

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI  
KUNCI PINTU JARAK JAUH DENGAN TRANSMISI  
GELOMBANG RADIO**

**SKRIPSI**

*Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik*



Disusun oleh :

**VIVI SESMA PUSPITA SARI**

**NIM. 0210630120**

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

repository.ub.ac.id

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI  
KUNCI PINTU JARAK JAUH DENGAN TRANSMISI  
GELOMBANG RADIO**

**SKRIPSI**

*Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik*



Disusun oleh :

**VIVI SESMA PUSPITA SARI**

**NIM. 0210630120**

**DOSEN PEMBIMBING:**

**Ir. Nurussa'adah**  
**NIP. 131 994 339**

**Ir. Nanang Sulistyanto**  
**NIP. 132 090 389**



**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI  
KUNCI PINTU JARAK JAUH DENGAN TRANSMISI  
GELOMBANG RADIO**

Disusun oleh :

**VIVI SESMA PUSPITA SARI**

**NIM. 0210630120**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 14 Juni 2007

**DOSEN PENGUJI:**

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

**Waru Djuriatno, ST. MT.**  
**NIP. 132 158 733**

**R. Arief Setyawan, ST. MT.**  
**NIP. 132 231 713**

**M. Rif'an, ST. MT.**  
**NIP. 132 283 659**

Mengetahui  
Ketua Jurusan Teknik Elektro

**Ir. Heru Nurwarsito, M. Kom**  
**NIP. 131 879 033**

## PENGANTAR

Alhamdulillah, Segala puji syukur kehadiran Allah SWT, atas rahmat, hidayah serta karunia-Nya, sehingga skripsi yang berjudul **“Perancangan dan Pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio”** dapat terselesaikan dengan baik.

Selama penyusunan skripsi ini, tidak terlepas dari bantuan, bimbingan, dorongan dan motivasi dari berbagai pihak. Oleh karena itu, dengan kesungguhan dan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Papa *Basuki*, Mama *Nanik Sukartini*, Kakak *Puspa Dewi Ratna Juwita* sekeluarga dan seluruh keluarga yang selalu mendo'akan dan selalu memberikan dukungan lahir dan bathin.
2. Bapak Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
3. Bapak Rudi Yuwono, ST selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
4. Ibu Ir. Nurussa'adah selaku dosen pembimbing yang telah memberikan bimbingan, saran serta pengarahan dalam penyusunan skripsi ini.
5. Bapak Ir. Nanang Sulistyanto selaku dosen pembimbing yang telah memberikan bimbingan, saran serta pengarahan dalam penyusunan skripsi ini.
6. Mas *Thully Swandaru* yang selalu sabar membimbingku dan menemaniku kemarin, hari ini dan esok. *We'll touch the dusk together*.
7. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
8. Seluruh Asisten dan Laboran Lab. Elka atas bantuan dan bimbingan dalam penyelesaian skripsi ini.
9. Teman-teman angkatan 2002 atas bantuan dan do'anya.
10. *Uching, Hesti, Rizka, Indry, Arik, Dek Fenty* dan Keluarga besar Armaga *is the best family*.
11. Semua pihak yang tidak dapat disebutkan satu persatu, sehingga skripsi ini dapat terselesaikan dengan sebaik-baiknya.

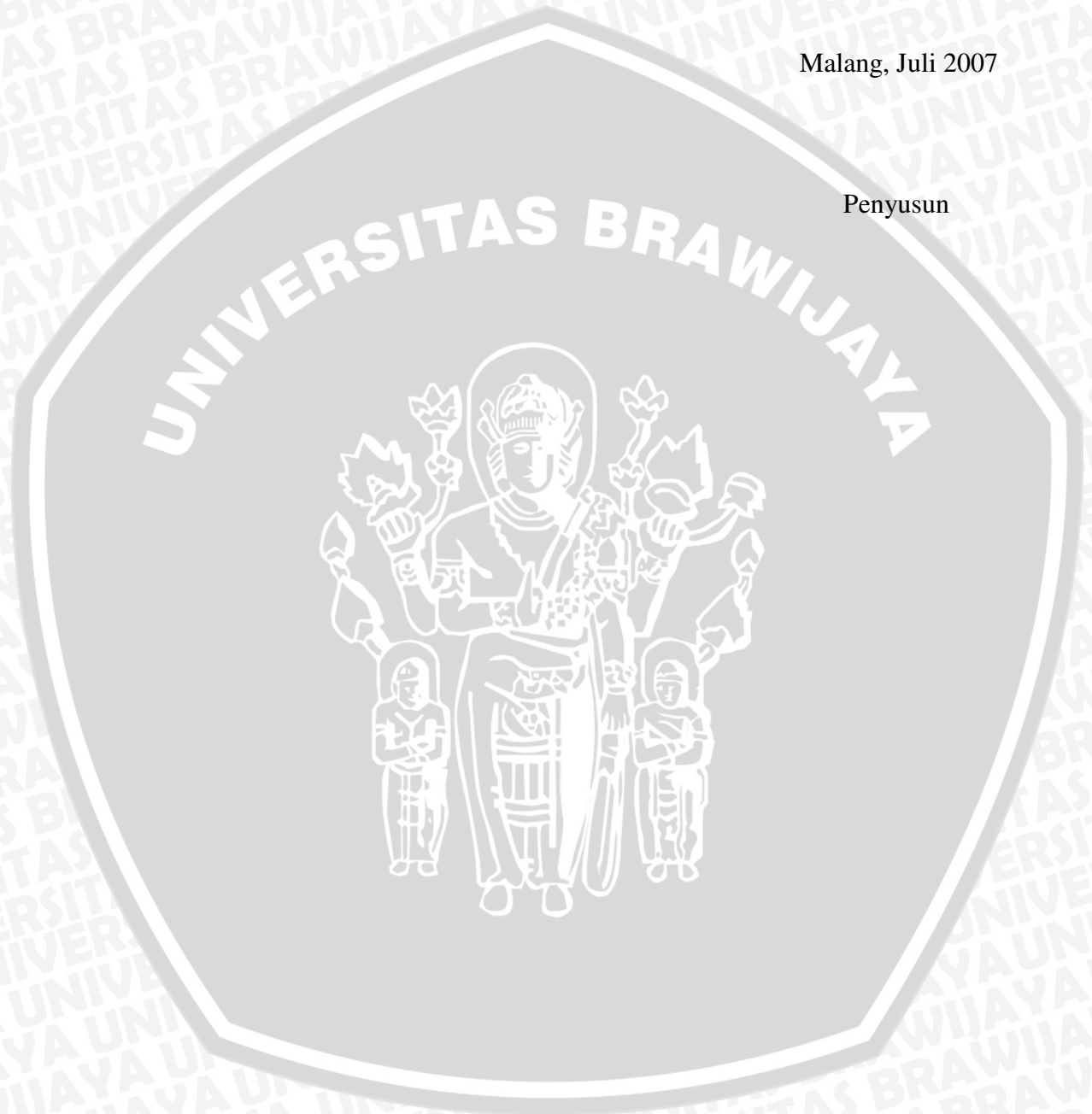
Penyusun menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan serta masih jauh dari sempurna, hal ini tidak lain karena keterbatasan materi

dan pengetahuan yang dimiliki penyusun. Oleh karena itu kritik serta saran sangat diharapkan demi kesempurnaan penyusunan skripsi ini.

Akhir kata, penyusun mengharapkan semoga skripsi ini dapat bermanfaat bagi rekan-rekan mahasiswa dan bagi seluruh pembaca pada umumnya.

Malang, Juli 2007

Penyusun

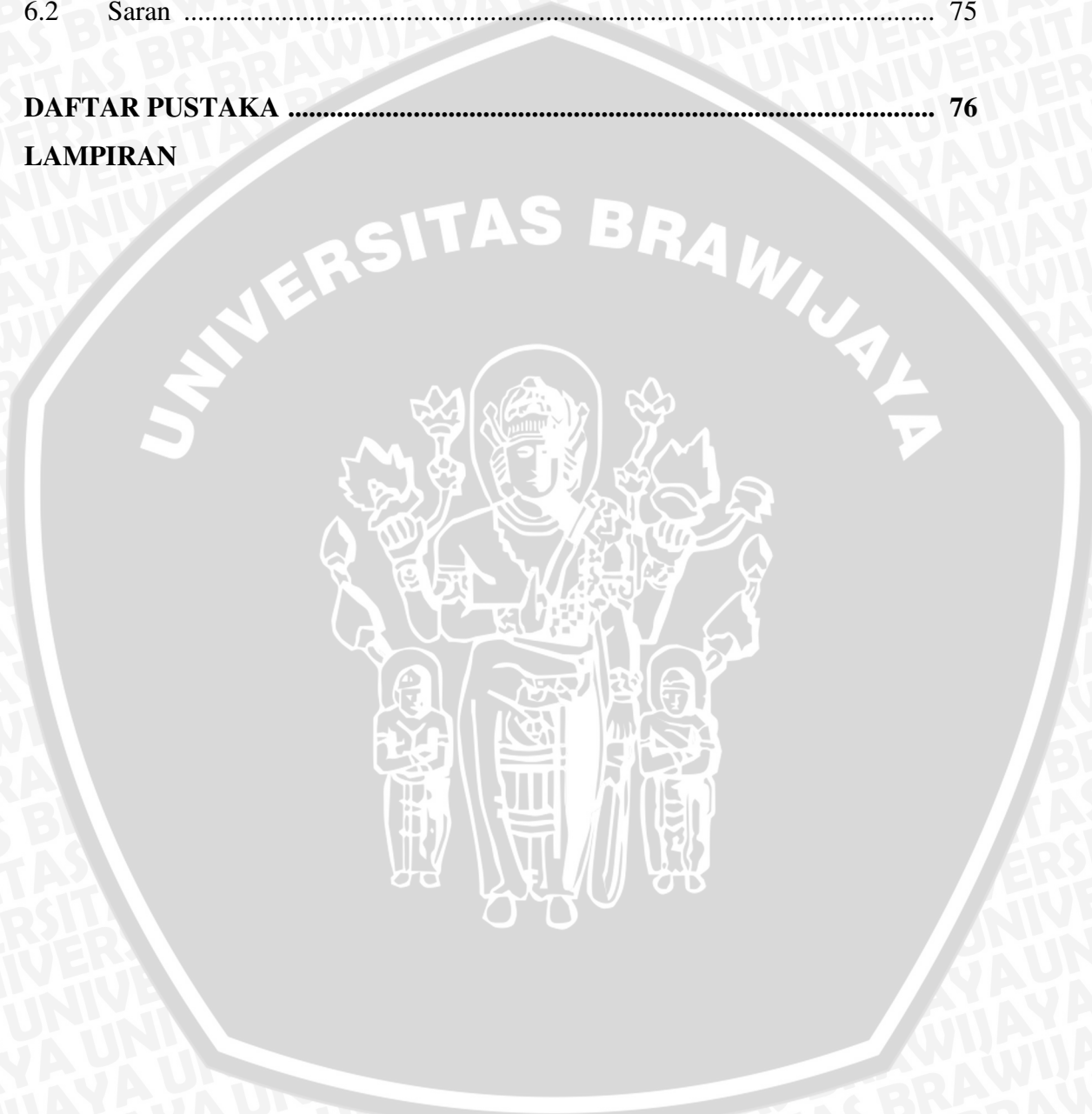


## DAFTAR ISI

<b>Halaman Judul .....</b>	<b>i</b>
<b>Halaman Pengesahan .....</b>	<b>iii</b>
<b>Pengantar .....</b>	<b>iv</b>
<b>Daftar Isi .....</b>	<b>vi</b>
<b>Daftar Gambar .....</b>	<b>ix</b>
<b>Daftar Tabel .....</b>	<b>xi</b>
<b>Daftar Lampiran .....</b>	<b>xii</b>
<b>Abstrak .....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Ruang Lingkup .....	3
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Dasar-dasar Perhubungan Radio.....	5
2.2 Elemen-elemen Sistem Radio.....	6
2.3 Modulasi Frekuensi.....	8
2.4 DTMF ( <i>Dual-Tone Multy-Frequency</i> ).....	10
2.4.1 <i>Encoder DTMF</i> .....	11
2.4.2 <i>Decoder DTMF</i> .....	13
2.5 RTC ( <i>Real Time Clock</i> ) .....	17
2.6 Mikrokontroler AT89S51 .....	19
2.6.1 Konfigurasi Pin .....	20
2.7 <i>Keypad</i> Matriks 4x3 .....	26
2.8 <i>Solenoid</i> .....	27
2.9 <i>D-Latch (Transparent Latch)</i> .....	28
2.10 Pemancar dan Penerima FM .....	29

<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>30</b>
3.1 Studi Literatur .....	30
3.2 Perancangan dan Pembuatan Alat.....	30
3.3 Pengujian Alat.....	31
<b>BAB IV PERENCANAN DAN PEMBUATAN ALAT .....</b>	<b>32</b>
4.1 Spesifikasi Alat .....	32
4.2 Perencanaan Sistem .....	32
4.3 Perencanaan Perangkat Keras ( <i>Hardware</i> ).....	35
4.3.1 Rangkaian DTMF <i>Encoder</i> dan Pemancar FM .....	35
4.3.2 Rangkaian Penerima FM dan DTMF <i>Decoder</i> .....	38
4.3.3 Rangkaian Pengolah Data Unit Pengunci Pintu .....	40
4.3.3.1 Rangkaian Reset .....	42
4.3.3.2 Rangkaian Osilator Kristal .....	44
4.3.4 Rangkaian <i>Real Time Clock</i> (RTC) .....	44
4.3.5 Rangkaian Sensor .....	45
4.3.6 Rangkaian <i>Driver Relay</i> dan <i>Solenoid</i> .....	46
4.3.7 Rangkaian Pengolah Data Unit Pengendali .....	48
4.3.8 Rangkaian <i>Keypad</i> .....	49
4.3.9 Rangkaian <i>D – Latch</i> dan Indikator LED ( <i>Light Emitting Diode</i> ) ....	53
4.3.10 Rangkaian Driver Alarm .....	54
4.4 Perencanaan Perangkat Lunak ( <i>Software</i> ) .....	56
4.4.1 Perencanaan Perangkat Lunak Sistem Pengendalian Kunci Pintu Manual .....	56
4.4.2 Perancangan Perangkat Lunak Sistem Pengontrol Keadaan Pintu ....	59
4.4.3 Perancangan Perangkat Lunak Sistem Penguncian Pintu Secara Otomatis.....	61
<b>BAB V PENGUJIAN DAN ANALISIS DATA .....</b>	<b>63</b>
5.1 Pengujian Rangkaian <i>Keypad</i> .....	63
5.2 Pengujian Rangkaian <i>Encoder</i> DTMF TP5088 .....	64
5.3 Pengujian Rangkaian <i>Decoder</i> DTMF MT8870 .....	65
5.4 Pengujian Pemancar dan Penerima FM .....	66
5.5 Pengujian Rangkaian <i>Real Time Clock</i> (RTC) .....	68

5.6	Pengujian <i>Driver Relay</i> dan <i>Solenoid</i> .....	70
5.7	Pengujian Sistem Secara Keseluruhan .....	71
<b>BAB VI PENUTUP .....</b>		<b>75</b>
6.1	Kesimpulan .....	75
6.2	Saran .....	75
<b>DAFTAR PUSTAKA .....</b>		<b>76</b>
<b>LAMPIRAN</b>		





## DAFTAR GAMBAR

Gambar 2.1	Diagram Blok Pemancar dan Penerima Radio .....	7
Gambar 2.2	Modulasi Frekuensi .....	9
Gambar 2.3	Sinyal DTMF tombol '1' .....	11
Gambar 2.4	Konfigurasi Penyemat IC TP5088 .....	11
Gambar 2.5	Konfigurasi Pin <i>Decoder</i> DTMF IC MT8870 .....	13
Gambar 2.6	Blok Diagram Fungsi <i>Decoder</i> DTMF IC MT8870 .....	14
Gambar 2.7	Diagram Waktu Penerima DTMF IC MT8870 .....	16
Gambar 2.8	Konfigurasi Pin <i>Real Time Clock</i> IC DS1302 .....	17
Gambar 2.9	Konfigurasi Pin RTC IC DS1302 .....	18
Gambar 2.10	Register-register IC RTC DS1302 .....	19
Gambar 2.11	Bentuk Chip Mikrokontroler AT89S51 .....	20
Gambar 2.12	Rangkaian <i>Power-on Reset</i> dan Rangkaian Ekivalen <i>Power-on Reset</i> .....	23
Gambar 2.13	Blok Diagram Mikrokontroler AT89S51 .....	25
Gambar 2.14	Susunan <i>Special Function Register</i> .....	26
Gambar 2.15	<i>Keypad</i> Matriks 4x3 .....	26
Gambar 2.16	<i>Solenoid Construction</i> .....	27
Gambar 2.17	Rangkaian <i>D – Latch</i> .....	28
Gambar 4.1	Blok Diagram Sistem .....	33
Gambar 4.2	Blok diagram Proses <i>Handshaking</i> Unit Pengendali dan Unit Pengunci Pintu .....	34
Gambar 4.3	Rangkaian Transistor <i>Switching</i> Pemancar FM .....	36
Gambar 4.4	Rangkaian Penyesuai <i>Encoder</i> DTMF TP5088 .....	37
Gambar 4.5	Rangkaian Pembangkit DTMF IC TP5088 .....	38
Gambar 4.6	Rangkaian Penerima DTMF IC MT8870 .....	39
Gambar 4.7	Rangkaian Unit Pengendali Sistem Pada Penerima .....	41
Gambar 4.8	Rangkaian Reset .....	43
Gambar 4.9	Rangkaian Osilator .....	44
Gambar 4.10	Rangkaian <i>Real Time Clock</i> (RTC) .....	45
Gambar 4.11	Rangkaian <i>Limitswitch</i> .....	46
Gambar 4.12	Rangkaian <i>Driver Relay</i> dan <i>Solenoid</i> .....	46
Gambar 4.13	Rangkaian Unit Pengolah Data Pada Pengendali .....	48

Gambar 4.14 Rangkaian *Keypad* Sistem Mikrokontroler .....50

Gambar 4.15 Sinyal DTMF Data1 .....51

Gambar 4.16 Sinyal Kosong Antara Data1 dan Data2 .....51

Gambar 4.17 Data DTMF 1 perintah .....52

Gambar 4.18 Rangkaian *D – Latch* Untuk LED Indikator .....54

Gambar 4.19 Rangkaian Driver Alarm .....55

Gambar 4.20 Diagram Alir Utama Pengendali Penguncian Pintu Secara Manual  
oleh *user* .....57

Gambar 4.21 Diagram Alir Proses Pengambilan Data DTMF .....57

Gambar 4.22 Diagram Alir Proses Pengunci Pintu .....58

Gambar 4.23 Diagram Alir Proses Mengaktifkan LED Indikator .....59

Gambar 4.24 Diagram Alir Utama Sistem Pengontrol Keadaan Pintu .....60

Gambar 4.25 Diagram Alir Proses Mengaktifkan Buzzer dan LED .....60

Gambar 4.26 Diagram Alir Proses Tutup Pintu .....61

Gambar 4.27 Diagram Alir Proses Mematikan Alarm .....61

Gambar 4.28 Diagram Alir Utama Sistem Penguncian Pintu Secara Otomatis .....62

Gambar 5.1 Blok Diagram Pengujian *Keypad* .....63

Gambar 5.2 Blok Diagram Pengujian *Encoder* DTMF .....64

Gambar 5.3 Sinyal DTMF Dengan Penekanan Tombol ‘6’ .....65

Gambar 5.4 Blok Diagram Pengujian DTMF .....66

Gambar 5.5 Blok Diagram Pengujian Pemancar dan Penerima FM .....67

Gambar 5.6 Hasil Pengujian Rangkaian Pada Jarak 5 meter .....67

Gambar 5.7 Hasil Pengujian Rangkaian Pada Jarak 10 meter .....68

Gambar 5.8 Hasil Pengujian Rangkaian Pada Jarak 15 meter .....68

Gambar 5.9 Diagram Blok Rangkaian Pengujian RTC DS1302 .....69

Gambar 5.10 Diagram Alir Pengujian RTC DS1302 .....69

Gambar 5.11 Rangkaian Pengujian *Driver Relay* dan *Solenoid* .....71

Gambar 5.12 Kondisi LED Indikator Sebelum Proses Membuka Pengunci Pintu ...72

Gambar 5.13 Kondisi LED Indikator Sesudah Proses Membuka Pengunci Pintu ....73



## DAFTAR TABEL

Tabel 2.1	Band Frekuensi Radio .....	6
Tabel 2.2	Frekuensi <i>Keypad</i> DTMF .....	10
Tabel 2.3	Tabel Kebenaran <i>Encoder</i> DTMF IC TP5088 .....	13
Tabel 2.4	Tabel Fungsi Dekoder DTMF IC MT8870 .....	15
Tabel 2.5	Fungsi – Fungsi Pin IC RTC DS1302 .....	17
Tabel 2.6	Fungsi Pengganti dari Port 3 .....	22
Tabel 4.1	Perencanaan Tombol <i>Keypad</i> .....	50
Tabel 4.2	Konfigurasi Pengiriman Data .....	53
Tabel 4.3	Perencanaan LED Indikator .....	54
Tabel 5.1	Hasil Pengujian <i>Keypad</i> .....	64
Tabel 5.2	Hasil Pengujian <i>Encoder</i> DTMF TP5088 .....	64
Tabel 5.3	Hasil Pengujian <i>Decoder</i> DTMF .....	66
Tabel 5.4	Hasil Pengujian RTC DS1302 Untuk Mengeluarkan Data Menit Pada Port 1 .....	70
Tabel 5.5	Hasil Pengujian RTC DS1302 Untuk Mengeluarkan Data Jam Pada Port 1 .....	70
Tabel 5.6	Hasil Pengujian Rangkaian <i>Driver Relay</i> dan <i>Solenoid</i> .....	71
Tabel 5.7	Hasil Pengujian Sistem Secara Keseluruhan .....	74

## DAFTAR LAMPIRAN

Lampiran 1	Skema Rangkaian .....	L1-1
Lampiran 2	<i>Listing</i> Program .....	L2-1
Lampiran 3	Foto Alat .....	L3-1
Lampiran 4	<i>Datasheet</i> .....	L4-1



## ABSTRAK

VIVI SESMA PUSPITA SARI, Juli 2007. Perencanaan dan Pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio. Skripsi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Malang. Dosen Pembimbing: Ir. Nurussa'adah dan Ir. Nanang Sulistyanto.

