

repository.ub.ac

ALAT PENAMPIL AKORD GITAR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh
FEBBIE ADHITYA GIRINDRO
NIM. 0110633031

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN ELEKTRO
MALANG
2007**

UNIVERSITAS BRAWIJAYA

ALAT PENAMPIL AKORD GITAR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh :

FEBBIE ADHITYA GIRINDRO

NIM. 0110633031

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Moch. Rif'an, ST, MT
NIP. 132 283 659

Waru Djuriatno, ST, MT
NIP. 132 158 733



ALAT PENAMPIL AKORD GITAR

Disusun oleh :

FEBBIE ADHITYA GIRINDRO

NIM. 0110633031

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 09 Agustus 2007

DOSEN PENGUJI

Ir. Ponco Siwindarto, MS
NIP. 131 837 966

Ir. M. Julius, St., MS
NIP. 131 124 655

Panca Mudjirahardjo, ST, MT
NIP: 132 288 163

Ir. Bambang Siswojo
NIP. 131 759 588

Mengetahui,

Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom

NIP. 131 879 033



PENGANTAR

Assalamualaikum Wr. Wb.

Puji syukur ke hadirat Allah SWT atas terselesaikannya penulisan Tugas Akhir ini, karena hanya atas perkenan dan petunjuk-Nya lah, maka segala sesuatu dapat terjadi. Kendala dan cobaan yang mengiringinya semata adalah jalan yang dibentangkanNya untuk menyadari betapa kerdil dan tidak berdayanya manusia dalam menghadapi takdirNya.

Tugas Akhir ini mengambil judul *“Alat Penampil Akord Gitar”*. Sangat sulit rasanya menyatakan bahwa penulis telah berhasil menyelesaikan Tugas Akhir ini tanpa bantuan banyak pihak. Pada kesempatan ini penulis mengucapkan terima kasih kepada :

- Kedua orang tuaku dan semua saudara-saudaraku yang telah memberi banyak dukungan.
- Bapak Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro dan Bapak Rudy Yowono, ST,M.Sc. selaku Sekretaris Jurusan Teknik Elektro.
- Bapak Ir. Ponco Siwindarto, MS dan sebagai KKDK Teknik Elektronika yang telah menerima dan mengesahkan judul skripsi ini.
- Bapak Moch. Rif'an, ST, MT dan Bapak Waru Djuriatno, ST, MT selaku dosen pembimbing yang senantiasa memberikan koreksi, masukan serta waktunya sehingga Tugas Akhir ini terselesaikan.
- Teman-teman angkatan 2001 yang telah membantu penulis menyelesaikan tugas akhir ini.
- Dan berbagai pihak yang tidak mungkin saya sebutkan satu persatu. Terima kasih telah memberikan dukungan demi terselesainya tugas akhir ini.

Semoga kebaikan dan amalnya dibalas oleh Allah SWT, Amiiiiin...

Penulis menyadari ketidak sempurnaan dalam penyusunan skripsi ini, sehingga penulis sangat mengharapkan kritik, saran, dan masukan dari pembaca. Akhir kata, semoga skripsi ini dapat memberikan manfaat dan berguna bagi pembaca khususnya mahasiswa Jurusan Teknik Elektro Universitas Brawijaya Malang.

Wassalamualikum, Wr. Wb.

Malang, 09 Agustus 2007

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	
LEMBAR PERSEMBAHAN	
PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
ABSTRAK	vii

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Sistematika Penulisan	2

BAB II TINJAUAN PUSTAKA

2.1. Akord	4
2.1.1. Macam-Macam Akord	5
2.1.2. Cara Menulis dan Membaca Akord	6
2.1.3. Contoh Akord	7
2.1.4. Frekuensi Gitar	7
2.2. ISD 25120	7
2.2.1. Mode Operasi	10
2.2.2. Merekam Pesan Dalam Mode <i>Push-Button</i>	11
2.2.3. Memainkan Ulang Pesan Dalam Mode <i>Push-Button</i>	12
2.2.4. Durasi	12
2.2.5. Resolusi Alamat	13
2.3. Mikrokontroler AT89S51	13
2.3.1. Konfigurasi Pin	15

2.3.2. Struktur dan Operasi <i>Port</i>	16
2.3.3. Organisasi Memori	17
2.3.4. Osilator	18
2.3.5. Reset	19
2.3.6. Register Fungsi Khusus	20
2.4. <i>Keypad</i>	21
2.5. LED	23
2.6. <i>Seven Segment</i>	23

BAB III METODE PENELITIAN

3.1. Studi Kepustakaan	25
3.2. Perancangan Alat	25
3.3. Pengujian Rangkaian Pada <i>Project Board</i>	25
3.4. Perancangan Perangkat Lunak	26
3.5. Pengujian Alat	26
3.6. Pengambilan Kesimpulan dan Saran	26
3.7. Penyusunan Buku Laporan Tugas Akhir	26

BAB IV PERENCANAAN ALAT

4.1. Perencanaan Sistem	27
4.1.1. Spesifikasi Alat	27
4.1.2. Blok Diagram Alat	27
4.1.3. Prinsip Kerja Alat	28
4.2. Perencanaan Perangkat Keras	28
4.2.1 Rangkaian LED Indikator	28
4.2.2 Rangkaian Mikrokontroler AT89S51	30
4.2.3 Rangkaian <i>Keypad</i>	34
4.2.4 Rangkaian <i>Display</i>	35
4.2.5 Rangkaian Pemutar Suara	37
4.3. Perencanaan Perangkat Lunak	39

BAB V PENGUJIAN ALAT

5.1. Pengujian Rangkaian <i>Keypad</i>	42
5.2. Pengujian Rangkaian LED Indikator	43

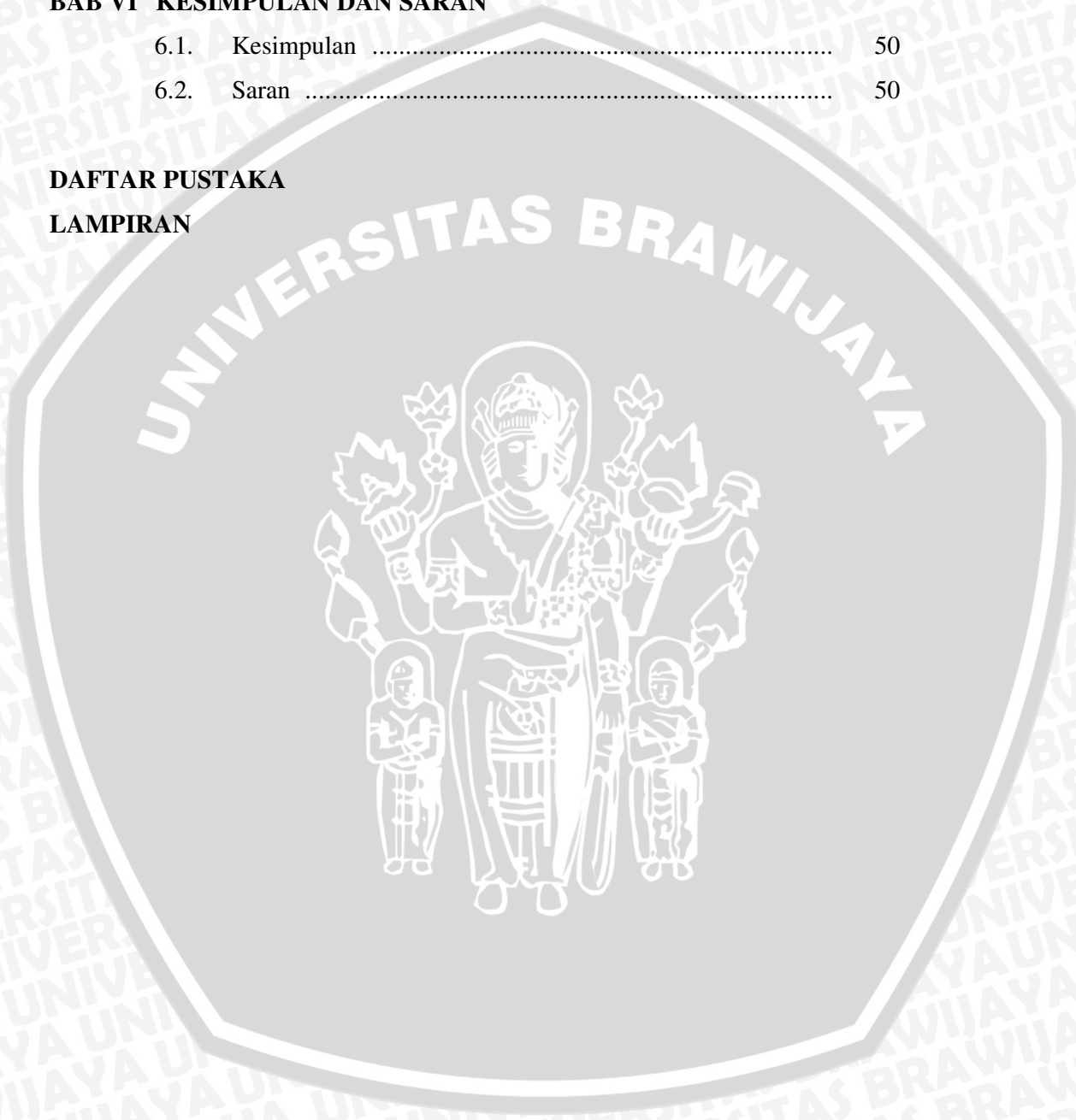
5.3. Pengujian Rangkaian *Display* 45
5.4. Pengujian Rangkaian Pemutar Suara ISD25120 46
5.5. Pengujian Rangkaian Keseluruhan Sistem 48

BAB VI KESIMPULAN DAN SARAN

6.1. Kesimpulan 50
6.2. Saran 50

DAFTAR PUSTAKA

LAMPIRAN

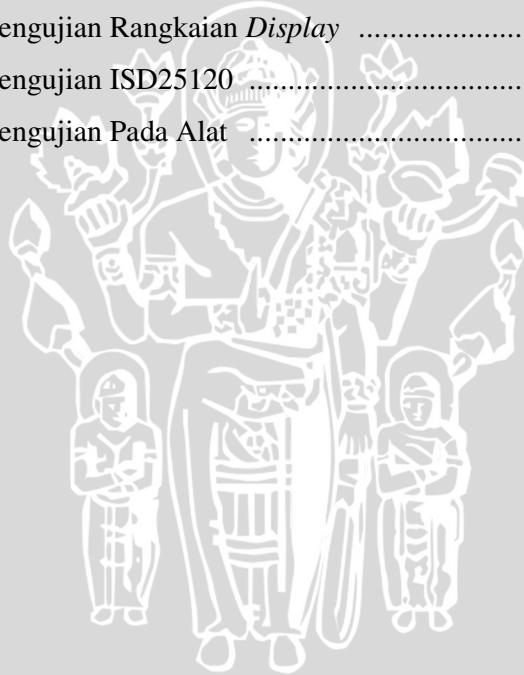


DAFTAR GAMBAR

Gambar 2.1.	Contoh Penulisan Akord Dan Pembacaannya	6
Gambar 2.2.	Contoh Akord	7
Gambar 2.3.	Blok Diagram ISD25120	8
Gambar 2.4.	Susunan Kaki-Kaki ISD25120	8
Gambar 2.5.	Blok Diagram AT89S51	14
Gambar 2.6.	Susunan Kaki AT89S51	15
Gambar 2.7.	Struktur Memori MCS-51	18
Gambar 2.8.	Rangkaian Osilator	18
Gambar 2.9.	(a) <i>Power On Reset</i>	19
	(b) Rangkaian Ekuivalen <i>Power On Reset</i>	19
Gambar 2.10.	Rangkaian <i>Keypad</i>	22
Gambar 2.11.	Lambang skematik LED	23
Gambar 2.12.	Tampilan <i>Seven Segment Common Anode</i>	24
Gambar 4.1.	Blok Diagram Keseluruhan Sistem	27
Gambar 4.2.	Rangkaian LED Indikator	29
Gambar 4.3.	Rangkaian Mikrokontroler AT89S51	30
Gambar 4.4.	Rangkaian Reset	33
Gambar 4.5.	Rangkaian Tombol Nada	33
Gambar 4.6.	Rangkaian <i>Keypad</i>	34
Gambar 4.7.	Rangkaian <i>Display</i>	36
Gambar 4.8.	Rangkaian Pemutar Suara ISD25120	38
Gambar 4.9.	Diagram Alir Perangkat Lunak Mikrokontroler 1	40
Gambar 4.10.	Diagram Alir Perangkat Lunak Mikrokontroler 2	41
Gambar 5.1.	Diagram Blok Pengujian Rangkaian <i>Keypad</i>	43
Gambar 5.2.	Diagram Blok Pengujian Rangkaian LED Indikator	44
Gambar 5.3.	Diagram Blok Pengujian Rangkaian <i>Display</i>	45
Gambar 5.4.	Modul IC ISD25120	47
Gambar 5.5.	Blok Diagram Keseluruhan Sistem	48

DAFTAR TABEL

Tabel 2.1.	Durasi Waktu dan <i>Sample Rate</i> ISD2500	12
Tabel 2.2.	Kapasitas Penyimpanan dan Resolusi ISD2500	13
Tabel 2.3.	Fungsi Khusus Port 3	17
Tabel 2.4.	Register Fungsi Khusus	20
Tabel 2.5.	Tabel Kebenaran IC 74C922	22
Tabel 4.1.	Tabel Kebenaran IC 74C922	35
Tabel 4.2.	Alamat dan Durasi Untuk Merekam Bunyi Akord C	39
Tabel 5.1.	Hasil Pengujian Rangkaian <i>Keypad</i>	43
Tabel 5.2.	Hasil Pengujian Rangkaian LED Indikator	44
Tabel 5.3.	Hasil Pengujian Rangkaian <i>Display</i>	45
Tabel 5.4.	Hasil Pengujian ISD25120	47
Tabel 5.5.	Hasil Pengujian Pada Alat	49



ABSTRAK

Febbie Adhitya Girindro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Agustus 2007, *Alat Penampil Akord Gitar.*

Dosen Pembimbing : Moch. Rif'an, ST, MT dan Waru Djuriatno, ST

Musik tidak dapat dipisahkan dalam kehidupan ini, banyak alat musik diciptakan termasuk gitar. Dari masa yang dahulu sampai sekarang gitar merupakan satu alat musik yang banyak digemari masyarakat, baik tua maupun muda. Untuk belajar memainkan alat musik ini yang sangat mendasar adalah mengerti tentang akord. Pada dasarnya akord gitar ada dua macam yaitu akord mayor dan akord minor. Dari akord-akord tersebut dapat dikembangkan menjadi bermacam-macam. Sering kita jumpai akord-akord gitar beserta penjariannya terdapat pada halaman tengah maupun belakang pada buku-buku gitar. Untuk itu para pemula akan kesulitan dalam menguasai permainan gitar dengan membolak-balik buku gitar. Untuk itu pada skripsi kali ini akan direalisasikan sebuah “Alat Penampil Akord Gitar” dengan sembilan variasi akord alat ini dapat menampilkan 108 macam akord gitar. Dengan menekan keypad untuk memilih akord, alat ini akan menampilkan akord tersebut dengan nyala LED dan seven segmen sebagai penunjuk untuk jari yang digunakan menekan. Alat ini menggunakan mikrokontroler sebagai pengendali utama. Selain itu alat ini juga menggunakan ISD untuk mengeluarkan bunyi dari akord gitar yang ditampilkan. Dengan alat ini diharapkan para pemula akan lebih mudah untuk belajar bermain gitar. Karena dengan alat ini para pemula akan lebih mengerti tentang akord dan posisi penjariannya.

Kata kunci : *Akord, mikrokontroler, LED, seven segment, ISD*

BAB I PENDAHULUAN

1.1. Latar Belakang

Pada masa dahulu sampai saat ini, gitar merupakan alat musik yang banyak digemari masyarakat, baik orang dewasa, remaja, maupun anak-anak, bahkan orang yang sudah berumur pun banyak yang hobi memainkan gitar. Harga gitar yang ada di pasaran sangat bervariasi, ada yang mahal ataupun murah tergantung dari jenis dan kualitasnya, bahkan dapat dibilang relatif terjangkau karena banyak sekali orang yang mempunyai alat musik satu ini.

Di samping itu, gitar merupakan alat musik yang sangat harmonis sehingga dapat digunakan untuk mengiringi berbagai macam aliran lagu. Banyak orang yang ingin belajar alat musik ini, dan pasti mereka menginginkan jalan termudah untuk melakukannya. Sering kali dijumpai orang yang belajar bermain gitar terhenti di tengah jalan karena menemui kesulitan dalam teknik bermain gitar, entah itu sulit untuk menghafal akordnya maupun menggunakannya. Salah satu hal terpenting dalam permainan gitar adalah mengetahui akord-akord gitar. Akord gitar ini berfungsi sebagai ritem atau pengiring lagu dalam permainan gitar. Akord gitar dasar terdiri dari dua jenis yaitu akord mayor dan akord mionor. Dari dua jenis tersebut dapat dikembangkan menjadi puluhan bahkan ratusan akord. Biasanya akord-akod gitar sering kita temukan pada bagian belakang buku-buku lagu. Untuk seorang pemula yang sedang belajar bermain gitar, tentulah tidak mudah untuk mempelajari bermacam-macam akord gitar sambil membolak-balik buku.

Berdasar hal tersebut diatas terpikirkan penulis membuat suatu alat penampil akord gitar yang praktis dalam pemakaiannya, dilengkapi dengan suara dari akord gitar yang ditampilkan dengan harapan dapat membantu mempermudah jalan untuk para pemula dalam menguasai permainan gitar.

1.2. Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana merancang dan membuat alat penampil akord gitar berbasis mikrokontroler AT89S51 yang mudah untuk digunakan.

2. Bagaimana merancang pengalamatan Mikrokontroler sehingga dapat digunakan sebagai pengontrol pada alat ini.
3. Bagaimana merancang variasi untuk akord gitar yang ditampilkan.

1.3. Tujuan

Tujuan pembuatan alat penampil akord gitar ini adalah untuk memudahkan para pemula untuk belajar bermain gitar, dengan menekan tombol-tombol akord yang disediakan, alat ini akan menampilkan akord-akord gitar yang diinginkan beserta variasinya dan juga menunjukkan letak jari-jari pada saat menekan senar gitar, selain itu alat ini juga dapat mengeluarkan bunyi dari akord yang ditampilkan.

1.4. Batasan Masalah

Adapun batasan masalah dalam perancangan dan pembuatan alat ini adalah sebagai berikut :

1. Pengendali sistem dilakukan oleh dua buah mikrokontroler AT89S51
2. Menggunakan LED dan *seven segment* sebagai keluaran tampilan
3. Suara yang diisikan ke dalam ISD adalah suara gitar akustik dengan senar string
4. Akord yang ditampilkan sebanyak 108 macam

1.5. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut :

Bab I : Pendahuluan

Menjelaskan latar belakang permasalahan, rumusan masalah, tujuan, ruang lingkup pembahasan dan sistematika pembahasan tugas akhir.

Bab II : Tinjauan Pustaka

Menjelaskan tentang teori-teori dasar penunjang perancangan dan pembuatan alat.

Bab III: Metodologi Penelitian

Menjelaskan tahap-tahap dan metode yang dilakukan dalam perencanaan pembuatan alat.

Bab IV: Perencanaan dan Pembuatan Alat

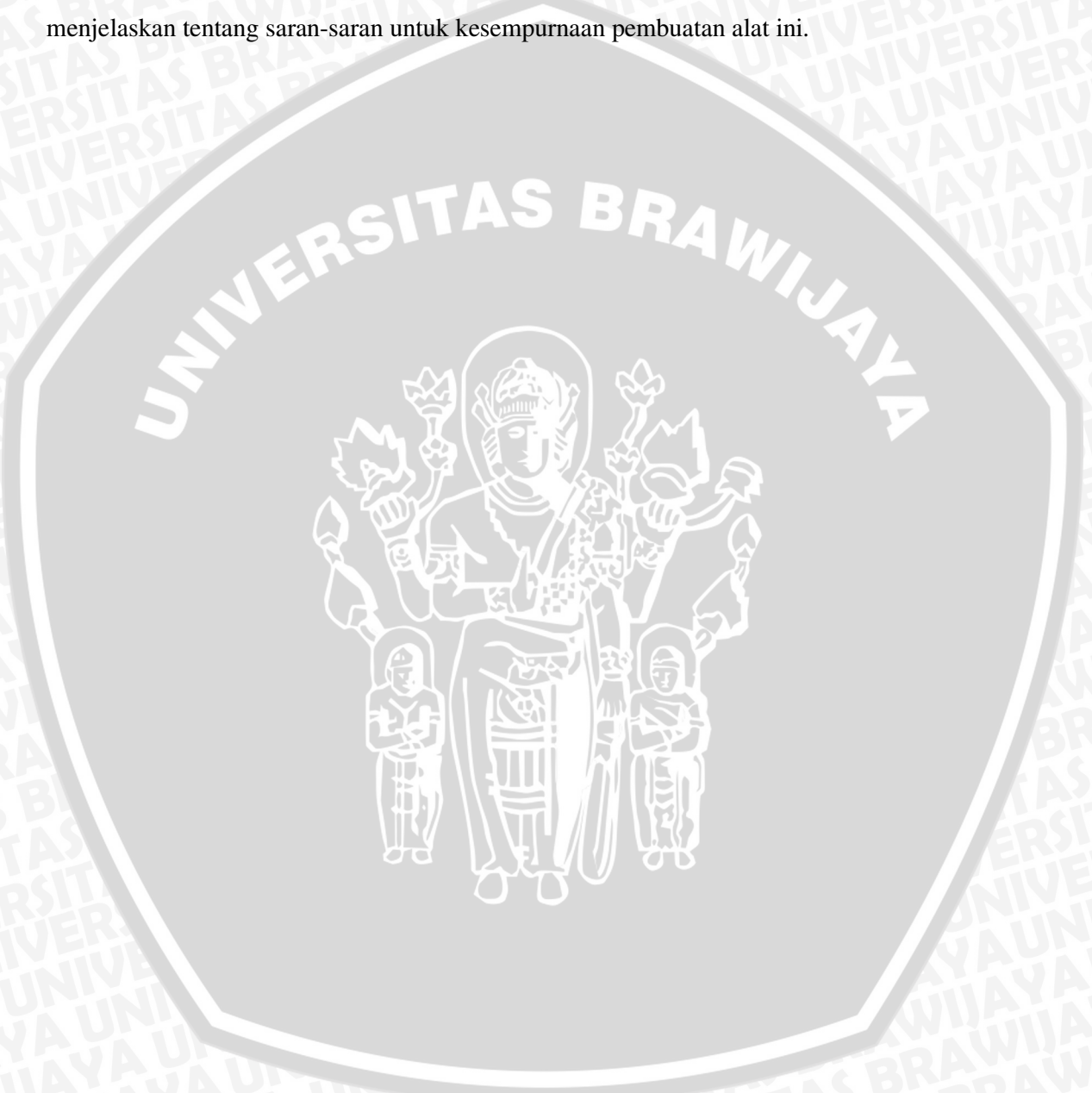
Menjelaskan spesifikasi, diagram blok dan prinsip kerja dari alat yang akan dirancang.

