini digunakan Modul LCD M1602 - 04, dimana di dalam modul ini sudah tersedia pengendali layar LCD (HD44780) yang dilengkapi memori, pola–pola karakter, dan antarmuka ke mikrokontroler. Modul LCD ini mempunyai konfigurasi 16 karakter dan 2 baris di setiap karakternya dibentuk oleh 8 baris pixel (1 baris pixel terakhir adalah kursor) dan 5 kolom pixel. LCD ini memiliki 8-bit untuk port data (pada perancangan ini hanya dipakai 4-bit untuk port data), 3-bit kontrol (RS, R/W, dan E), 3 pin catu daya (VCC, VLCD, dan GND) dan 2 pin untuk backlight (Anoda dan Katoda).

2.6.3 Inisialisasi LCD

Berdasarkan teknik antarmuka dengan mikrokontroler, terdapat dua jenis inisialisasi LCD. LCD harus melakukan proses inisialisasi sebelum siap untuk menerima berbagai macam data input.

Pada perancangan Tugas Akhir, LCD menggunakan inisialisasi 8 bit. Adapun dua jenis inisialisasi pada LCD, yaitu :

1. Inisialisasi 4 bit

   Proses standar (yang selalu terjadi) saat inisialisasi LCD dilakukan sebelum masuk ke pengaturan fungsi dan mode. Sistem menunggu selama 15 milisekon atau lebih setelah sumber daya mencapai tegangan 4,5 volt agar HD44780 siap berhubungan dengan mikrokontroler. Pengiriman dan pembacaan data dikirimkan sebanyak 2 kali dengan waktu tunda 4,1 milisekon atau lebih. Pengaturan fungsi digunakan untuk mengatur panjang data, jumlah baris, dan *font*, sedangkan pengaturan mode digunakan untuk mengatur arah pergeseran kursor dan *display*.

2. Inisialisasi 8 bit

   Proses inisialisasi LCD jenis ini hampir sama dengan proses inisialisasi 4 bit. Perbedaan terletak pada penulisan atau pembacaan data. Jika pada inisialisasi 4 bit penulisan atau pembacaan data dilakukan sebanyak dua kali untuk 8 bit data, maka pada inisialisasi 8 bit cukup dilakukan satu sekali.

2.6.4 Struktur Memori LCD

LCD M1602 - 04 memiliki beberapa jenis memori yang digunakan untuk menyimpan atau memproses data-data yang akan ditampilkan pada layar LCD. Adapun memori-memori tersebut adalah :

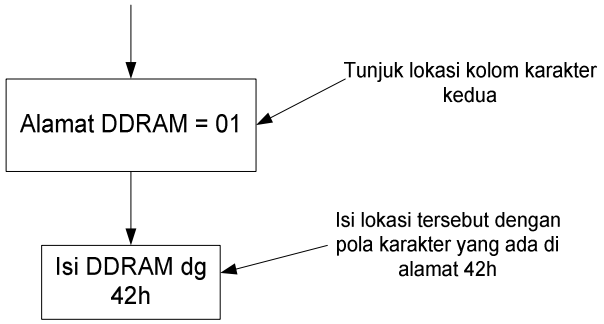
a. DDRAM

   Merupakan memori tempat karakter yang ditampilkan berada. Data pada memori ini berfungsi untuk menunjuk lokasi pola karakter yang akan ditampilkan pada layar LCD.

b. CGRAM

   Merupakan memori untuk menggambarkan pola sebuah karakter. RAM bersifat tidak permanen, isi memori ini akan hilang jika sumber daya dimatikan sehingga pola karakter juga akan hilang.

Posisi tampilan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alamat	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
DDRAM	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF



Gambar 2.8 Diagram Alir Penulisan Karakter

c. CGROM

Merupakan memori untuk menggambarkan pola sebuah karakter dan pola tersebut sudah ditentukan secara permanen. ROM bersifat permanen, sehingga pola karakter tidak akan hilang jika sumber daya dimatikan.

2.7 Bahasa *Assembly*

Assembler adalah program komputer yang men-*translitrasi* program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi programer untuk memprogram mesin komputer daripada menggunakan serangkaian 0 dan 1 (bahasa mesin) yang panjang dan rumit.

2.7.1 Konstruksi Program *Assembly*

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan *extension* .ASM dengan 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yakni bagian label, bagian *mnemonic*, bagian *operand*. Program sumber (*source code*) dibuat dengan program editor seperti *notepad* atau Editor DOS, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *assembler*. Hasil kerja program *assembler*

adalah "*program objek*" dan juga "*assembly listing*". Ketentuan penulisan *source code* adalah sebagai berikut:

1. Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk *operand* yang lebih dari satu masing-masing *operand* dipisahkan dengan koma.
2. Bagian-bagian tersebut tidak harus semuanya ada dalam sebuah baris, jika ada satu bagian yang tidak ada maka spasi atau TAB sebagai pemisah bagian tetap harus ditulis.
3. Bagian label ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung label maka label tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian label dan bagian *mnemonic*.