Rumah tinggal merupakan hal yang sangat utama bagi individu yang tinggal didalamnya, apalagi jika rumah tersebut sangat luas. Sistem pengunci pintu jarak jauh ini akan memberikan kemudahan bagi *user* mengontrol keamanan rumahnya, terutama dalam hal penguncian pintu. Cukup dengan menekan tombol *keypad* yang telah disediakan, *user* dapat melakukan penguncian pintu dengan mudah dan cepat.

Pengendali kunci pintu ini mempunyai dua bagian penting, yaitu unit pengendali dan unit pengunci pintu. Unit pengendali dioperasikan oleh *user*, sedangkan unit pengunci pintu diletakkan pada masing-masing pintu. Komunikasi antara keduanya menggunakan modulasi frekuensi dengan frekuensi yang sama dan menggunakan sistem *half-duplex*. Unit pengendali terdiri atas pengendali utama, sistem pengkode data dan sekaligus dekoder data yang diterima. Pemancar dan penerima FM sebagai sarana pengiriman data dari unit pengendali ke unit pengunci pintu atau sebaliknya. *Keypad* digunakan sebagai data masukan sesuai dengan perintah *user*. LED dan buzzer digunakan sebagai indikator operasi yang dilakukan. Sedangkan unit pengunci pintu terdiri atas pengendali utama dan juga sistem pengkodean data yang keduanya mempunyai fungsi yang sama seperti dalam unit pengendali. Selain itu, di dalam unit pengendali terdapat RTC sebagai pengatur waktu, mekanik pengunci pintu dan dua sensor berupa limitswitch untuk mengetahui keadaan pintu dan keadaan pengunci pintu.

Sistem yang dirancang ini mampu mengendalikan 10 pintu dengan jarak jangkauan maksimum 10 meter. Sistem ini disertai dengan alarm indikator keadaan pintu. Dimana jika sensor pintu pada unit pengunci pintu menunjukkan pintu dalam keadaan terbuka, maka data tersebut akan dikirimkan ke unit pengendali untuk mengaktifkan buzzer sebagai alarm. Selain itu, sistem ini juga dilengkapi dengan penguncian pintu secara otomatis pada pukul 9 malam.

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Sistem pengamanan termasuk prioritas utama dalam kehidupan. Bagi setiap pemilik rumah tinggal yang mempunyai luas cukup besar, seringkali mengalami permasalahan dalam pengamanan khususnya dalam sistem penguncian pintu. Untuk membuka atau menutup pengunci pintu tersebut, pemilik rumah harus berjalan mendekati masing-masing pintu rumah. Hal ini tidak akan menjadi suatu masalah jika jarak pintu saling berdekatan. Semakin luas rumah yang dimiliki, maka akan semakin jauh jarak masing-masing pintu dan jika tetap menggunakan pengunci pintu *konvensional* untuk mengontrol semua pintu apakah pengunci pintu dalam keadaan terkunci atau tidak, tentunya akan memerlukan waktu yang lama.

Untuk menyikapi permasalahan tersebut, maka banyak usaha yang dapat dilakukan. Saat ini, banyak peralatan elektronika dirancang untuk memenuhi kebutuhan masyarakat akan sistem pengamanan terutama sistem pengamanan rumah yang *modern*, mulai dari sistem pengendali penguncian pintu dari satu tempat dengan transmisi data secara langsung yaitu dengan menggunakan kabel atau pengendali penguncian pintu secara *wireless*, misalnya sistem pengendali jarak jauh dengan inframerah. Tetapi sistem pengendali tersebut mempunyai banyak kelemahan, antara lain kerumitan instalasi kabel sebagai media pengiriman data, juga kesulitan pengontrolan jika terjadi kerusakan. Jika menggunakan sistem pengendali dengan inframerah pengguna harus mengarahkan pengendalinya ke sensor penerima, selain itu panjang gelombang inframerah hanya sebesar 0,75-1000 mikron. Untuk mengatasi kelemahan tersebut maka dilakukan berbagai pengembangan, salah satunya adalah remote kontrol yang memanfaatkan gelombang radio untuk pengiriman data.

Teknologi remote kontrol dengan transmisi gelombang radio yang dapat mengendalikan berbagai macam peralatan elektronika telah dikenal dan banyak dimanfaatkan dalam setiap kebutuhan. Dengan adanya remote kontrol tersebut pengguna tidak perlu mendekati peralatan elektronika untuk menekan tombol-tombol sesuai dengan kebutuhan. Hanya dengan menekan tombol-tombol dalam remote kontrol yang telah diatur fungsinya, kita dapat lebih mudah dan praktis menjalankan piranti apapun. Selain itu, keuntungan menggunakan remote kontrol dengan transmisi gelombang radio ini antara lain karena gelombang radio mempunyai panjang

gelombang yang cukup jauh yaitu mencapai 1-10 meter, gelombang radio dapat menembus benda padat misalnya dinding, sehingga sangat memungkinkan untuk memanfaatkan gelombang radio sebagai pengendali jarak jauh.

Memperhatikan dari permasalahan tersebut di atas, maka akan dibuat suatu model/*prototype* pengendali kunci pintu jarak jauh yang digunakan sebagai sistem keamanan rumah yang memanfaatkan gelombang radio. Pada dasarnya ada beberapa proses modulasi sinyal dalam pentransmisian data, misalnya modulasi amplitudo (AM). Modulasi amplitudo (AM) ini mempunyai bandwidth yang lebar, namun modulasi amplitudo ini sensitif terhadap noise (Chattopadhyay, 1989: 370-371). Selain itu daya yang diperlukan pemancar AM juga cukup besar. Sistem modulasi yang lain adalah modulasi frekuensi (FM). Jika dibandingkan dengan modulasi amplitudo, modulasi frekuensi ini mempunyai bandwidth yang lebih lebar. Modulasi frekuensi ini juga memberikan beberapa kelebihan antara lain: lebih tahan terhadap noise, daya yang dibutuhkan pemancar juga lebih kecil (Simanjuntak, 1993: 89-90). Dengan demikian, proses modulasi yang dipilih untuk komunikasi data dalam sistem ini adalah sistem modulasi frekuensi (FM).

Sinyal yang akan dikirimkan melalui pemancar FM tersebut adalah sinyal suara yang merupakan kode-kode suara. Kode-kode tersebut dapat diperoleh dari sistem pengkodean yang biasa digunakan, yaitu DTMF (*Dual Tone Multy-Frequency*). DTMF merupakan salah satu sistem pengkodean data yang terbentuk dari dua frekuensi yang dapat menghasilkan 16 sistem konfigurasi pengkodean yang akurat. Pengiriman data pada DTMF ini secara paralel. Komponen DTMF ini murah dan rangkaian didalamnya juga cukup sederhana. Dengan demikian, sistem pengkodean yang digunakan dalam sistem ini adalah DTMF (*Dual Tone Multy-Frequency*).

Agar dapat bekerja dengan baik, dalam sistem ini diperlukan pengaturan pengkodean data-data masukan demi keamanan pengiriman data, yaitu adanya kode-kode DTMF selain data yang diinginkan oleh *user*. Diperlukan juga sistem pengaturan sensor yang selanjutnya digunakan untuk mengaktifkan indikator masing-masing pengunci pintu yang berguna untuk memberikan kemudahan bagi pengguna mengetahui keadaan pengunci pintu rumahnya dengan mudah, apakah pengunci pintu rumahnya dalam keadaan terkunci atau membuka tanpa harus mendatangi masing-masing pintu. Sehingga dapat dilihat bahwa proses yang akan dijalankan cukup rumit. Untuk mengatasi permasalahan tersebut diperlukan mikrokontroler sebagai pusat pengendali seluruh sistem.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka rumusan masalah yang akan dibahas antara lain sebagai berikut:

1. Bagaimana merancang sistem pengkodean masing-masing pintu.
2. Bagaimana merancang sistem pengiriman data dengan memanfaatkan transmisi gelombang radio.
3. Bagaimana merancang sistem pengkodean sehingga dapat mengaktifkan kunci elektronik.
4. Bagaimana merancang sistem pengunci elektronik sehingga dapat memberikan indikasi kepada pengguna.

## 1.3 Ruang Lingkup

Dalam perencanaan dan pembuatan alat yang berdasarkan dari permasalahan yang ada, maka pembatasan masalah yang diajukan dalam skripsi ini antara lain:

1. Sistem pengkodean menggunakan DTMF.
2. Pembahasan ditekankan pada pembuatan sistem serta proses transfer data melalui gelombang radio.

## 1.4 Tujuan

Tujuan penyusunan skripsi ini adalah untuk merealisasikan sebuah alat yang dapat memudahkan pengguna untuk melakukan penguncian pintu dalam jarak jauh dan sekaligus memberikan kemudahan untuk dapat memeriksa keadaan pengunci pintu setiap saat melalui satu sistem yaitu Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio.

## 1.5 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

### **BAB I PENDAHULUAN**

Memuat Latar Belakang, Rumusan Masalah, Ruang Lingkup Permasalahan, Tujuan dan Sistematika Penulisan.

### **BAB II TINJAUAN PUSTAKA**

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.



### **BAB III METODOLOGI PENULISAN**

Berisi tentang metode penelitian dan perencanaan alat serta pengujian.

### **BAB IV PERENCANAAN DAN PEMBUATAN ALAT**

Membahas perancangan dan perealisasiian Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio.

### **BAB V PENGUJIAN ALAT**

Memuat hasil pengujian terhadap alat yang telah direalisasikan.

### **BAB VI PENUTUP**

Memuat kesimpulan dan saran-saran.

UNIVERSITAS BRAWIJAYA



## BAB II

### TINJAUAN PUSTAKA

Perancangan dan Pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio ini akan bekerja jika pada salah satu tombol pengontrol (*remote*) ditekan dan kemudian mengirimkan frekuensi dari gelombang radio yang menunjukkan kode yang spesifik. Frekuensi tersebut akan diterima oleh penerima dan diolah menjadi kode biner yang selanjutnya dapat dikirimkan ke perintah yang sesuai.

Dalam perancangan dan pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio ini sangat dibutuhkan pemahaman tentang berbagai hal yang mendukung. Pemahaman ini bermanfaat untuk merancang perangkat keras dan perangkat lunak dari sistem yang akan dirancang. Dasar teori yang mendukung pemahaman tersebut antara lain:

1. Dasar-dasar Perhubungan Radio
2. Elemen-elemen Sistem Radio
3. Modulasi Frekuensi (FM)
4. DTMF (*Dual-Tone Multi-Frequency*)
5. RTC (*Real Time Clock*)
6. Mikrokontroler AT89S51
7. *Keypad Matriks 4x3*
8. *Solenoid*
9. *D – Latch (Transparent Latch)*
10. Pemancar dan Penerima FM

#### 2.1 Dasar - dasar Perhubungan Radio

Sistem komunikasi memancarkan informasi dalam bentuk sinyal listrik yang menyajikan pembicaraan, musik, gambar, televisi, data ilmiah, bisnis dan sebagainya. Bentuk gelombang dari sinyal ini sangat kompleks dan selalu berubah, tetapi spektrum frekuensi sinyal-sinyal tersebut biasanya terbatas pada lebar pita tertentu, baik oleh alam dari sumber sinyal ataupun oleh filter-filter dalam peralatan transmisi. Karena sinyal ini mencakup pita frekuensi yang melebar sampai ke beberapa hertz, maka sinyal tersebut tidak dapat dipancarkan dalam bentuk aslinya melewati lintasan transmisi biasa oleh karena tidak mungkin untuk memisahkannya pada ujung penerima. Suatu saluran transmisi terpisah atau jalur radio terpisah untuk setiap sinyal tidak layak, baik dari segi

ekonomi maupun dari segi praktis. Karena itu sistem komunikasi keseluruhan harus memungkinkan adanya suatu transmisi simultan dari sejumlah sinyal, baik dengan menggeserkan mereka dalam beberapa bagian yang berbeda dari spektrum frekuensinya ataupun dengan cara mengirimkan sampel-sampel sinyal atas dasar penggiliran waktu. (Krauss dan Bostian, 1990:1-2)

Dalam perhubungan radio, mengingat gelombang elektromagnetis dipancarkan pada ruang yang dipergunakan secara bersama, maka sangat besar kemungkinannya untuk terjadi saling mengganggu. Agar gangguan ini dapat dicegah, maka alokasi frekuensi ditentukan bagi tiap daerah, sehingga perhubungan radio dapat terawasi. Klasifikasi frekuensi-frekuensi radio ditunjukkan dalam Tabel 2.1.

**Tabel 2.1** Band Frekuensi Radio

Nama	Frekuensi	Panjang Gelombang	Nama
<i>Very Low Frequency</i>	VLF	Kurang dari 30 kHz	Gelombang Myriametrik
<i>Low Frequency</i>	LF	30-300 kHz	Gelombang kilometer
<i>Medium Frequency</i>	MF	300-3000 kHz	Gelombang hektometer
<i>High Frequency</i>	HF	3-30 MHz	Gelombang decameter
<i>Very High Frequency</i>	VHF	30-300 MHz	Gelombang meter
<i>Ultra High Frequency</i>	UHF	300-3000 MHz	Gelombang decimeter
<i>Super High Frequency</i>	SHF	3-30 GHz	Gelombang sentimeter
<i>Extremely high Frequency</i>	EHF	30-300 GHz	Gelombang millimeter

-----  
**Sumber:** Suhana dan Shoji, 2002:183-184

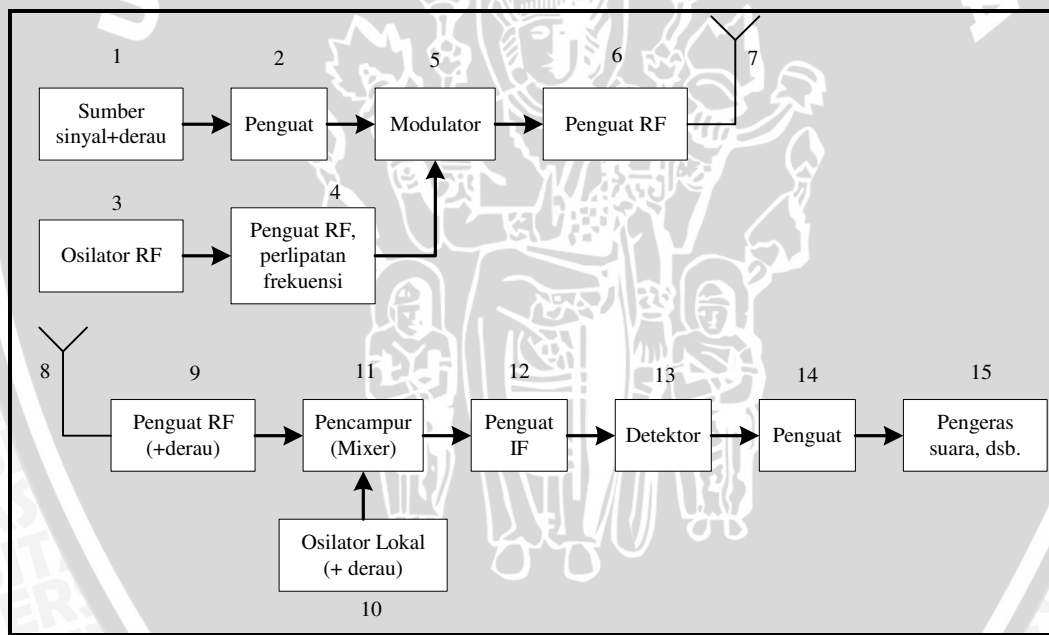
## 2.2 Elemen-elemen Sistem Radio

Suatu proses dimana pesan asli diubah menjadi suatu bentuk baru yang cocok untuk transmisi radio dinamakan modulasi. Proses modulasi mengakibatkan adanya beberapa sifat (seperti amplitudo, frekuensi atau fase) dari pembawa berfrekuensi tinggi, yang harus diubah dari harga-harga tanpa modulasi sebesar harga yang sebanding dengan harga sesaat sinyal pemodulasi. Dalam penerima, proses ini dibalikkan dalam detektor yang menemukan kembali sinyal asli.

Gambar 2.1 menunjukkan diagram blok yang disederhanakan dari pemancar dan penerima radio untuk menggambarkan pemrosesan sinyal yang terjadi. Fungsi masing-masing blok diterangkan di bawah ini.

1. Sumber dari sinyal pesan mungkin suatu mikrofon, *pick up* atau kamera televisi, atau alat lain yang mengubah informasi yang diinginkan menjadi sinyal listrik.
2. Sinyal tersebut diperkuat dan sering dilewatkan melalui filter lewat bawah (*low pass*) untuk membatasi lebar pita.

3. Osilator RF menentukan frekuensi pembawa atau kelipatannya.
4. Suatu atau beberapa tingkat penguat menaikkan tingkat daya sinyal dari osilator ke harga yang diperlukan untuk masuk ke modulator.
5. Modulator menggabungkan sinyal dan komponen-komponen frekuensi pembawa untuk menghasilkan salah satu jenis gelombang termodulasi.
6. Penguatan tambahan mungkin diperlukan setelah modulasi untuk membawa tingkat daya sinyal pada harga masukan ke antena yang diinginkan.
7. Antena pemancar mengubah energi RF menjadi gelombang elektromagnet dengan polarisasi yang diinginkan.
8. Antena penerima dapat bersifat omnidireksional (ke segala arah) untuk maksud pelayanan umum ataupun sangat terarah untuk komunikasi titik ke titik.
9. Tingkat penguat RF menaikkan daya sinyal ke tingkat yang cocok untuk masukan ke pencampuran (*mixer*) dan membantu mengisolasi osilator lokal dari antena.



**Gambar 2.1** Diagram Blok Pemancar dan Penerima Radio

Sumber: Krauss dan Bostian, 1990: 3

10. Osilator lokal dalam penerima ditala untuk menghasilkan frekuensi  $f_{LO}$  yang berbeda dengan frekuensi sinyal datang  $f_{RF}$  sebesar frekuensi intermedia (perantara)  $f_{IF}$ ; yakni  $f_{LO}$  dapat sama dengan  $f_{RF} + f_{IF}$  atau  $f_{RF} - f_{IF}$ .