Bab V : Pengujian dan Analisis

Menjelaskan pengujian alat dan analisis terhadap data hasil pengujian menggunakan teori yang ada.

Bab VI: Kesimpulan dan Saran

Menjelaskan kesimpulan dari perancangan dan pembuatan alat serta menjelaskan tentang saran-saran untuk kesempurnaan pembuatan alat ini.



BAB II

TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan alat ini, maka perlu penjelasan dan uraian teori penunjang yang digunakan.

Teori-teori penunjang yang akan dijelaskan dalam bab ini adalah :

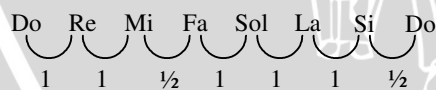
1. Akord
2. ISD 25120
3. Mikrokontroler AT89S51
4. Keypad
5. LED
6. *Seven Segment*

2.1. Akord

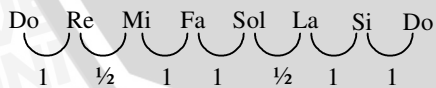
Akord adalah kumpulan tiga nada atau lebih yang dimainkan secara bersamaan terdengar harmonis. Akord bisa dimainkan secara terputus-putus ataupun secara bersamaan. Akord ini digunakan untuk mengiringi suatu lagu. Ketika anda memetik tiga senar gitar C, E, dan G secara bersamaan, ini berarti anda sudah memainkan akord. Contoh alat musik lainnya yang bisa memainkan akord adalah piano, organ dan *electone*.

Tangga nada dasar akord berasal dari tangga nada diatonis dengan jarak sebagai berikut:

- Tangga nada Diatonis Mayor



- Tangga nada Diatonis Minor



Dari dua tangga nada diatonis tersebut dapat dikembangkan menjadi berbagai macam akord, antara lain akord mayor, akord minor, akord mayor 7, akord minor 7, akord *augmented*, akord *suspended*, dan masih banyak yang lainnya. Akord yang sering dipakai dalam suatu lagu yang sederhana adalah akord mayor dan akord minor. Akord lainnya digunakan untuk memperindah atau mengubah kualitas suatu lagu. Penyisipan akord yang berbeda akan memberikan efek rasa yang berbeda dalam iringan suatu lagu.

Akord yang memiliki nama berbeda namun bila dimainkan bersuara sama disebut akord enharmonis. Contohnya: akord Cb (Ces mayor) dengan B (B mayor). Akord tersebut adalah akord dasar, akord tersebut bisa dibolak-balik urutannya (disebut balikan pertama dan balikan kedua). Misalnya: C on E (C/E). Ini berarti kita harus memainkan akord dengan urutan E-G-C bukan C-E-G. C on E adalah balikan pertama dari akord dasar C. Balikan keduanya adalah C on G (C/G) yaitu G-E-C. Untuk menentukan sebuah akord dipakai rumus nada dasar 1-3-5, yang dimaksud 1-3-5 adalah nada pertama, ketiga dan kelima. Misalkan kita akan menentukan akord C caranya adalah C dijadikan nada pertama, nada ketiga adalah E, dan nada kelima adalah G. Jadi akord C adalah akord yang didalamnya terdapat nada C, E, G.

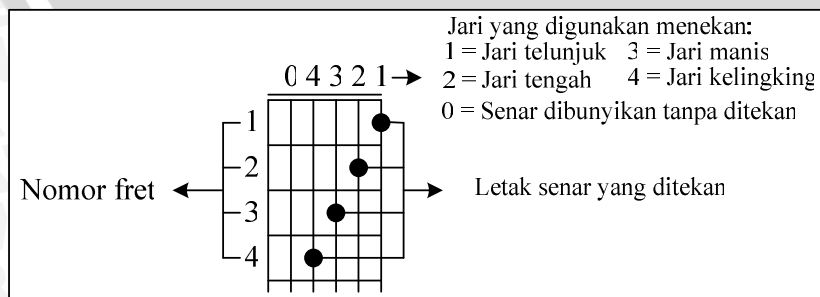
2.1.1. Macam-Macam Akord:

- Akord Mayor (may)
Akord mayor adalah akord yang nada dasarnya adalah 1-3-5 dan *interval* antara nadanya 2 – 1½. Contoh akord mayor adalah C (C-E-G).
- Akord Minor (m)
Akord minor adalah akord yang nada dasarnya adalah 1-3-5 dan *interval* antara nadanya 1½ - 2. Apabila anda sudah tahu suatu akord mayor misalnya: C mayor maka anda bisa mengetahui pula akord minornya (Cm) yaitu dengan cara menurunkan nada yang ada ditengah (nada ketiga) sebanyak setengah *interval*. Sehingga didapat akord Cm adalah C-Es (E diturunkan setengah menjadi Es)-G.
- Akord *Dominant Seventh* (7th)
Akord *dominant seventh* adalah akord mayor yang ditambahi nada ketujuh dari nada dasar (1-3-5-7), *interval* nadanya adalah 2 - 1½ - 1½. Contoh akord *dominant seventh* adalah: C7 terdiri atas nada C-E-G-Bb.
- Akord Mayor *Seventh* (may7)
Akord mayor *seventh* adalah akord mayor yang ditambahi nada ketujuh dari nada dasar (1-3-5-7), *interval* nadanya adalah 2 - 1½ - 2. Contoh akord mayor *seventh* adalah: Cmay7 terdiri atas nada C-E-G-B.
- Akord Minor *Seventh* (m7)
Akord minor *seventh* adalah akord minor yang ditambahi nada ketujuh dari nada dasar (1-3-5-7), *interval* nadanya adalah 1½ - 2 - 1½. Contoh akord minor *seventh* adalah: Cm7 terdiri atas nada C-Eb-G-Bb.

- Akord *Augmented* (+)
Akord *augmented* adalah akord yang nada dasarnya adalah 1-3-5 dan *interval* antara nadanya 2 – 2. Contoh akord *augmented* adalah: Caug atau C+ terdiri atas nada C-E-Ab.
- Akord *Suspended Fourth* (sus4)
Akord *suspended fourth* adalah akord yang nada dasarnya adalah 1-3-5 dan *interval* antar nadanya adalah 2½ - 1. Apabila anda sudah tahu suatu akord mayor misalnya: C mayor maka anda bisa mengetahui pula akord *suspended fourth*-nya (Csus4) yaitu dengan cara menaikkan nada yang ada ditengah (nada ketiga) sebanyak setengah *interval*. Sehingga akord Csus4 terdiri atas nada C-F (E dinaikkan setengah menjadi F)-G.
- Akord *Dominant Ninth* (9th)
Akord mayor *ninth* adalah akord mayor yang ditambahi nada ketujuh dan nada kesembilan dari nada dasar (1-3-5-7-9), *interval* nadanya adalah 2 - 1½ - 1½ - 2. Contoh akord *dominant ninth* adalah: C9 terdiri atas nada C-E-G-Bb-D.
- Akord *Dominant Eleventh* (11th)
Akord mayor *eleventh* adalah akord mayor yang ditambahi nada ketujuh, nada kesembilan dan nada kesebelas dari nada dasar (1-3-5-7-9-11), *interval* nadanya adalah 2 - 1½ - 1½ - 2 - 1½. Contoh akord *dominant eleventh* adalah: C11 terdiri atas nada C-E-G-Bb-D-F.

2.1.2. Cara Menulis dan Membaca Akord

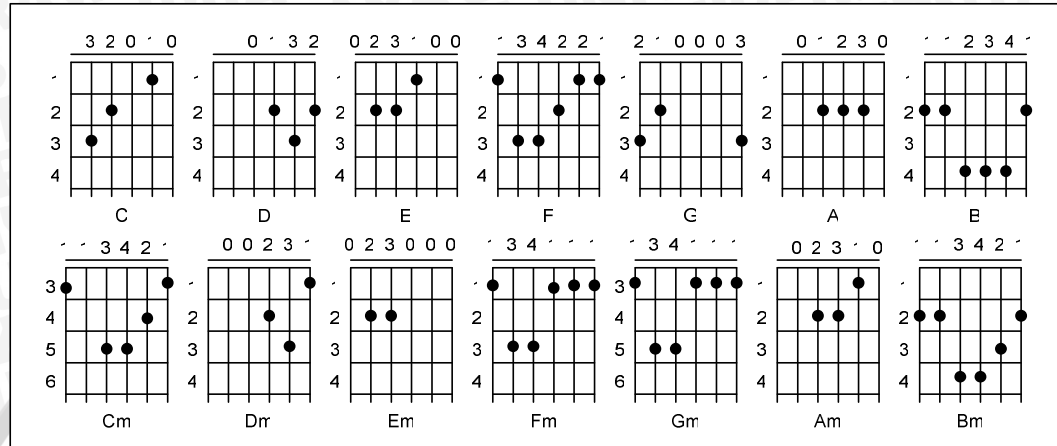
Dalam bermain gitar ada cara untuk menuliskan akord gitar, adapula cara membaca dari akord gitar tersebut. Contoh penulisan akord gitar dan cara membacanya ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Contoh Penulisan Akord dan Pembacaannya

2.1.3. Contoh Akord

Akord gitar ada banyak macamnya bahkan bisa sampai ratusan macam. Akord yang sering dimainkan para pemain gitar adalah akord mayor dan akord minor. Gambar 2.2 adalah beberapa contoh dari akord-akord untuk bermain gitar.



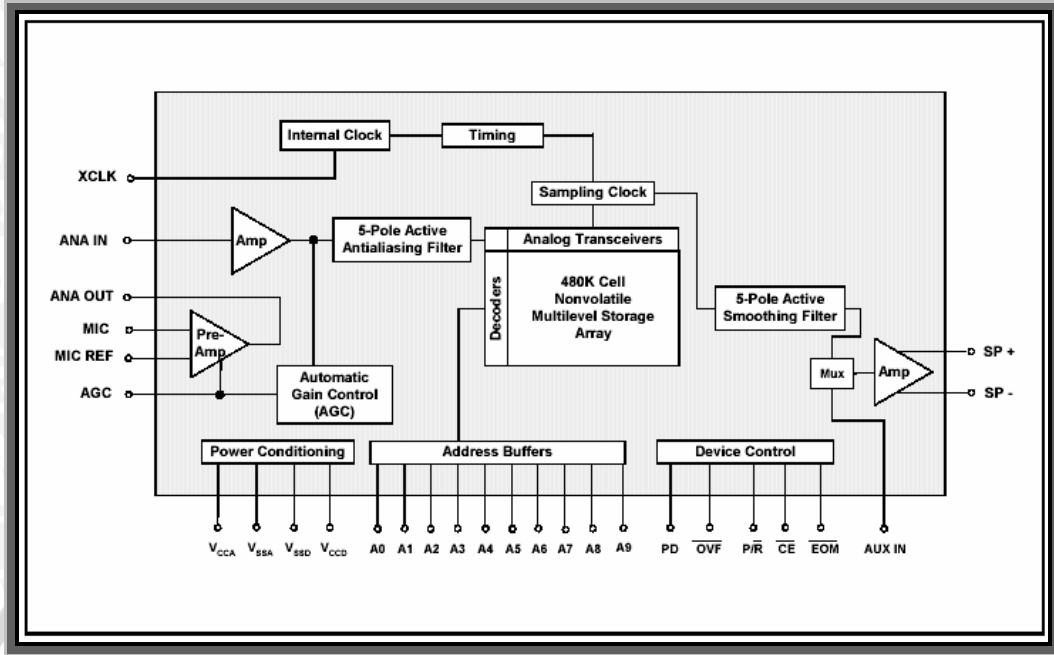
Gambar 2.2 Contoh Akord

2.1.4. Frekuensi Gitar

Telinga manusia mampu menangkap suara mulai dari frekuensi 20Hz sampai dengan 20kHz. Sedangkan gitar mempunyai bermacam-macam frekuensi, dari tiap senar menghasilkan frekuensi yang berbeda, selain itu jika ditekan di fret tertentu frekuensi yang dihasilkan juga akan berbeda. Frekuensi standar gitar yaitu nada A pada fret ke-5 senar E (senar1) adalah 440Hz. Untuk senar yang tidak ditekan, pada senar pertama (E) frekuensinya adalah 329.6Hz, senar kedua (B) adalah 246.9Hz, senar ketiga (G) adalah 196Hz, senar keempat frekuensinya adalah 146.8, frekuensi senar kelima (A) adalah 110Hz, dan senar ke-6 (E) frekuensinya adalah 82.4Hz. (www.phys.unsw.edu.au)

2.2. ISD 25120

ISD25120 adalah IC produksi ISD, IC ini merupakan jenis EEPROM (*Electrical Erasable Programmable ROM*). EEPROM adalah jenis ROM yang dapat diprogram ulang secara elektrik, tanpa menggunakan sinar ultra violet. IC ISD25120 ini dapat merekam pesan maksimal 120 detik dan dapat dikaskade sehingga pesan yang disimpan dapat diperpanjang sesuai dengan keinginan. Peralatan CMOS yang ada di dalamnya adalah *chip oscillator*, *microphone preamplifier*, *automatic gain control*, *antialiasing filter*, *smoothing filter*, *speaker amplifier*. Blok diagram dari ISD25120 ditunjukkan dalam Gambar 2.3.

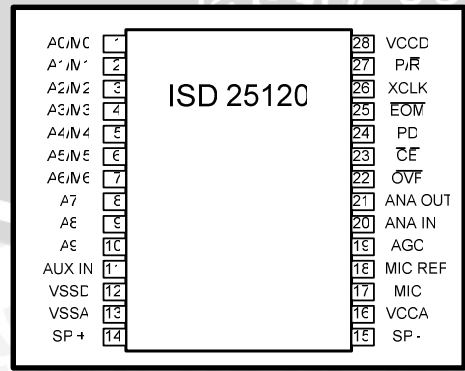


Gambar 2.3 Blok Diagram ISD25120

(Sumber : Data Sheet ISD2500)

ISD25120 ini kompatibel dengan mikrokontroler. Alamat dan jalur kendali dapat dihubungkan dengan mikrokontroler, sehingga mengijinkan penyimpanan dan pengalamatan yang kompleks.

Kelebihan dari ISD Chip Recorder adalah penggunaan *nonvolatile memory* yang mampu menyimpan data sampai 100 tahun dalam keadaan tidak diberikan catu daya. Perekaman ulang dapat dilakukan sampai 100.000 kali. Susunan kaki-kaki ISD25120 ditunjukkan dalam Gambar 2.4.



Gambar 2.4 Susunan Kaki-Kaki ISD25120

(Sumber : Data Sheet ISD2500)

Fungsi dari masing-masing kaki IC ISD 25120 dijelaskan sebagai berikut:

- a) *Power Down (PD)*
Apabila suara direkam atau diputar ulang, PD dapat diaktifkan supaya ISD 25120 mendapat konsumsi daya rendah. Jika EOM rendah selama terjadi kondisi *overflow*, PD harus berlogika “1” untuk mereset *address* supaya kembali ke awal perekam *playback*.
- b) *Chip Enable Input (CE)*
Untuk mengakses ISD 25120 maka pin CE harus berlogika “0”. Apabila pin CE berlogika “1” maka ISD 25120 tidak dapat mengakses dan *auxiliary input* terhubung langsung ke amplifier speaker.
- c) *Playback Record Input (P/R)*
Pada mode rekam pin P/R harus berlogika “0” dan apabila *playback* maka P/R harus berlogika “1”.
- d) *End of Message Output (EOM)*
Penanda ini secara otomatis disisipkan pada akhir penyimpanan pesan. EOM output akan rendah selama periode T_{EOM} pada akhir pesan telah melampaui 120 detik (*device full*).
- e) *Address / Mode Input (A0 – A9)*
Pin ini mempunyai dua fungsi tergantung pada *level* dari dua *Most Significant Bit* (MSB). Jika salah satu atau keduanya dari dua MSB berlogika rendah, semua input dianggap sebagai bit alamat dan digunakan sebagai awal alamat untuk proses *record* atau *playback cycle* terbaru.
- f) *Voltage Input* (V_{CCA} , V_{CCD}) dan *Ground Input* (V_{SSA} , V_{SSD})
Digunakan untuk dua input (untuk analog dan untuk digital) agar dapat memperkecil fase yang timbul.
- g) *Overflow Output (OVF)*
Pin ini akan *LOW* apabila batas akhir memori terlewati.
- h) *Microphone Output (MIC)*
Input *microphone* akan mentransfer sinyal yang masuk ke amplifier yang berada di dalam IC. Rangkaian AGC mengontrol penguatan antara – 15 sampai 24 dB. Microphone eksternal harus dihubungkan dengan kondensator kopling ke pin ini. Nilai kapasitor kopling bersama dengan impedansi internal (normalnya 10 K Ω) akan menentukan frekuensi *cutt – off* rendah dari filter internal.

- i) *Microphone Reference Input* (MIC REF)
MIC REF input merupakan *inverting input* yang berhubungan dengan *preamplifier*.
- j) *Automatic Gain Control* (AGC)
AGC menyesuaikan penguatan *output* dari *preamplifier* yang dapat digunakan oleh pemakai.
- k) *Analog Output* (ANA OUT)
Pin ini merupakan output dari *preamplifier* yang dapat digunakan oleh pemakai.
- l) *Analog Input* (ANA IN)
Analog input akan mentransfer sinyal ke *chip* untuk perekaman. Jika digunakan *microphone* maka ANA OUT harus dihubungkan ke pin ANA IN melalui kondensator kopling. Nilai kapasitor ini dengan impedansi input ANA IN (normalnya 3 K Ω) akan menentukan frekuensi *cut – off* filter.
- m) *Auxiliary Input* (AUX IN)
Auxiliary input dimultiplex melewati output amplifier dan dikeluarkan ke *speaker* jika CE dalam keadaan *high*, P/R dan *playback* tidak aktif.

2.2.1. Mode Operasi

Untuk menjalankan mode operasi ISD 25120 ada dua hal penting, yaitu:

- Semua operasi dilakukan dengan inisialisasi pada alamat 000H. ketika terjadi perubahan dari perekaman ke memainkan ulang alamat pointer direset menuju pada alamat 000H.
- Mode operasi aktif. Jika CE berlogika rendah, A8 dan A9 berlogika *high*.
- Mode operasi yang digunakan di dalam ISD 25120 adalah:

➤ M0 (*Message Cueing*)

Digunakan pemakai untuk melompati pesan atau menuju kepesan yang lain tanpa mengetahui secara pasti alamat data pada pesan yang baru.

➤ M1 (*Delete EOM Markers*)

Untuk menggabungkan beberapa pesan yang telah direkam. EOM diset hanya pada saat pesan yang direkam telah melebihi batas kapasitas perekaman, sehingga yang terputus-putus dapat tersambung satu dengan yang lainnya pada waktu memainkan ulang data hasil perekaman.

➤ M2 (*Unused*)

Pada waktu mengaktifkan mode operasi dihubungkan ke ground.

➤ M3 (*Message Looping*)

M3 untuk memainkan berulang-ulang pesan yang terekam secara otomatis yang dimulai pada alamat 000H. Pesan akan dimainkan ulang secara utuh dari awal sampai akhir tanpa membuat OVF berlogika rendah.

➤ M4 (*Consecutive Addressing*)

Merekam dan memainkan ulang beberapa pesan secara berurutan.

➤ M5 (*CE-Level Activated*)

Untuk menghentikan proses memainkan ulang pesan. Pada waktu CE diberikan logika 'low', maka proses dihentikan, setelah CE diberikan logika 'high' proses dilanjutkan dan akan berhenti ketika CE berlogika 'low' kembali.

➤ M6 (*Push-Button Mode*)

Dalam mode ini beberapa pin mempunyai fungsi alternatif, yaitu:

✓ Pin CE (*Start/Pause*)

Ketika CE menuju logika 'low' maka proses perekaman atau memainkan ulang pesan berlangsung (tergantung dari kondisi masukan P/R apabila berlogika 'high' terjadi proses memainkan ulang pesan, apabila berlogika 'low' maka terjadi proses perekaman).

✓ Pin PD (*Stop/Reset*)

Ketika PD menuju logika 'high', proses dihentikan (perekaman atau memainkan ulang pesan), alamat direset menuju alamat awal.

✓ Pin EOM (*Run*)

Sebagai penunjuk adanya proses perekaman atau memainkan ulang data.

2.2.2. Merekam Pesan Dalam Mode *Push-Button*

Proses perekaman dalam mode *Push-Button* diuraikan sebagai berikut:

- Pin PD diberikan logika rendah.
- Pin P/R diberikan logika rendah
- Pin CE diberikan logika rendah sesaat, proses perekaman dimulai, EOM menuju logika tinggi sebagai indikator berlangsungnya proses perekaman.
- Pin CE diberikan logika rendah sesaat, perekaman berhenti, EOM berlogika rendah, data *pointer* tidak direset.
- Pin CE diberikan logika rendah sesaat, perekaman berlangsung kembali.
- Ketika perekaman secara sekuensial ketika CE diberikan logika rendah atau apabila pin PD diberikan logika tinggi.

2.2.3. Memainkan Ulang Pesan Dalam Mode *Push-Button*

Proses memainkan ulang pesan terekam dalam mode *Push-Button* adalah:

- Pin PD diberikan logika rendah.
- Pin P/R diberikan logika tinggi.
- Pin CE diberikan logika rendah, proses memainkan ulang pesan berlangsung, EOM menuju logika tinggi sebagai indikator jalannya proses memainkan ulang pesan.
- Ketika pin CE diberikan logika rendah sesaat maka proses berhenti, ketika diberikan logika rendah sesaat lagi maka proses permainan ulang pesan. Proses tersebut diatas dapat diulang-ulang terus dan suatu saat berhenti ketika pin PD diberikan logika tinggi dan terjadi *overflow*. Ketika *overflow* pemberian logika rendah pin CE mengakibatkan alamat direset menuju alamat awal dan proses memainkan ulang pesan dimulai dari alamat awal.

2.2.4. Durasi

Untuk memenuhi macam dari sistem yang diperlukan, ISD seri 2500 mengeluarkan *sigle-chip* dalam 60, 75, 90, dan 120 detik. Durasi untuk berbagai tipe ISD2500 ditunjukkan dalam Tabel 2.1.