2.7.2 Instruksi MCS-51 yang Digunakan

Beberapa instruksi yang digunakan dalam penyusunan program sensor kendaraan adalah sebagai berikut :

1. EQU

   Konstanta Data yang dideklarasikan dengan nilai pada variabel tertentu.
2. ORG

   Digunakan untuk menunjukkan lokasi memori tempat instruksi atau perintah yang ada di bawahnya disimpan.
3. CLR

   Memberikan nilai "nol" pada bit tertentu.
4. SETB

   Memberikan nilai "satu" pada bit tertentu.
5. CALL

   Instruksi melakukan lompatan dengan area sebesar 2 KByte.
6. LCALL

   Sama seperti instruksi CALL, hanya instruksi ini digunakan jika label yang dipanggil letaknya lebih dauh dari 2 Kbyte
7. MOV

   Instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua dan disimpan di variabel pada kode operasi pertama.

8. MOVC

   Instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua ditambah carry dan disimpan di variabel pada kode operasi pertama.

9. JMP

   Melakukan lompatan dan menjalankan program yang berada di alamat yang ditentukan oleh label tertentu.

10. SJMP

    Melakukan lompatan untuk jarak yang pendek (*Short JMP*).

11. LJMP

    Melakukan lompatan untuk jarak yang jauh (*Long JMP*).

12. JNB

    Kebalikan dari JB, JNB merupakan instruksi untuk beralih ke alamat tertentu jika Bit tertentu bernilai “nol”.

13. JNC

    Fungsinya identik dengan JNB.

14. JC

    Melakukan suatu loncatan menuju “Tujuan” bila CF=1. “Tujuan” dapat berupa nama label atau alamat memori. Pada lompatan bersyarat ini, besaarnya lpmpatan tidak dapat melebihi -128 dan +127 byte.

15. DJNZ

    Melakukan pengurangan pada Rn (R0…R7) dengan 1 dan lompat ke alamat yang ditentukan apabila hasilnya bukan 00. Apabila hasilnya telah mencapai 00, maka program akan terus menjalankan instruksi dibawahnya.

16. CJNE

    Instruksi ini melakukan perbandingan antara data tujuan dan data sumber serta melakukan lompatan ke alamat yang ditentukan jika hasil perbandingan tidak sama.

17. RET

    Instruksi ini melakukan lompatan ke alamat yang disimpan dalam SP dan SP-1. Instruksi ini biasa digunakan pada saat kembali dari subrutin yang dipanggil dengan instruksi ACALL atau LCALL.

18. ANL

    Logika AND untuk variabel tertentu. Perintah ANL bekerja dengan melakukan operasi AND antara variabel yang ditentukan dengan nilai

tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

19. ORL

    Logika OR untuk variabel tertentu. Perintah ORL bekerja dengan melakukan operasi OR antara variabel yang ditentukan dengan nilai tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

20. LOOP

    Untuk melakukan suatu proses yang berulang (Looping) dengan CX sebagai counternya. Perintah LOOP akan mengurangkan CX dengan 1 kemudian dilihat apakah CX telah mencapai nol. Proses looping akan selesai jika register CX telah mencapai nol. Oleh karena inilah maka jika kita menjadikan CX=0 pada saat pertama kali, kita akan mendapatkan suatu pengulangan yang terbanyak karena pengurangan 0-1 =-1(FFFF).

21. INC

    Menambahkan nilai variabel yang ditunjuk dengan 1 dan hasilnya disimpan di variabel tersebut.

22. SUBB

    Mengurangkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya disimpan di variabel tersebut.

23. ADD

    Menjumlahkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya disimpan di variabel tersebut.

24. ADDC

    Menjumlahkan nilai variabel yang ditunjuk dengan nilai lain dan ditambah dengan nilai carry yang kemudian hasilnya disimpan di variabel yang ditunjuk tersebut.

25. NOP

    No Operation, yang berarti tidak melakukan apa – apa. Tujuan dari instruksi ini adalah hanya untuk menambahkan siklus pulsa pewaktu sebesari 1 siklus.

26. END

    END biasanya diletakkan di akhir baris dari file program sumber *assembler* sebagai tanda akhir pernyataan (*statement*) bagi program *assembler* dalam melakukan proses *assembly*.