11. Pencampuran merupakan alat tidak linear yang menggeserkan sinyal yang diterima pada  $f_{RF}$  ke frekuensi intermedia  $f_{IF}$ . Modulasi pada pembawa yang diterima juga diubah ke frekuensi intermedia.
12. Penguat  $IF$  menaikkan sinyal ke tingkat yang cocok untuk deteksi dan menyediakan sebagian besar pemilahan frekuensi yang diperlukan untuk “melewatkan” sinyal yang diperlukan dan menyaring keluar (*filter*) sinyal-sinyal yang tidak diinginkan yang terdapat dalam keluaran pencampur.
13. Detektor memulihkan sinyal pesan asli dari masukan  $IF$  termodulasi.
14. Penguat audio dan video menaikkan tingkat daya keluaran detektor ke harga yang cocok untuk menggerakkan pengeras suara, tabung televisi, atau alat keluaran lainnya.
15. Alat keluaran mengubah informasi sinyal kembali ke bentuk aslinya (gelombang suara, gambar dan sebagainya).

Sebagai tambahan ke sinyal yang diinginkan yang diproses oleh penerima, derau listrik ditambahkan dalam alur transmisi, dan dibangkitkan dalam penguat RF, osilator lokal, pencampuran dan lain-lainnya.

### 2.3 Modulasi Frekuensi

Modulasi frekuensi dihasilkan kalau penyimpangan  $\delta\omega$  dari frekuensi sesaat  $\omega(t)$  dari frekuensi pembawa  $\omega_c$  berbanding langsung dengan amplitudo sesaat tegangan pemodulasi.

Karena

$$\omega(t) = \frac{d\phi}{dt} = \omega_c + \frac{d\theta(t)}{dt} \quad (2-1)$$

penyimpangan  $\delta\omega$  dari  $\omega(t)$  dari  $\omega_c$  diberikan oleh

$$\delta\omega(t) = \omega(t) - \omega_c = \frac{d\theta(t)}{dt} \quad (2-2)$$

Dalam modulasi frekuensi  $\delta\omega(t)$  dibuat sebanding dengan tegangan pemodulasi  $v_m(t)$ ; yakni

$$\delta\omega(t) = k_\omega v_m(t) \quad (2-3)$$

di mana  $k_\omega$  merupakan kepekaan (sensitivitas) modulator dalam rad/det/V.

Karena  $\theta(t)$  dan  $\delta\omega$  dihubungkan oleh Persamaan (2-2), maka

$$\theta(t) = \int_0^t k_\omega v_m(t) dt + \theta(0) \quad (2-4)$$

Bentuk gelombang yang dimodulasikan diberikan oleh

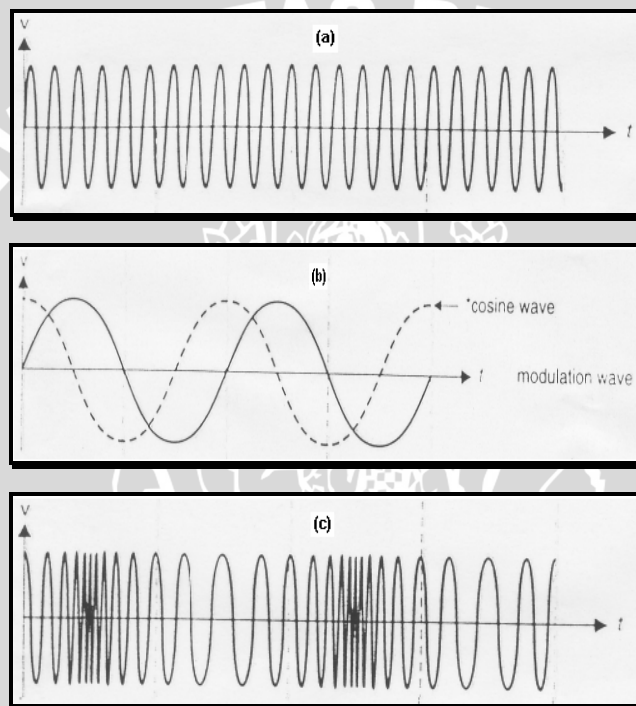
$$F(t) = A(t) \cos[\omega_c t + \theta(t)] = A(t) \cos \phi(t) \quad (2-5)$$

Jika Persamaan (2-4) disubstitusikan ke dalam Persamaan (2-5) dengan  $\theta(0)$  dimisalkan sama dengan nol maka menghasilkan persamaan untuk gelombang termodulasi frekuensi:

$$F_{FM}(t) = A \cos[\omega_c t + k_f \int_0^t v_m(t) dt] \quad (2-6)$$

(Krauss dan Bostian, 1990: 262)

Gelombang yang termodulasi frekuensi ditunjukkan dalam Gambar 2.2.



**Gambar 2.2** Modulasi Frekuensi (a). Gelombang Pembawa (*carrier*) (b). Gelombang yang termodulasi frekuensi (c). Gelombang FM

**Sumber:** Hioki, 1996: 50

Dalam perancangan “Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio” ini sistem modulasi yang digunakan adalah modulasi frekuensi, karena jika dibandingkan dengan modulasi amplitudo sistem modulasi frekuensi ini mempunyai beberapa keuntungan. Modulasi frekuensi (FM) timbul dari upaya menghilangkan kekurangan dari modulasi amplitudo. Dalam FM, amplitudo gelombang dijaga tetap. Jadi perubahan dalam amplitudo karena derau dihilangkan pada pendeteksian. Akibatnya penerimaan FM jauh kurang deraunya dan dengan kualitas yang sama dalam penerimaan, daya pemancar untuk FM jauh lebih sedikit

dibandingkan daya untuk AM. Lebar pita saluran FM jauh lebih besar daripada lebar saluran AM, selain itu FM juga memungkinkan reproduksi *Hi-Fi* untuk suara dan musik sehingga FM menjadi sarana komunikasi dengan kemurnian tinggi (*Hi-Fi*). (Chattopadhyay, 1989 : 370-371).

#### 2.4 DTMF (*Dual-Tone Multi-Frequency*)

DTMF merupakan kependekan dari *Dual Tone Multi Frequency*. DTMF merupakan sebuah tone yang terbentuk dari dua lapisan frekuensi. Masing-masing merupakan frekuensi pilihan yang mudah untuk desain filter dan mudah untuk mengirimkan *tone-tone* pada sebuah line telepon. DTMF didesain untuk mengirimkan kode-kode yang menggunakan mikrofon. Masing-masing beep (atau digit yang ditekan pada telepon) diubah menjadi dua frekuensi secara bersama-sama, yang ditumpangkan pada amplitudo.

*Keypad* DTMF berbentuk matrik 4x4, dengan masing-masing baris menyatakan frekuensi rendah, dan masing-masing kolom menyatakan frekuensi tinggi. Dengan menekan sebuah tombol, misalnya tombol ‘1’ akan mengirimkan sebuah sinyal *sinusoida* dari kedua frekuensi 697 Hz dan 1209 Hz. Kedua sinyal merupakan alasan mengapa mereka disebut multifrekuensi. Sinyal ini kemudian didekodekan oleh pensaklaran untuk mengetahui tombol mana yang ditekan. Tabel 2.2 menunjukkan frekuensi *keypad* DTMF.

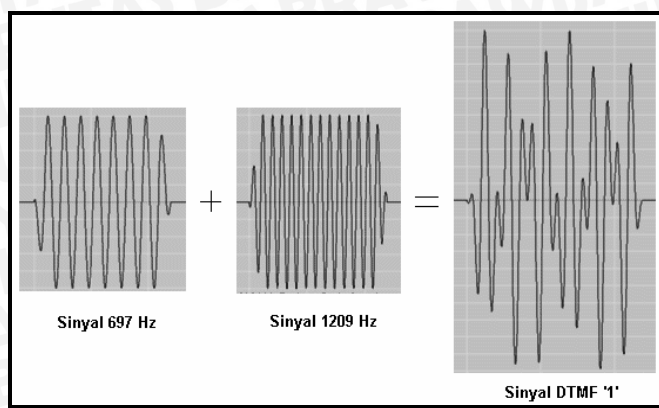
Tabel 2.2 Frekuensi *Keypad* DTMF

		Upper Band			
		1209 Hz	1336 Hz	1477 Hz	1633 Hz
Lower Band	697 Hz	1	2	3	A
	770 Hz	4	5	6	B
	852 Hz	7	8	9	C
	941 Hz	*	0	#	D

Frequency/Key Matrix for DTMF Key Pads

Sumber: Hissen, 2007: 1

Sesuai dengan Tabel 2.2, untuk menghasilkan sinyal DTMF pada tombol ‘1’ harus mencampurkan sebuah sinyal murni 697 Hz dengan sinyal murni 1209 Hz, sinyal DTMF dengan penekanan pada tombol ‘1’ ditunjukkan dalam Gambar 2.3.

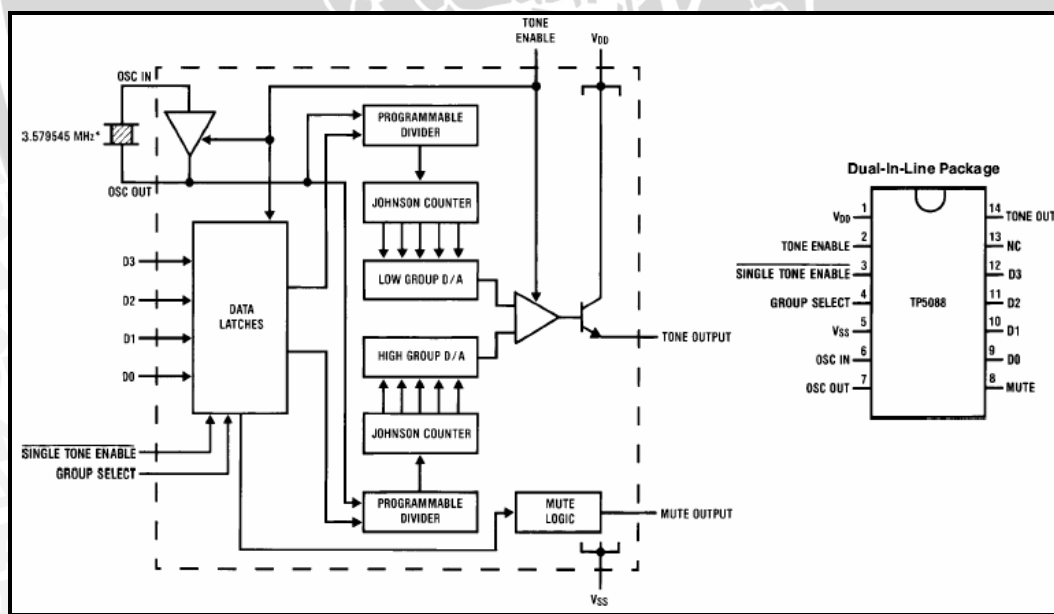


**Gambar 2.3** Sinyal DTMF tombol '1'

**Sumber:** Hissen, 2007: 1

### 2.4.1 Encoder DTMF

Dalam perancangan sistem ini digunakan IC *encoder* DTMF tipe TP5088. DTMF IC TP5088 ini merupakan jenis *encoder* DTMF yang merupakan perangkat CMOS yang dapat menghasilkan *tone-dialing* yang dalam aplikasinya dapat digunakan pada hubungan telepon. Data masukan akan dikodekan secara langsung tanpa memerlukan konversi untuk mensimulasikan masukan yang dibutuhkan oleh pembangkit DTMF standart. IC *encoder* DTMF ini mempunyai 14 penyemat dengan konfigurasi penyemat seperti ditunjukkan dalam Gambar 2.4 berikut ini.



**Gambar 2.4** Konfigurasi Penyemat IC TP5088

**Sumber:** National Semiconductor, 1991: 1-3



Secara keseluruhan, fungsi beberapa penyemat IC TP5088 *encoder* DTMF adalah sebagai berikut:

a.  **$V_{DD}$  dan  $V_{SS}$**

Merupakan penyemat catu daya,  $V_{DD}$  dihubungkan dengan catu daya positif dan  $V_{SS}$  pada *ground*.

b. ***Osc In* dan *Osc Out***

Kaki ini dihubungkan dengan kristal atau sumber osilator dari luar. Nilai kristal yang digunakan biasanya sebesar 3.579545 MHz.

c. ***Tone Enable***

Masukan ini mempunyai sebuah resistor *pull-up internal*. Ketika *tone-enable* dalam kondisi *low*, osilator akan dihentikan, sehingga *tone-generator* non aktif dan tidak ada keluaran pada transistor. Perubahan logika “0” ke “1” pada *tone-enable* menahan data dari D0-D3. Osilator akan mulai bekerja dan *tone-generator* kontinyu sampai *tone-enable* kembali ke keadaan *low*.

d. ***Mute***

Keluaran ini berfungsi untuk mengeluarkan arus ke  $V_{SS}$  pada saat *tone-enable* dalam kondisi *low* dan tidak ada *tone* yang dibangkitkan. Pin ini berlogika “0” pada saat *tone-enable high*.

e. **D0-D3**

Pin ini merupakan masukan untuk data biner yang dikodekan, yang ditahan pada tepi naik oleh *tone-enable*.

f. ***Tone Out***

Keluaran ini merupakan *open-emitter* dari transistor NPN, kolektor internal dihubungkan ke  $V_{DD}$ . Ketika beban resistor eksternal dihubungkan *tone out* ke  $V_{SS}$ , maka tegangan pada kaki ini merupakan penjumlahan dari *group tone low* dan *high*. Pada saat tidak membangkitkan *tone output* transistor ini mati.

g. ***Single Tone Enable***

Masukan ini mempunyai resistor *pull-up internal*. Pada keadaan normal kaki ini dihubungkan ke  $V_{DD}$ .

h. ***Group Select***

Kaki ini digunakan untuk menyeleksi *group frekuensi* atas atau bawah, ketika perangkat ini dalam mode *tone tunggal*.

Tabel fungsi proses perubahan data biner menjadi sinyal DTMF dari IC TP5088 ditunjukkan dalam Tabel 2.3.

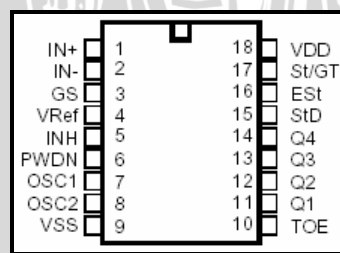
Tabel 2.3 Tabel Kebenaran *Encoder* DTMF IC TP5088

Keyboard Equivalent	Data Inputs				TONE ENABLE	TONES OUT		MUTE
	D3	D2	D1	D0		f <sub>L</sub> (Hz)	f <sub>H</sub> (Hz)	
X	X	X	X	X	0	0V	0V	0V
1	0	0	0	1	↘	697	1209	O/C
2	0	0	1	0	↘	697	1336	O/C
3	0	0	1	1	↘	697	1477	O/C
4	0	1	0	0	↘	770	1209	O/C
5	0	1	0	1	↘	770	1336	O/C
6	0	1	1	0	↘	770	1477	O/C
7	0	1	1	1	↘	852	1209	O/C
8	1	0	0	0	↘	852	1336	O/C
9	1	0	0	1	↘	852	1477	O/C
0	1	0	1	0	↘	941	1336	O/C
*	1	0	1	1	↘	941	1209	O/C
#	1	1	0	0	↘	941	1477	O/C
A	1	1	0	1	↘	697	1633	O/C
B	1	1	1	0	↘	770	1633	O/C
C	1	1	1	1	↘	852	1633	O/C
D	0	0	0	0	↘	941	1633	O/C

Sumber: National Semiconductor, 1991: 4

#### 2.4.2 Decoder DTMF

*Decoder* DTMF adalah rangkaian yang dapat merespon kode-kode sinyal DTMF ke dalam bentuk sinyal yang diinginkan. Salah satunya jenis *decoder* DTMF dalam bentuk IC 18 pin adalah IC MT8870. Fungsinya adalah untuk mendeteksi atau menterjemahkan sinyal suara menjadi kode biner 4 bit. Gambar 2.5 menunjukkan konfigurasi penyemat dari *decoder* DTMF IC MT8870.



Gambar 2.5 Konfigurasi Pin *Decoder* DTMF IC MT8870

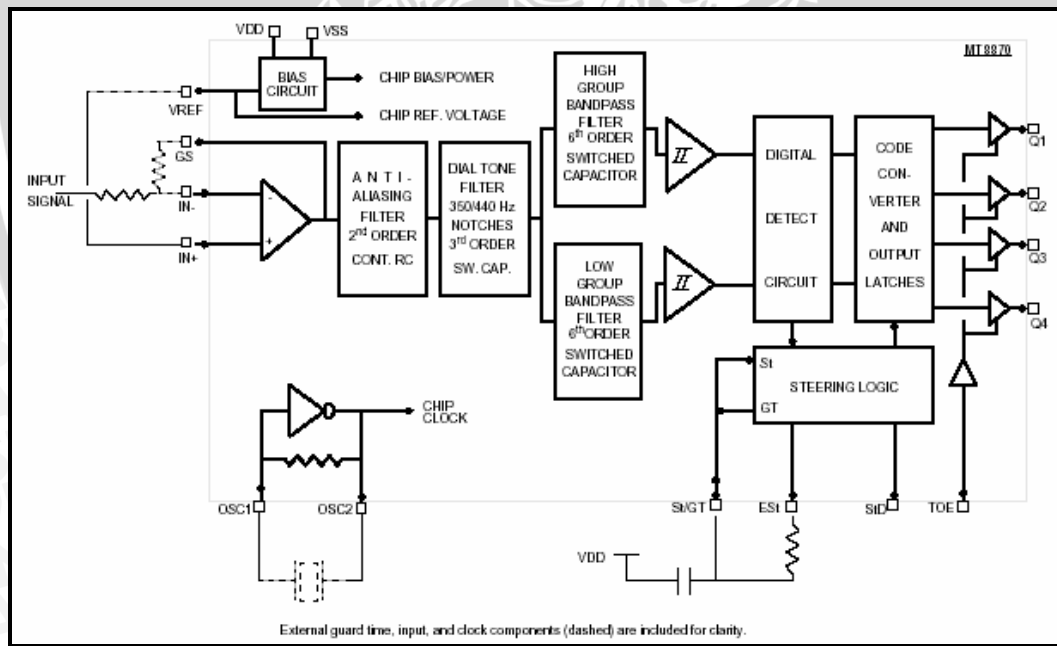
Sumber: Mitel, 1995: 2

Fungsi beberapa penyemat secara keseluruhan dari *decoder* DTMF IC MT8870 adalah sebagai berikut:

- a. **IN+** = *Non inverting input op-amp*
- b. **IN-** = *Inverting input op-amp*

- c. **GS** = Menghubungkan keluaran penguat differensial dengan resistor umpan balik.
- d. **Vref** = *Voltage* referensi (keluaran), besarnya  $V_{DD}/2$ .
- e. **INH** = Mengeluarkan data DTMF A, B, C, D
- f. **PWDN** = *Power down* (masukan), aktif *high*
- g. **OSC 1** = *Clock Input*
- h. **OSC 2** = *Clock output*, frekuensi keluaran 3.579545 MHz.
- i. **Vss** = *Ground*, 0 V.
- j. **TOE** = *Tristate Output Enable*, jika logika *high* mengeluarkan data Q1-Q4.
- k. **Q1-Q4** = Jika TOE aktif, memberikan kode DTMF yang diterima (Tabel 2.4)  
Jika TOE logika *low*, data keluaran adalah *high impedance*.
- l. **StD** = Jika logika tinggi pada pin ini, maka data keluaran valid.
- m. **ESt** = Saat nilai masukan besar, maka keluaran logika tinggi.
- n. **St/GT** = *Steering/Guard Time*, menentukan kecepatan nada.
- o. **V<sub>DD</sub>** = Supply tegangan, +5V.

Gambar 2.6 merupakan blok diagram fungsi dari *decoder* DTMF IC MT8870.



**Gambar 2.6** Blok Diagram Fungsi *Decoder* DTMF IC MT8870

Sumber: Mitel, 1995: 1

MT8870 merupakan chip tunggal penerima DTMF yang ditambah dengan teknologi filter kapasitor-saklar dan perhitungan algoritma digital untuk periode pengukuran.

Untuk membantu fleksibilitas desain, pertama kali sinyal masukan DTMF di *buffer* oleh masukan op-amp untuk menyesuaikan penguatan dan memilih konfigurasi masukan. Masukan melalui filter aktif RC *low pass* untuk menjalankan fungsi yang sebenarnya. Penekanan *tone* 350Hz dan 440Hz kemudian ditolak filter takik kapasitor-saklar orde ketiga. Sinyal, tetap dalam bentuk berkebalikan, kemudian dipisah kedalam komponen frekuensi *low* dan *high* oleh dua kapasitor-saklar orde enam dan *pass filter*. Masing-masing komponen *tone* kemudian disaring oleh filter keluaran dan diolah menjadi sinyal kotak oleh komparator dengan batas yang kuat.