Tabel 2.1. Durasi Waktu dan *Sample Rate* ISD2500

Tipe	Durasi (detik)	Input Sample Rate (kHz)
ISD 2560	60	8
ISD 2575	75	6,4
ISD 2590	90	5,3
ISD 25120	120	4

ISD25120 ini mempunyai *sampling clock*, frekuensi *sampling clock* ini besarnya dua kali dari frekuensi yang akan disampling. Dari *datasheet* diketahui bahwa frekuensi *sample*-nya adalah 4 KHz, jadi frekuensi yang di-*sampling* maksimal adalah 2KHz. Hasil output yang dihasilkan ISD25120 ini adalah sebagai berikut:

Dari *datasheet* diketahui memori internal = 480.000 byte

$$\frac{480000\text{byte}}{120\text{s}} = 4000\text{byte/s}$$

$$\frac{4000\text{byte/s}}{4000\text{Hz}} = 1\text{byte/sampling}$$

Jika frekuensi *sampling* ada 4000Hz, maka ada 4000 data hasil *sampling* tiap detiknya.

$$\text{bitrate} = 8\text{bit} \times 4000 = 32\text{kbps}$$

Jadi output dari ISD25120 ini adalah 32kbps, 4KHz, sedangkan suara standar MPEG1.0 layer 3 atau biasa disebut MP3 adalah 128kbps dan 44.1KHz, maka suara yang dihasilkan ISD25120 secara teori kurang layak untuk mengeluarkan bunyi dari gitar, tetapi dalam perncangan nanti akan dibuat semaksimal mungkin agar suara yang dihasilkan lebih bagus.

2.2.5. Resolusi Alamat

Resolusi alamat menyatakan kapasitas panjang waktu penyimpanan pesan dalam satuan milidetik. resolusi alamat untuk beberapa jenis IC ISD 2500 *Chip Recorder Series* ditunjukkan dalam tabel 2.2. kapasitas penyimpanan menyatakan panjang durasi waktu penyimpanan masing-masing IC.

Tabel 2.2. Kapasitas Penyimpanan dan Resolusi ISD2500

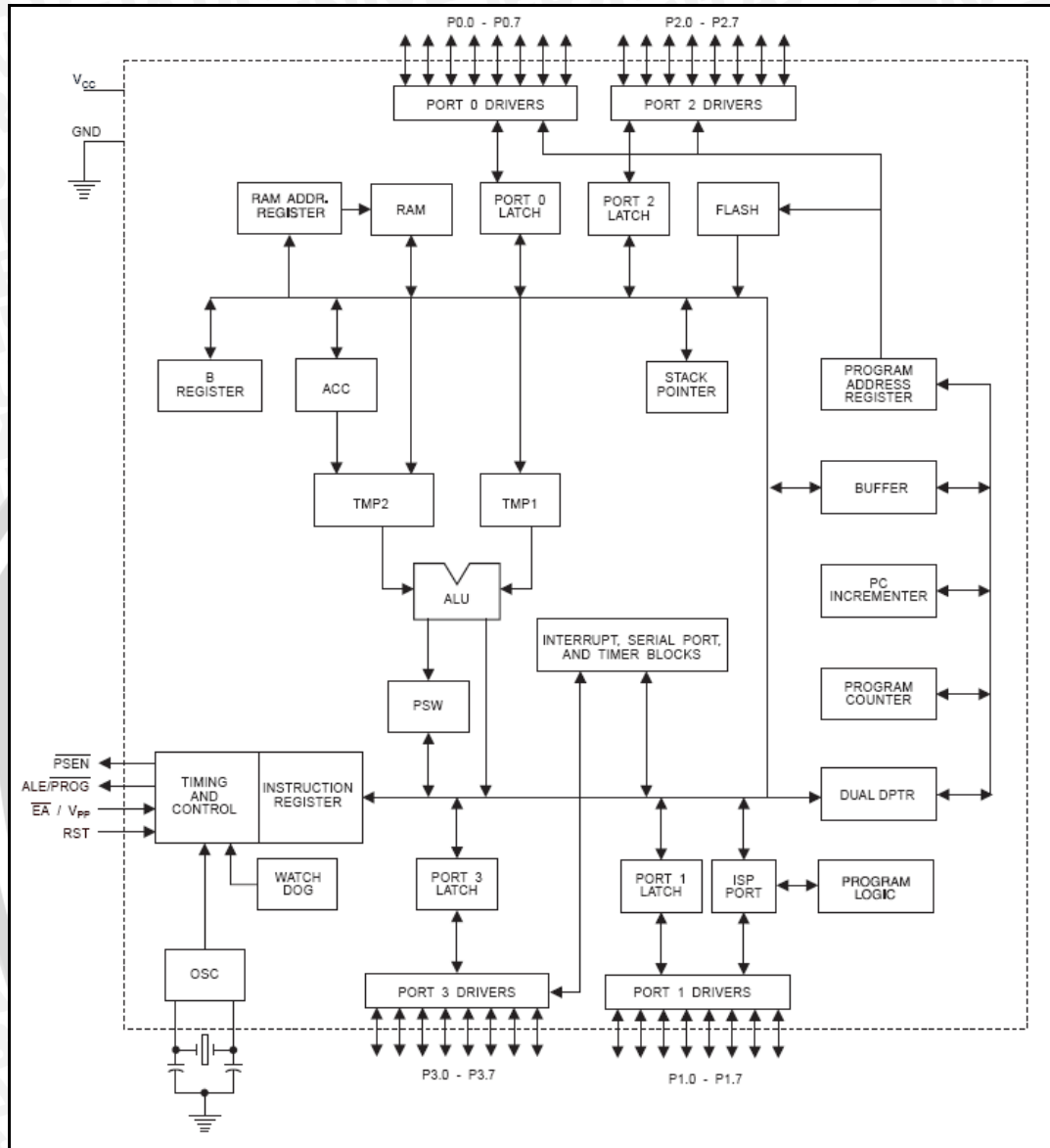
Type	Kapasitas Penyimpanan (detik)	Resolusi (milidetik)
ISD 2560	60	100
ISD 2575	75	125
ISD 2590	75	150
ISD 25120	120	200

2.3. Mikrokontroler AT89S51

Mikrokontroler AT89S51 merupakan mikrokontroler 8 *bit* kompatibel dengan standar industri MCS-51TM baik atas segi pemrograman maupun kaki tiap pin. Mikrokontroler AT89S51 mempunyai 4 *kbyte* PROM (*Flash Programmable and Erasable Read Only Memory*). Pada dasarnya mikrokontroler adalah terdiri atas mikroprosesor, *timer*, dan *counter*, perangkat I/O dan *internal memory*. Mikrokontroler termasuk perangkat yang sudah didisain dalam bentuk *chip* tunggal. Pada dasarnya mikrokontroler mempunyai fungsi yang sama dengan mikroprosesor yaitu untuk mengontrol suatu kerja sistem. Selain itu mikrokontroler juga dikemas dalam satu *chip* (*single chip*). Didalam mikrokontroler juga terdapat CPU, ALU, PC, SP, dan *register* seperti dalam mikroprosesor, tetapi juga ditambah dengan perangkat-perangkat lain seperti ROM, RAM, PIO, SIO, *counter* dan sebuah rangkaian *clock*. Mikrokontroler didisain dengan instruksi-instruksi lebih luas dan 8 *bit* instruksi yang digunakan membaca data instruksi dari *internal memory* ke ALU.

Sebagai suatu sistem kontrol mikrokontroler AT89S51 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor

didalamnya tidak terdapat keduanya. Diagram blok AT89S51 ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Diagram Blok AT89S51
(Sumber: AT89S51 Data Sheets, 1997: 4-30)

Diagram blok AT89S51 terdiri dari:

1. 8 bit CPU dengan *register A (accumulator)* dan *match register (register B)*.
2. 16 bit *Program Counter (PC)* dan *Data Pointer (DPTR) register*.
3. 8 bit *Program Status Word (PSW) register*, 8 bit *Stack Pointer*.
4. *Internal ROM* dan *EPROM* dengan kapasitas 4 *kbyte*.
5. *Internal RAM* dengan kapasitas 128 *byte* yang digunakan untuk 4 buah *register bank*, yang masing-masing terdiri dari 8 *register*, 16 *byte*, yang mana dapat

dieksekusi pada masing-masing *bit* secara *independent* (*Bit Addesable*) dan sebagai *memory variable* 8 *bit*.

6. 32 *input/output* yang disusun pada 4 *port* (*port* 0 – *port* 3).
7. 2 buah 16 *bit timer/counter* :T0 dan T1.
8. *Full Duplex Serial Data Communication* :SBUF.
9. *Control Register* :TCON, TMOD, PCON, IP dan IE.
10. 2 eksternal *interrupt* dan 3 *internal interrupt*.
11. Oscilator dan *Clock Circuit*.

Selain memiliki *feature* yang terdapat pada AT89C51, Mikrokontroler AT89S51 memiliki beberapa *feature* tambahan, yaitu:

1. *Watchdog Timer*
2. *Dual Data pointer*

2.3.1. Konfigurasi Pin

Masing-masing kaki dalam mikrokontroler AT89S51 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89S51, perancangan aplikasi mikrokontroler AT89S51 akan lebih mudah merencanakan dan membuat sistem yang dirancang. AT89S51 mempunyai 40 pin, susunan masing-masing pin dapat dilihat dalam Gambar 2.6.

1	P1.0	VCC	40
2	P1.1	P0.0 (AD0)	39
3	P1.2	P0.1 (AD1)	38
4	P1.3	P0.2 (AD2)	37
5	P1.4	P0.3 (AD3)	36
6	P1.5	P0.4 (AD4)	35
7	P1.6	P0.5 (AD5)	34
8	P1.7	P0.6 (AD6)	33
9	RST	P0.7 (AD7)	32
10	P3.0 (RXD)	EA / VPP	31
11	P3.1 (TXD)	ALE / PRG	30
12	P3.2 (INT0)	PSEN	29
13	P3.3 (INT1)	P2.7 (A15)	28
14	P3.4 (T0)	P2.6 (A14)	27
15	P3.5 (T1)	P2.5 (A13)	26
16	P3.6 (WR)	P2.4 (A12)	25
17	P3.7 (RD)	P2.3 (A11)	24
18	XTAL 2	P2.2 (A10)	23
19	XTAL 1	P2.1 (A9)	22
20	GND	P2.0 (A8)	21

Gambar 2.6 Susunan Kaki AT89S51

(Sumber: AT89S51 Data Sheets, 1997: 4-29)

Adapun fungsi dari masing-masing pin pada AT89S51 adalah:

- Port* 1 (Pin 1..8), berfungsi sebagai *port I/O* biasa.
- Pin 9 (RST), pulsa transisi dari rendah ke tinggi yang diumpankan ke pin RST akan mereset AT89S51. Pin ini dihubungkan dengan rangkaian *power on reset*.

- ❑ *Port 3* (Pin 10..17), port paralel 8 bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TXD (*Transmit Data*), RXD (*Receive Data*), INT0 (*Interrupt 0*), INT1 (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), WR (*Write*), RD (*Read*). Apabila fungsi pengganti tidak digunakan, pin-pin ini dapat digunakan sebagai port *I/O* biasa.
- ❑ Pin 18 (XTAL1), merupakan pin masukan ke rangkaian osilator internal. Osilator kristal dan sumber osilator luar dapat digunakan.
- ❑ Pin 19 (XTAL2), merupakan pin masukan ke rangkaian osilator internal. Pin ini dipakai bila menggunakan osilator kristal.
- ❑ Pin 20 (*Ground*), dihubungkan ke VSS atau *ground*.
- ❑ *Port 2* (Pin 21..28), port paralel 8 bit dua arah, dapat digunakan sebagai port *I/O* 8 bit biasa dan digunakan untuk mengirim *byte* alamat bila digunakan untuk mengakses memori eksternal.
- ❑ Pin 29 (*PSEN/Program Store Enable*), merupakan pengontrol yang digunakan untuk mengakses program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi.
- ❑ Pin 30 (ALE), digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- ❑ Pin 31 (EA), bila pin diberikan logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari ROM/EPROM. Bila diberikan logika rendah, mikrokontroler akan melaksanakan instruksi dari memori program luar.
- ❑ *Port 0* (Pin 32..39), merupakan *port* paralel 8 bit *open drain* dua arah. *Port 0* dapat digunakan sebagai *port I/O* biasa dan dapat juga digunakan untuk memultiplek alamat dengan data pada waktu mengakses memori *eksternal*.
- ❑ Pin 40 (VCC), dihubungkan ke VCC (+5 volt).

2.3.2. Struktur dan Operasi *Port*

Mikrokontroler AT89S51 memiliki 4 buah *port*. Setiap *port* memiliki 8 buah jalur *I/O* yang bersifat bidirectional. Beberapa karakteristik *port* mikrokontroler AT89S51 dijelaskan secara singkat berikut ini:

- ❑ *Port 0* dapat digunakan sebagai *port I/O* biasa. Pada fungsi sebagai *port 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.

- ❑ *Port 1* merupakan *port I/O* biasa. *Port* ini mempunyai internal *pull-up* dan berfungsi sebagai *output* dengan memberikan logika 1. Sebagai *output port* ini dapat memberikan *output sink* keempat buah *input TTL*. *Port 1* memiliki kemampuan untuk menyangga beban sampai 20 mA dan dapat mengemudikan *LED* secara langsung.
- ❑ *Port 2* merupakan *port I/O* biasa, *port* ini mempunyai internal *pull-up* dan berfungsi sebagai *output* dengan memberikan logika 1. Sebagai *output port* ini dapat memberikan *output sink* ke empat buah *TTL*.
- ❑ *Port 3*, sebagai *I/O* biasa *port 3* mempunyai sifat yang sama dengan *port 1* maupun *port 2*. sedangkan sebagai fungsi khusus *port-port* ini mempunyai penjelasan seperti yang terlihat dalam Tabel 2.3.

Tabel 2.3 Fungsi Khusus Port 3

Jalur	Fungsi Alternatif
P3.0	<i>Input</i> data serial RXD
P3.1	<i>Output</i> data serial TXD
P3.2	<i>Input</i> Interupsi Eksternal INT0
P3.3	<i>Input</i> Interupsi Eksternal INT1
P3.4	<i>Input</i> Counter T0
P3.5	<i>Input</i> Counter T1
P3.6	<i>Eksternal Data Memory Write Strobe</i>
P3.7	<i>Eksternal Data Memory Read Strobe</i>

(Sumber : Atmel Corp, 2000)

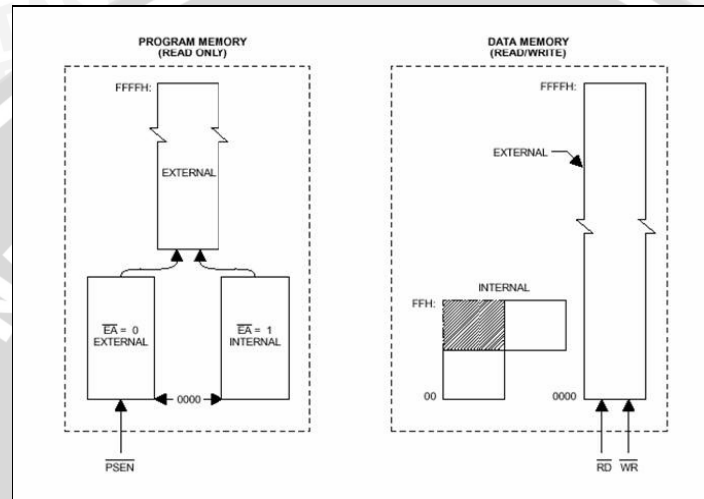
2.3.3. Organisasi Memori

Mikrokontroler MCS-51 memiliki pembagian ruang alamat untuk program dan data. Memori data diakses oleh alamat 8 bit, tetapi alamat data 16 bit juga dapat dihasilkan mikrokontroler melalui register DPTR (*Data Pointer Register*). Alamat data dan program yang bisa dialamati oleh mikrokontroler adalah sebesar 64 *kilobyte* yaitu dari alamat 0000H-FFFFH.

PSEN adalah sinyal yang digunakan untuk pembacaan memori program eksternal. Mikrokontroler MCS-51 mempunyai dua buah alternatif untuk pembacaan memori program yaitu internal dan eksternal. Pembacaan memori program eksternal dengan menset pin *EA* pada logika 0 dan pembacaan memori program internal pin *EA* diset pada logika 1.

AT89S51 memiliki *RAM* internal 128 *byte* (00H-7FH) yang dapat digunakan untuk menampung data-data yang diperlukan dalam pemrograman. *RAM* internal tersebut dapat diklasifikasikan sebagai berikut : 80 *byte general purpose* (30H-7FH), 32 *byte* (00H – 1FH) sebagai *register bank* yang dapat dimanfaatkan seperti *RAM* biasa, dan 16 *byte* (20H-2FH) *bit adresable*.

Gambar 2.7 memperlihatkan struktur memori MCS-51. *ROM* internal AT89S51 jenis *flash EEPROM* sebesar 4 *kilobyte* dapat diprogram ulang sebanyak 1000 kali.



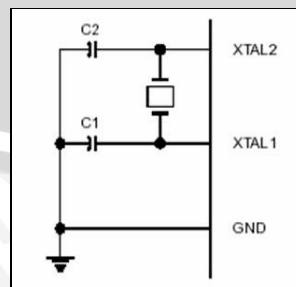
Gambar 2.7 Struktur Memori MCS-51

(Sumber : AT89S51 Architectural Overview, 1997:2-4)

2.3.4. Osilator

Mikrokontroler AT89S51 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber pewaktuan (*clock*) bagi CPU.

Untuk menggunakan osilator internal diperlukan sebuah kristal atau resonator keramik antara pin XTAL1 dan pin XTAL2 dan sebuah kapasitor ke *ground*. Untuk kristalnya dapat digunakan kristal dengan frekuensi sampai dengan 24 MHz. Gambar 2.8 menunjukkan rangkaian osilator yang digunakan.

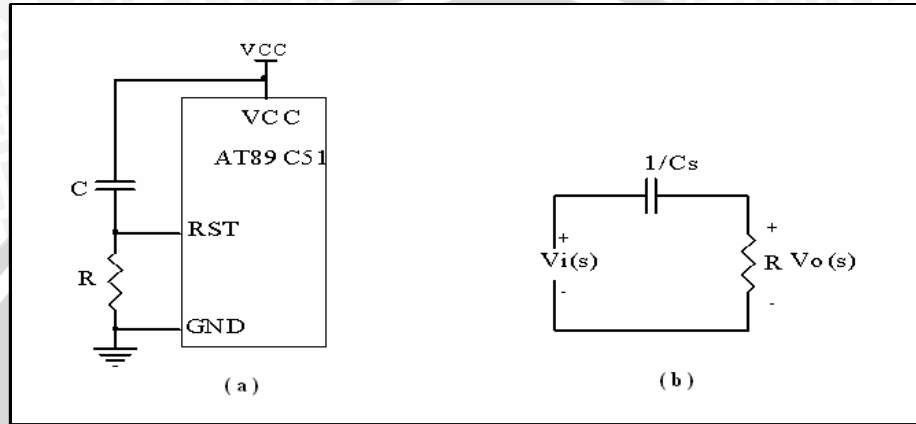


Gambar 2.8. Rangkaian Osilator

Sumber : AT89S51 Data Sheets, 1997: 4-32

2.3.5. Reset

Rangkaian *reset* dibutuhkan untuk me-*reset* mikrokontroler pada saat *power on*. Tegangan berlogika tinggi selama 2 siklus mesin dibutuhkan untuk me-*reset* MCU pada saat dihidupkan. Rangkaian *reset* terdiri dari resistor dan kapasitor yang dihubungkan seperti dalam Gambar 2.9.



Gambar 2.9 (a) Rangkaian *Power On Reset*
 (b) Rangkaian Ekuivalen *Power On Reset*
 (Sumber : Atmel, 1997: 2-63)

Dalam rangkaian ekuivalen seperti ditunjukkan dalam Gambar 2.9 (b) maka didapatkan :

$$V_o(s) = \frac{R}{R + 1/C_s} \cdot V_i(s) = \frac{RC_s}{RC_s + 1} \cdot V_i(s)$$

dengan tegangan V_i adalah tegangan V_{CC} yaitu 5V, dimana dalam fungsi Laplace adalah $\frac{5}{s}$ sehingga :

$$V_o = \frac{RC_s}{RC_s + 1} \cdot \frac{5}{s} = \frac{RC}{RC_s + 1} \cdot 5 = 5 \cdot \left(\frac{1}{s + 1/RC} \right)$$

$$V_o(t) = 5 \cdot e^{-t/RC}, \text{ sehingga } \ln \frac{5}{V_o(t)} = \frac{t}{RC}, \text{ sehingga } t = R \cdot C \left(\ln \frac{5}{V_o(t)} \right)$$

dengan nilai V_o adalah tegangan logika nominal yang diijinkan oleh pin RST (Atmel, 1997: 4-37), dimana $V_o = 0,7 \times V_{CC} = 0,7 \times 5 \text{ volt} = 3,5 \text{ volt}$, maka :

$$t = R \cdot C \frac{\left(\log \frac{5}{V_o} \right)}{\log e} = R \cdot C \frac{\left(\log \frac{5}{3,5} \right)}{\log e}, \text{ maka } t = 0,357 R \cdot C$$

2.3.6. Register Fungsi Khusus

Register fungsi khusus (*Special Function Register*, SFR) terletak pada 128 *byte* bagian atas memori data internal. Wilayah SFR ini terletak pada alamat 80_H sampai FF_H. Register-register ini hanya dapat diakses dengan pengalamatan langsung, baik per bit maupun per *byte*.

SFR berisi register-register dengan fungsi tertentu. Masing-masing register seperti ditunjukkan dalam Tabel 2.4 yang meliputi simbol, nama, dan alamatnya serta keadaannya dalam nilai biner pada saat terjadi *power-on reset*

Beberapa kegunaan register fungsi khusus, yang penting dijelaskan sebagai berikut:

- ❑ *Accumulator* (ACC); merupakan register untuk penambahan dan pengurangan. Perintah *mnemonic* untuk mengakses akumulator disederhanakan sebagai register A.
- ❑ Register B; merupakan register yang berfungsi untuk melayani operasi perkalian dan pembagian.
- ❑ PSW; terdiri dari beberapa bit status yang menggambarkan kejadian di akumulator sebelumnya, yang pengkondisian keadaan akumulator tersebut melalui *flag register* yang terdiri dari *carry flag*, *auxiliary carry flag*, *parity flag*, *overflow flag*, dua bit pemilih *bank*, dan dua *flag* yang dapat didefinisikan sendiri oleh pemakai.
- ❑ *Control Register*: terdiri atas register yang mempunyai fungsi kontrol. Terdapat dua register khusus untuk mengontrol sistem interupsi, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat register khusus, yaitu register TMOD (*Timer/counter Mode Control*) dan register TCON (*Timer/counter Control*).