Dua sinyal kotak yang dihasilkan diletakkan pada rangkaian digital yang merupakan suatu pengukuran algoritma hitungan dan periode rata-ratanya. Referensi clock yang akurat diperoleh dari kristal *external* 3.58MHz.

**Tabel 2.4** Tabel Fungsi Dekoder DTMF IC MT8870

Digit	TOE	INH	Est	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L	undetected, the output code will remain the same as the previous detected code			
B	H	H	L				
C	H	H	L				
D	H	H	L				

L = Logika Low

H = Logika High

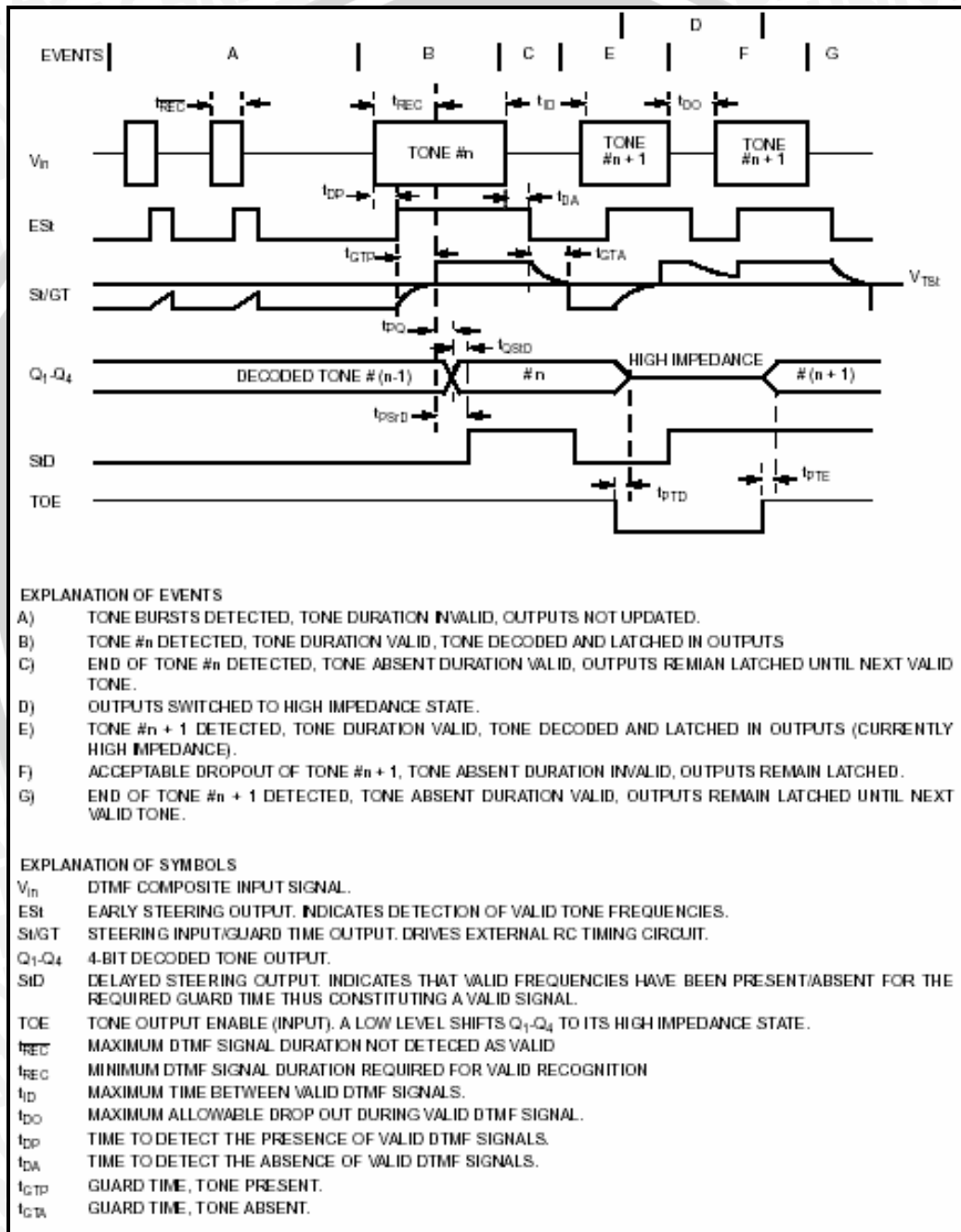
Z = High Impedance

X = Don't Care

-----  
**Sumber:** Mitel,1995: 4

Timing diagram yang ditunjukkan dalam gambar 2.7 menggambarkan sinyal kotak dari setiap waktu sesuai dengan deteksi digital dari pasangan nada DTMF. Dikatakan frekuensi valid dari masing-masing *group tone* jika keluaran *Early Steering* (Est) adalah *high*. Waktu yang diperlukan untuk mendeteksi adanya pasangan *tones valid*,  $t_{DP}$ , merupakan fungsi algoritma *decoder*, frekuensi *tone* dan bagian logika *decoder* sebelumnya. Est mengindikasikan bahwa kedua *tone* dari frekuensi sebenarnya

telah dideteksi dan diterima oleh rangkaian timing RC. Jika kedua *tone* menunjukkan untuk waktu deteksi minimum,  $t_{GTP}$ , ditentukan oleh saluran RC eksternal, sinyal DTMF didekodekan dan data hasil (Tabel 2.4) di-*latch* ke register output. Keluaran *Delayed Steering* (StD) diberi logika *high* dan menunjukkan bahwa data baru telah siap. Waktu yang diperlukan untuk menerima sinyal DTMF valid,  $t_{REC}$ , sama dengan jumlah dari  $t_{DP}$  dan  $t_{GTP}$ . Tabel fungsi *decoder* DTMF ditunjukkan dalam Tabel 2.4.

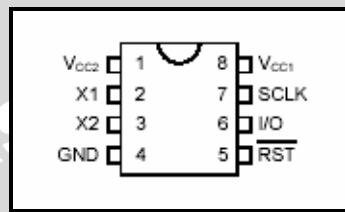


**Gambar 2.7** Diagram Waktu Penerima DTMF IC MT8870

Sumber: Mitel,1995: 11

## 2.5 RTC (*Real Time Clock*)

RTC merupakan rangkaian dan kalender dalam sebuah IC lengkap dengan baterai *back up*-nya, bahkan rangkaian jam dalam IC tersebut dapat tetap bekerja walaupun *power supply* sudah dimatikan. RTC yang digunakan adalah tipe DS1302. IC ini dapat menghitung detik, menit, jam, tanggal, bulan dan tahun. Memiliki 31 x 8 RAM dan dalam transfer datanya baik proses baca atau tulis dapat dilakukan dengan cara *single-byte* maupun *multiple-byte*. DS1302 memiliki 8 buah pin, dimana konfigurasi pinnya dapat dilihat dalam Gambar 2.8.



**Gambar 2.8** Konfigurasi Pin *Real Time Clock* IC DS1302

**Sumber:** Dallas Semiconductor, 2001:1

Adapun fungsi dari masing – masing pin dijelaskan dalam Tabel 2.5.

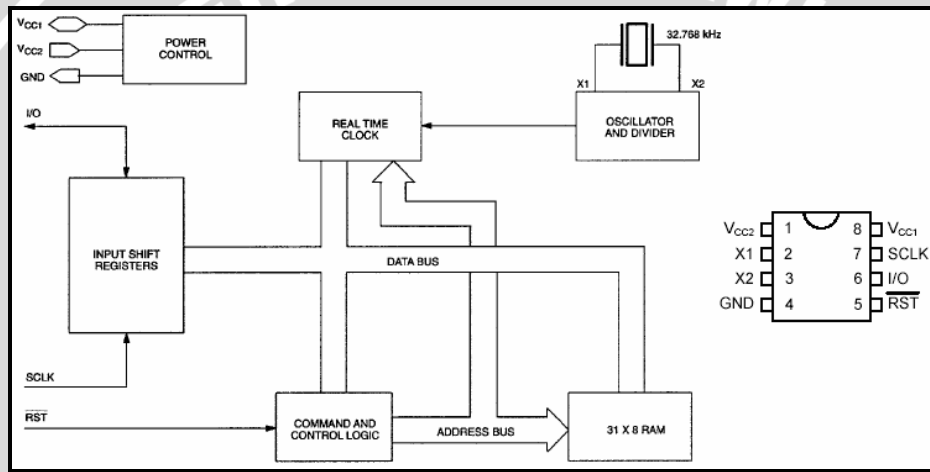
**Tabel 2.5** Fungsi – Fungsi Pin IC RTC DS1302

NO	SIMBOL	KETERANGAN
1.	VCC2	Pin <i>power supply</i> utama dalam konfigurasi <i>supply dual</i> . VCC1 dihubungkan pada sumber <i>back-up</i> dan memelihara kerja dari DS1302 jika <i>power supply</i> utama tidak mampu untuk mencatu. Jika VCC2 lebih besar dari VCC1+0,2Volt, maka DS1302 akan dicatu oleh VCC2. Jika VCC2 lebih kecil dari VCC1, maka VCC1 yang akan mencatu DS1302.
2.	X1	Koneksi untuk 32,768 kHz <i>Quartz Crystal Standard</i> . Osilator internal dirancang untuk beroperasi dengan <i>crystal</i> yang mempunyai beban kapasitansi sebesar 6 pF. Dapat juga digunakan osilator eksternal 32,768 kHz dimana pin X1 dihubungkan ke osilator eksternal dan pin X2 diambangkan.
3.	X2	
4.	GND	<i>Ground</i> .
5.	$\overline{RST}$	<i>Input</i> . Sinyal $\overline{RST}$ harus dinyatakan tinggi selama proses baca atau tulis. Pin ini mempunyai 40 k $\Omega$ resistor <i>pull-down internal</i> yang terhubung ke <i>ground</i> .
6.	I/O	<i>Input/Push-pull output</i> . Merupakan jalur data <i>bidirectional</i> untuk 3-wire <i>interface</i> dan mempunyai 40 k $\Omega$ resistor <i>pull-down internal</i> yang

		terhubung singkat.
7.	SCLK	<i>Input.</i> Digunakan untuk perpindahan data sinkron pada serial. Pin ini mempunyai 40 k $\Omega$ resistor <i>pull-down internal</i> yang terhubung ke <i>ground</i> .
8.	VCC1	Digunakan sebagai <i>charger</i> daya dari DS1302 dan dihubungkan ke baterai.

-----  
**Sumber:** Dallas Semiconductor, 2001:3

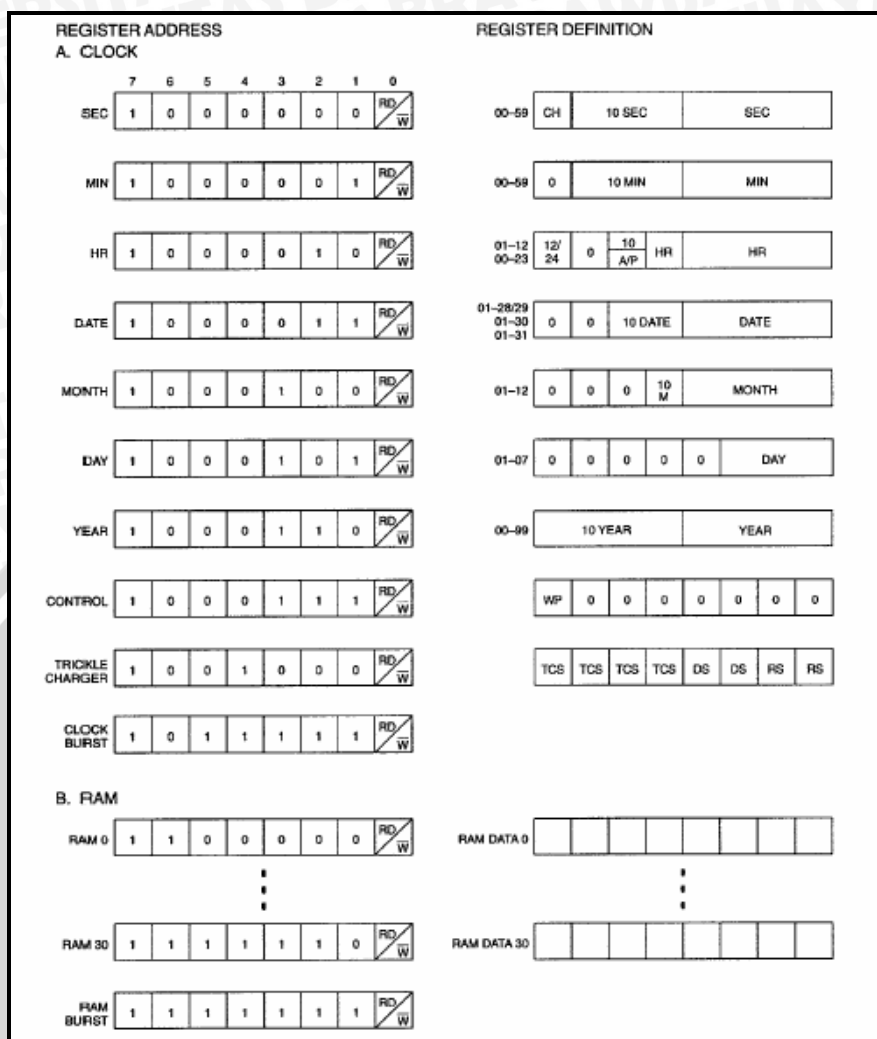
IC DS1302 mempunyai elemen-elemen utama yaitu *shift register*, *control logic*, *oscillator*, *real time clock* dan RAM. Blok diagram IC DS1302 ditunjukkan dalam Gambar 2.9.



**Gambar 2.9** Konfigurasi Pin RTC IC DS1302

-----  
**Sumber:** Dallas Semiconductor, 2001:3

Informasi penanggalan dan waktu diperoleh dengan pembacaan byte register yang sesuai. Gambar 2.10 menunjukkan register – register RTC. Penanggalan dan waktu di set dengan penulisan byte register yang sesuai. Isi dari register penanggalan dan waktu adalah dalam format *binary-coded decimal* (BCD).



Gambar 2.10 Register-register IC RTC DS1302

Sumber: Dallas Semiconductor, 2001:8

## 2.6 Mikrokontroler AT89S51

Mikrokontroler AT89S51 merupakan mikrokomputer CMOS 8 bit dengan 8 Kbyte **Flash Programmable dan Erasable Read Only Memory (PEROM)**. Mikrokontroler ini berteknologi **non volatile** kerapatan tinggi dari Atmel yang kompatibel dengan mikrokontroler standar industri MCS-51 baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah.

Spesifikasi penting AT89S51 antara lain:

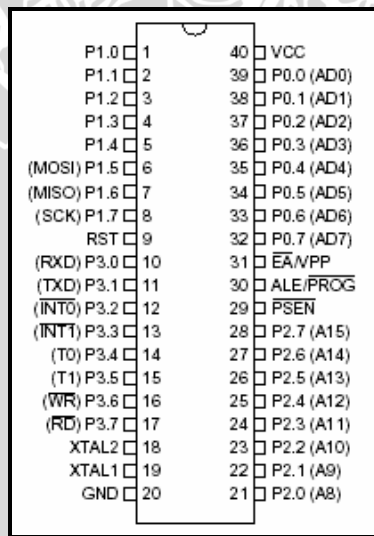
1. Kompatibel dengan keluarga mikrokontroler MCS-51 sebelumnya.
2. 4K Byte **In-system Programmable (ISP)** flash memori dengan kemampuan 1000 kali baca/tulis.
3. *Range* operasi 4 V – 5.5 V.
4. Bekerja dengan rentang 0 Hz – 33 MHz.



5. *Three-level Program Memory Lock.*
6. 128 x 8 bit RAM *internal.*
7. 32 jalur I/O yang dapat diprogram.
8. Dua buah 16 bit *Timer/Counter.*
9. Enam sumber *interrupt.*
10. Saluran *Full-Duplex* serial UART
11. *Low-power Idle and Power-down Modes*
12. *Interrupt Recovery from Power-down Mode*
13. *Watchdog Timer*
14. *Dual Data Pointer*
15. *Power-off Flag*
16. *Fast Programming Time*
17. *Flexible ISP Programming (Byte and Page Mode)*

### 2.6.1 Konfigurasi Pin

Gambar 2.11 menunjukkan bentuk chip dari mikrokontroler AT89S51.



**Gambar 2.11** Bentuk Chip Mikrokontroler AT89S51

**Sumber:** Atmel, 2003: 2

Berikut ini merupakan penjelasan dari masing-masing pin:

1. **VCC**  
Merupakan sumber tegangan
2. **GND**  
Merupakan *ground* sumber tegangan

3. **VDD**

Merupakan sumber tegangan untuk 42-PDIP

4. **PWRVDD**

Merupakan sumber tegangan untuk 42 - PDIP yang hanya terhubung dengan I/O *Pad Drivers*.

5. **PWRGND**

Merupakan *ground* untuk 42-PDIP yang hanya terhubung dengan I/O *Pad Drivers*.

6. **Port 0**

Port 0 merupakan saluran (bus) *input/output* 8 bit *open drain*. Sebagai port output, masing-masing pin dapat memasukkan 8 masukan TTL. Port 0 dapat digunakan sebagai multipleks bus alamat rendah dan bus data selama adanya akses ke memori program eksternal. *External pull-up* diperlukan selama proses verifikasi.

7. **Port 1**

Port 1 menjadi saluran (bus) dua arah *input/output* 8 bit. Dengan *internal pull-up* yang dapat digunakan untuk berbagai keperluan dan dapat mengendalikan empat *input* TTL. Port ini juga digunakan sebagai saluran alamat pada saat pemrograman dan verifikasi.

Pin port	Fungsi Alternatif
P1.5	MOSI (digunakan untuk <i>IN-System programming</i> )
P1.6	MISO (digunakan untuk <i>IN-System programming</i> )
P1.7	SCK (digunakan untuk <i>IN-System programming</i> )

8. **Port 2**

Port 2 menjadi saluran (bus) *input/output* dua arah 8 bit dengan *internal pull-up*. Saat pengambilan data dari program memori eksternal atau selama mengakses data memori eksternal dengan menggunakan alamat 16 bit (MOVX @ DPTR). Sedangkan pada saat mengakses ke data memori eksternal menggunakan alamat 8 bit (MOVX @ R1), port 2 mengeluarkan isi dari P2 ke **Special Function Register**.

9. **Port 3**

Port 3 merupakan saluran (bus) I/O 8 bit dua arah dengan *internal pull up* yang memiliki fungsi pengganti. Ketika logika 1 diberikan ke port 3, maka *pull up* internal akan menseset port pada kondisi high dan port 3 dapat digunakan sebagai

saluran input. Selain itu, sebagian dari port 3 dapat berfungsi sebagai sinyal kontrol pada saat proses pemrograman dan verifikasi. Adapun fungsi penggantinya ditunjukkan dalam Tabel 2.6

**Tabel 2.6** Fungsi Pengganti dari Port 3

Bit	Nama	Fungsi Alternatif
P 3. 0	RXD	Untuk menerima data port serial
P 3. 1	TXD	Untuk mengirim data port serial
P 3. 2	INT0'	Interupsi eksternal 0
P 3. 3	INT1'	Interupsi eksternal 1
P 3. 4	T0	Input eksternal waktu/pencacah 0
P 3. 5	T1	Input eksternal waktu/pencacah 1
P 3. 6	WR'	Jalur menulis memori data eksternal strobe
P 3. 7	RD'	Jalur membaca memori data eksternal strobe

Sumber: Atmel, 2003: 5

**10. RST**

Merupakan masukan reset yang aktif dengan memberikan input high selama 2 siklus. Rangkaian *power-on reset* diperlukan untuk me-reset mikrokontroler secara otomatis setiap catu daya on. Dalam gambar 2.12 menunjukkan rangkaian ekuivalen untuk rangkaian fungsi reset dimana ketika catu daya diaktifkan (Vcc), rangkaian reset akan menahan logika pin RST dengan jangka waktu yang ditentukan oleh lamanya pengisian C. Sehingga didapat persamaan untuk perhitungan waktu tunda reset sebagai berikut:

$$Vo(t) = \frac{R}{R + \frac{1}{sC}} \cdot Vi(t)$$

$$Vo(t) = \frac{sCR}{sCR + 1} \cdot Vi(t)$$

Dengan tegangan masukan  $V_i$  adalah tegangan Vcc, yaitu 5V, dalam fungsi Laplace adalah  $5/s$ .



$$Vo(t) = \frac{sCR}{sCR + 1} \cdot \frac{5}{s}$$

$$Vo(t) = \frac{cR}{sCR + 1} \cdot 5$$

$$Vo(t) = 5 \left[ \frac{1}{s + \frac{1}{RC}} \right]$$

$$Vo(t) = 5 \cdot e^{-\frac{t}{RC}}$$

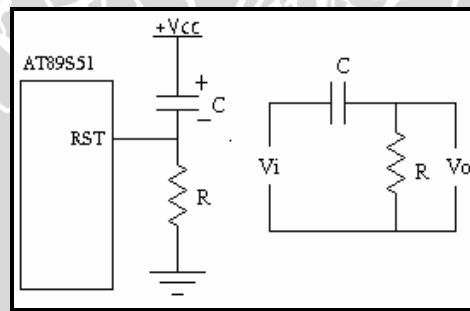
$$\frac{5}{Vo(t)} = e^{\frac{t}{RC}}$$

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

$$t = R \cdot C \left[ \ln \frac{5}{Vo(t)} \right] \tag{2-7}$$

Dengan  $V_o$  adalah tegangan logika minimal yang diijinkan oleh pin RST, dimana  $V_o = 0.7$ ,  $V_{cc} = 3.5V$   $t = 0.357.R.C$ .