Tabel 2.4 Register Fungsi Khusus

Simbol	Nama	Alamat
ACC	Akumulator	E0H
B	Register B	F0H
PSW	Program Status Word	D0H
SP	Stack Pointer	81H
DPL	Data Pointer byte rendah	82H
DPH	Data Pointer byte tinggi	83H

P0	Port 0	80H
P1	Port 1	90H
P2	Port 2	A0H
P3	Port 3	B0H
IP	Interrupt Priority Control	B8H
IE	Interrupt Enable Control	A8H
TMOD	Timer/counter Mode Control	89H
TCON	Timer/counter Control	88H
TH0	Timer/counter 0 high byte	8CH
TL0	Timer/counter 0 low byte	8AH
TH1	Timer/counter 1 high byte	8DH
TL1	Timer/counter 0 low byte	8BH
SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

Sumber : Malik, 1997: 19-20

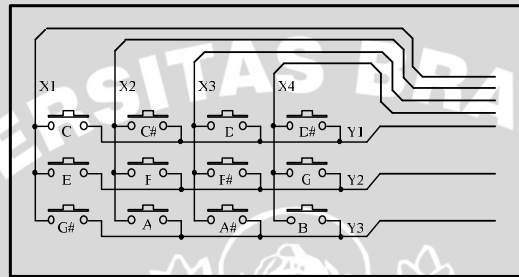
- ❑ *Stack Pointer (SP)*; merupakan register 8 bit yang diletakkan di alamat manapun pada RAM internal. Isi regiter ini ditambah sebelum data disimpan, selama instruksi *PUSH* dan *CALL*. Pada saat reset, register SP diinisialisasi pada alamat 07_H, sehingga *stack* akan dimulai pada lokasi 08_H.
- ❑ *Data Pointer (DPTR)* terdiri dari dua register, yaitu untuk byte tinggi (*Data Pointer High, DPH*) dan untuk byte rendah (*Data Pointer Low, DPL*). Fungsinya dimaksudkan untuk menahan alamat 16-bit. DPTR dapat dimanipulasi sebagai register 16-bit atau sebagai dua buah register 8-bit.
- ❑ Port 0 sampai Port 3 merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, dan 3. Masing-masing register ini dapat dialamati *per-bit* maupun *per-byte*.

2.4. Keypad

Matrik *keypad* sering digunakan pada rangkaian yang memerlukan tombol yang banyak. *Keypad* adalah sarana untuk memasukkan data. Misalnya, matriks 4x3 berarti

ada 12 kali kemungkinan keluaran. Banyaknya kemungkinan tersebut sesuai dengan banyaknya susunan antara 4 kolom dan 3 baris.

Pada rangkaian *keypad* ini difungsikan untuk memberi masukan data melalui 12 tombol yang tersedia pada papan *keypad* tersebut. Tombol-tombol ini mewakili dari 12 akord gitar dasar. *Keypad* ini berfungsi untuk menterjemahkan penekanan pada salah satu tombol ke dalam bentuk biner. Berikut ini rangkaian matriks keypad 4x3, yang terdiri dari 4 kolom (X1, X2, X3, X4) dan 3 baris (Y1, Y2, Y3).



Gambar 2.10 Rangkaian Keypad

IC 74C922 merupakan sebuah encoder CMOS yang memberikan logika keluaran 4 bit dari keypad. IC encoder 16 ke 4 ini dapat diimplementasikan dengan sebuah external clock atau external kapasitor. Encoder ini sudah memiliki *chip pull up* yang diijinkan sampai dengan 50 KΩ. DA (Data Available output) akan berlogika *high* saat terdapat data keluaran dari keypad yang valid (keypad ditekan), Data Available akan kembali berlogika *low* saat keypad dilepas. Sirkuit *debouncing* internal hanya membutuhkan sebuah kapasitor external yang terhubung di pin KM (*Keybounce Mask*).

Tabel 2.5. adalah tabel kebenaran IC encoder 74C922.

Tabel 2.5. Tabel Kebenaran IC 74C922

Switch positior	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Y ⁻ X ⁻	Y ⁻ X ²	Y ⁻ X ³	Y ⁻ X ⁴	Y ² X ⁻	Y ² X ²	Y ² X ³	Y ² X ⁴	Y ³ X ⁻	Y ³ X ²	Y ³ X ³	Y ³ X ⁴	Y ⁴ X ⁻	Y ⁴ X ²	Y ⁴ X ³	Y ⁴ X ⁴
D																
A	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-
B	0	0	-	-	0	0	-	-	0	0	-	-	0	0	-	-
C	0	0	0	0	-	-	-	-	0	0	0	0	-	-	-	-
D	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

Sumber : Datasheet Fairchild MM74C922

2.5. LED (Light Emitting Diode)

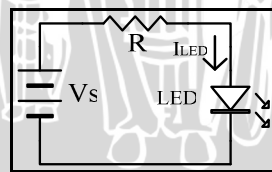
LED (*Light Emitting Diode*) adalah salah satu komponen elektronika yang dapat mengeluarkan cahaya jika dicatu. LED-LED ini mempunyai macam-macam warna, seperti merah, hijau, kuning dan biru.

LED mempunyai penurunan tegangan lazimnya dari 1,5 V sampai 2,5 V untuk arus diantara 10 sampai 150 mA. Penurunan tegangan tergantung dari arus LED, warna, kelonggaran, dan sebagainya. Kalau tak ada hal lain yang ditentukan, gunakanlah tegangan jatuh nominal 2 V pada saat memperbaiki atau menganalisa rangkaian-rangkaian LED. Jika ingin merancang, sebaiknya memperhatikan *datasheet*, karena tegangan LED mempunyai kelonggaran yang besar. Biasanya arus LED ada diantara 10 sampai 50 mA karena daerah ini memberikan cahaya yang cukup untuk banyak pemakaian.

Kecemerlangan LED tergantung dari arusnya. Idealnya, cara terbaik untuk mengendalikan kecemerlangan ialah dengan menjalankan LED dengan sumber arus. Cara berikutnya yang terbaik setelah sumber arus adalah tegangan catu yang besar dan resistansi seri yang besar. Persamaan untuk menentukan Resistor (R) pembatas arus diberikan oleh:

$$R = \frac{V_S - V_{LED}}{I_{LED}}$$

Makin besar tegangan sumber, makin kecil pengaruh V_{LED} . Dengan kata lain V_S yang besar menghilangkan pengaruh perubahan pada tegangan LED. (Malvino, 1999:96-98). Gambar 2.11 menunjukkan lambing skematik LED.



Gambar 2.11 Lambang Skematik LED

2.6. Seven Segment

Seven segment adalah gabungan dari beberapa LED (*Light Emitting Diode*) yang dapat dicatu secara individu atau masing - masing. Warna yang ditampilkan pada *Seven segment* ini umumnya adalah warna merah. LED-LED tersebut tersusun menyerupai angka delapan, jika dicatu semuanya maka akan menampilkan angka delapan. Seperti namanya, ada tujuh segmen LED yang dapat diatur kombinasinya untuk menampilkan angka dan huruf.

Ada dua macam hubungan pada *seven segment*, yaitu :

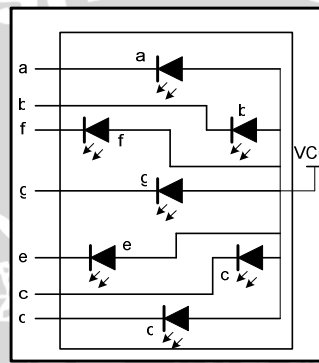
1. *Common anoda*

Pada hubungan ini semua *anoda* dihubungkan menjadi satu ke positif catu daya dan katodanya difungsikan sebagai masukan.

2. *Common katoda*

Pada hubungan ini semua *katoda* dihubungkan menjadi satu ke *ground* dan anodanya sebagai masukan.

Dalam penggunaan *seven segmen* ini, harus dipasang resistor seri terhadap setiap LED untuk membatasi arus yang mengalir.



Gambar 2. 12 Tampilan *seven segment Common Anode*

BAB III

METODOLOGI PENELITIAN

Dalam perancangan dan pembuatan Alat Penampil Akord Gitar ini memakai metode sebagai berikut:

1. Studi kepustakaan.
2. Penentuan spesifikasi alat.
3. Perancangan alat.
4. Pengujian rangkaian pada *project board*.
5. Perancangan perangkat lunak.
6. Pengujian alat.
7. Penyusunan buku laporan tugas akhir.

3.1. Studi Kepustakaan

Pertama kali adalah studi literatur atau studi pustaka. Bahan pustaka yang dibutuhkan dikumpulkan selengkap-lengkapnyanya. Bahan pustaka yang dibutuhkan adalah yang membahas mengenai akord, LED, *seven segment*, transistor, mikrokontroler dan perancangannya baik perangkat keras maupun perangkat lunak khususnya untuk AT89S51, aplikasi untuk IC yang berfungsi menghasilkan suara yaitu ISD2590, dan bahan pustaka penunjang lainnya.

3.2. Perancangan Alat

Berdasarkan studi literatur tahap selanjutnya yaitu perencanaan alat. Perancangan alat ini disesuaikan dengan fungsi dari komponen-komponen yang akan digunakan sesuai dengan literatur yang dipelajari sehingga terbentuk rangkaian elektronik yang siap direalisasikan.

3.3. Pengujian Rangkaian Pada *Project Board*

Sebelum direalisasikan dalam bentuk PCB, rangkaian yang telah dirancang secara matematis perlu dicoba terlebih dahulu dengan *project board*. Pengujian dapat diketahui dengan multimeter lewat *point-point* (titik-titik) keluaran yang dapat dideteksi. Jika terjadi kesalahan atau perubahan jenis dan nilai komponen dapat dilakukan dengan mudah.

3.4. Perancangan Perangkat Lunak

Setelah perangkat keras dirancang sesuai dengan perencanaan, maka langkah selanjutnya adalah perancangan perangkat lunak. Perangkat lunak ini difungsikan untuk mengatur keseluruhan sistem yang terdiri atas beberapa perangkat keras sehingga sistem ini dapat bekerja dengan baik. Pembuatan dimulai dari pembuatan *Lay Out* PCB, pemasangan komponen dan pembuatan perangkat lunak yang mendukung sistem.

3.5. Pengujian Alat

Pengujian alat terdiri atas dua bagian, yaitu pengujian rangkaian secara per blok dan secara rangkaian keseluruhan.

1. Pengujian tiap blok rangkaian dilaksanakan dengan menguji tiap blok alat untuk mengetahui apakah blok-blok tersebut telah memenuhi spesifikasi yang telah ditentukan.
2. Pengujian rangkaian keseluruhan dilaksanakan dengan membuat modul sistem yang bekerja sesuai kondisi yang sebenarnya, sehingga dapat diketahui ketepatan dari kerja alat yang dirancang.

3.6. Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh sesuai dengan spesifikasi yang ditentukan saat dilakukan perancangan, maka alat telah memenuhi harapan.

3.7. Penyusunan Buku Laporan Tugas Akhir

Penyusunan buku laporan dibuat dengan mengikuti sistematika pembahasan yang telah dibuat.

BAB IV PERENCANAAN ALAT

Dalam bab ini menjelaskan mengenai spesifikasi alat, perancangan perangkat keras dari Alat Penampil Akord Gitar, yang meliputi diagram blok rangkaian, dan cara kerja rangkaian. Selain perancangan perangkat keras akan dijelaskan juga mengenai perancangan perangkat lunak.

Dalam perancangan alat, pemilihan komponen ditekankan pada komponen yang mudah didapatkan dipasaran sehingga diperoleh kemudahan dalam pembuatan dan apabila terjadi kerusakan dapat mudah diganti.

4.1. Perencanaan Sistem

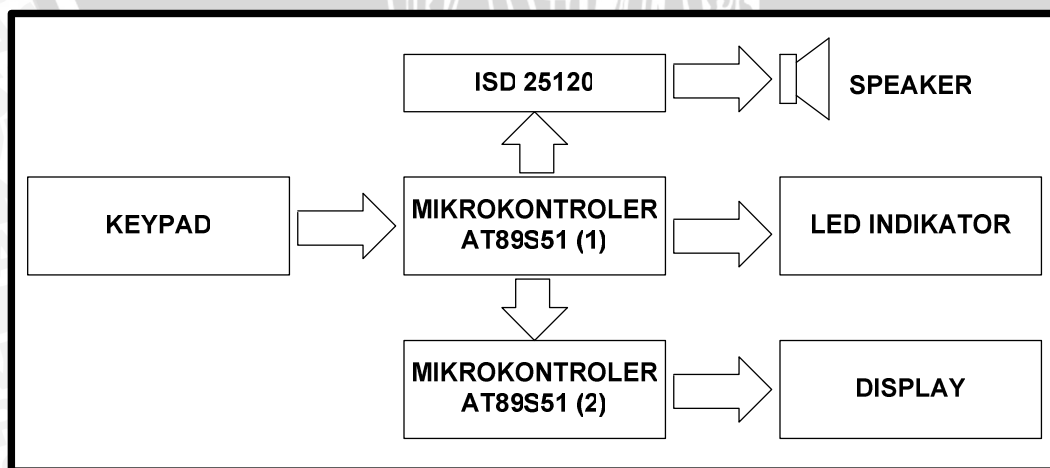
4.1.1. Spesifikasi Alat

Perancangan alat ini mempunyai beberapa spesifikasi diantaranya:

1. Secara umum Alat Penampil Akord Gitar ini menggunakan LED sebagai indikator untuk variasi akord, *seven segment* dan LED sebagai *display*, mikrokontroler AT89S51 dan ISD25120
2. Akord gitar yang dapat ditampilkan sebanyak 108 buah
3. Keluaran bunyi dari alat untuk mengetahui nada dari akord tersebut
4. Alat ini menggunakan catu daya DC 5 volt

4.1.2. Blok Diagram Alat

Blok diagram Alat Penampil Akord Gitar ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Blok Diagram Keseluruhan Sistem

4.1.3. Prinsip Kerja Alat

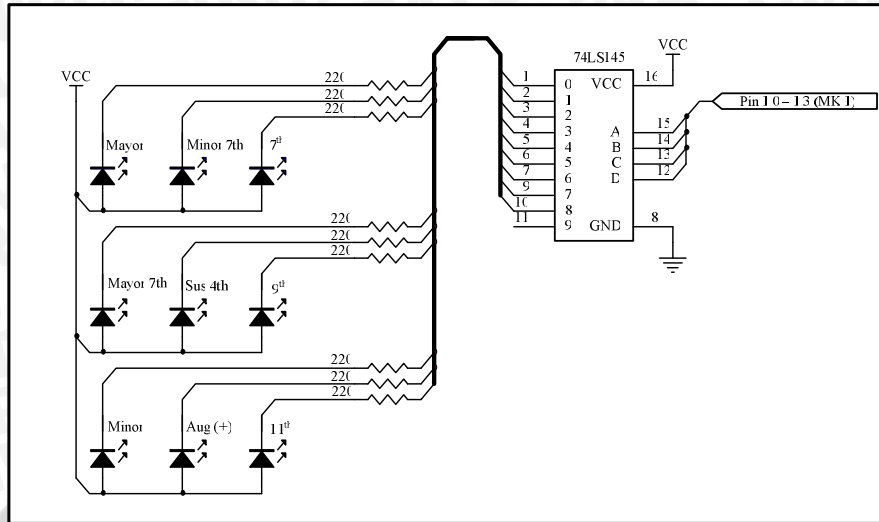
Pada dasarnya fungsi alat ini adalah untuk mempermudah para pemula untuk mengetahui letak jari-jari dari akord yang diinginkan. Sebelum sistem ini dijalankan pertama kali akord yang ingin ditampilkan dipilih melalui 12 keypad yang tersedia. Sistem kerja alat ini diawali dari keypad yang ditekan akan memberikan masukan ke mikrokontroler, mikrokontroler yang digunakan dalam perancangan ini adalah dua buah mikrokontroler AT89S51. Semua proses pengolahan data dilakukan oleh mikrokontroler AT89S51. Data yang telah diolah oleh mikrokontroler tersebut akan ditampilkan melalui display berupa *seven segment* yang menunjukkan angka, angka-angka disini berfungsi untuk menunjukkan jari yang digunakan menekan dan nyala LED-LED untuk posisi senar gitar yang harus ditekan, kemudian ada 9 LED Indikator yang akan menyala untuk menunjukkan ada 9 variasi akord gitar. Selain dari itu tombol nada untuk menampilkan bunyi dari akord tersebut. Untuk menyimpan suara gitar digunakan ISD25120 sebagai IC penyimpan suara yang akan menampilkan bunyi dari akord gitar. ISD25120 mempunyai kemampuan penyimpanan suara dengan durasi 120 detik yang dioperasikan dalam *mode address* bit artinya setiap kata yang direkam mempunyai *address* sendiri.

4.2. Perencanaan Perangkat Keras

4.2.1. Rangkaian LED Indikator

LED Indikator berfungsi untuk menampilkan variasi dari akord gitar yang sedang ditampilkan. Ada 9 buah LED yang disusun dalam bentuk matriks 3x3, LED ini akan menyala sesuai dengan variasi yang ditampilkan.

Untuk menghemat pin pada mikrokontroler maka digunakan IC pendekode BCD ke desimal 74LS145 untuk menyalakan LED indikator. IC74LS145 ini memiliki empat saluran masukan dan sepuluh saluran keluaran yang dapat dikontrol oleh mikrokontroler. Katoda dari setiap LED dihubungkan dengan pin-pin keluaran dari IC74145 dan bagian anoda dihubungkan dengan Vcc (+5V). Input masukan dari IC74145 dihubungkan dengan P1.0-1.3 mikrokontroler 1. Jika mikrokontroler mengirim data ke IC74145, maka pin-pin keluaran dari IC74LS145 ini akan mengeluarkan kondisi tertentu di tiap pin, jika pin tersebut mengeluarkan sinyal “*low*”, maka LED yang bersangkutan akan menyala, sebaliknya jika mengirimkan sinyal “*high*” maka LED tidak akan menyala. Rangkaian untuk LED Indikator dalam perancangan ini dapat dilihat dalam Gambar 4.2



Gambar 4.2 Rangkaian LED Indikator

Dalam Gambar 4.2 penggunaan *pin* dapat dijelaskan secara detail sebagai berikut:

1. Pin *Input* A, B, C, D
Dihubungkan ke P1.0-1.3 mikrokontroler 1 digunakan sebagai *input*
2. Pin *output* 0-8
Dihubungkan ke katoda tiap LED untuk *output* variasi dari akord gitar
3. Pin VCC dan GND dihubungkan ke catu positif dan negatif.

LED sendiri mempunyai karakteristik, $V_{led} = 1,7\text{Volt}$ dan $I_{LED} = 15\text{mA}$. Resistor yang diberikan berdasarkan tegangan pada LED dan batas arus maksimum LED. Dari rangkaian diatas dengan menggunakan Hukum Tegangan Kirchoff didapatkan rumus:

$$V_{CC} = V_R + V_{led}$$

$$V_R = 5 \text{ Volt} - 1.7 \text{ Volt} = 3,3 \text{ Volt}$$

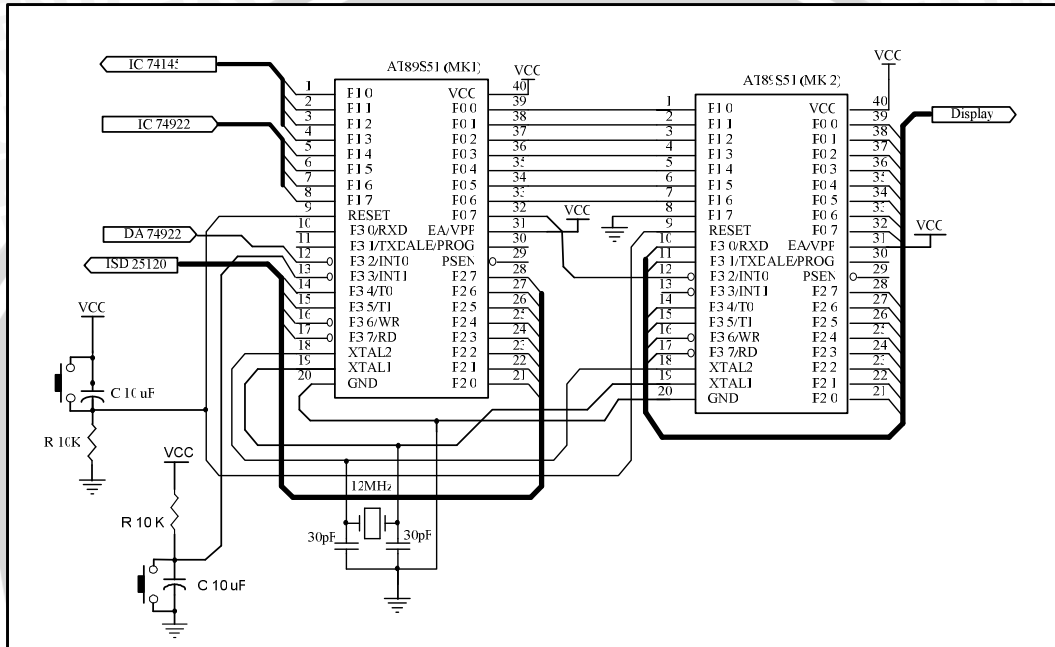
$$R = \frac{V_R}{I_{led}}$$

$$R = \frac{3,3}{15.10^{-3}} = 220 \Omega$$

Sehingga resistor yang digunakan untuk membatasi arus pada LED sebesar 220Ω.

4.2.2. Rangkaian Mikrokontroler AT89S51

Pada rangkaian ini komponen utamanya adalah unit mikrokontroler tipe AT89S51 kompatibel dengan keluarga MCS-51. Komponen ini merupakan sebuah *chip* tunggal sebagai pengolah data. Pemilihan mikrokontroler jenis ini karena mudah diperoleh di pasaran. Pin-pin pada AT89S51 dihubungkan pada rangkaian pendukung untuk membentuk suatu sistem minimum. Rangkaian mikrokontroler ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangkaian Mikrokontroler AT89S51

Dalam Gambar 4.3 penggunaan *port* dapat dijelaskan secara detail sebagai berikut:

1. *Port 0* IC1:
 P0.0 – 0.6 dihubungkan dengan P1.0 - 1.6 dari IC2 untuk komunikasi paralel
 P0.7 digunakan sebagai pengontrol jalur komunikasi kedua mikrokontroler
2. *Port 1* IC1:
 P1.0 – P1.3 digunakan sebagai saluran *output* ke IC 74LS145
 P1.4 -1.7 digunakan sebagai saluran *input* dari IC 74C922
3. *Port 2* IC1:
 P2.0 – P2.7 digunakan sebagai *output address bus* alamat A0 - A7 pada ISD25120

4. *Port 3 IC1:*
 - P3.2 digunakan sebagai *interrupt* untuk proses pengontrolan IC74C922
 - P3.3 digunakan sebagai *interrupt* untuk tombol nada
 - P3.4 – 3.5 digunakan sebagai *output address bus* alamat A8 – A9 pada ISD25120
 - P3.6 – 3.7 dihubungkan dengan PD dan CE pada ISD25120
5. *Port 0 IC2:*
 - P0.0 – 0.6 digunakan sebagai *output* ke *seven segment*
 - P0.7 digunakan sebagai *output* LED display
6. *Port 1 IC2:*
 - P1.0 – 1.6 dihubungkan dengan P0.0 - 0.6 dari IC1 untuk komunikasi paralel
 - P1.7 dihubungkan dengan *ground*
7. *Port 2 IC2:*
 - P2.0 – 2.7 digunakan sebagai *output* LED display
8. *Port 3 IC2:*
 - P3.0 – 3.1 digunakan sebagai *output* ke Transistor BC557
 - P3.2 digunakan sebagai *interrupt* untuk proses pengontrolan komunikasi data kedua mikrokontroler
9. XTAL1 dan XTAL2
Digunakan sebagai *input* dari rangkaian osilator kristal. Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator *internal* yang telah tersedia dalam *chip* AT89S51. Untuk menentukan frekuensi osilatornya cukup dengan menghubungkan kristal dalam pin 19 (X_1) dan pin 18 (X_2) serta dua buah kapasitor ke *ground*.

Pada aplikasinya nanti, mikrokontroler ini mempunyai fungsi utama menghasilkan satu *interval* waktu dengan durasi tepat 1 detik dengan menggunakan fungsi *timer*. Hasil dari fungsi *timer* ini ditentukan sepenuhnya oleh waktu yang diperlukan mikrokontroler untuk satu siklus mesin yaitu 12 kali periode kristal. Karena itu dipilih kristal dengan frekuensi kelipatan 12 agar didapatkan periode satu siklus mesin yang bulat.

Pada perancangan ini kristal digunakan bersama-sama oleh kedua mikrokontroler, dipilih kristal 12 MHz sehingga waktu untuk satu siklus mesin adalah $12/12 \cdot 10^6$ atau sama dengan 1 μ s.

Besarnya kapasitansi C3 dan C4 disesuaikan dengan spesifikasi dalam lembar data AT89S51 yaitu 30 ± 10 pF. Agar dapat menepatkan kerja kristal pada frekuensi tepat 12MHz, maka pada pin XTAL1 kapasitor dibuat variabel keluar dari batas toleransinya. Hal ini untuk mengatasi nilai toleransi dari kristal yang digunakan. Nilai dari masing-masing kapasitor pembentuk osilator adalah $C3 = 30$ pF, $C4 = 30$ pF.

10. Reset

Reset digunakan untuk me-reset mikrokontroler AT89S51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset kapasitor dihubungkan dengan V_{CC} dan sebuah resistor yang dihubungkan ke ground.

Karena kristal yang digunakan mempunyai frekuensi sebesar 12 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}}$$

$$T = \frac{1}{12 \text{ MHz}} \text{ s} = 8,33 \times 10^{-8} \text{ s}$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk me-reset mikrokontroler adalah :

$$\begin{aligned} t_{\text{reset}(\text{min})} &= T \times \text{periode yang dibutuhkan} \\ &= 8,33 \times 10^{-8} \times 24 = 2 \mu\text{s} \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2 μs untuk me-reset. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C.

$$t = RC \ln \frac{5}{V_0(t)}$$

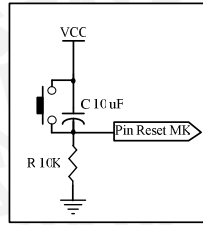
Dengan nilai V_0 adalah tegangan logika nominal yang diperlukan oleh pin RST (Atmel, 1997:4-37), dimana:

$$V_0 = 0,7 \times V_{CC} = 0,7 \times 5 \text{ volt} = 3,5 \text{ volt}$$

$$t = RC \ln \frac{5}{3,5}$$

$$t = 0,357 RC = 0,357 \times 10000 \Omega \times 10 \cdot 10^{-6} = 35,7 \text{ ms}$$

Jadi dengan nilai komponen $R = 10 \text{ k}\Omega$ dan nilai $C = 10 \mu\text{F}$ sudah dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler. Rangkaian reset ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Rangkaian Reset

11. Tombol Nada

Tombol nada ini berfungsi untuk mengeluarkan nada dari akord yang sedang ditampilkan dalam perancangan ini digunakan satu tombol nada, jika dalam mendengarkan nada dari akord yang ditampilkan kurang jelas, maka tombol ini dapat ditekan berulang-ulang sehingga bunyi dari akord tersebut akan diulang sesuai penekanan tombol nada tadi. Waktu minimal logika rendah yang dibutuhkan untuk *interrupt* mikrokontroler adalah 0.25us.

Jadi waktu minimal inilah yang dipakai sebagai pedoman dalam menentukan besarnya R dan C.

$$t = RC \ln \frac{5}{V_0(t)}$$

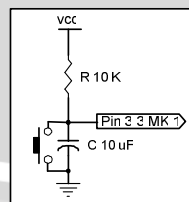
Dengan nilai V_0 adalah tegangan nominal yang diperlukan oleh pin INT1 dimana:

$$V_0 = 0,7 \times V_{CC} = 0,7 \times 5 \text{ volt} = 3,5 \text{ volt}$$

$$t = RC \ln \frac{5}{3,5}$$

$$t = 0,357 R.C = 0,357 \times 10000 \Omega \times 10.10^{-6} = 35.7 \text{ ms}$$

Jadi dengan nilai komponen $R=10 \text{ k}\Omega$ dan $C=10 \mu\text{F}$ sudah dapat memenuhi syarat waktu minimal yang dibutuhkan mikrokontroler. Rangkaian tombol nada ditunjukkan dalam gambar 4.5.

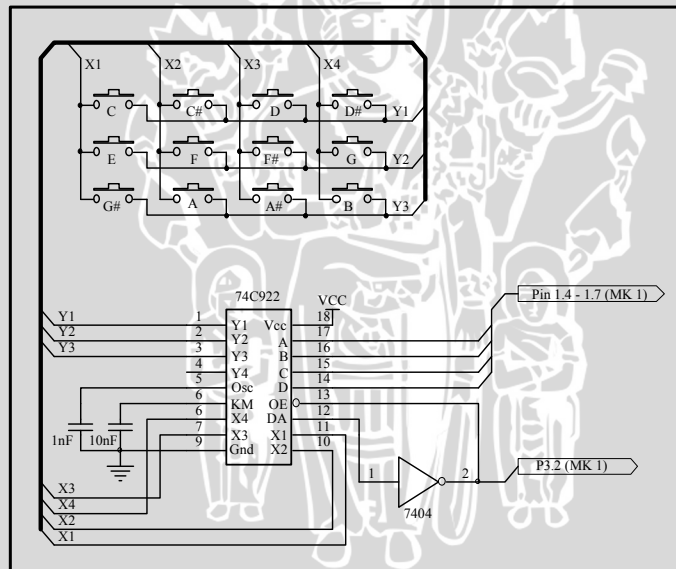


Gambar 4.5 Rangkaian Tombol Nada

4.2.3. Rangkaian Keypad

Pada rangkaian *keypad* ini difungsikan untuk memberi masukan data melalui 12 tombol yang tersedia pada papan *keypad* tersebut. Tombol-tombol ini mewakili dari 12 akord gitar dasar. *Keypad* ini berfungsi untuk menterjemahkan penekanan pada salah satu tombol ke dalam bentuk biner.

Dalam perancangan ini *keypad* dihubungkan ke IC74C922. Masukan dari kolom *keypad* dihubungkan ke pin X1-X4, sedangkan baris dari *keypad* disambungkan ke Y1-Y3 IC74C922. Data yang dihasilkan dikeluarkan dari pin A, B, C, dan D menuju port 1 mikrokontroler 1 (P1.4-P1.7). Pin Osc dan KM dihubungkan dengan dua buah kapasitor dengan perbandingan 1:10 ke *ground*. OE keluar menuju mikrokontroler menunggu DA, sedangkan OE dan DA kondisi logikanya terbalik, sehingga digunakan gerbang inverting 7404 dari DA ke OE. Keluaran dari OE 74C922 menuju pin 3.2 mikrokontroler 1 (INT 0). Rangkaian keypad ditunjukkan dalam gambar 4.6.



Gambar 4.6 Rangkaian Keypad

IC 74C922 merupakan sebuah encoder CMOS yang memberikan logika keluaran 4 bit dari keypad. IC encoder 16 ke 4 ini dapat diimplementasikan dengan sebuah *external clock* atau *external* kapasitor. Encoder ini sudah memiliki *chip pull up* yang diijinkan sampai dengan 50 K Ω . DA (*Data Available output*) akan berlogika *high* saat terdapat data keluaran dari *keypad* yang ditekan, *Data Available* akan kembali berlogika *low* saat *keypad* dilepas. Sirkuit *debouncing* internal hanya membutuhkan

sebuah kapasitor *external* yang terhubung di pin KM (*Keybounce Mask*). Tabel 4.1. adalah tabel kebenaran dari IC Encoder 74C922.

Tabel 4.1. Tabel Kebenaran IC 74C922

Switch positior	0 Y ₁ X ₁	1 Y ₁ X ₂	2 Y ₁ X ₃	3 Y ₁ X ₄	4 Y ₂ X ₁	5 Y ₂ X ₂	6 Y ₂ X ₃	7 Y ₂ X ₄	8 Y ₃ X ₁	9 Y ₃ X ₂	10 Y ₃ X ₃	11 Y ₃ X ₄	12 Y ₄ X ₁	13 Y ₄ X ₂	14 Y ₄ X ₃	15 Y ₄ X ₄
D A T A	A	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
A	B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
O	C	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1
U	D	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Sumber : Datasheet Fairchild MM74C922

4.2.4. Rangkaian *Display*

Rangkaian *display* berfungsi untuk menampilkan variasi dari akord gitar sesuai dengan penekanan tombol yang dilakukan pada rangkaian *keypad*. Rangkaian *display* terdiri dari LED dan *seven segment* yang dihubungkan secara *common anoda*, dimana anoda dari setiap LED dan *seven segment* dihubungkan dengan kolektor dari *switching transistor*. Katoda dari tiap-tiap LED dan *seven segment* dihubungkan dengan *port* keluaran mikrokontroler (*Port 0* dan *Port 2*) melalui resistor 220 Ohm.

Switching transistor dipakai untuk melakukan proses *scanning* dengan cara menghidupkan satu per satu transistor ini dengan cepat. Untuk proses *scanning* transistor yang dipakai disini adalah transistor PNP BC557 yang merupakan transistor untuk keperluan umum (*general purpose transistor*). Basis dari transistor ini dihubungkan dengan pin-pin keluaran mikrokontroler 2 (P3.0 – P3.1 dan P3.4 – P3.7).

Dari data yang diketahui:

$$I_{LED} = 15 \text{ mA}$$

Untuk satu senar, maksimal LED yang menyala adalah 6 buah, jadi:

$$I_C = 6 \times 15\text{mA} = 90\text{mA}$$

Transistor menggunakan BC557 dengan $\beta = 490$

$$V_{BE} = -0.65 \text{ Volt}$$

untuk mendapatkan I_{LED} tersebut dibutuhkan I_B sebesar:

$$I_B = \frac{I_{LED}}{\beta} = \frac{90 \times 10^{-3}}{490} = 184 \mu\text{A}$$

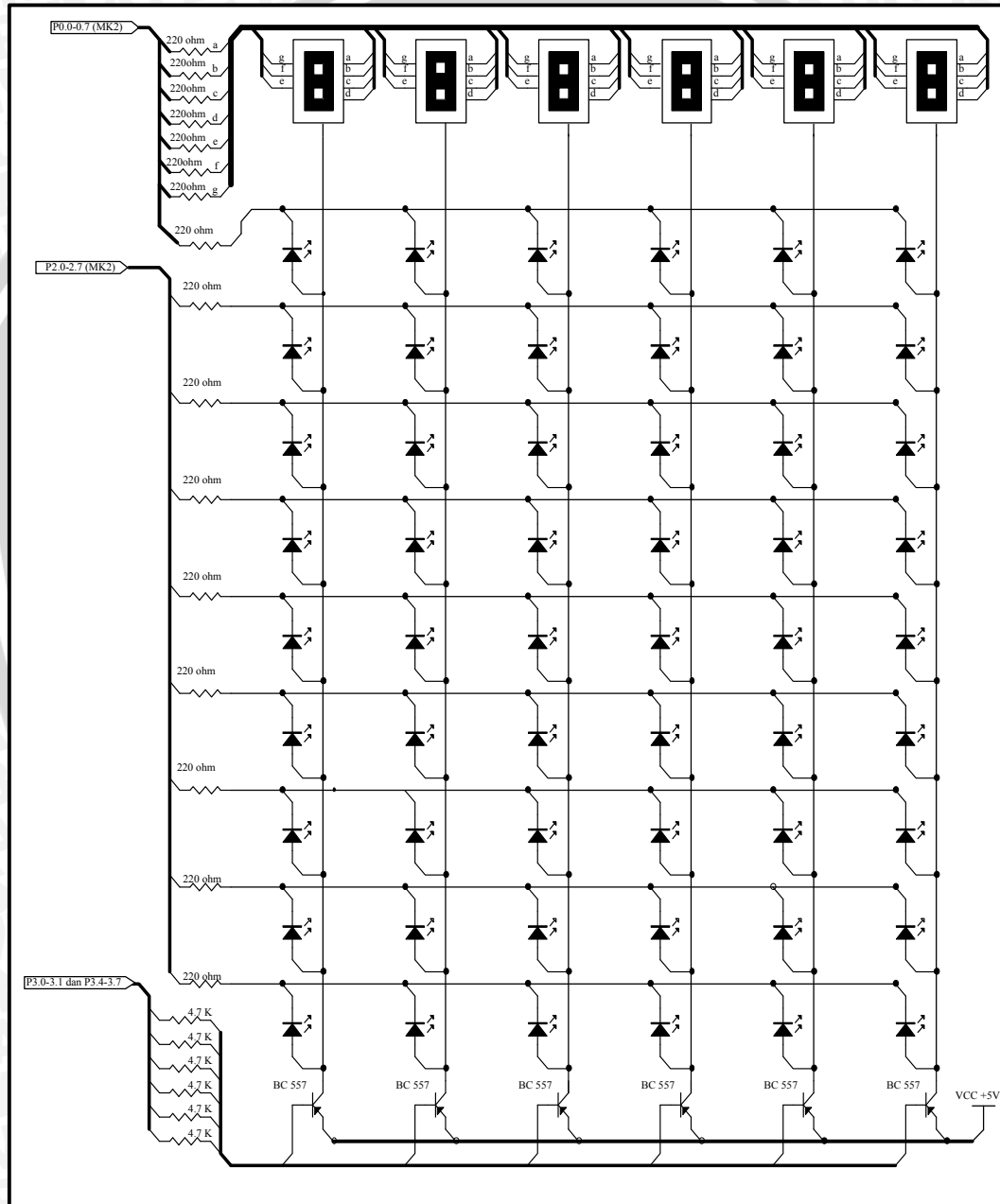
maka,

$$R_B = \frac{V_{EE} + V_{EB}}{I_B}$$

$$= \frac{5 + 0,65}{184 \times 10^{-6}} = 30.7 k\Omega$$

sehingga dalam perancangan dipilih nilai resistor yang mendekati sebesar 30.7 kΩ.

Gambar rangkaian display ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Rangkaian Display

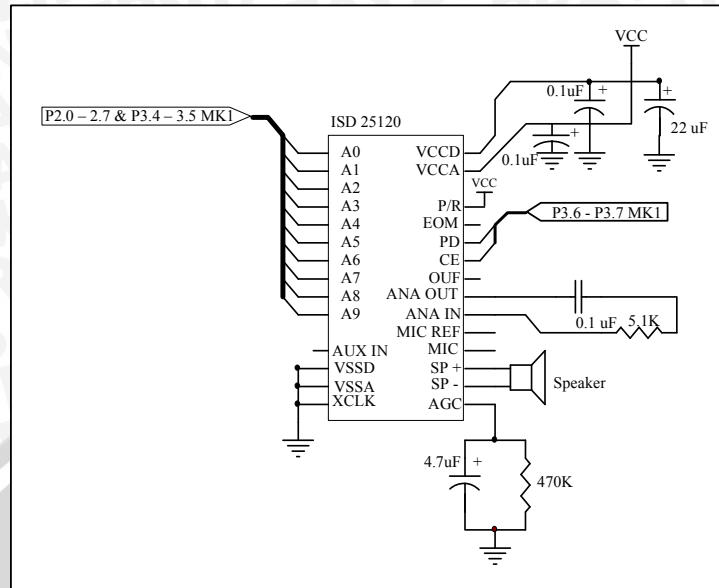
Cara kerja rangkaian *display* ini adalah sebagai berikut:

Pada saat pertama kali mikrokontroler diaktifkan atau pada kondisi sesudah *reset*, semua *port* AT89C51 adalah dalam kondisi *high*. Pada kondisi ini, maka semua *switching* transistor dalam keadaan *off*, sehingga semua LED dan *seven segment* tidak menyala. Apabila kemudian mikrokontroler mengirimkan sinyal *low* pada basis transistor, maka sekarang transistor dalam kondisi *on*. Dalam keadaan ini, nyala tidaknya LED dan *seven segment* tergantung pada sinyal yang dikirimkan pada bagian katoda-nya (*Port 1* dan *Port 3* pada mikrokontroler). Jika sinyal yang dikirimkan pada LED atau *seven segment* tersebut adalah *high* (logika 1), maka LED dan *segment* yang bersangkutan tidak akan menyala. Namun jika sinyal yang dikirimkan adalah *low* (logika 0), maka LED dan *segment* yang bersangkutan akan menyala.

4.2.5. Rangkaian Pemutar Suara

Rangkaian pemutar suara pada perancangan ini berfungsi untuk mengeluarkan bunyi dari akord gitar yang sedang ditampilkan, akord gitar yang dapat ditampilkan oleh alat ini sebanyak 108 macam, karena dengan jumlah sebanyak 108 sudah mencukupi untuk para pemula dalam bermain gitar. Dengan asumsi tiap bunyi akord gitar membutuhkan waktu satu detik, maka IC pemutar suara yang digunakan dengan durasi minimal 108 detik, sedangkan dipasaran tidak terdapat IC yang berdurasi 108 detik. Sedangkan IC penyimpan suara yang dekat dengan 108 detik adalah 120 detik.

Oleh karena itu dalam perancangan ini menggunakan ISD25120. ISD25120 merupakan IC yang kompatibel dengan mikrokontroler AT89S51. Sebelumnya IC ini sudah diisi dengan bunyi dari akord gitar yang diperlukan untuk perancangan dari alat ini.



Gambar 4.8 Rangkaian Pemutar Suara ISD25120

Dalam Gambar 4.8 penggunaan pin dapat dijelaskan secara detail sebagai berikut:

1. Pin A0 - A9 dihubungkan dengan P2.0 - 2.7 dan P3.4 - 3.5 pada mikrokontroler pertama sebagai jalur data.
2. Pin PD dan CE dihubungkan dengan P3.6 - 3.7 pada mikrokontroler pertama sebagai kontrol untuk ISD
3. Pin VCCD dan VCCA dihubungkan ke VCC +5Volt dan Pin VSSA dan VSSD dihubungkan ke *ground* untuk dua input (untuk analog dan untuk digital) agar dapat memperkecil fase yang timbul
4. Pin SP+ dan SP- dihubungkan ke *speaker* untuk mengeluarkan bunyi yang disimpan dalam ISD
5. AGC digunakan untuk menyesuaikan penguatan *output* dari *preamplifier* yang digunakan
6. Pin ANA OUT dan ANA IN dihubungkan melalui sebuah resistor dan kondensator kopling. Nilai kapasitor ini dengan impedansi input ANA IN (normalnya 3 K Ω) akan menentukan frekuensi *cut - off* filter

Sebagaimana diketahui durasi penyimpanan IC ini selama 120 detik, maka suara yang akan disimpan tidak boleh melebihi dari durasi yang diijinkan. ISD25120 memiliki 10 bit jalur alamat dan memiliki lokasi alamat sebanyak 600 bit. Dari *datasheet* diketahui tiap detiknya mempunyai jumlah alamat sebanyak 5 *bit*.

Untuk penyimpanan suara, perdetiknya digunakan jumlah alamat yang dibutuhkan sebanyak 5 bit, total jumlah alamat yang dibutuhkan adalah 496 bit.