Untuk memastikan keabsahan reset, logika *high* pada pin ini harus ditahan lebih lama dari 2 siklus mesin ditambah waktu mulai hidup (*start on*) osilator.



Gambar 2.12 Rangkaian Power-on Reset dan Rangkaian Ekivalen Power-on Reset

**12. PSEN**

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

**13. EA/VPP**

EA harus dihubungkan dengan GND untuk mengaktifkan *device* untuk mengambil kode dari memori program eksternal yang beralamat 0000H sampai FFFFH. EA seharusnya dihubungkan dengan VCC untuk eksekusi program internal. Pada saat Flash Programming pin ini akan mendapat tegangan 12 Volt (VPP).



**14. XTAL 1**

Merupakan *input* untuk *inverting osilator amplifier*.

**15. XTAL 2**

Merupakan *output* dari *inverting osilator amplifier*.

Blok diagram dari mikrokontroler AT89S51 ditunjukkan dalam Gambar 2.13.

AT89S51 mempunyai struktur memori yang terdiri atas:

**1. RAM Internal**

Memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara. RAM Internal terdiri atas:

**a. Register Banks**

AT89S51 mempunyai 8 buah register yang terdiri atas R0 hingga R7. Kedelapan buah register ini selalu terletak pada alamat 00H hingga 07H pada setiap kali sistem direset. Namun posisi R0 hingga R7 dapat dipindah ke bank 1 (08 hingga 0FH), bank 2 (10H hingga 17H) atau bank 3 (18H hingga 1FH) dengan mengatur bit RS0 dan RS1.

**b. Bit Addressable RAM**

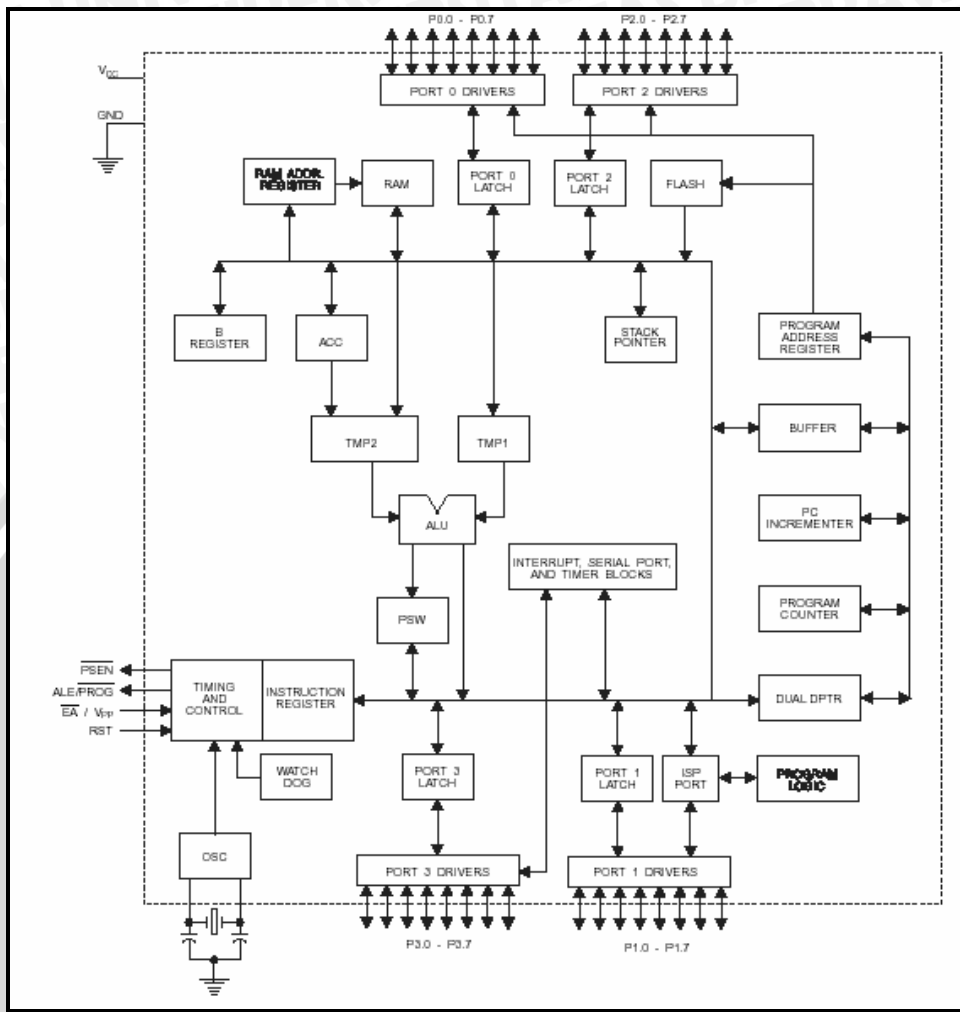
RAM pada alamat 20H hingga 2FH dapat diakses secara pengalamatan bit (*bit addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat di-*set*, di-*clear*, di-AND dan di-OR. Dalam aplikasinya, lokasi yang dapat diakses dengan pengalamatan bit ini dapat juga digunakan untuk menandai suatu lokasi bit tertentu, baik berupa Register Fungsi Khusus yang dapat dialamati secara bit (termasuk register I/O), ataupun lokasi-lokasi tertentu yang dapat dialamati secara bit.

**c. RAM Keperluan Umum**

RAM Keperluan Umum dimulai dari alamat 30H sampai 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang menunjukkan lokasi yang dialamati.

Sedangkan pengalamatan secara tak langsung pada lokasi dari RAM Internal ini adalah akses data dari memori ketika alamat tersebut tersimpan dalam suatu Register R0 atau R1. R0 dan R1 adalah dua buah register pada mikrokontroler berarsitektur MCS51 yang dapat digunakan sebagai pointer dari sebuah lokasi memori pada RAM Internal. Pengalamatan secara tidak langsung

biasa digunakan untuk mengakses beberapa lokasi memori dengan letak yang beraturan.



Gambar 2.13 Blok Diagram Mikrokontroler AT89S51

Sumber: Atmel, 2003: 3

## 2. *Special Function Register (Register Fungsi Khusus)*

Memori yang berisi register-register yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut, seperti *timer*, serial dan lain-lain. AT89S51 mempunyai 26 *Special Function Register (Register Fungsi Khusus)* yang terletak pada alamat 80H sampai dengan FFH seperti ditunjukkan dalam Gambar 2.14. Beberapa dari register-register ini juga mampu dialamati dengan pengalamatan bit sehingga dapat dioperasikan seperti yang ada pada RAM yang lokasinya dapat dialamati dengan pengalamatan bit. Berikut ini Gambar 2.14 menunjukkan susunan *Special Function Register* dan susunan RAM internal AT89S51.

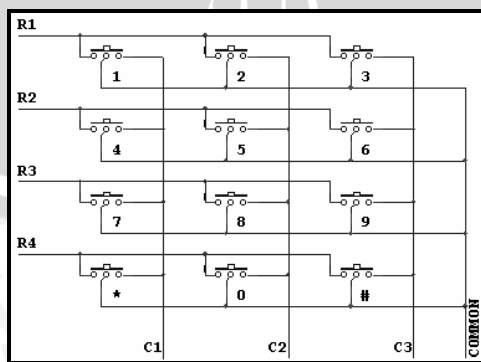
0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDTRST XXXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

Gambar 2.14 Susunan Special Function Register

Sumber: Atmel, 2003: 6

### 2.7 Keypad Matriks 4x3

Keypad matriks merupakan sebuah input device sederhana yang dapat menghasilkan suatu kode ke sistem bila kita tekan salah satu tombolnya, sebagai hasil dari pertemuan dua jalur. Kedua jalur tersebut berupa jalur baris (sumbu mendatar) dan jalur kolom (sumbu tegak). Keypad matriks 4x3 disini artinya terdapat 4 jalur baris dan 3 jalur kolom sehingga akan membentuk matriks 4x3 sama dengan menghasilkan 12 kode yang berbeda. Keypad matriks 4x3 ditunjukkan dalam Gambar 2.15.



Gambar 2.15 Keypad Matriks 4x3

Sumber: Eko Putra, 2005: 117

*Keypad* ini bekerja menggunakan sistem *scanning*. *Scanning* ini dilakukan dengan menggunakan IC dekoder khusus atau menggunakan mikrokontroler. Jika diantarmukakan dengan mikrokontroler, program dalam mikrokontroler akan mengatur proses *scanning* tersebut. Pertama akan dikirim sebuah sinyal aktif ke sebuah kolom, selanjutnya mikrokontroler akan membaca secara bergantian mulai dari baris pertama sampai baris keempat, untuk menyeleksi tombol mana yang aktif. Hal ini akan berulang-ulang sampai kolom ketiga, kemudian kembali lagi ke kolom pertama.

## 2.8 Solenoid

*Solenoid* merupakan komponen elektro mekanik yang merubah energi listrik menjadi gerak. *Solenoid* adalah lilitan kawat yang berintikan besi dan dapat bersifat magnet jika dialiri arus listrik DC. Arus yang mengalir pada lilitan kawat akan membuat medan magnet. Gaya magnet dibutuhkan untuk menggerakkan besi penarik yang bergerak secara berubah-ubah (menekan atau menarik).

Jika  $L$  adalah panjang *solenoid* dan  $N$  banyaknya lilitan loop atau banyaknya loop per satuan panjang adalah  $N/L$ . Sehingga medan yang dihasilkan oleh masing-masing loop dapat ditentukan dengan menggunakan Persamaan (2-8) dan (2-9) dibawah ini.

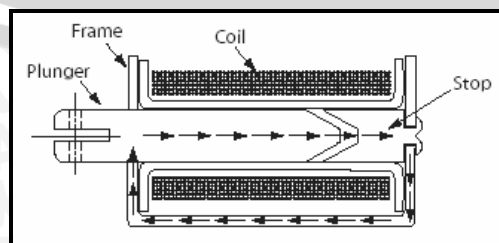
Jika *solenoid* panjang untuk titik ditengah-tengah:

$$B = \frac{\mu_o IN}{L} \quad (2-8)$$

Jika titik disalah satu ujung *solenoid*:

$$B = \frac{\mu_o IN}{2L} \quad (2-9)$$

Dari persamaan diatas sudah jelas bahwa kerja *solenoid* dipengaruhi oleh besarnya arus yang masuk dikalikan dengan banyaknya lilitan dan dibagi oleh panjang dari *solenoid* tersebut. Gambar konstruksi *solenoid* ditunjukkan dalam Gambar 2.16.



**Gambar 2.16** Solenoid Construction

Sumber: Wilde, 2007: 1

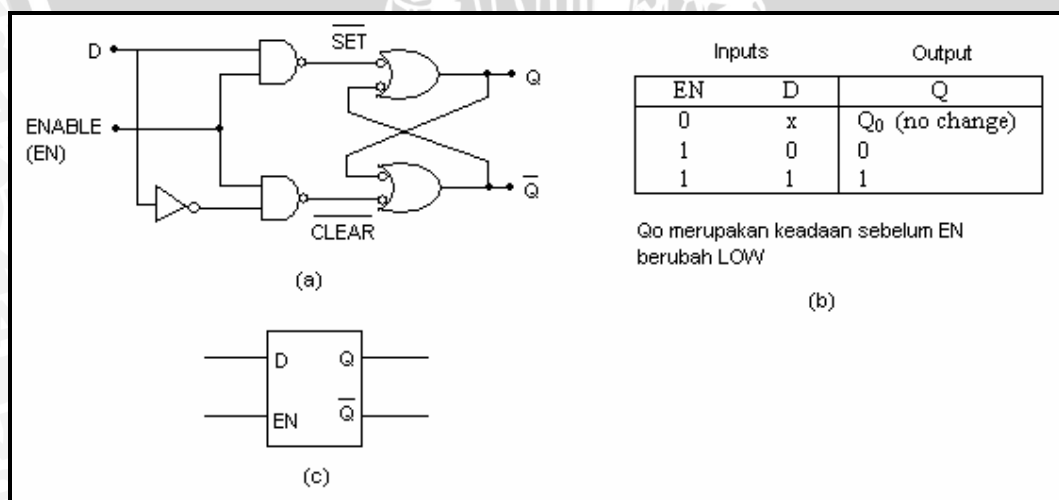


## 2.9 D – Latch (Transparent Latch)

Rangkaian *D – Latch* merupakan rangkaian logika yang berfungsi sebagai penahan data. Data masukan *D* akan ditahan oleh *D – Latch* apabila masukan *enable* (*EN*) berlogika rendah, sebaliknya apabila masukan *enable* berlogika tinggi maka data akan diteruskan. Operasi kerja dari *D – Latch* dapat dijelaskan sebagai berikut:

1. Pada saat *EN* berlogika tinggi, data masukan *D* akan menghasilkan logika rendah pada masukan  $\overline{SET}$  dan masukan  $\overline{CLEAR}$  pada *latch* gerbang NAND dalam Gambar 2.17a, dengan tujuan agar mendapatkan keluaran *Q* sama dengan masukan *D*. Apabila masukan *D* berubah dengan masukan *EN* masih berlogika tinggi, maka keluaran *Q* akan mengikuti perubahan dari masukan *D*. Pada mode ini, *D – Latch* dapat dikatakan sebagai “transparent”.
2. Pada saat masukan *EN* menjadi berlogika rendah, maka kedua masukan pada *latch* gerbang NAND akan berlogika tinggi, sehingga keluaran *Q* dan  $\overline{Q}$  akan tetap sama dengan level logika yang diberikan sebelum masukan *EN* menjadi berlogika rendah. Walaupun masukan *D* berubah, keluaran akan tetap ditahan sesuai dengan level logika sebelumnya dan tidak akan berubah selama masukan *EN* berlogika rendah. Pada mode ini, *D – Latch* dapat dikatakan sebagai “latched”.

Operasi kerja dari *D – Latch* ditunjukkan oleh tabel kebenaran dalam Gambar 2.17b. Sedangkan simbol logika untuk *D – Latch* ditunjukkan dalam Gambar 2.17c.



**Gambar 2.17** Rangkaian *D – Latch* (a). Struktur Rangkaian (b). Tabel Kebenaran (c). Simbol Logika

**Sumber:** Tocci, 1988: 195

## 2.10 Pemancar dan Penerima FM

Pemancar merupakan sebuah devais elektronika yang berfungsi untuk membawa dan memproses sinyal informasi untuk ditransmisikan. Metode yang dipakai dalam pengiriman sinyal adalah dengan menumpangkan sinyal informasi pada sinyal yang mempunyai frekuensi lebih tinggi, memperkuat sinyal dan gelombang yang dihasilkan diteruskan ke antena pemancar. Sinyal frekuensi tinggi tersebut berfungsi membawa sinyal informasi dari pesawat pengirim ke pesawat penerima, sehingga disebut sinyal pembawa (*carrier*). Proses penumpangkan sinyal informasi pada sinyal pembawanya disebut dengan proses modulasi dan gelombang yang dihasilkan disebut sinyal termodulasi.

Pesawat penerima radio berfungsi memisahkan sebuah sinyal radio yang dikehendaki dari semua sinyal radio lain yang mungkin diterima oleh antenna, dan menolak semua sinyal lain tersebut. Sinyal yang dipisahkan tersebut kemudian diperkuatnya sampai ke suatu tingkat yang dapat digunakan. Akhirnya, sinyal suara dipisahkannya dari pembawa (*carrier*) radio, dan diteruskannya ke pemakai. (Roody, Dennis dan Coolen, 1984: 235).

Pemancar dan penerima gelombang radio yang digunakan dalam sistem ini memanfaatkan modul yang telah ada yaitu modul *microphone wireless*. Modul ini menggunakan sistem modulasi frekuensi (FM). Untuk dapat memanfaatkannya secara maksimal maka perlu penyesuaian menurut spesifikasi modul yang ada, antara lain:

- |                              |                        |
|------------------------------|------------------------|
| 1. Catu Daya                 | 3V atau 1.5V DC        |
| 2. Daya Pengiriman           | 10 mWatt               |
| 3. <i>Frequency Range</i>    | 88 – 108 MHz           |
| 4. <i>Frequency Response</i> | 100 – 10000 Hz         |
| 5. Impedansi Input           | 600 $\Omega$           |
| 6. Jarak Efektif             | 15 – 30 m              |
| 7. <i>Directivity</i>        | <i>uni-directional</i> |

## BAB III

### METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu Perancangan dan Pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio yang bertujuan agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan. Jenis data yang digunakan dalam desain ini meliputi data sekunder dan data primer. Data sekunder adalah data yang diperoleh dari databook masing-masing komponen. Pemilihan komponen diutamakan yang ada di pasaran. Sedangkan data primer mempunyai data spesifikasi alat dari hasil pengujian. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut.

#### 3.1 Studi Literatur

Studi literatur dilakukan dengan mengambil data/informasi yang berkaitan dengan perancangan dan pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio. Studi literatur yang digunakan meliputi:

1. Sistem pengkodean data.
2. Sistem pengiriman sinyal jarak jauh.
3. Karakteristik mikrokontroler.
4. Sistem elektronika pendukung sistem.
5. Perangkat lunak yang mengendalikan sistem.

#### 3.2 Perancangan dan Pembuatan Alat

Sebelum melakukan perencanaan dan perealisasiian alat, maka ditentukan spesifikasi alat yang akan dibuat. Adapun spesifikasi alat yang akan direalisasikan sebagai berikut:

1. Data masukan dari *keypad* matriks 4 x 3
2. Mikrokontroler AT89S51 sebagai unit pengolah data
3. *Encoder* dan *decoder* data menggunakan IC DTMF TP5088 dan MT8870
4. Pengunci yang digunakan berupa *solenoid*
5. Menggunakan RTC sebagai pengatur waktu
6. LED dan *buzzer* sebagai indikator keadaan pintu dan indikator keberhasilan proses penguncian pintu

Selanjutnya dilakukan perhitungan dan perancangan tiap-tiap blok rangkaian dengan mengacu pada data-data dari buku data komponen elektronika. Dari data yang diperoleh, dilakukan berbagai analisa dan perhitungan untuk mencapai hasil optimal dari komponen yang digunakan. Selain itu agar sesuai dengan karakteristik masing-masing komponen yang meliputi kemampuan arus yang dilewatkan, catu tegangan, sinyal masukan, serta sinyal keluaran yang dihasilkan. Berbagai perlakuan di atas digunakan untuk dapat mendukung perancangan dan pembuatan keseluruhan sistem.

Proses perancangan alat adalah sebagai berikut:

1. Pembuatan blok diagram rangkaian.
2. Perancangan rangkaian pada tiap-tiap blok dan melakukan perhitungan yang diperlukan dalam penyusunan rangkaian sehingga terbentuk skema rangkaian.
3. Desain dan pembuatan papan rangkaian tercetak dengan menggunakan software Protel 99.
4. Perakitan komponen pada papan rangkaian tercetak.
5. Pembuatan perangkat lunak.

### **3.3 Pengujian Alat**

Pengujian alat ini bertujuan untuk mengetahui apakah alat hasil perancangan dapat bekerja sesuai dengan spesifikasi yang direncanakan. Proses pengujian alat dilakukan secara bertahap. Yang pertama, dilakukan pengujian rangkaian tiap blok sehingga dapat diketahui hasil perancangan masing-masing blok telah berfungsi sesuai dengan perancangan. Setelah masing-masing blok dapat berfungsi dengan baik, selanjutnya dilakukan pengujian secara keseluruhan dengan menyambungkan blok perangkat keras dan mengoperasikan sistem untuk mendapatkan hasil analisis. Hasil analisis tersebut akan dijadikan dasar untuk menarik kesimpulan dari proses perancangan sistem ini. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penelitian.

## BAB IV

### PERENCANAAN DAN PEMBUATAN ALAT

Bab ini membahas mengenai perancangan dan pembuatan Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio. Dalam perencanaan alat ini dilakukan bertahap blok demi blok untuk memudahkan analisis sistem setiap bagian maupun sistem secara keseluruhan. Perencanaan dan pembuatan sistem ini terdiri atas dua bagian yaitu perencanaan perangkat keras dan perencanaan perangkat lunak. Beberapa aspek lain yang perlu dijelaskan dalam pembahasan bab ini adalah penentuan spesifikasi dari sistem yang dirancang, blok diagram dan prinsip kerja sistem.

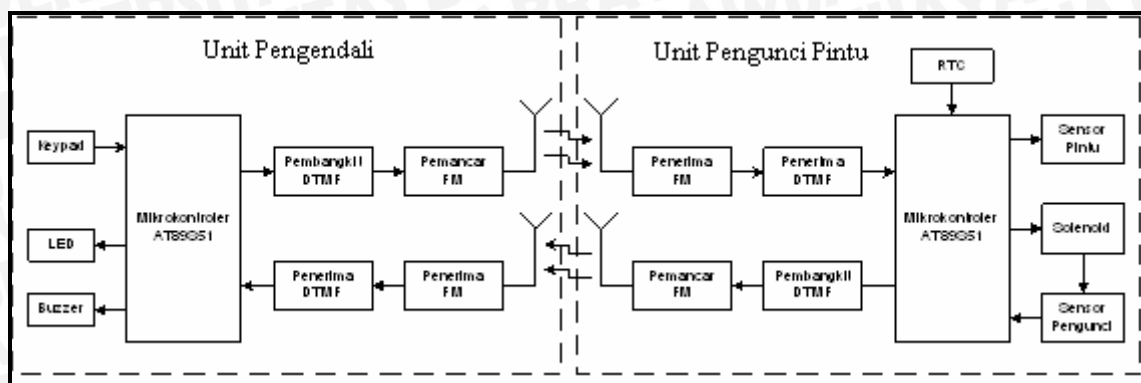
#### 4.1 Spesifikasi Alat

Beberapa hal yang perlu diperhatikan dalam melakukan perencanaan dan pembuatan alat ini adalah penentuan spesifikasi sistem yang akan dibuat. Adapun spesifikasi sistem yang akan dibuat adalah sebagai berikut:

1. Data masukan dari *keypad* matriks 4 x 3
2. Mikrokontroler AT89S51 sebagai unit pengolah data
3. *Encoder* dan *decoder* data menggunakan IC DTMF TP5088 dan MT8870
4. Pengunci yang digunakan berupa *solenoid*
5. Menggunakan RTC sebagai pengatur waktu
6. LED dan *buzzer* sebagai indikator keadaan pintu dan indikator keberhasilan proses penguncian pintu

#### 4.2 Perencanaan Sistem

Dalam perencanaan ini, tahap pertama adalah pembuatan blok diagram sistem. Dengan adanya blok diagram sistem ini, maka akan diketahui perangkat keras yang akan digunakan serta cara kerja dari sistem yang akan dibuat. Blok diagram Sistem Pengendali Kunci Pintu Jarak Jauh Dengan Transmisi Gelombang Radio ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Blok Diagram Sistem

Sistem Pengendali Kunci Pintu Dengan Transmisi Gelombang Radio ini mempunyai dua bagian penting, yaitu unit pengendali yang dioperasikan oleh *user* dan bagian unit pengunci pintu yang diletakkan pada setiap pintu. Agar dapat beroperasi dengan baik maka frekuensi kerja antara unit pengendali dan unit pengunci pintu harus sama. Mekanisme kerja dari sistem ini diawali dari penekanan tombol – tombol *keypad* sebagai masukan.

Setiap tombol *keypad* dihubungkan ke mikrokontroler sebagai pengolah data pada unit pengendali. Pengiriman sinyal dari unit pengendali ke penerima menggunakan pembangkit DTMF sebagai pembangkit sinyal DTMF yang selanjutnya akan ditransmisikan melalui pemancar FM.

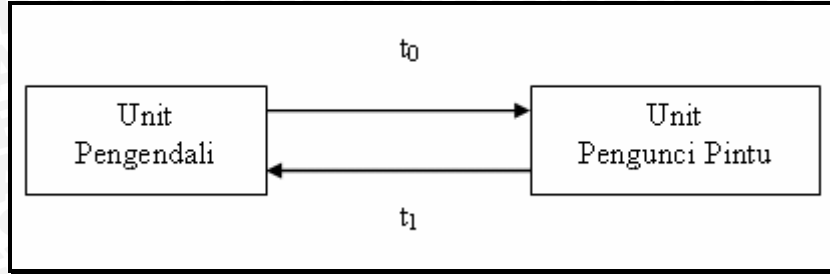
Pada unit pengunci pintu, sinyal radio tersebut dikirimkan ke penerima DTMF untuk didekodekan menjadi kode-kode DTMF dan selanjutnya ke mikrokontroler sebagai pengendali sistem pada unit pengunci pintu. RTC sebagai pengatur waktu. Sensor pintu digunakan untuk mengecek keadaan pintu, apakah dalam keadaan membuka atau menutup. *Solenoid* digunakan sebagai mekanik elektronika pengunci pintu. Kerja dari *solenoid* ini mempengaruhi sensor pengunci pintu. Semua proses kerja dari RTC, sensor dan *solenoid* diolah oleh mikrokontroler dan selanjutnya ditransmisikan kembali ke unit pengendali dengan sistem komunikasi yang sama dengan pengiriman sinyal ke unit pengunci pintu.

Pada unit pengendali ini *user* dapat mengetahui laporan keberhasilan operasi yang baru saja dilakukan melalui indikator yang ada. Indikator yang digunakan adalah *Light Emitting Diode* (LED). Sedangkan *buzzer* digunakan sebagai indikator alarm sensor pintu.

Mekanisme komunikasi data dari sistem ini adalah *half-duplex* sehingga digunakan dua buah pemancar dan penerima FM sebagai sarana pengiriman data. Kedua

repository.ub.ac.id

pemancar FM memiliki frekuensi yang sama. Untuk dapat mengirimkan data dengan baik maka pemancar FM ini diaktifkan secara bergantian, sedangkan penerima FM diaktifkan secara terus-menerus. Gambar 4.2 menunjukkan blok diagram proses *handshaking* unit pengendali dan unit pengunci pintu.



**Gambar 4.2** Blok diagram Proses *Handshaking* Unit Pengendali dan Unit Pengunci Pintu

Proses *handshaking* unit pengendali dan unit pengunci pintu dalam sistem ini terdapat beberapa kondisi, antara lain:

1. Jika unit pengendali mendapat perintah untuk mengirim data ke unit pengunci pintu ( $t_0$ ) maka pemancar FM pada unit pengendali diaktifkan selama proses pengiriman data. Kondisi pemancar FM pada unit pengunci pintu adalah non aktif. Jika data yang diinginkan oleh *user* dapat diterima dengan baik oleh unit pengunci pintu, selanjutnya data tersebut akan digunakan untuk melakukan perintah sesuai dengan yang diberikan oleh *user*. Hasil kerja dari perintah tersebut oleh unit pengunci pintu akan dikirimkan kembali ke unit pengendali sebagai laporan ( $t_1$ ). Proses pengiriman data yang terjadi adalah sama dengan proses pengiriman data dari unit pengendali ke unit pengunci pintu, yaitu dengan mengaktifkan pemancar FM pada unit pengunci pintu dan memberikan kondisi non aktif untuk pemancar FM pada unit pengendali.
2. Jika terdapat kesalahan penekanan tombol *keypad* oleh *user* dalam pemberian perintah, maka unit pengendali akan mengaktifkan buzzer sebagai pengingat. Selanjutnya *user* dapat mengulang penekanan tombol sesuai dengan perintah yang diinginkan.
3. Jika data yang diterima oleh unit pengunci pintu tidak sempurna, maka data tersebut tidak akan diproses oleh unit pengunci pintu. Kemudian unit pengunci pintu menunggu sampai ada data yang valid.

### 4.3 Perencanaan Perangkat Keras (*Hardware*)

#### 4.3.1 Rangkaian DTMF *Encoder* dan Pemancar FM

Pemancar FM dalam sistem ini digunakan untuk mengirimkan data dengan memanfaatkan media transmisi gelombang radio FM. Paket data yang dikirim melalui *encoder* DTMF merupakan data digital. Agar data tersebut dapat ditransmisikan melalui udara atau gelombang radio yang termodulasi frekuensi, maka data-data itu diubah ke dalam bentuk sinyal DTMF. *Encoder* DTMF ini mampu membangkitkan 16 pasang nada atau frekuensi yang mewakili bentuk-bentuk data digital tersebut. Dengan pemancar FM ini, nantinya pengiriman paket data dilakukan. Dalam hal ini tidak dibahas secara terperinci perancangan pemancar FM, namun penulis disini hanya memanfaatkan rangkaian pemancar FM yang telah ada di pasaran.

Pemancar FM yang digunakan adalah pemancar FM dari *microphone wireless*. Pemancar FM ini menggunakan sistem *half-duplex*, maka diperlukan rangkaian *transistor switching*. Rangkaian ini berfungsi untuk mengaktifkan pemancar FM.

Dari datasheet mikrokontroler AT89S51 diketahui bahwa arus keluaran mikrokontroler pada saat *high* adalah  $-60 \mu\text{A}$ . Sedangkan arus yang diperlukan rangkaian pemancar FM aktif sebesar  $75,9 \text{ mA}$ . Dalam perancangan ini digunakan transistor BC550 dan BC560 yang mempunyai spesifikasi sebagai berikut:

Dari *datasheet* transistor BC550 diketahui bahwa:

$$\begin{aligned} I_{C(\max)} &= 100 \text{ mA} \\ V_{BE(\max)} &= 0,9\text{V} \\ \beta_{(\min)} &= 110 \text{ dan } \beta_{(\max)} = 800 \end{aligned}$$

Dari *datasheet* transistor BC560 diketahui bahwa:

$$\begin{aligned} I_{C(\max)} &= 100 \text{ mA} \\ V_{BE(\max)} &= 0,9\text{V} \\ \beta_{(\min)} &= 110 \text{ dan } \beta_{(\max)} = 800 \end{aligned}$$

Transistor tersebut akan memerlukan arus basis untuk memastikan transistor BC560 akan tetap saturasi pada level arus pemancar FM, sehingga arus basis transistor BC550 harus lebih besar daripada arus pemancar FM dibagi dengan penguatan transistor BC560.

Transistor BC560 akan saturasi jika:

$$I_{B1} = I_{C2} \geq \frac{I_{C1}}{\beta_1} \quad (4-1)$$



$$I_{B1} = I_{C2} \geq \frac{75,9 \times 10^{-3}}{296,65}$$

$$I_{C2\min} = 0,256 \text{ mA}$$

Dari transistor BC550 diperoleh:

$$I_{B2} = \frac{I_{C2}}{\beta_2}$$

$$I_{B2} = \frac{0,256 \times 10^{-3}}{296,65}$$

(4-2)

$$I_{B2} = 8,63 \times 10^{-4} \text{ mA}$$

Sehingga nilai resistor pembatas arus pada basis adalah:

$$V_{OH(\max)} - I_{B2} \cdot R_{B2} - V_{BE} = 0$$

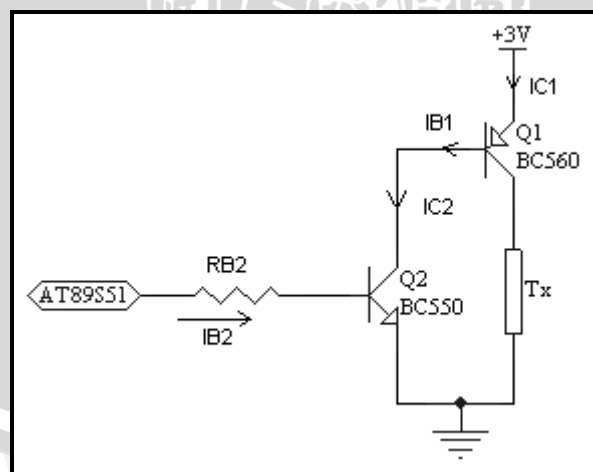
$$R_{B2(\min)} = \frac{V_{OH(\max)} - V_{BE}}{I_{B2}}$$

(4-3)

$$R_{B2(\min)} = \frac{5 - 0,7}{8,63 \times 10^{-4}}$$

$$R_{B2(\min)} = 4,98 \text{ k}\Omega$$

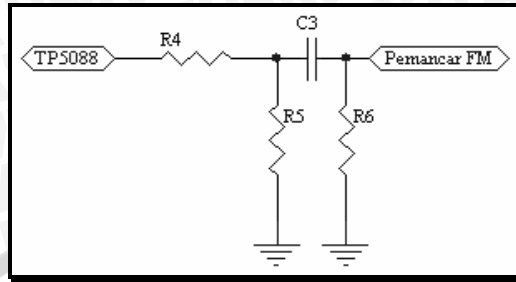
Dalam perancangan ini digunakan nilai  $R_{B2} = 82 \text{ k}\Omega$  yang akan memberikan arus  $I_{B2} = 52,4 \mu\text{A}$ . Dengan demikian, arus keluaran mikrokontroler masih dapat mencukupi rangkaian transistor *switching*. Gambar rangkaian transistor *switching* untuk pemancar FM ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangkaian Transistor *Switching* Pemancar FM

Dari pengukuran dapat diketahui bahwa tegangan keluaran rangkaian pembangkit DTMF TP5088 sebesar 1,4 Vpp dan tegangan masukan maksimum pemancar FM sebesar 1 Vpp. Sesuai dengan spesifikasi pemancar FM diketahui bahwa impedansi input 600  $\Omega$ , sehingga sebelum data ditransmisikan melalui pemancar FM

data DTMF harus diperkecil hingga kurang dari 1 V<sub>pp</sub> dan mempunyai impedansi 600Ω. Rangkaian tambahan yang dipasang pada pin *tone output encoder* DTMF TP5088 ditunjukkan dalam Gambar 4.4.



**Gambar 4.4** Rangkaian Penyesuai *Encoder* DTMF TP5088

Resistor R<sub>4</sub> sebesar 680 Ω dan R<sub>5</sub> sebesar 1,2 kΩ merupakan pembagian tegangan yang berfungsi untuk memperkecil sinyal DTMF sehingga masukan pemancar FM menjadi:

$$V_{IN} = \frac{R_5}{R_4 + R_5} \cdot V \quad (4-4)$$

$$V_{IN} = \frac{1200}{680 + 1200} \cdot 1,4V_{pp} = 0,89V_{pp}$$

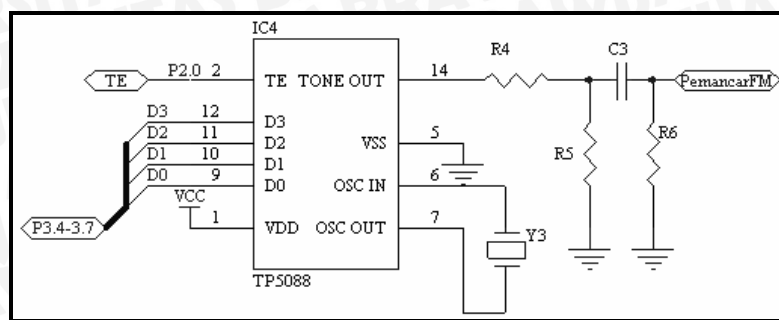
Resistor R<sub>6</sub> sama dengan R<sub>5</sub> sebesar 1,2 kΩ dipasang paralel untuk mendapatkan impedansi 600 Ω sesuai dengan spesifikasi pemancar FM.

$$Z = \frac{R_5 \cdot R_6}{R_5 + R_6} \quad (4-5)$$

$$Z = \frac{1200 \cdot 1200}{1200 + 1200} = 600\Omega$$

Sedangkan kapasitor C<sub>3</sub> sebesar 100 uF merupakan kapasitor kopling yang digunakan untuk menahan tegangan DC dan hanya melewatkan tegangan AC yang akan ditransmisikan melalui pemancar FM.

Rangkaian pembangkit DTMF memiliki komponen utama IC TP5088 yang memerlukan masukan dari mikrokontroler AT89S51. Pin D<sub>0</sub> - D<sub>3</sub> dan *Tone Enable* yang merupakan pin - pin masukan dari IC TP5088 dihubungkan dengan Port 3.4 - 3.7 dan Port 2.0 dari mikrokontroler. Untuk membangkitkan IC TP5088 ini membutuhkan XTAL 3.579545 MHz. Gambar rangkaian pembangkit DTMF dapat dilihat dalam Gambar 4.5.



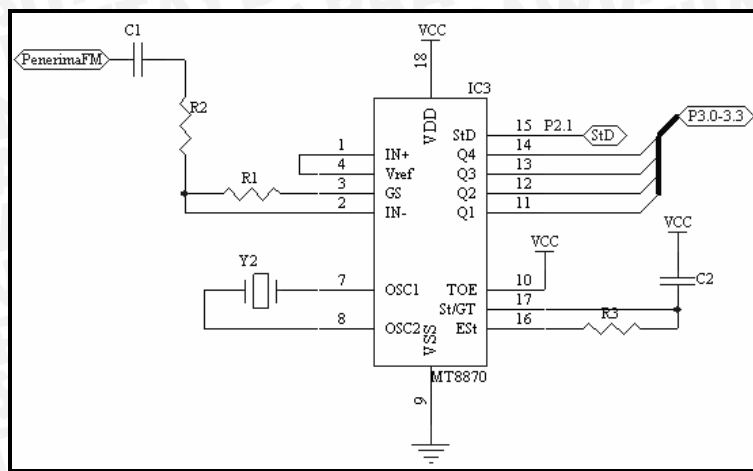
Gambar 4.5 Rangkaian Pembangkit DTMF IC TP5088

### 4.3.2 Rangkaian Penerima FM dan DTMF Decoder

Radio penerima FM dalam sistem ini digunakan untuk menerima sinyal informasi yang telah dikirimkan oleh pemancar. Penerima ini memisahkan sebuah sinyal informasi yang dikehendaki dari sinyal radio lain yang memungkinkan diterima oleh antena dan menolak sinyal lain tersebut. Sinyal informasi tersebut berupa paket data yang dipisahkan dari sinyal pembawa radio kemudian diteruskan untuk diubah kembali ke dalam bentuk data digital. Dalam hal ini tidak dibahas secara terperinci perancangan penerima FM, namun penulis disini hanya memanfaatkan rangkaian penerima FM yang telah ada di pasaran.

Penerima FM yang digunakan adalah penerima FM dari *microphone wireless*. Penerima FM ini menggunakan sistem *half-duplex*. Rangkaian penerima FM ini selalu dalam keadaan aktif tidak seperti rangkaian pemancar FM yang memerlukan transistor *switching* untuk mengaktifkannya.

Pengembalian sinyal berupa *tone* DTMF menjadi data-data biner kembali menggunakan IC *decoder* DTMF MT8870. IC *decoder* MT8870 merupakan *chip* yang memang dibuat khusus sebagai pendekode sinyal DTMF dengan penambahan beberapa komponen penunjang dalam proses kerjanya. Gambar 4.6 menunjukkan perancangan rangkaian DTMF yang menggunakan IC MT8870.



Gambar 4.6 Rangkaian Penerima DTMF IC MT8870

Untuk prinsip kerjanya, sinyal nada yang diterima oleh masukan IC MT8870 akan diterjemahkan dan menghasilkan keluaran berupa kode biner 4 bit yang akan menunjukkan pasangan nada yang diterimanya. Untuk mengetahui kode-kode dan pasangan nada yang berkorespondensi dengannya dapat dilihat dalam Tabel 2.4.