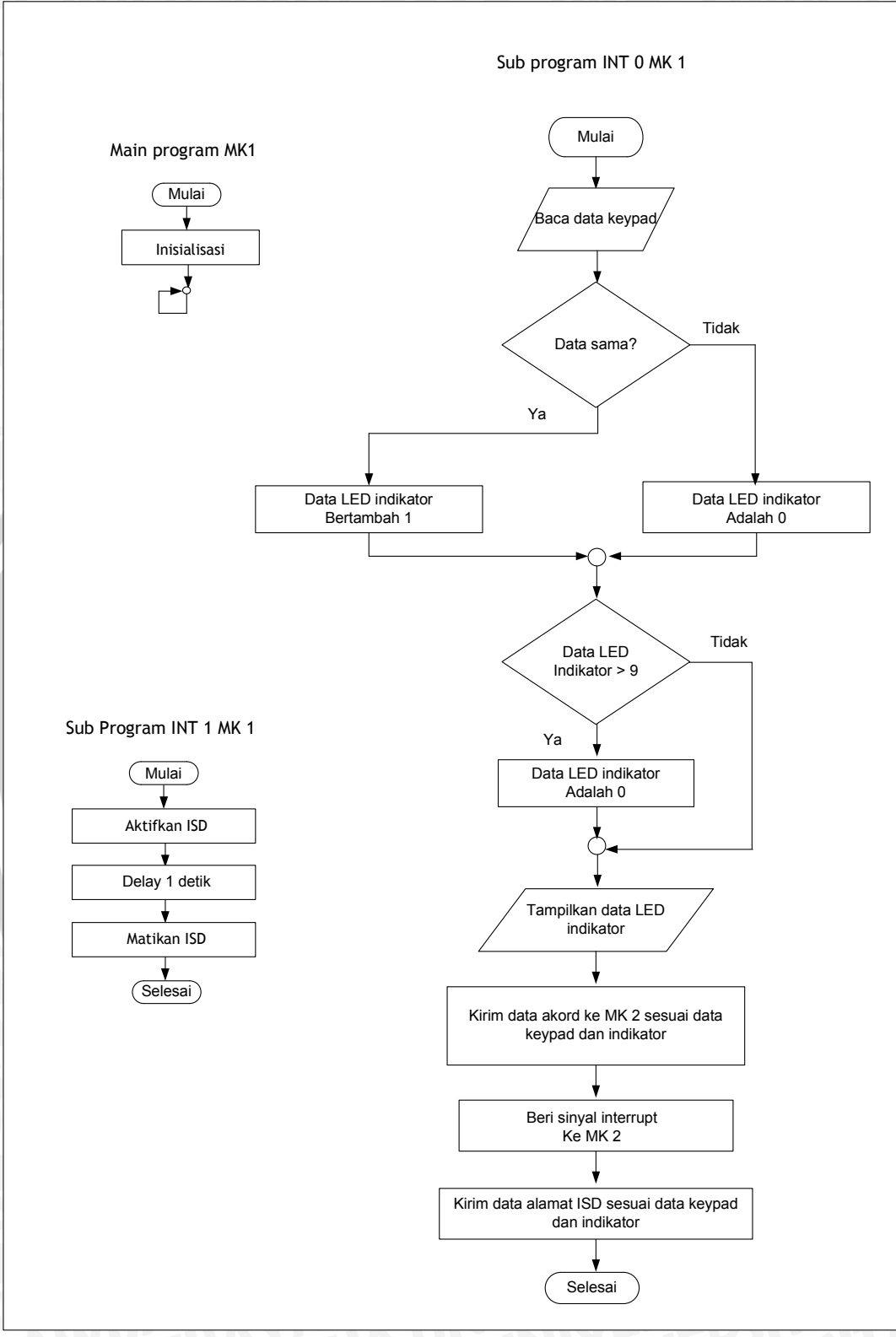
Tabel 4.2. Alamat dan Durasi Untuk Merekam Bunyi Akord C

Alamat (desimal)	Durasi (detik)	Suara yang direkam
0-5	1	C
6-10	1	C May7
11-15	1	Cm
16-20	1	Cm7
21-25	1	C Sus
26-30	1	C aug(+)
31-35	1	C7
36-40	1	C9
41-45	1	C11

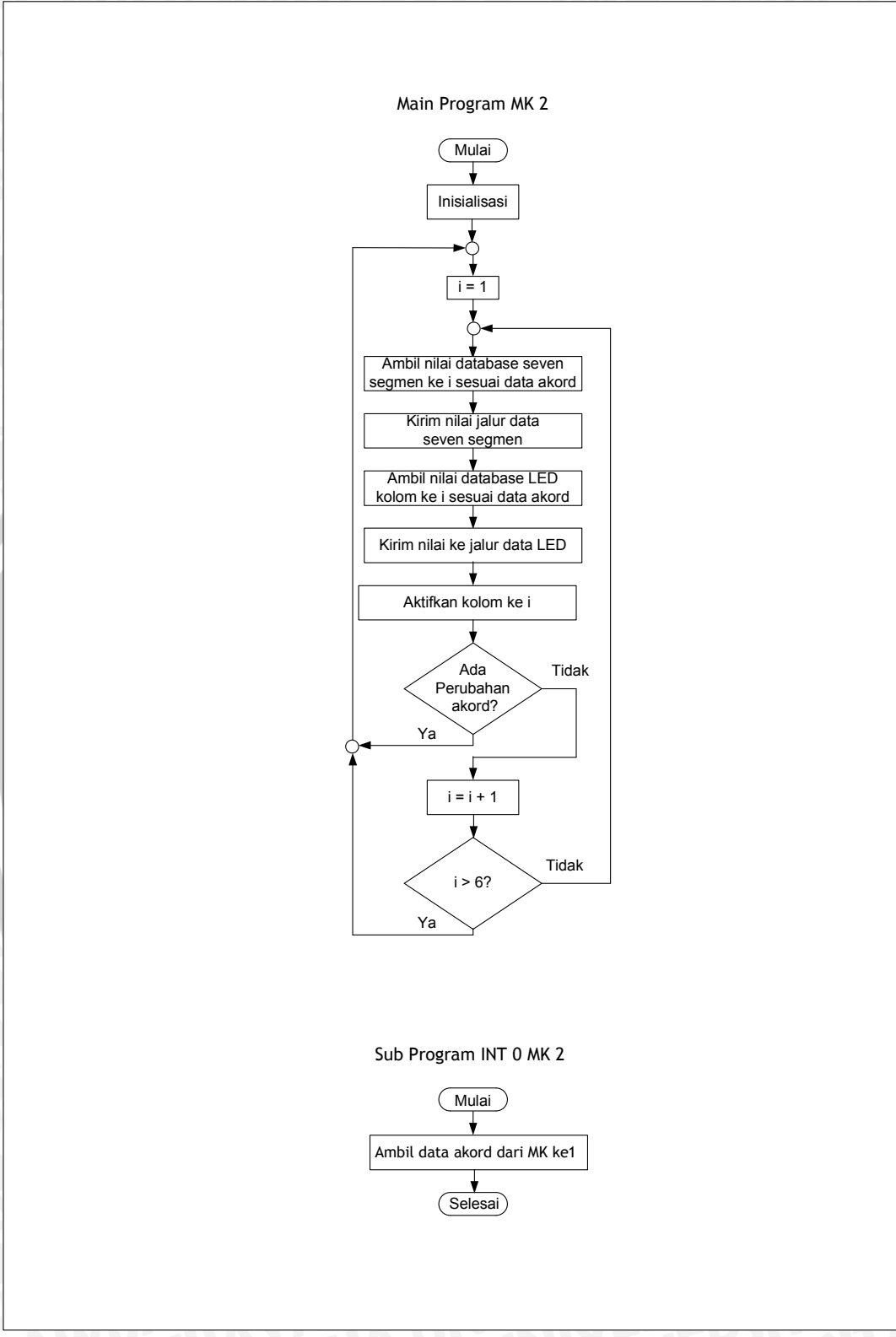
4.3. Perencanaan Perangkat Lunak

Hal lain yang sangat penting dalam membangun sistem ini adalah perangkat lunak yang akan dijalankan di dalam *personal computer* untuk mengolah data yang didapat dari unit masukan untuk kemudian melakukan respon tertentu yang diinginkan sesuai dengan spesifikasi sistem yang telah ditentukan.

Bahasa pemrograman yang digunakan dalam perencanaan perangkat lunak ini adalah bahasa pemrograman *Assembler*. *Flowchart* untuk perancangan ini dapat dilihat dalam Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Diagram Alir Perangkat Lunak Mikrokontroler 1



Gambar 4.10 Diagram Alir Perangkat Lunak Mikrokontroler 2

BAB V

PENGUJIAN ALAT

Bab ini berisi penjelasan prosedur pengujian dari alat yang telah dirancang guna mengetahui sistem dapat bekerja dengan baik sesuai dengan perencanaan. Pengujian dilakukan dengan memberikan perubahan pada masukan blok rangkaian dan mengamati keluaran dari blok rangkaian yang diuji tersebut. Data hasil pengujian yang diperoleh nantinya akan dianalisis untuk dijadikan acuan dalam mengambil kesimpulan.

Pengujian dilakukan pada tiap-tiap blok sistem. Adapun blok-blok yang diuji adalah :

- ✓ Pengujian Rangkaian *Keypad*
- ✓ Pengujian Rangkaian LED Indikator
- ✓ Pengujian Rangkaian *Display*
- ✓ Pengujian Rangkaian Pemutar Suara ISD25120
- ✓ Pengujian Rangkaian Keseluruhan Sistem

5.1. Pengujian Rangkaian *Keypad*

➤ Tujuan

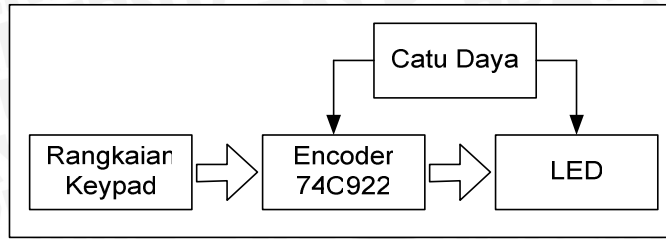
Untuk mengetahui apakah rangkaian *Keypad* dapat bekerja dengan baik.

➤ Peralatan yang Digunakan

- Rangkaian *Keypad*
- Encoder 74C922
- LED 4 buah
- Resistor 330 Ω , 4 buah
- Catu daya

➤ Langkah Pengujian

- Merangkai blok pengujian *keypad* dan Encoder 74C922 seperti dalam Gambar 5.1
- Menghubungkan catu daya ke rangkaian pengujian
- Menekan tombol pada *keypad* dan mencatat data *output* berupa tampilan pada LED



Gambar 5.1. Diagram Blok Pengujian Rangkaian *Keypad*

➤ Hasil Pengujian dan Analisis

Hasil pengujian rangkaian *Keypad* ditunjukkan dalam Tabel 5.1.

Tabel 5.1. Hasil Pengujian Rangkaian *Keypad*

Penekanan tombol <i>keypad</i>	Input Pin IC74C922							LED yang menyala			
	Y1	Y2	Y3	X1	X2	X3	X4	A	B	C	D
	C	1	0	0	1	0	0	0	0	0	0
C#	1	0	0	0	1	0	0	1	0	0	0
D	1	0	0	0	0	1	0	0	1	0	0
D#	1	0	0	0	0	0	1	1	1	0	0
E	0	1	0	1	0	0	0	0	0	1	0
F	0	1	0	0	1	0	0	1	0	1	0
F#	0	1	0	0	0	1	0	0	1	1	0
G	0	1	0	0	0	0	1	1	1	1	0
G#	0	0	1	1	0	0	0	0	0	0	1
A	0	0	1	0	1	0	0	1	0	0	1
A#	0	0	1	0	0	1	0	0	1	0	1
B	0	0	1	0	0	0	1	1	1	0	1

Berdasarkan hasil pengujian rangkaian *keypad* pada Tabel 5.1 didapatkan bahwa sinyal yang keluar pada LED sama dengan penekanan tombol pada rangkaian *keypad*. Dengan demikian rangkaian *keypad* telah berfungsi dengan baik.

5.2. Pengujian Rangkaian LED Indikator

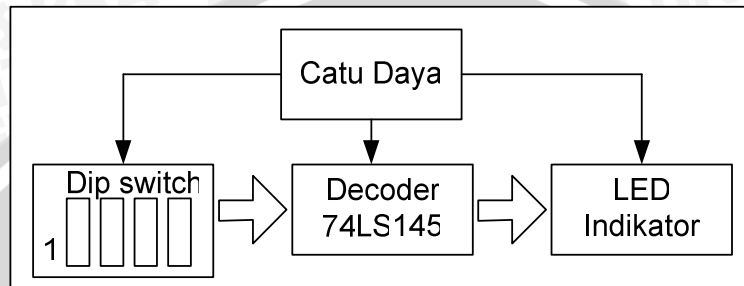
➤ Tujuan

Untuk mengetahui apakah rangkaian LED Indikator dapat bekerja dengan baik.

➤ Peralatan yang Digunakan

- Rangkaian LED Indikator
- Decoder 74LS145
- Catu daya
- *Dip switch*

- Langkah Pengujian
 - Merangkai blok pengujian LED Indikator dan Decoder 74LS145 seperti dalam Gambar 5.2
 - Menghubungkan catu daya ke rangkaian pengujian
 - Memberikan kombinasi sinyal 'low' dan 'high' pada pin-pin masukan IC74LS145



Gambar 5.2. Diagram Blok Pengujian Rangkaian LED Indikator

- Hasil Pengujian dan Analisis

Hasil pengujian rangkaian LED Indikator ditunjukkan dalam Tabel 5.2.

Tabel 5.2. Hasil Pengujian Rangkaian LED Indikator

Masukan pada pin IC74LS145				LED Yang Menyala
A	B	C	D	
0	0	0	0	7 th
1	0	0	0	Minor 7 th
0	1	0	0	Mayor
1	1	0	0	9 th
0	0	1	0	Sus 4 th
1	0	1	0	Mayor 7 th
0	1	1	0	11 th
1	1	1	0	Aug (+)
0	0	0	1	Minor

Berdasarkan hasil pengujian rangkaian LED Indikator pada Tabel 5.2 didapatkan bahwa jika masukan *dip switch* menghasilkan sinyal 'low', maka LED tersebut akan menyala. Dengan demikian rangkaian LED Indikator telah berfungsi dengan baik.

5.3. Pengujian Rangkaian *Display*

➤ Tujuan

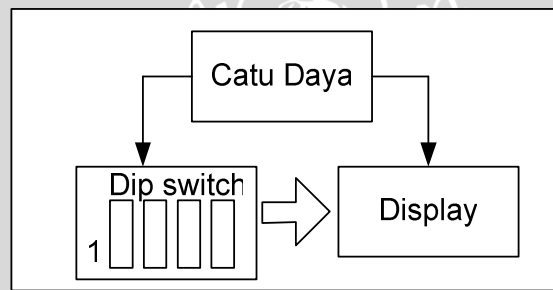
Untuk mengetahui apakah rangkaian *display* dapat bekerja dengan baik.

➤ Peralatan yang Digunakan

- Rangkaian *display*
- Catu daya
- Dip switch

➤ Langkah Pengujian

- Merangkai blok pengujian rangkaian *display* seperti dalam Gambar 5.3
- Menghubungkan catu daya ke rangkaian pengujian
- Memberikan sinyal 'low' pada anoda dari setiap LED dan *seven segment*
- Memberikan sinyal 'low' pada basis transistor BC557 secara bergantian



Gambar 5.3. Diagram Blok Pengujian Rangkaian *Display*

➤ Hasil Pengujian dan Analisis

Hasil pengujian rangkaian *Display* ditunjukkan dalam Tabel 5.3.

Tabel 5.3. Hasil Pengujian Rangkaian *Display*

Masukan pada basis transistor BC557						LED dan <i>seven segment</i> yang Menyala
1	2	3	4	5	6	
0	1	1	1	1	1	Senar 6
1	0	1	1	1	1	Senar 5
1	1	0	1	1	1	Senar 4
1	1	1	0	1	1	Senar 3
1	1	1	1	0	1	Senar 2
1	1	1	1	1	0	Senar 1

Berdasarkan hasil pengujian rangkaian *display* pada Tabel 5.3 didapatkan bahwa LED dan *seven segment* akan menyala bila masukan pada basis transistor yang berhubungan dengan LED dan *seven segment* tersebut adalah 'low'. Dengan demikian rangkaian *diplay* telah berfungsi dengan baik.

5.4. Pengujian Rangkaian Pemutar Suara ISD25120

➤ Tujuan

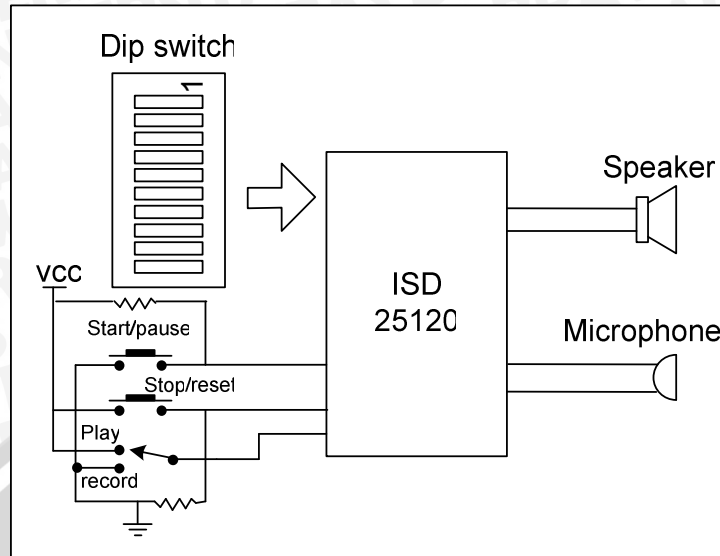
Pengujian rangkaian pemutar suara ISD25120 dilakukan untuk mengetahui apakah IC ISD25120 dapat dioperasikan untuk menyimpan suara dan memutar suara yang telah direkam.

➤ Peralatan yang Digunakan

- Rangkaian ISD 25120
- Catu daya.
- Speaker
- *Microphone*
- *Dip Switch*

➤ Langkah Pengujian

- Merangkai rangkaian pengujian pemutar/perekam suara ISD25120 seperti ditunjukkan dalam Gambar 5.4
- Menghubungkan catu daya ke rangkaian pengujian
- Alamat penyimpan suara yang akan direkam di-*set* terlebih dahulu dengan menggunakan *Dip switch*
- Untuk merekam suara, pilih saklar pada posisi *RECORD*, kemudian tekan tombol *START*, setelah selesai merekam tekan tombol *STOP*
- Untuk memutar suara pilih saklar pada posisi *PLAY*, kemudian tekan tombol *START*. setelah selesai merekam tekan tombol *STOP*
- Untuk berhenti merekam sesaat atau berhenti memutar ulang sesaat tekan tombol *PAUSE*



Gambar 5.4. Modul IC ISD25120

➤ Hasil Pengujian dan Analisis

Tabel 5.4 menunjukkan pemilihan alamat untuk merekam suara yang direkam serta durasi penyimpanannya. Dari hasil pengujian, ISD25120 dapat merekam dan memutar suara yang direkam.

Tabel 5.4 Hasil Pengujian ISD25120

Alamat (desimal)	Durasi (detik)	Suara yang direkam
0	1	C
6	1	C May7
11	1	C Min
16	1	C Min7
21	1	C Sus
26	1	C aug(+)
31	1	C7
36	1	C9
41	1	C11

Pengujian IC ISD25120 dilakukan hanya untuk mengetahui kemampuan merekam dan memutar suara IC ISD25120. Proses perekaman suara dilakukan sesuai dengan prosedur yang telah disebutkan diatas. Sedangkan untuk pengalamatan setiap

suara yang akan disimpan harus disesuaikan dengan lamanya waktu yang diperlukan untuk merekam. Jika ada kesalahan dalam menentukan alamat akan mengakibatkan suara yang telah disimpan akan masuk ke dalam alamat suara berikutnya dan mengakibatkan suara yang telah direkam terpotong. Berdasarkan hasil pengujian yang telah dilakukan dengan memperlihatkan prosedur yang telah ditetapkan, IC ISD25120 dapat bekerja dengan baik, karena mampu merekam dan memutar suara yang direkam.

5.5. Pengujian Rangkaian Keseluruhan Sistem

➤ Tujuan

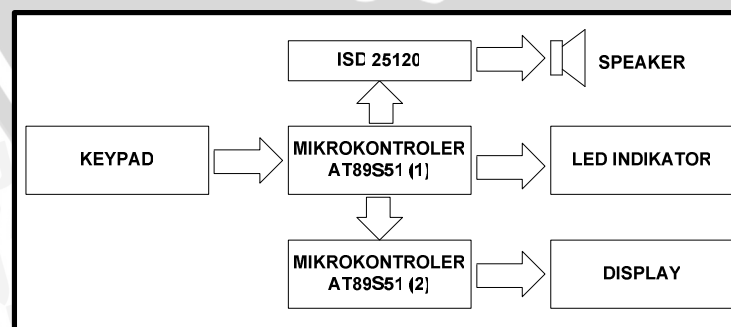
Pengujian keseluruhan sistem bertujuan untuk mengetahui kinerja sistem secara keseluruhan, apakah telah berhasil sesuai dengan spesifikasi yang diharapkan.

➤ Peralatan yang Digunakan

- Rangkaian *Keypad*
- Rangkaian Minimum Sistem
- Rangkaian *Display*
- Rangkaian ISD25120
- Rangkaian LED Indikator

➤ Langkah Pengujian

- Merangkai sistem seperti dalam gambar 5.5
- Mengaktifkan catu daya
- Beri masukan pada keypad satu per satu hingga semua variasi akord gitar ditampilkan
- Beri masukan pada tombol nada
- Amati keluarannya.




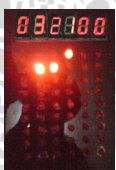




Gambar 5.5 Blok Diagram Keseluruhan Sistem

➤ Hasil Pengujian dan Analisis

Tabel 5.5 menunjukkan hasil pengujian yang dilakukan secara acak terhadap alat:

Tabel 5.5. Hasil Pengujian Pada Alat

Tombol yang ditekan	Jumlah penekanan	LED indikator yang menyala	Tampilan display	Bunyi akord jika tombol nada ditekan	Keterangan
C	1x	Mayor		C Mayor	Sesuai
A	1x	Mayor		A Mayor	Sesuai
A	3x	Minor		A Minor	Sesuai
E	1x	Mayor		E Mayor	Sesuai
G	1x	Mayor		G Mayor	Sesuai
D	1x	Mayor		D Mayor	Sesuai

Dari hasil pengujian didapatkan bahwa sistem dapat bekerja sesuai dengan spesifikasi yang telah ditentukan. Pengujian yang sama dilakukan terhadap semua tombol pada *keypad*.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran dari perencanaan dan pembuatan *Alat Penampil Akord Gitar*.

6.1. Kesimpulan

Berdasarkan pada perancangan, hasil dari pengujian, serta pengamatan yang telah dilakukan, maka dapat diambil beberapa kesimpulan:

1. Alat Penampil Akord Gitar ini dapat menampilkan 108 buah akord gitar.
2. Pada display, *seven segment* yang digunakan mampu menunjukkan jari-jari yang digunakan untuk menekan dan LED yang menyala menunjukkan posisi senar gitar yang harus ditekan.
3. Alat Penampil Akord Gitar ini dapat menampilkan 9 macam variasi akord
4. Jika tombol nada tidak ditekan maka alat ini tidak akan mengeluarkan bunyi dari akord gitar yang sedang ditampilkan.
5. Pengujian keseluruhan dari sistem berjalan dengan baik, di mana sistem dapat menampilkan akord gitar pada *display* sesuai dengan penekanan tombol pada *keypad*.

Dari pengujian keseluruhan ini dapat disimpulkan bahwa alat penampil akord gitar ini sudah bekerja dengan baik.

6.2. Saran

Ada beberapa hal yang perlu diperhatikan dalam pengembangan alat ini dikemudian hari. Meskipun alat ini sudah dapat bekerja dengan baik dan sesuai dengan sistem yang direncanakan masih ada hal-hal yang perlu ditingkatkan, diantaranya:

1. Menambah jumlah tampilan akord gitar sehingga akord gitar yang bisa ditampilkan akan lebih bervariasi.
2. Dapat dikembangkan dengan menambahkan variasi penekanan dari setiap akord gitar yang ditampilkan.

DAFTAR PUSTAKA

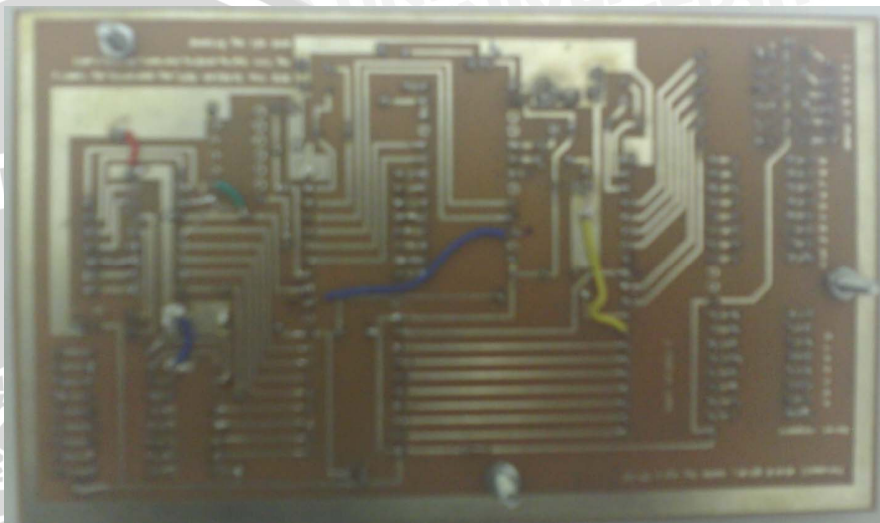
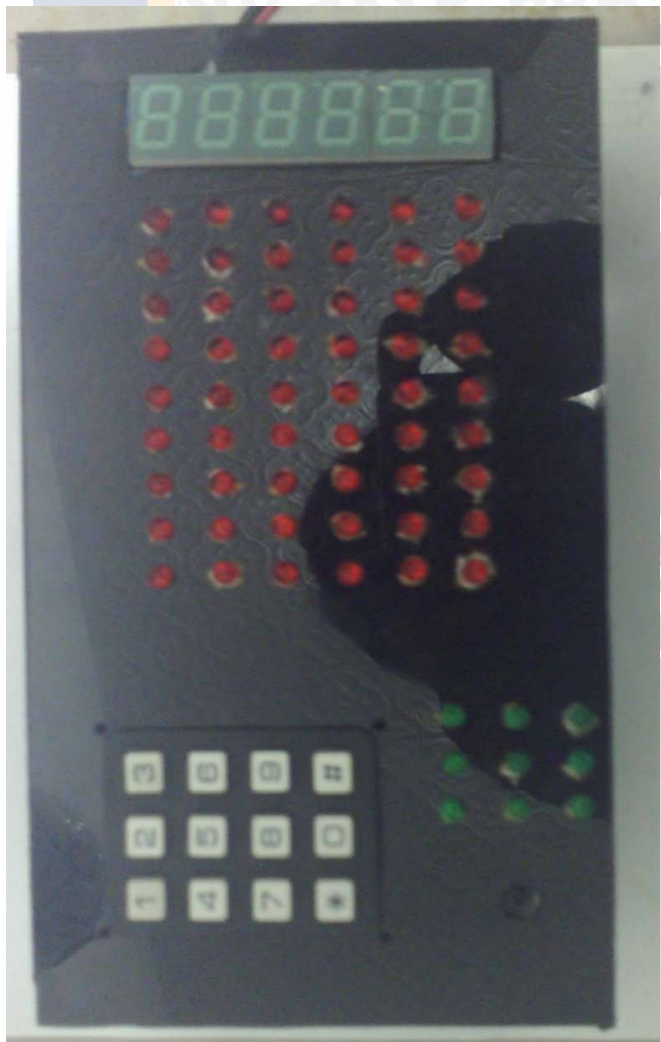
- Anonymous, 2002. *AT89S51 8 bit microcontroller wiyh 4 Byte Flash (online)*. Atmel Cooperation. <http://www.atmel.com/89S51.pdf>. Tanggal Akses: 05 Januari 2007.
- Anonymous, 2006. Memahami akor gitar
<http://pocastella.com/ebo/>. Tanggal Akses: 04 Juli 2007.
- Anonymous, 2007. Bursa Musik Populer volume 36 Th.ke III.
- Atmel, 1997, *Flash Microcontroller: Architectural Overview*, Atmel Inc. www.atmel.com. USA.2006. Tanggal Akses: 05 Januari 2007.
- Lee, Samual C. 1976. *Digital Circuits and Logic Design*. Prentice Hall. Englewood, USA.
- Malvino. 1994. *Prinsip-prinsip Elektronika*. Jilid I dan II. Edisi Ketiga. Alih bahasa: Barnawi. Penerbit Erlangga. Jakarta.
- Putra, Agfianto Eko, 2004. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi), Edisi Kedua, Cetakan Pertama*. Yogyakarta: Gava Media.
- Roger L, Tokheim, 1990. *Elektronika Digital, Edisi Kedua*, Terjemahan Sutrisno. Jakarta: Erlangga.
- Wasito S. 1996. *Data Sheet Book 1*. Jakarta: PT Elex Media Komputindo
- _____, 1997, <http://www.phys.unsw.edu.au/>. Tanggal Akses: 07 Juli 2007.
- _____, 1999, <http://www.fairchildsemi.com/>. Tanggal Akses: 15 Juli 2007.
- _____, 2000, <http://www.alldatasheet.com/>. Tanggal Akses: 15 Juli 2007.
- _____, 2000, <http://www.guitar-pro.com/>. Tanggal Akses: 07 Juli 2007.
- _____, 2004, <http://www.digitech.com/>. Tanggal Akses: 07 Juli 2007.

UNIVERSITAS BRAWIJAYA

LAMPIRAN A

Gambar Alat

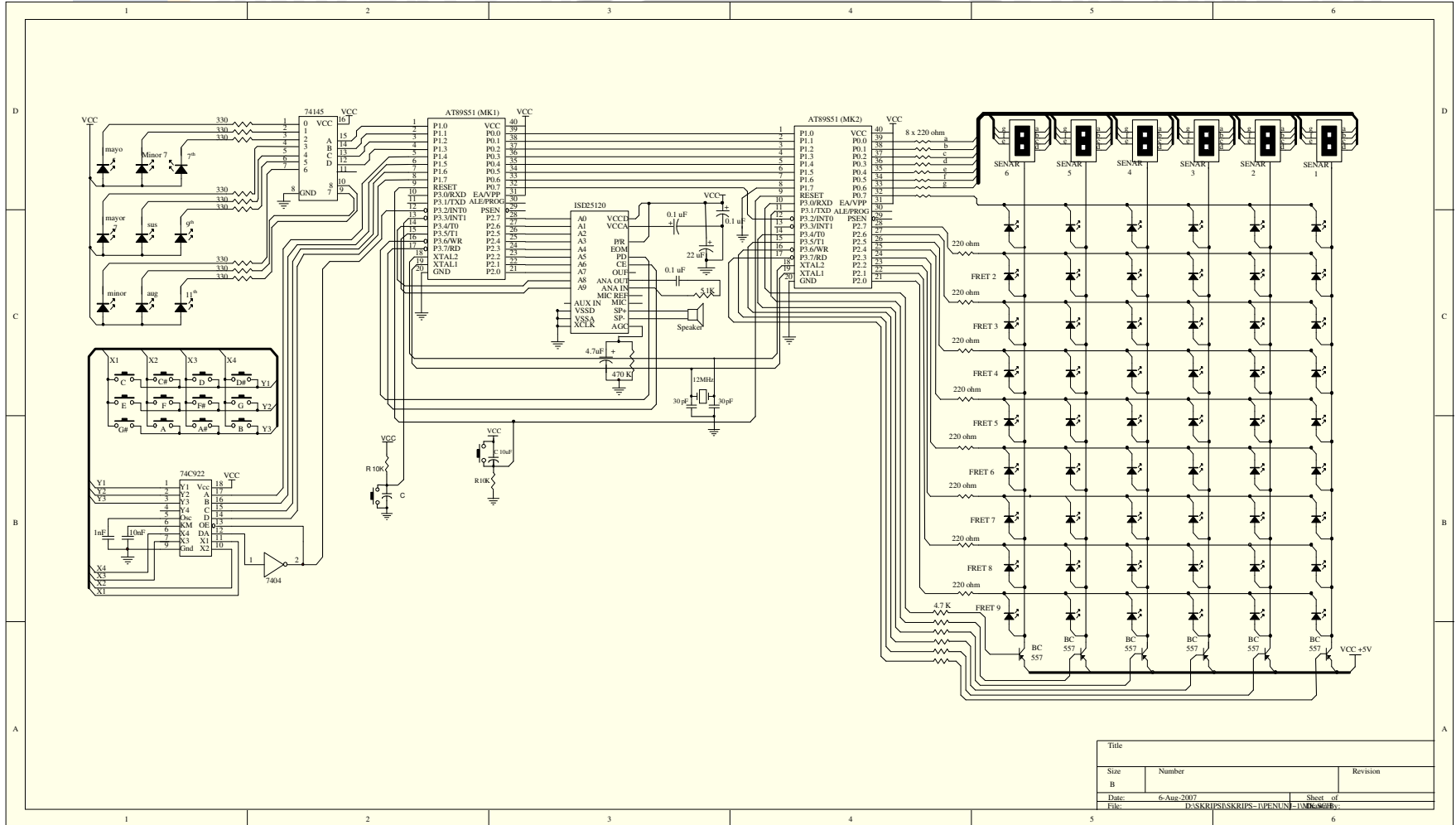
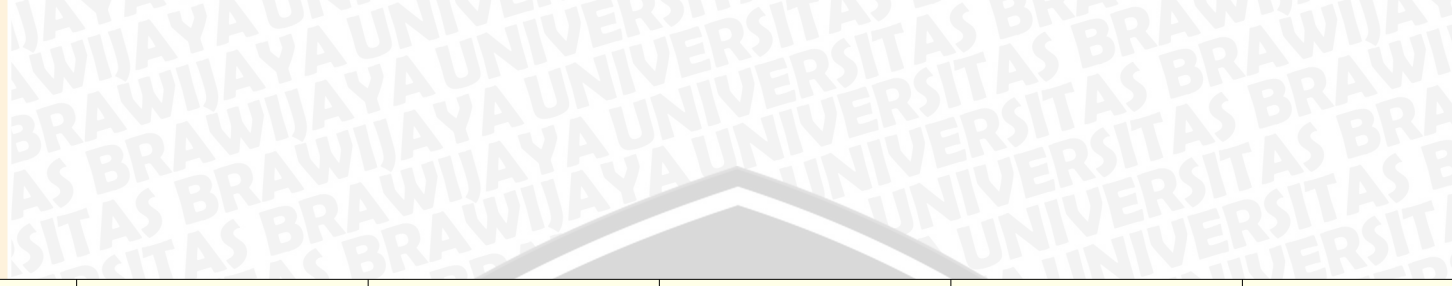






LAMPIRAN B

Gambar Rangkaian



Title		
Size	Number	Revision
B		
Date:	6-Aug-2007	Sheet of
File:	D:\SKRIPSI\SKRIPS-LUPENUNJ-LUMBU\6061	





LAMPIRAN C

Perangkat Lunak (Software)

```

;-----
; MK 1
;-----
;***DEFINISI***
KIRIM_INTERUPSI BIT P0.7
PD_ISD BIT P3.6
CE_ISD BIT P3.7
;***ORIGIN***
ORG 000H
LJMP MAIN
ORG 003H
LJMP PENANGANAN_INT_0
ORG 00BH
LJMP PENANGANAN_IT_0
ORG 013H
LJMP PENANGANAN_INT_1
ORG 01BH
LJMP PENANGANAN_IT_1
ORG 033H
;***MAIN***
MAIN:
LCALL INISIALISASI
TUNGGU_INTERUPSI:
SJMP TUNGGU_INTERUPSI
;***INISIALISASI***
INISIALISASI:
MOV IE,#08FH
MOV TCON,#005H
MOV TMOD,#011H
MOV TL0,#0B0H
MOV TH0,#03CH
MOV TL1,#0B0H
MOV TH1,#03CH
MOV P0,#0FFH
MOV P1,#0FFH
MOV P2,#04EH
MOV P3,#0EFH
MOV R7,#0FFH
MOV R6,#0FFH
MOV R5,#0FFH
RET
;***INTERRUPT 0***
PENANGANAN_INT_0:
CLR EX0
MOV A,P1
ANL A,#0F0H
MOV R6,A
CLR C
SUBB A,R7
JZ KEYPAD_SAMA
MOV R5,#000H
LJMP PENGAKTIFAN_LED
KEYPAD_SAMA:
INC R5
CJNE R5,#009H,PENGAKTIFAN_LED
MOV R5,#000H
PENGAKTIFAN_LED:
MOV A,R5
ORL A,#0F0H
MOV P1,A
MOV A,R6
MOV R7,A
AKTIF_AKORD_C:
CJNE R6,#000H,AKTIF_AKORD_CX
MOV A,R5
MOV DPTR,#AKORD_C
MOVC A,@A+DPTR
MOV P0,A
MOV A,R5
MOV DPTR,#ISD_LOW_C
MOVC A,@A+DPTR
MOV P2,A
MOV A,R5
MOV DPTR,#ISD_HIGH_C
MOVC A,@A+DPTR
MOV P3,A
LJMP PENGAKTIFAN_INTERUPSI
AKTIF_AKORD_CX:
CJNE R6,#010H,AKTIF_AKORD_D
MOV A,R5
MOV DPTR,#AKORD_CX
MOVC A,@A+DPTR
MOV P0,A
MOV A,R5
MOV DPTR,#ISD_LOW_CX
MOVC A,@A+DPTR
MOV P2,A
MOV A,R5
MOV DPTR,#ISD_HIGH_CX
MOVC A,@A+DPTR
MOV P3,A
LJMP PENGAKTIFAN_INTERUPSI
AKTIF_AKORD_D:
CJNE R6,#020H,AKTIF_AKORD_DX
MOV A,R5
MOV DPTR,#AKORD_D
MOVC A,@A+DPTR
MOV P0,A
MOV A,R5
MOV DPTR,#ISD_LOW_D
MOVC A,@A+DPTR
MOV P2,A
MOV A,R5
MOV DPTR,#ISD_HIGH_D
MOVC A,@A+DPTR
MOV P3,A
LJMP PENGAKTIFAN_INTERUPSI
AKTIF_AKORD_DX:
CJNE R6,#030H,AKTIF_AKORD_E
MOV A,R5
MOV DPTR,#AKORD_DX
MOVC A,@A+DPTR
MOV P0,A
MOV A,R5
MOV DPTR,#ISD_LOW_DX
MOVC A,@A+DPTR
MOV P2,A
MOV A,R5
MOV DPTR,#ISD_HIGH_DX
MOVC A,@A+DPTR
MOV P3,A
LJMP PENGAKTIFAN_INTERUPSI
AKTIF_AKORD_E:
CJNE R6,#040H,AKTIF_AKORD_F

```



```

MOV     A,R5
MOV     DPTR,#AKORD_E
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_E
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_E
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_F:

```

CJNE    R6,#050H,AKTIF_AKORD_FX
MOV     A,R5
MOV     DPTR,#AKORD_F
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_F
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_F
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_FX:

```

CJNE    R6,#060H,AKTIF_AKORD_G
MOV     A,R5
MOV     DPTR,#AKORD_FX
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_FX
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_FX
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_G:

```

CJNE    R6,#070H,AKTIF_AKORD_GX
MOV     A,R5
MOV     DPTR,#AKORD_G
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_G
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_G
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_GX:

```

CJNE    R6,#080H,AKTIF_AKORD_A

```

```

MOV     A,R5
MOV     DPTR,#AKORD_GX
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_GX
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_GX
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_A:

```

CJNE    R6,#090H,AKTIF_AKORD_AX
MOV     A,R5
MOV     DPTR,#AKORD_A
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_A
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_A
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_AX:

```

CJNE    R6,#0A0H,AKTIF_AKORD_B
MOV     A,R5
MOV     DPTR,#AKORD_AX
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_AX
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_AX
MOVC    A,@A+DPTR
MOV     P3,A
LJMP
PENGAKTIFAN_INTERUPSI

```

AKTIF_AKORD_B:

```

CJNE    R6,#0B0H,INT_0_KELUAR
MOV     A,R5
MOV     DPTR,#AKORD_B
MOVC    A,@A+DPTR
MOV     P0,A
MOV     A,R5
MOV     DPTR,#ISD_LOW_B
MOVC    A,@A+DPTR
MOV     P2,A
MOV     A,R5
MOV     DPTR,#ISD_HIGH_B
MOVC    A,@A+DPTR
MOV     P3,A

```

PENGAKTIFAN_INTERUPSI:

```

NOP
SETB   KIRIM_INTERUPSI
INT_0_KELUAR:
MOV     R4,#005H

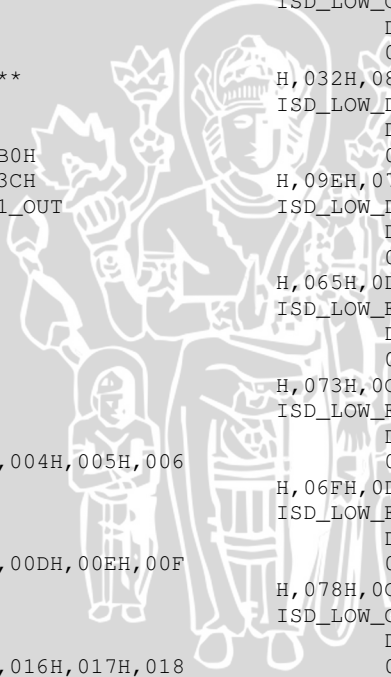
```



```

SETB TR0 AKORD_G:
RETI DB
;***INTERRUPT TIMER 0*** 040H,041H,042H,043H,044H,045
PENANGANAN_IT_0: H,046H,047H,048H
CLR ET0 AKORD_GX:
MOV TL0,#0B0H DB
MOV TH0,#03CH 049H,04AH,04BH,04CH,04DH,04E
DJNZ R4,IT_0_OUT H,04FH,050H,051H
CLR TR0 AKORD_A:
SETB EX0 DB
IT_0_OUT: 052H,053H,054H,055H,056H,057
SETB ET0 H,058H,059H,05AH
RETI AKORD_AX:
;***INTERRUPT 1*** DB
PENANGANAN_INT_1: 05BH,05CH,05DH,05EH,05FH,060
CJNE R5,#0FFH,INT_1_DO H,061H,062H,063H
LJMP INT_1_OUT AKORD_B:
INT_1_DO: DB
CLR EX1 064H,065H,066H,067H,068H,069
CLR CE_ISD H,06AH,06BH,06CH
CLR PD_ISD ISD_LOW_C:
SETB CE_ISD DB
MOV R3,#00CH 000H,060H,0D0H,008H,0A8H,058
SETB TR1 H,0F8H,024H,094H
INT_1_OUT: ISD_LOW_CX:
RETI DB
;***INTERRUPT TIMER 1*** 074H,0CCH,01CH,0BCH,042H,0E2
PENANGANAN_IT_1: H,032H,08AH,06AH
CLR ET1 ISD_LOW_D:
MOV TL1,#0B0H DB
MOV TH1,#03CH 0DAH,006H,0A6H,056H,0F6H,02E
DJNZ R3,IT_1_OUT H,09EH,07EH,0C1H
CLR TR1 ISD_LOW_DX:
SETB PD_ISD DB
SETB EX1 011H,0B1H,049H,069H,039H,085
IT_1_OUT: H,065H,0D5H,00DH
SETB ET1 ISD_LOW_E:
RETI DB
;***DATABASE*** 0ADH,05DH,0FDH,023H,093H,058
AKORD_C: H,073H,0CBH,01BH
DB ISD_LOW_F:
001H,002H,003H,004H,005H,006 DB
H,007H,008H,009H 0BBH,047H,0E7H,037H,08FH,0E2
AKORD_CX: H,06FH,0DFH,000H
DB ISD_LOW_FX:
00AH,00BH,00CH,00DH,00EH,00F DB
H,010H,011H,012H 0A0H,050H,0F0H,028H,098H,02E
AKORD_D: H,078H,0C4H,014H
DB ISD_LOW_G:
013H,014H,015H,016H,017H,018 DB
H,019H,01AH,01BH 0B4H,04CH,0EEH,03CH,082H,085
AKORD_DX: H,062H,0D2H,00AH
DB ISD_LOW_GX:
01CH,01DH,01EH,01FH,020H,021 DB
H,022H,023H,024H 0AAH,05AH,0FAH,026H,096H,058
AKORD_E: H,076H,0CEH,01EH
DB ISD_LOW_A:
025H,026H,027H,028H,029H,02A DB
H,02BH,02CH,02DH 0BEH,041H,0E1H,031H,089H,0E2
AKORD_F: H,069H,0D9H,005H
DB ISD_LOW_AX:
02EH,02FH,030H,031H,032H,033 DB
H,034H,035H,036H 0A5H,055H,0F5H,02DH,09DH,02E
AKORD_FX: H,07DH,0C3H,013H
DB ISD_LOW_B:
037H,038H,039H,03AH,03BH,03C DB
H,03DH,03EH,03FH 0B3H,04BH,0EBH,03BH,087H,085
H,067H,0D7H,00FH

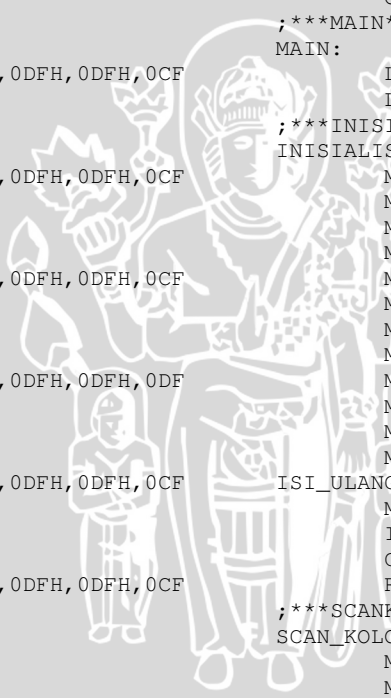
```



```

ISD_HIGH_C:          ;-----
                   DB          ; MK 2
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   H,OCFH,OCFH,OCFH
                   ;***DEFINISI***
ISD_HIGH_CX:        DATA_0_SENAR_5      EQU      20H
                   DB          EQU      21H
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   DATA_0_SENAR_6      EQU      22H
                   H,OCFH,OCFH,OCFH
                   DATA_1_SENAR_6      EQU      23H
ISD_HIGH_D:        DATA_0_SENAR_1      EQU      24H
                   DB          EQU      25H
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   DATA_0_SENAR_2      EQU      26H
                   H,OCFH,OCFH,OCFH
                   DATA_1_SENAR_2      EQU      27H
ISD_HIGH_DX:        DATA_0_SENAR_3      EQU      28H
                   DB          EQU      29H
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   DATA_0_SENAR_4      EQU      2AH
                   H,OCFH,OCFH,OCFH
                   DATA_1_SENAR_4      EQU      2BH
ISD_HIGH_E:        ;***ORIGIN***
                   DB          ORG      000H
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   LJMP      MAIN
                   H,OCFH,OCFH,OCFH
                   ORG      003H
ISD_HIGH_F:        LJMP      PENANGANAN_INT_0
                   DB          ORG      00BH
                   OCFH,OCFH,OCFH,OCFH,OCFH,OCF
                   LJMP      PENANGANAN_IT_0
                   H,OCFH,OCFH,OCFH
                   ORG      033H
ISD_HIGH_FX:        ;***MAIN***
                   DB          MAIN:
                   ODFH,ODFH,ODFH,ODFH,ODFH,OCF
                   LCALL     INISIALISASI
                   H,ODFH,ODFH,ODFH
                   LJMP      SCAN_KOLOM
ISD_HIGH_G:        ;***INISIALISASI***
                   DB          INISIALISASI:
                   ODFH,ODFH,ODFH,ODFH,ODFH,OCF
                   MOV      IE,#083H
                   H,ODFH,ODFH,ODFH
                   MOV      TCON,#001H
ISD_HIGH_GX:        MOV      TMOD,#001H
                   DB          MOV      TL0,#018H
                   ODFH,ODFH,ODFH,ODFH,ODFH,OCF
                   MOV      TH0,#0FCH
                   H,ODFH,ODFH,ODFH
                   MOV      P0,#0FFH
ISD_HIGH_A:        MOV      P1,#0FFH
                   DB          MOV      P2,#0FFH
                   ODFH,ODFH,ODFH,ODFH,ODFH,ODF
                   MOV      P3,#0FFH
                   H,ODFH,ODFH,ODFH
                   MOV      R7,#0FFH
ISD_HIGH_AX:        MOV      R6,#0FFH
                   DB          MOV      R0,#020H
                   ODFH,ODFH,ODFH,ODFH,ODFH,OCF
                   ISI_ULANG:
                   H,ODFH,ODFH,ODFH
                   MOV      @R0,#0FFH
ISD_HIGH_B:        INC      R0
                   DB          CJNE     R0,#02CH,ISI_ULANG
                   ODFH,ODFH,ODFH,ODFH,ODFH,OCF
                   RET
                   H,ODFH,ODFH,ODFH
                   ;***SCANKOLOM***
                   END          SCAN_KOLOM:
                                   MOV      R7,#0FFH
                                   MOV      R5,#0FDH
                                   MOV      R0,#020H
                   TIAP_KOLOM:
                                   MOV      P0,@R0
                                   INC      R0
                                   MOV      P2,@R0
                                   INC      R0
                                   MOV      A,R5
                                   MOV      P3,A
                                   RR      A
                                   MOV      R5,A
                                   SETB   TR0
                   TUNDA:
                                   JNB     TF0,TUNDA
                                   MOV      P3,#0FFH
                                   CJNE     R7,#0FFH,SCAN_KOLOM
                                   CJNE     R0,#02CH,TIAP_KOLOM
                                   LJMP     SCAN_KOLOM