Komponen kapasitor pada masukan IC MT8870 berfungsi sebagai kopling dan penahan sinyal DC agar tidak masuk ke dalam rangkaian.  $R_2$  merupakan resistor umpan balik menentukan penguatan dari rangkaian penerima DTMF sama dengan satu. Dari datasheet *decoder* DTMF IC MT8870 diketahui bahwa level sinyal masukan yang diperlukan sebesar:

$$V_{\min} = 27,5 \text{ mVrms} = 38,8 \text{ mV}$$

$$V_{\max} = 869 \text{ mVrms} = 1228,95 \text{ mV}$$

Dari hasil pengukuran dapat diketahui bahwa tegangan keluaran penerima FM sebesar 800 mV. Sehingga untuk dapat menerima sinyal dengan baik, *decoder* DTMF tidak memerlukan penguatan. Maka besar nilai  $R_2$  sama dengan nilai  $R_1$  untuk menghasilkan penguatan tegangan satu. Besarnya nilai  $R_1 = R_2 = 100 \text{ k}\Omega$  sesuai dengan karakteristik IC dekoder MT8870.

Penguatan tegangan IC dekoder MT8870 adalah:

$$A_v = \frac{R_2}{R_1} = \frac{100\text{k}\Omega}{100\text{k}\Omega} = 1 \tag{4-6}$$

Penentuan waktu tunda ditentukan oleh resistor  $R_3$  dan  $C_2$  dari saat kombinasi nada diterima hingga dikeluarkan. Besar  $R_3$  dan  $C_2$  berturut-turut adalah 390 k $\Omega$  dan 100 nF. Untuk waktu tunda DTMF yang diperlukan DTMF untuk menentukan adanya nada baru  $t_{GTP}$  (Grand Time Tone Present) dapat ditentukan dengan persamaan:

$$t_{GTP} = (R_3 \cdot C_2) \ln \left[ \frac{V_{DD}}{V_{DD} - V_{TSI}} \right] \quad (4-7)$$

Didalam databook besarnya 2.2V dan maksimal 2.5V dengan  $V_{DD}$  besarnya 5V. maka sesuai dengan persamaan diatas, untuk nilai maksimumnya sebesar:

$$\begin{aligned} t_{GTP} &= (390 \cdot 10^3 \times 100 \cdot 10^{-9}) \ln \left[ \frac{5}{(5 - 2.5)} \right] \\ &= 0.027 \text{ s} \end{aligned}$$

dan nilai minimalnya sebesar:

$$\begin{aligned} t_{GTP} &= (390 \cdot 10^3 \times 100 \cdot 10^{-9}) \ln \left[ \frac{5}{(5 - 2.2)} \right] \\ &= 0.0226 \text{ s} \end{aligned}$$

Untuk menentukan besarnya  $t_{GTA}$  (Guard Time Tone Abstract) maksimal, yaitu waktu tunda yang diperlukan untuk mendeteksi bahwa nada yang ditekan telah dilepaskan, dapat ditentukan dengan persamaan:

$$\begin{aligned} t_{GTA} &= (R_3 \cdot C_2) \ln \left[ \frac{V_{DD}}{V_{TSI}} \right] \\ t_{GTA} &= (390 \cdot 10^3 \times 100 \cdot 10^{-9}) \ln \left[ \frac{5}{2.5} \right] \\ &= 0.027 \text{ s} \end{aligned} \quad (4-8)$$

dan nilai minimalnya sebesar:

$$\begin{aligned} t_{GTA} &= (390 \cdot 10^3 \times 100 \cdot 10^{-9}) \ln \left[ \frac{5}{(2.2)} \right] \\ &= 0.0320 \text{ s} \end{aligned}$$

Diagram waktu dari penerima DTMF IC MT8870 ini ditunjukkan dalam Gambar 2.7.

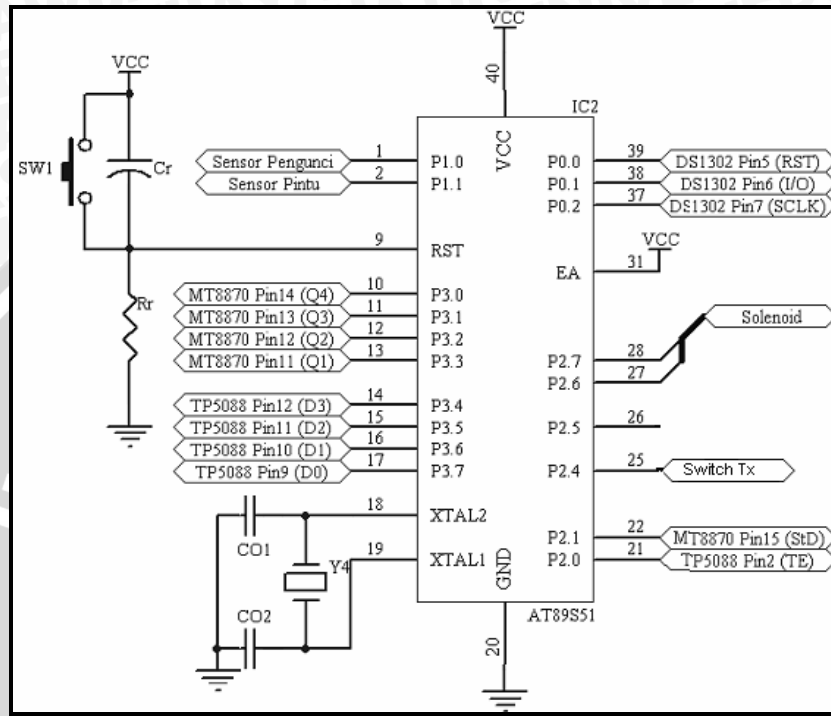
Supaya dapat menerjemahkan sinyal DTMF yang valid, maka pada masukan osilator dari IC MT8870 ini menggunakan kristal sebesar 3.579545 MHz. Untuk setiap kali *decoder* berhasil menerjemahkan sinyal yang diterimanya maka *decoder* akan menghasilkan kode biner dan juga *steering logic* yang menandakan sebuah sinyal telah selesai diterjemahkan. Sinyal ini yang akan digunakan untuk memberi tanda kepada mikrokontroler bahwa telah terdapat data baru pada masukan.

### 4.3.3 Rangkaian Pengolah Data Pada Unit Pengunci Pintu

Mikrokontroler AT89S51 merupakan salah satu komponen utama dalam sistem pengendali kunci pintu jarak jauh dengan transmisi gelombang radio. Mikrokontroler ini berfungsi sebagai pusat pengendali sistem. Jenis mikrokontroler ini dipilih karena

jumlah memori dan pin I/O yang mencukupi untuk penggunaan pengendali kunci pintu jarak jauh.

Sebagai pusat pengolahan data dan pengontrolan alat, pin-pin mikrokontroler AT89S51 dihubungkan pada rangkaian pendukung membentuk sistem minimum yang ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Rangkaian Unit Pengendali Sistem Pada Penerima

Mikrokontroler AT89S51 mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran. Pada perancangan ini pin-pin yang digunakan adalah:

1. Port 0.0 – Port 0.2 sebagai jalur data dari RTC.
2. Port 1.0 – Port 1.1 sebagai jalur data masukan dari sensor.
3. Port 2.0 sebagai jalur data keluaran ke pembangkit DTMF (Pin *Tone Enable*)
4. Port 2.1 sebagai jalur data masukan dari penerima DTMF (Pin *StD*).
5. Port 2.4 sebagai jalur data transistor *switching* pemancar FM.
6. Port 2.6 – 2.7 sebagai jalur data *solenoid*.
7. Port 3.0 – 3.3 sebagai jalur data masukan dari penerima DTMF (Pin *Q4 – Q1*).
8. Port 3.4 – 3.7 sebagai jalur data keluaran ke pembangkit DTMF (Pin *D0 – D3*)

Untuk mengaktifkan mikrokontroler AT89S51 maka perlu diberikan tegangan +5V pada pin 40 (Vcc) dan tegangan 0V (*ground*) pada pin 20 (GND). Disamping itu

diperlukan juga pengaktifan osilator yang terdapat pada mikrokontroler (XTAL1 dan XTAL2). Untuk mengaktifkan osilator tersebut dalam perancangan ini digunakan kristal 12 MHz dan kapasitor 30 pF. Digunakannya kristal 12 MHz untuk memperoleh kecepatan pelaksanaan instruksi per siklus sebesar 1 mikrodetik ( $(1/12 \text{ MHz}) \times 12$  siklus perioda).

Mikrokontroler menggunakan rangkaian *power on reset*. Untuk pin *reset* (RST) diberi rangkaian reset yang terdiri dari sebuah kapasitor 10  $\mu\text{F}$  dan resistor 4,7 k $\Omega$ . Saat rangkaian mikrokontroler diberi catu daya, rangkaian reset akan menahan logika tinggi beberapa saat pada pin RST. Rangkaian reset tersebut akan mereset mikrokontroler pada saat *power on*.

Dari datasheet mikrokontroler AT89S51 diketahui bahwa besarnya  $I_{OL}$  dan  $I_{OH}$  secara tipikal untuk setiap pin adalah:

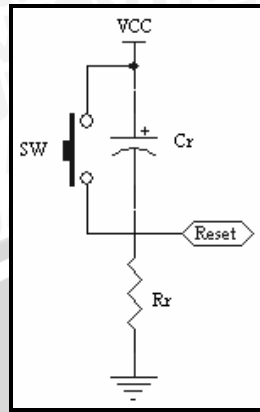
$I_{OL}$ (Port 0)	= 3,2 mA
$I_{OL}$ (Port 1, 2, 3)	= 1,6 mA
$I_{OH}$ (Port 0)	= -60 $\mu\text{A}$
$I_{OH}$ (Port 1, 2, 3)	= -800 $\mu\text{A}$

Dalam perancangan ini, pin-pin output yang aktif pada saat logika *high* sebanyak 7 pin yaitu P2.0 dan P3.4 - P3.7 untuk jalur data ke pembangkit DTMF, P2.4 untuk jalur data transistor *switching* pemancar FM dan P2.6 - P2.7 yang dihubungkan ke *relay*. Besarnya  $I_{OH}$  untuk P3.4 - P3.7 masing-masing sebesar 0,01 mA, untuk P2.0 sebesar 0,02 mA, untuk P2.4 sebesar 0,0524 mA dan untuk P2.6 - P2.7 masing - masing sebesar 0,0567 mA, sehingga jumlah arus yang keluar pada saat output mikrokontroler berlogika *high* adalah sebesar  $[(0,01 \text{ mA} \times 4) + (0,02 \text{ mA} \times 1) + (0,0524 \text{ mA} \times 1) + (0,0567 \text{ mA} \times 2)] = 0,2258 \text{ mA}$ . Sedangkan pada logika *low* tidak terdapat pin - pin output yang aktif.

#### 4.3.3.1 Rangkaian Reset

Mikrokontroler menggunakan rangkaian *power-on reset*, yang mereset mikrokontroler secara otomatis setiap kali catu daya dinyalakan. Rangkaian *power-on reset* tersusun dari sebuah kondensator 10  $\mu\text{F}$  dan resistor 4k7  $\Omega$ . Saat catu daya dinyalakan rangkaian reset akan menahan logika tinggi pada pin RST untuk jangka waktu tertentu. Jangka waktu tersebut ditentukan oleh pengosongan muatan pada kondensator. Logika tinggi pada pin RST harus bertahan selama lebih dari dua siklus

mesin ditambah waktu mulai hidup osilator untuk menjamin keabsahan reset. Gambar rangkaian *power-on reset* ditunjukkan dalam Gambar 4.8 berikut ini.



Gambar 4.8 Rangkaian Reset

Dengan mengacu pada Persamaan (2-7) untuk menentukan nilai R dan C dalam rangkaian reset ini memberikan waktu penundaan (logika tinggi) selama minimal  $2 \mu\text{F}$  ditambah dengan lamanya waktu mulai hidup (*start on*) osilator. Total waktu yang diperlukan adalah:

$$\begin{aligned}
 t \text{ untuk waktu tunda 2 siklus mesin} &= 2 \mu\text{s} \\
 t \text{ untuk waktu tunda } \textit{start on} \text{ osilator} &= 13,206 \mu\text{s} \\
 \text{total waktu tunda minimal yang diperlukan} &= 15,206 \mu\text{s}
 \end{aligned}$$

Dalam perancangan waktu tunda yang digunakan adalah selama 15 ms.

Kemudian untuk nilai kapasitor (C) dapat ditetapkan lebih dulu agar dalam perhitungan dapat mempermudah dalam mencari nilai resistansinya, maka untuk nilai C diberikan kapasitas sebesar  $10 \mu\text{F}$ , sehingga didapatkan:

$$\begin{aligned}
 t &= 0.357 \cdot R \cdot C \\
 R &= \frac{t}{0.357 \cdot C} \\
 &= \frac{15 \cdot 10^{-3}}{0.357 \cdot 10 \cdot 10^{-6}} \\
 &= 4201,68 \Omega \\
 R &= 4201.68 \Omega \approx 4700 \Omega
 \end{aligned} \tag{4-9}$$

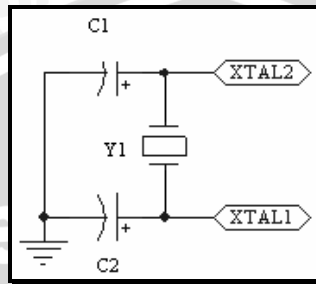
Dari hasil perhitungan waktu reset didapat bahwa rangkaian reset tersebut dapat memberikan keabsahan terhadap waktu tunda yang lebih besar hampir 1000 kali dari nilai minimum untuk lamanya kondisi reset, hal ini diberikan agar mikrokontroler dapat diyakinkan benar-benar untuk dapat mempersiapkan segala sesuatunya baik dalam hal



setting up maupun dalam pengesetan alamat register dan I/O (*self test on start*) awal sebelum mengeksekusi program yang akan dijalankan.

#### 4.3.3.2 Rangkaian Osilator Kristal

Rangkaian osilator kristal yang digunakan adalah seperti ditunjukkan dalam Gambar 4.9.



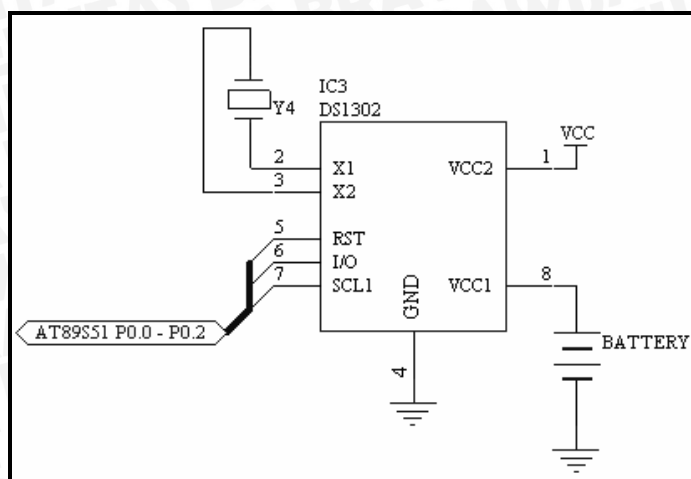
Gambar 4.9 Rangkaian Osilator

Rangkaian ini berfungsi sebagai sumber clock sinyal kepada AT89S51. kristal yang terpasang menggunakan frekuensi 12 MHz, sementara nilai kapasitansi  $C_1$  dan  $C_2$  sebesar 30 pF yang sesuai dengan ketentuan ATMEL yaitu apabila mempergunakan osilator kristal maka diperlukan  $C_1$  dan  $C_2$  dengan nilai sebesar 30 pF  $\pm$  10 pF.

Untuk mengaktifkan osilator tersebut dalam perancangan ini digunakan kristal 12 MHz dan kapasitor 30 pF. Digunakannya kristal 12 MHz ini untuk memperoleh kecepatan pelaksanaan instruksi per siklus sebesar 1 mikrodetik ( $(1/12 \text{ MHz}) \times 12$  siklus perioda).

#### 4.3.4 Rangkaian Real Time Clock (RTC)

RTC pada alat ini digunakan mengatur tanggal, bulan, tahun, detik, menit, maupun jam yang diinginkan. Hubungan antara serial RTC dengan Mikrokontroler AT89S51 ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Rangkaian Real Time Clock (RTC)

DS1302 memiliki pin daya ganda untuk primer dan suply daya cadangan, yaitu VCC1 dihubungkan ke Baterai dan VCC2 dihubungkan ke catu daya 5 Volt. Rangkaian RTC menggunakan kristal 32.768 kHz sebagai osilator. Pin RST', I/O, dan SCLK dihubungkan dengan mikrokontroler pada Port 0.0 – 0.2. Pin – pin ini tidak memerlukan tahanan *pull – down* karena terdapat *internal pull – down resistor 40 kΩ*.

#### 4.3.5 Rangkaian Sensor

Sensor yang digunakan ada 2 macam, yaitu sensor pintu dan sensor pengunci pintu. Sensor pintu berfungsi sebagai indikator keadaan pintu, apakah dalam keadaan membuka atau menutup. Jika pada batas waktu yang telah ditentukan sebelumnya pintu tersebut masih dalam keadaan membuka maka sensor pintu ini akan memberikan respon. Sedangkan sensor pengunci pintu bekerja jika ada respon dari *solenoid*. Respon kedua sensor ini selanjutnya akan dikirimkan ke mikrokontroler untuk ditransmisikan ke bagian pengendali sebagai laporan keberhasilan operasi penguncian pintu untuk sensor pengunci pintu dan sebagai indikator perintah.

Sensor yang dipakai dalam perancangan ini adalah *limitswitch* yang diletakkan di kusen atau bingkai pintu. Agar *limitswitch* ini dapat berfungsi dengan baik maka harus memenuhi persyaratan berikut ini:

$$V_{in} < 0,2 V_{cc} - 0,1$$

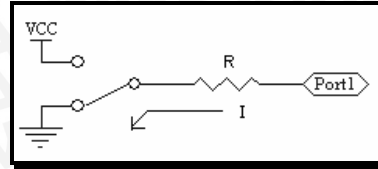
$$V_{in} < 0,2 \times 5V - 0,1$$

$$V_{in} < 0,9 V$$

Dari mikrokontroler diketahui besar  $I_{ILmax} = 50 \mu A$ , sehingga diperlukan hambatan maksimum sebesar:

$$R_{\max} = \frac{V}{I} = \frac{0,9V}{50 \cdot 10^{-6}} = 18k\Omega \quad (4-10)$$

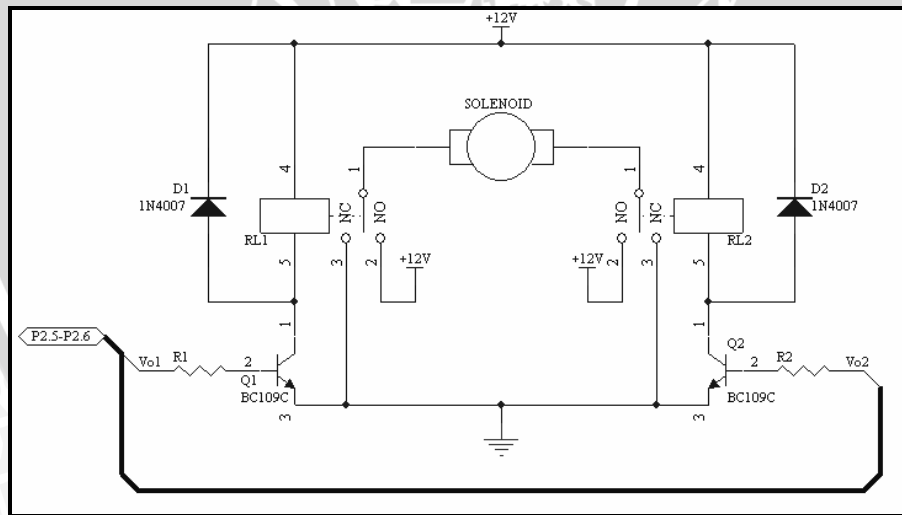
Dalam perancangan ini digunakan nilai  $R = 10k\Omega$  sehingga *limitswitch* aktif.



Gambar 4.11 Rangkaian Limitswitch

### 4.3.6 Rangkaian Driver Relay dan Solenoid

Rangkaian *driver* digunakan untuk menggerakkan *relay* 12 volt yang akan dihubungkan ke *solenoid*. Pada saat output1 mikrokontroler berlogika *high* dan output2 mikrokontroler berlogika *low*, maka kontak dalam *relay1* terhubung ke tegangan 12 volt dan kontak dalam *relay2* akan terhubung ke GND sehingga *solenoid* akan bergerak kesuatu arah. Sedangkan apabila output1 mikrokontroler berlogika *low* dan output2 mikrokontroler berlogika *high*, maka kontak dalam *relay1* terhubung ke GND dan kontak dalam *relay2* akan terhubung ke tegangan 12 volt sehingga *solenoid* akan bergerak ke arah sebaliknya.