```



```

;***INTERRUPT 0***
PENANGANAN_INT_0:
    CLR     EX0
    MOV     R6, P1
    DEC     R6
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_6
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_6, A
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_5
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_5, A
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_4
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_4, A
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_3
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_3, A
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_2
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_2, A
    MOV     A, R6
    MOV     DPTR, #DB_0_SENAR_1
    MOVC    A, @A+DPTR
    MOV     DATA_0_SENAR_1, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_6
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_6, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_5
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_5, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_4
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_4, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_3
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_3, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_2
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_2, A
    MOV     A, R6
    MOV     DPTR, #DB_1_SENAR_1
    MOVC    A, @A+DPTR
    MOV     DATA_1_SENAR_1, A
    MOV     R7, 000H
    SETB   EX0
    RETI

;***INTERRUPT TIMER 0***
PENANGANAN_IT_0:
    CLR     ET0
    MOV     TL0, #018H
    MOV     TH0, #0FCH
    CLR     TR0
    SETB   ET0
    RETI

;***DATABASE***
DB_0_SENAR_6:

```

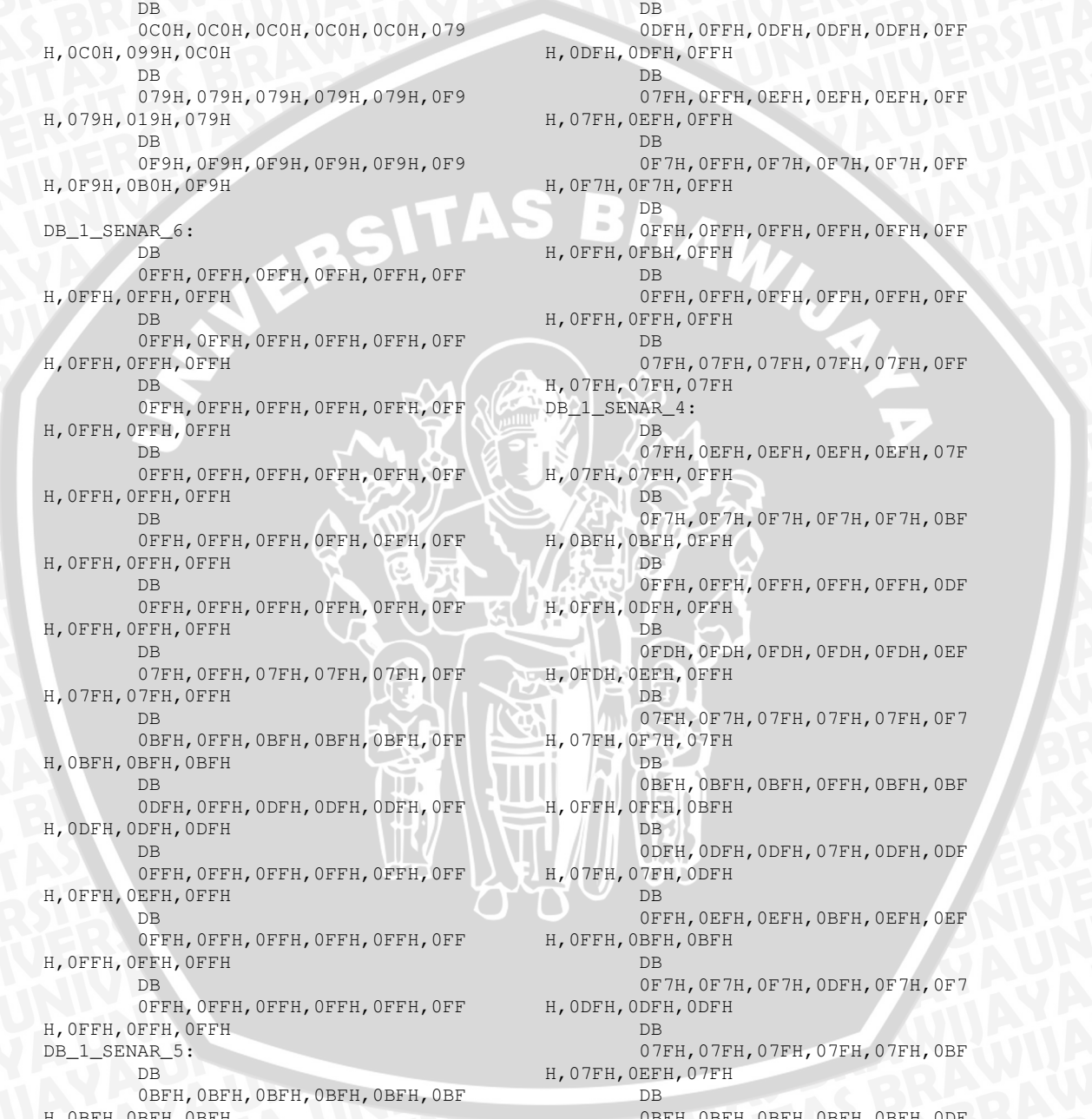


DB
 079H, 079H, 079H, 079H, 079H, 0FF
 H, 079H, 079H, 079H
 DB
 0F9H, 0F9H, 0F9H, 0F9H, 0F9H, 0FF
 H, 0F9H, 0A4H, 0F9H
 DB_0_SENAR_4:
 DB
 0A4H, 0B0H, 0B0H, 0B0H, 0A4H, 0A4
 H, 0A4H, 0F9H, 0FFH
 DB
 0A4H, 0B0H, 0B0H, 0B0H, 0A4H, 099
 H, 0A4H, 0F9H, 0FFH
 DB
 0C0H, 0C0H, 0C0H, 0C0H, 0C0H, 099
 H, 0C0H, 0F9H, 0C0H
 DB
 0A4H, 0B0H, 0B0H, 0B0H, 0A4H, 099
 H, 0B0H, 0F9H, 079H
 DB
 0B0H, 0B0H, 0B0H, 0B0H, 0B0H, 099
 H, 0B0H, 0F9H, 0F9H
 DB
 099H, 099H, 099H, 079H, 0B0H, 099
 H, 079H, 079H, 0F9H
 DB
 099H, 099H, 099H, 0F9H, 0B0H, 099
 H, 0F9H, 0F9H, 0F9H
 DB
 0C0H, 099H, 099H, 0F9H, 0B0H, 099
 H, 0C0H, 0F9H, 099H
 DB
 099H, 099H, 099H, 0F9H, 0B0H, 099
 H, 0F9H, 0F9H, 099H
 DB
 0A4H, 0A4H, 0A4H, 0A4H, 0A4H, 0A4
 H, 0A4H, 0F9H, 0F9H
 DB
 0A4H, 0B0H, 0B0H, 0B0H, 0A4H, 099
 H, 0B0H, 0C0H, 0B0H
 DB
 0A4H, 0B0H, 0B0H, 0B0H, 0A4H, 099
 H, 0B0H, 079H, 0B0H
 DB_0_SENAR_3:
 DB
 0C0H, 0A4H, 099H, 0F9H, 0B0H, 079
 H, 099H, 0B0H, 0B0H
 DB
 0B0H, 0A4H, 099H, 0F9H, 0B0H, 0A4
 H, 099H, 0B0H, 0B0H
 DB
 0F9H, 0F9H, 0A4H, 0A4H, 0F9H, 0A4
 H, 0A4H, 0B0H, 0A4H
 DB
 0B0H, 0A4H, 099H, 0F9H, 0B0H, 0A4
 H, 0F9H, 0B0H, 0B0H
 DB
 079H, 0F9H, 0C0H, 0C0H, 099H, 0A4
 H, 079H, 0B0H, 0B0H
 DB
 0A4H, 0A4H, 079H, 079H, 099H, 0A4
 H, 0A4H, 0A4H, 0B0H
 DB
 0A4H, 0B0H, 0F9H, 0F9H, 099H, 0A4
 H, 0A4H, 0A4H, 0B0H
 DB
 0C0H, 0B0H, 0F9H, 0F9H, 099H, 0A4
 H, 0C0H, 0A4H, 0A4H

DB
 0A4H, 0B0H, 0F9H, 0F9H, 099H, 0A4
 H, 0A4H, 0A4H, 0A4H
 DB
 0B0H, 079H, 0B0H, 0C0H, 0B0H, 0A4
 H, 0C0H, 0A4H, 0C0H
 DB
 0B0H, 0A4H, 099H, 079H, 0B0H, 0A4
 H, 079H, 024H, 079H
 DB
 0B0H, 0A4H, 099H, 0F9H, 0B0H, 0A4
 H, 0F9H, 0B0H, 0F9H
 DB_0_SENAR_2:
 DB
 079H, 099H, 0A4H, 0A4H, 099H, 079
 H, 079H, 0B0H, 099H
 DB
 099H, 099H, 0A4H, 0A4H, 099H, 0B0
 H, 0F9H, 0B0H, 099H
 DB
 0B0H, 0F9H, 0B0H, 079H, 0B0H, 0B0
 H, 079H, 0B0H, 079H
 DB
 099H, 099H, 0A4H, 0A4H, 099H, 0B0
 H, 099H, 0B0H, 0A4H
 DB
 0C0H, 0F9H, 0C0H, 099H, 0C0H, 0B0
 H, 099H, 0B0H, 0A4H
 DB
 079H, 079H, 079H, 079H, 079H, 0B0
 H, 079H, 079H, 0A4H
 DB
 0F9H, 0A4H, 0F9H, 0F9H, 0F9H, 0B0
 H, 0F9H, 0F9H, 0A4H
 DB
 0C0H, 0A4H, 0F9H, 0F9H, 0F9H, 0B0
 H, 0C0H, 0F9H, 079H
 DB
 0F9H, 0A4H, 0F9H, 0F9H, 0F9H, 0B0
 H, 0F9H, 0F9H, 0F9H
 DB
 099H, 0B0H, 079H, 079H, 099H, 0B0
 H, 0B0H, 0F9H, 0B0H
 DB
 099H, 099H, 0A4H, 0A4H, 099H, 0B0
 H, 099H, 030H, 099H
 DB
 099H, 099H, 0A4H, 0A4H, 099H, 0B0
 H, 099H, 0B0H, 099H
 DB_0_SENAR_1:
 DB
 0C0H, 0F9H, 0F9H, 0F9H, 0F9H, 0C0
 H, 0C0H, 0B0H, 079H
 DB
 0F9H, 0F9H, 0F9H, 0F9H, 0F9H, 079
 H, 0C0H, 0B0H, 0F9H
 DB
 0A4H, 0F9H, 079H, 079H, 099H, 0F9
 H, 0B0H, 0B0H, 0B0H
 DB
 0F9H, 0F9H, 0F9H, 0F9H, 0F9H, 0F9
 H, 0F9H, 0B0H, 099H
 DB
 0C0H, 0F9H, 0C0H, 0C0H, 0C0H, 0F9
 H, 0C0H, 0B0H, 099H
 DB
 079H, 0C0H, 079H, 079H, 079H, 079
 H, 079H, 099H, 099H

DB
 0F9H, 079H, 0F9H, 0F9H, 0F9H, 0F9
 H, 0F9H, 099H, 099H
 DB
 099H, 0F9H, 0F9H, 0F9H, 0F9H, 0F9
 H, 079H, 099H, 0FFH
 DB
 0F9H, 0F9H, 0F9H, 0F9H, 0F9H, 0F9
 H, 0F9H, 099H, 0FFH
 DB
 0C0H, 0C0H, 0C0H, 0C0H, 0C0H, 079
 H, 0C0H, 099H, 0C0H
 DB
 079H, 079H, 079H, 079H, 079H, 0F9
 H, 079H, 019H, 079H
 DB
 0F9H, 0F9H, 0F9H, 0F9H, 0F9H, 0F9
 H, 0F9H, 0B0H, 0F9H
 DB_1_SENAR_6:
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 07FH, 0FFH, 07FH, 07FH, 07FH, 0FF
 H, 07FH, 07FH, 0FFH
 DB
 0BFH, 0FFH, 0BFH, 0BFH, 0BFH, 0FF
 H, 0BFH, 0BFH, 0BFH
 DB
 0DFH, 0FFH, 0DFH, 0DFH, 0DFH, 0FF
 H, 0DFH, 0DFH, 0DFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB_1_SENAR_5:
 DB
 0BFH, 0BFH, 0BFH, 0BFH, 0BFH, 0BF
 H, 0BFH, 0BFH, 0BFH
 DB
 0DFH, 0DFH, 0DFH, 0DFH, 0DFH, 0FF
 H, 0DFH, 0DFH, 0DFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH

DB
 0F7H, 0F7H, 0F7H, 0F7H, 0F7H, 0FF
 H, 0F7H, 0F7H, 0FFH
 DB
 07FH, 0FBH, 07FH, 07FH, 07FH, 0FF
 H, 07FH, 0FBH, 0FFH
 DB
 0BFH, 0BFH, 0BFH, 0BFH, 0BFH, 0FF
 H, 0BFH, 0BFH, 0FFH
 DB
 0DFH, 0FFH, 0DFH, 0DFH, 0DFH, 0FF
 H, 0DFH, 0DFH, 0FFH
 DB
 07FH, 0FFH, 0EFH, 0EFH, 0EFH, 0FF
 H, 07FH, 0EFH, 0FFH
 DB
 0F7H, 0FFH, 0F7H, 0F7H, 0F7H, 0FF
 H, 0F7H, 0F7H, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FBH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FF
 H, 0FFH, 0FFH, 0FFH
 DB
 07FH, 07FH, 07FH, 07FH, 07FH, 0FF
 H, 07FH, 07FH, 0FFH
 DB_1_SENAR_4:
 DB
 07FH, 0EFH, 0EFH, 0EFH, 0EFH, 07F
 H, 07FH, 07FH, 0FFH
 DB
 0F7H, 0F7H, 0F7H, 0F7H, 0F7H, 0BF
 H, 0BFH, 0BFH, 0FFH
 DB
 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0DF
 H, 0FFH, 0DFH, 0FFH
 DB
 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0EF
 H, 0FDH, 0EFH, 0FFH
 DB
 07FH, 0F7H, 07FH, 07FH, 07FH, 0F7
 H, 07FH, 0F7H, 07FH
 DB
 0BFH, 0BFH, 0BFH, 0FFH, 0BFH, 0BF
 H, 0FFH, 0FFH, 0BFH
 DB
 0DFH, 0DFH, 0DFH, 07FH, 0DFH, 0DF
 H, 07FH, 07FH, 0DFH
 DB
 0FFH, 0EFH, 0EFH, 0BFH, 0EFH, 0EF
 H, 0FFH, 0BFH, 0BFH
 DB
 0F7H, 0F7H, 0F7H, 0DFH, 0F7H, 0F7
 H, 0DFH, 0DFH, 0DFH
 DB
 07FH, 07FH, 07FH, 07FH, 07FH, 0BF
 H, 07FH, 0EFH, 07FH
 DB
 0BFH, 0BFH, 0BFH, 0BFH, 0BFH, 0DF
 H, 0BFH, 0FFH, 0BFH
 DB
 0DFH, 0DFH, 0DFH, 0DFH, 0DFH, 0EF
 H, 0DFH, 0FFH, 0DFH
 DB_1_SENAR_3:
 DB
 0FFH, 0DFH, 0EFH, 0BFH, 0EFH, 0FF
 H, 0BFH, 0BFH, 0BFH



DB
 0F7H,0EFH,0F7H,0DFH,0F7H,07F
 H,0DFH,0DFH,0DFH
 DB
 07FH,07FH,07FH,07FH,07FH,0BF
 H,07FH,0EFH,07FH
 DB
 0FDH,0FBH,0FDH,0F7H,0FDH,0DF
 H,0F7H,0F7H,0BFH
 DB
 0FFH,0DFH,0FFH,0FFH,07FH,0EF
 H,0FFH,0FBH,0DFH
 DB
 07FH,07FH,0FFH,0FFH,0BFH,07F
 H,07FH,07FH,0EFH
 DB
 0BFH,0BFH,07FH,07FH,0DFH,0BF
 H,0BFH,0BFH,0F7H
 DB
 0FFH,0DFH,0BFH,0BFH,0EFH,0DF
 H,0FFH,0DFH,07FH
 DB
 0EFH,0EFH,0DFH,0DFH,0F7H,0EF
 H,0EFH,0EFH,0BFH
 DB
 07FH,0FFH,07FH,0FFH,07FH,07F
 H,0FFH,0F7H,0FFH
 DB
 0BFH,07FH,0BFH,0FFH,0BFH,0BF
 H,0FFH,0FFH,0FFH
 DB
 0DFH,0BFH,0DFH,07FH,0DFH,0DF
 H,07FH,07FH,07FH
 DB_1_SENAR_2:
 DB
 0FFH,0EFH,0DFH,0DFH,0F7H,0FF
 H,0FFH,0BFH,0BFH
 DB
 0F7H,0F7H,0EFH,0EFH,0FBH,07F
 H,07FH,0DFH,0DFH
 DB
 0BFH,07FH,0BFH,0FFH,0BFH,0BF
 H,0FFH,0EFH,0FFH
 DB
 0FDH,0FDH,0FBH,0FBH,0FEH,0DF
 H,0FDH,0F7H,07FH
 DB
 0FFH,0DFH,0FFH,0BFH,0FFH,0EF
 H,0BFH,0FBH,0BFH
 DB
 0FFH,0FFH,0FFH,0FFH,0FFH,07F
 H,0FFH,0FFH,0DFH
 DB
 07FH,07FH,07FH,07FH,07FH,0BF
 H,07FH,07FH,07FH

DB
 0FFH,0BFH,0BFH,0BFH,0BFH,0DF
 H,0FFH,0BFH,0FFH
 DB
 0DFH,0DFH,0DFH,0DFH,0DFH,0EF
 H,0DFH,0DFH,07FH
 DB
 07FH,07FH,0FFH,0FFH,0BFH,07F
 H,07FH,0EFH,0BFH
 DB
 0BFH,0BFH,07FH,07FH,0DFH,0BF
 H,0BFH,0FFH,0DFH
 DB
 0DFH,0DFH,0BFH,0BFH,0EFH,0DF
 H,0DFH,07FH,0EFH
 DB_1_SENAR_1:
 DB
 0FFH,0BFH,0BFH,0BFH,0BFH,0FF
 H,0FFH,0BFH,0FFH
 DB
 0DFH,0DFH,0DFH,0DFH,0DFH,0FF
 H,0FFH,0DFH,07FH
 DB
 07FH,07FH,0FFH,0FFH,0BFH,07F
 H,07FH,0EFH,0BFH
 DB
 0F7H,0F7H,0F7H,0F7H,0F7H,0BF
 H,0F7H,0F7H,0DFH
 DB
 0FFH,0DFH,0FFH,0FFH,0FFH,0DF
 H,0FFH,0FBH,0EFH
 DB
 0FFH,0FFH,0FFH,0FFH,0FFH,0FF
 H,0FFH,0BFH,0F7H
 DB
 07FH,0FFH,07FH,07FH,07FH,07F
 H,07FH,0DFH,0FBH
 DB
 0BFH,07FH,0BFH,0BFH,0BFH,0BF
 H,0FFH,0EFH,0FFH
 DB
 0DFH,0BFH,0DFH,0DFH,0DFH,0DF
 H,0DFH,0F7H,0FFH
 DB
 0FFH,0FFH,0FFH,0FFH,0FFH,0FF
 H,0FFH,0FBH,0FFH
 DB
 0FFH,0FFH,0FFH,0FFH,0FFH,07F
 H,0FFH,0FFH,0FFH
 DB
 07FH,07FH,07FH,07FH,07FH,0BF
 H,07FH,07FH,07FH
 END





LAMPIRAN D

Akord Gitar

	MAYOR	MAYOR 7	MINOR	MINOR 7	SUS 4	AUG (+)	7	9	11
C									
C#									
D									
D#									
E									
F									

Sumber : Bursa Musik Populer Vol.36 thn ke III
www.guitar-pro.com

	MAYOR	MAYOR 7	MINOR	MINOR 7	SUS 4	AUG (+)	7	9	11
F#									
G									
G#									
A									
A#									
B									

Sumber : Bursa Musik Populer Vol.36 thn ke III
www.guitar-pro.com

UNIVERSITAS BRAWIJAYA

LAMPIRAN E

Datasheet Komponen