Gambar 4.12 Rangkaian Driver Relay dan Solenoid

Dari *datasheet* mikrokontroler AT89S51 diketahui bahwa pada saat arus keluaran mikrokontroler sebesar  $60 \mu A$  maka tegangan keluaran minimum dijamin sebesar 2,4 V. Sedangkan dari *datasheet relay* yang mempunyai spesifikasi hampir

sama dengan *relay* yang digunakan diketahui bahwa arus kerja *relay* sebesar 21,8 mA. Sehingga dari data-data tersebut nilai  $\beta_{\min}$  yang diperlukan adalah:

$$\beta_{\min} = \frac{I_{\text{Relay}}}{I_{OH}} \quad (4-11)$$

$$\begin{aligned} \beta_{\min} &= \frac{21,8 \cdot 10^{-3}}{60 \cdot 10^{-6}} \\ &= 363,33 \end{aligned}$$

Sehingga transistor yang digunakan harus memiliki  $\beta$  lebih besar dari 363,33 dan arus  $I_C$  yang lebih besar dari 21,8 mA. Transistor BC109C memenuhi spesifikasi di atas karena transistor ini memiliki  $\beta_{\min}$  sebesar 420 dan arus kolektor maksimum 100 mA.

Dari *datasheet* transistor BC109C diketahui bahwa:

$$\begin{aligned} I_{C(\max)} &= 100 \text{ mA} \\ V_{BE(\max)} &= 0,7 \text{ V} \\ \beta_{(\min)} &= 420 \text{ dan } \beta_{(\max)} = 800 \end{aligned}$$

Transistor BC109C akan memerlukan arus basis untuk memastikan transistor akan tetap saturasi pada level arus *relay*, sehingga arus basis transistor harus lebih besar daripada arus *relay* dibagi dengan penguatan minimum transistor. Sehingga:

$$I_B \geq \frac{I_{\text{Relay}}}{\beta_{\min}} \quad (4-12)$$

$$I_B \geq \frac{21,8 \cdot 10^{-3}}{420}$$

$$I_B \geq 51,9 \mu\text{A}$$

Sehingga sesuai dengan Persamaan (4-3) nilai resistor pembatas arus pada basis adalah:

$$V_{OH(\min)} - I_B \cdot R_B - V_{BE(\max)} = 0$$

$$R_{B_{\max}} = \frac{V_{OH(\min)} - V_{BE(\max)}}{I_B}$$

$$= \frac{(2,4 - 0,7) \text{ V}}{0,0519 \text{ mA}}$$

$$= 32,75 \text{ k}\Omega$$

Dalam perancangan ini digunakan nilai  $R_B = 30 \text{ k}\Omega$  yang akan memberikan arus

$$I_B = 56,67 \mu\text{A}.$$

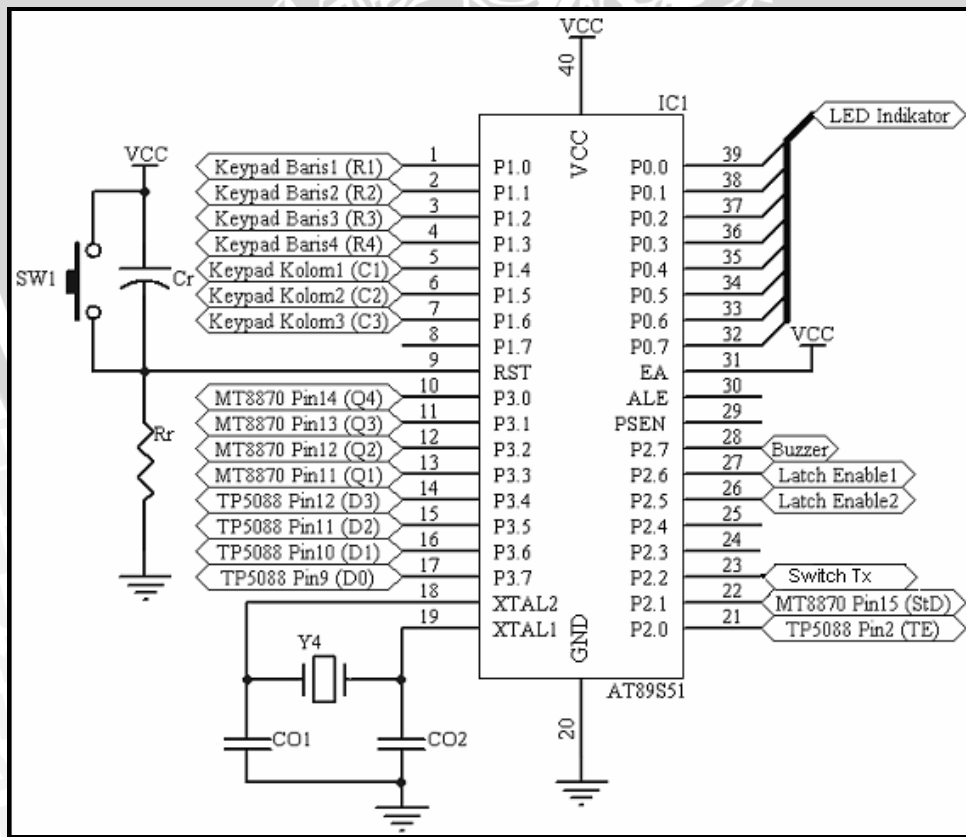
### 4.3.7 Rangkaian Pengolah Data Pada Unit Pengendali

Mikrokontroler yang digunakan pada pengendali ini mempunyai jenis yang sama pada bagian penerima, yaitu AT89S51. Rangkaian reset, osilator, dan cara mengaktifkan mikrokontroler ini telah dibahas dalam unit pengolah data pada penerima.

Pada perancangan ini konfigurasi pin yang dipakai adalah sebagai berikut:

1. Port 0.0 – Port 0.7 sebagai jalur data untuk LED indikator.
2. Port 1.0 – Port 1.6 sebagai jalur pengambilan kode dari rangkaian *keypad*.
3. Port 2.0 sebagai jalur data ke pembangkit DTMF (Pin *Tone Enable*).
4. Port 2.1 sebagai jalur data ke penerima DTMF (Pin *StD*).
5. Port 2.2 sebagai jalur data transistor *switching* pemancar FM.
6. Port 2.5 – 2.6 sebagai jalur data ke *D – Latch* (Pin *Latch Enable*).
7. Port 2.7 sebagai jalur data ke *buzzer*.
8. Port 3.0 – Port 3.3 sebagai jalur data ke penerima DTMF (Pin Q4 – Q1).
9. Port 3.4 – 3.7 sebagai jalur data ke pembangkit DTMF (Pin D3 – D1).

Blok rangkaian yang menjelaskan perancangan tersebut ditunjukkan dalam Gambar 4.13.



Gambar 4.13 Rangkaian Unit Pengolah Data Pada Pengendali

Dari datasheet mikrokontroler AT89S51 diketahui bahwa besarnya  $I_{OL}$  dan  $I_{OH}$  secara tipikal untuk setiap pin adalah:

$I_{OL}$ (Port 0)	= 3,2 mA
$I_{OL}$ (Port 1, 2, 3)	= 1,6 mA
$I_{OH}$ (Port 0)	= -60 $\mu$ A = 0,06 mA
$I_{OH}$ (Port 1, 2, 3)	= -800 $\mu$ A = 0,8 mA

Dalam perancangan ini, pin-pin output yang aktif pada saat logika *high* sebanyak 6 pin yaitu P2.1 dan P3.0 – P3.3 untuk jalur data ke pembangkit DTMF, P2.2 untuk jalur data transistor *switching* pemancar FM, P2.5 – 2.6 untuk jalur data ke masukan *Latch Enable* dan P2.7 yang dihubungkan ke *buzzer*. Besarnya  $I_{OH}$  untuk P3.0 - P3.3 masing-masing sebesar 0,01 mA, untuk P2.1 sebesar 0,02 mA, untuk P2.2 sebesar 0,0524 mA, untuk P2.5 – 2.6 masing – masing sebesar 0,02 mA dan untuk P2.7 sebesar 0,063 mA. Sehingga jumlah arus yang keluar pada saat output mikrokontroler berlogika *high* adalah sebesar  $[(0,01 \text{ mA} \times 4) + (0,02 \text{ mA} \times 1) + (0,0524 \text{ mA} \times 1) + (0,02 \text{ mA} \times 2) + (0,063 \text{ mA} \times 1)] = 0,2154 \text{ mA}$ . Sedangkan pin-pin output yang aktif pada logika *low* sebanyak 12 pin yaitu P0.0 - P0.7 dihubungkan dengan LED indikator dan P1.0 - P1.3 dihubungkan ke *keypad*. Besarnya  $I_{OL}$  untuk P0.0 - P0.7 masing – masing sebesar 0,4 mA dan untuk P1.0 - P1.3 masing – masing sebesar 0,008 mA. Sehingga jumlah arus yang masuk pada saat pin output aktif pada logika *low* adalah sebesar  $[(0,4 \text{ mA} \times 8) + (0,008 \text{ mA} \times 4)] = 3,232 \text{ mA}$ .

#### 4.3.8 Rangkaian Keypad

Rangkaian *keypad* berfungsi sebagai unit masukan dari mikrokontroler yang diantarmukakan ke port 1 pada sistem mikrokontroler. *Keypad* yang digunakan adalah *keypad* 4 baris x 3 kolom dan terdiri dari 12 buah tombol. *Keypad* 4x3 memiliki 2 buah terminal, masing-masing terminal dari setiap tombol dihubungkan ke kelompok baris dan kelompok kolom. Rangkaian *keypad* ditunjukkan dalam Gambar 4.14.

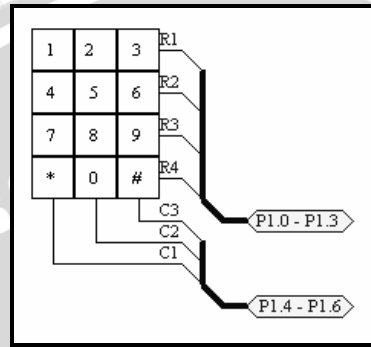
Metode yang digunakan untuk membaca data dari *keypad* menggunakan metode *scanning*. Pada metode ini, setiap kolom secara berurutan diberi logika rendah (di *scan* dengan mengeluarkan logika 0 ke setiap kolom satu demi satu) ketika pada saat yang sama 4 baris diambil. Jika sebuah tombol ditekan pada saat kolom yang bersangkutan berlogika rendah, maka logika rendah ini akan terhubung dengan pin input melalui saklar yang terhubung dan baris yang bersangkutan akan berlogika rendah. Dalam

perancangan ini, periode *scanning* yang digunakan sebesar 1 ms. Sehingga frekuensi *scanning* dalam tiap detiknya adalah:

$$f = \frac{1}{T} \quad (4-13)$$

jika  $T = 1 \text{ ms}$ , maka:

$$f = \frac{1}{0,001} = 1000 \text{ Hz}$$



**Gambar 4.14** Rangkaian *Keypad* Sistem Mikrokontroler

Dengan rangkaian *keypad* yang telah ditunjukkan dalam Gambar 4.14, masing - masing tombol *keypad* direncanakan seperti ditunjukkan dalam Tabel 4.1.

**Tabel 4.1** Perencanaan Tombol *Keypad*

Nama Tombol	Keterangan
Key 1	Pintu 1
Key 2	Pintu 2
Key 3	Pintu 3
Key 4	Pintu 4
Key 5	Pintu 5
Key 6	Pintu 6
Key 7	Pintu 7
Key 8	Pintu 8
Key 9	Pintu 9
Key #	Perintah "Tutup Kunci"
Key 0	Pintu 10
Key *	Perintah "Buka Kunci"

Demi keamanan dalam pengiriman data, maka data yang dikirimkan dikonfigurasi secara khusus. Konfigurasi data tersebut adalah sebagai berikut:

Data	Kode Pintu	Kode Perintah	Data
------	------------	---------------	------

Contoh:

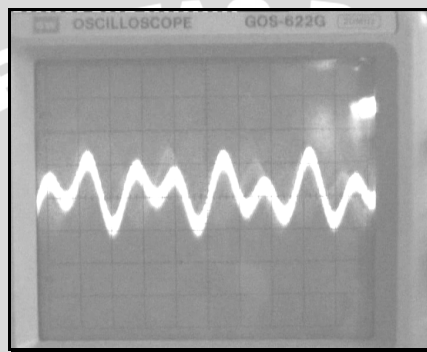
Jika *user* melakukan operasi penguncian pada pintu 1, maka data yang dikirimkan adalah:

0	1	#	2
---	---	---	---

Sehingga data yang akan diterima oleh unit pengunci pintu adalah data 16 byte dari konfigurasi data yang terkirim. Dengan mengambil contoh diatas maka konfigurasi data yang akan diterima adalah sebagai berikut:

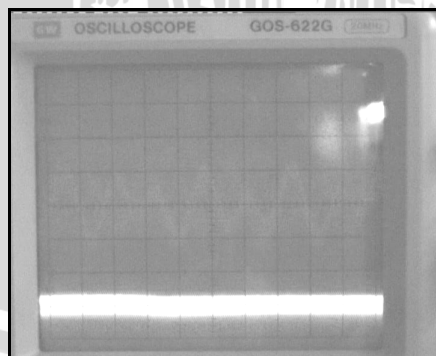
Data Terkirim	0	1	#	2
Kode DTMF	1010	0001	1100	0010

Keempat data tersebut akan diterima oleh unit pengunci pintu satu-persatu. Data DTMF pertama (sinyal DTMF '0') yang diterima akan disimpan ke alamat Data1. Gambar 4.15 menunjukkan gambar sinyal DTMF Data1 yang diterima oleh unit pengunci pintu.



**Gambar 4.15** Sinyal DTMF Data1

Kemudian sinyal DTMF untuk Data2 (sinyal DTMF '1') dapat diterima setelah unit pengunci pintu mendeteksi adanya sinyal kosong antara sinyal DTMF Data1 dan Data2. Demikian selanjutnya untuk sinyal DTMF data berikutnya, sehingga keempat sinyal DTMF dapat diterima dengan baik oleh unit pengunci pintu.

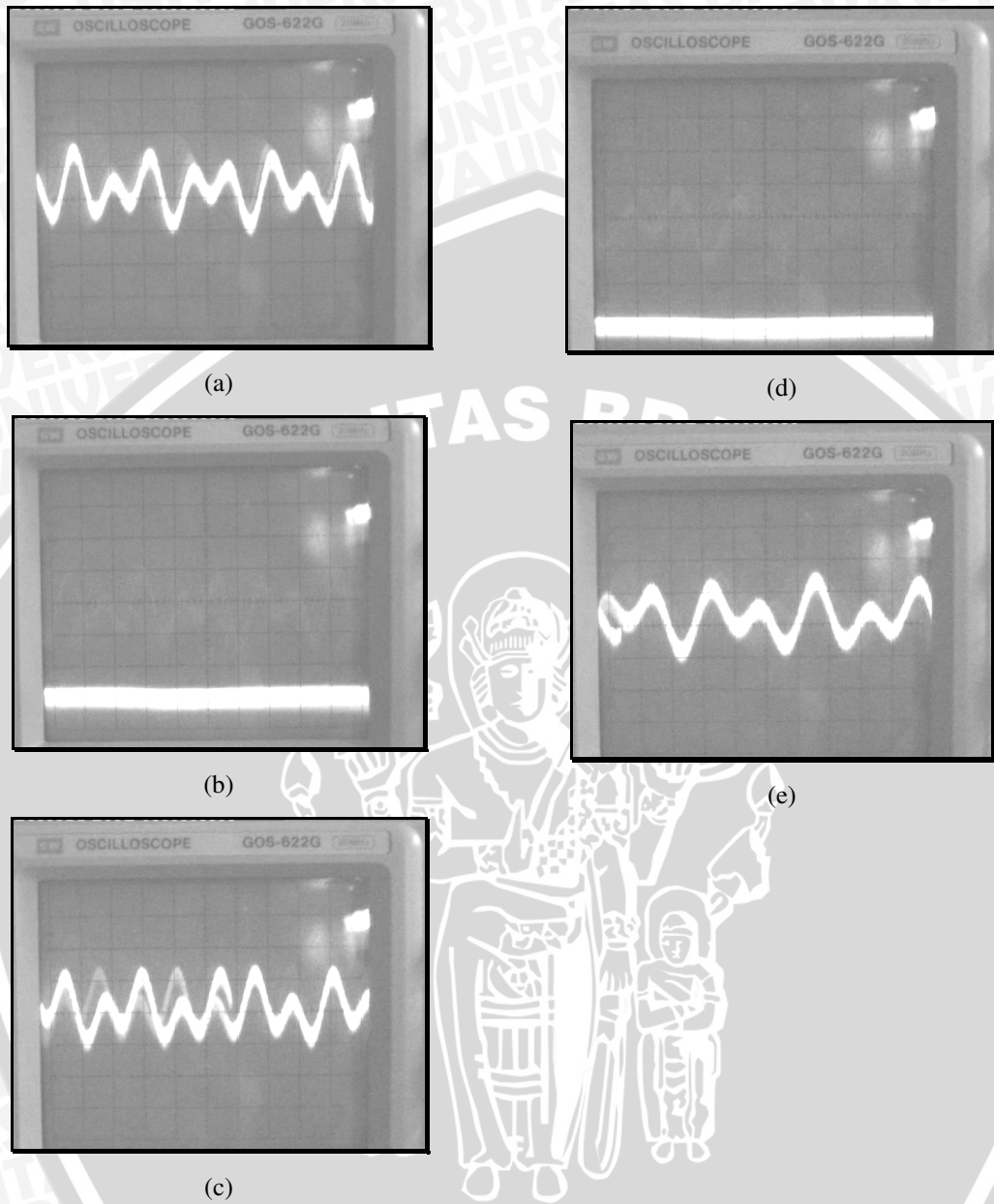


**Gambar 4.16** Sinyal Kosong Antara Data1 dan Data2

Tiga sinyal DTMF data berikutnya secara berurutan diterima oleh unit pengunci pintu dan disimpan ke alamat Data1+1 (sinyal DTMF '1'), Data1+2 (sinyal DTMF '#'),



dan Data1+3 (sinyal DTMF '2'). Maka bentuk sinyal-sinyal DTMF berikutnya yang diterima oleh unit pengunci pintu ditunjukkan dalam Gambar 4.17 berikut ini.



**Gambar 4.17** Data DTMF 1 perintah (a). Sinyal DTMF Data2 (b). Sinyal Kosong Antara Data2 dan Data3 (c). Sinyal DTMF Data3 (d). Sinyal Kosong Antara Data3 dan Data4 (e). Sinyal DTMF Data4

Selanjutnya data-data tersebut oleh unit pengunci pintu dibandingkan dengan data-data yang ada dalam memori. Jika data tersebut benar, maka data tersebut digunakan untuk mengaktifkan *solenoid* sesuai dengan perintah yang diinginkan. Konfigurasi data yang akan dikirim secara keseluruhan ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Konfigurasi Pengiriman Data

Nama Tombol	Konfigurasi Data				Data
	Data	Kode Pintu	Kode Perintah		
			“Tutup”	“Buka”	
Key 1	0	1	#	*	2
Key 2	1	2	#	*	3
Key 3	2	3	#	*	4
Key 4	3	4	#	*	5
Key 5	4	5	#	*	6
Key 6	5	6	#	*	7
Key 7	6	7	#	*	8
Key 8	7	8	#	*	9
Key 9	8	9	#	*	0
Key 0	9	0	#	*	1

#### 4.3.9 Rangkaian *D – Latch* dan Indikator LED (*Light Emitting Diode*)

Dalam perancangan sistem ini LED digunakan sebagai indikator keberhasilan penguncian pintu dan indikator keadaan pintu. Dalam perancangan alat ini menggunakan 16 buah LED sebagai indikatornya. 8 buah LED yang pertama sebagai indikator keberhasilan penguncian pintu dan 8 buah LED berikutnya sebagai indikator keadaan pintu.

Berdasarkan databook mikrokontroler AT89S51 dapat diketahui bahwa besarnya  $I_{OL}$  dan  $V_{OH}$  tipikal tiap pin adalah:

$$I_{OL} = 3,2 \text{ mA} \quad (\text{Port 0})$$

$$V_{OH} = 0,45 \text{ V} \quad (\text{Port 0})$$

$$\text{Maksimum } I_{OL} \text{ Port 0} = 26 \text{ mA}$$

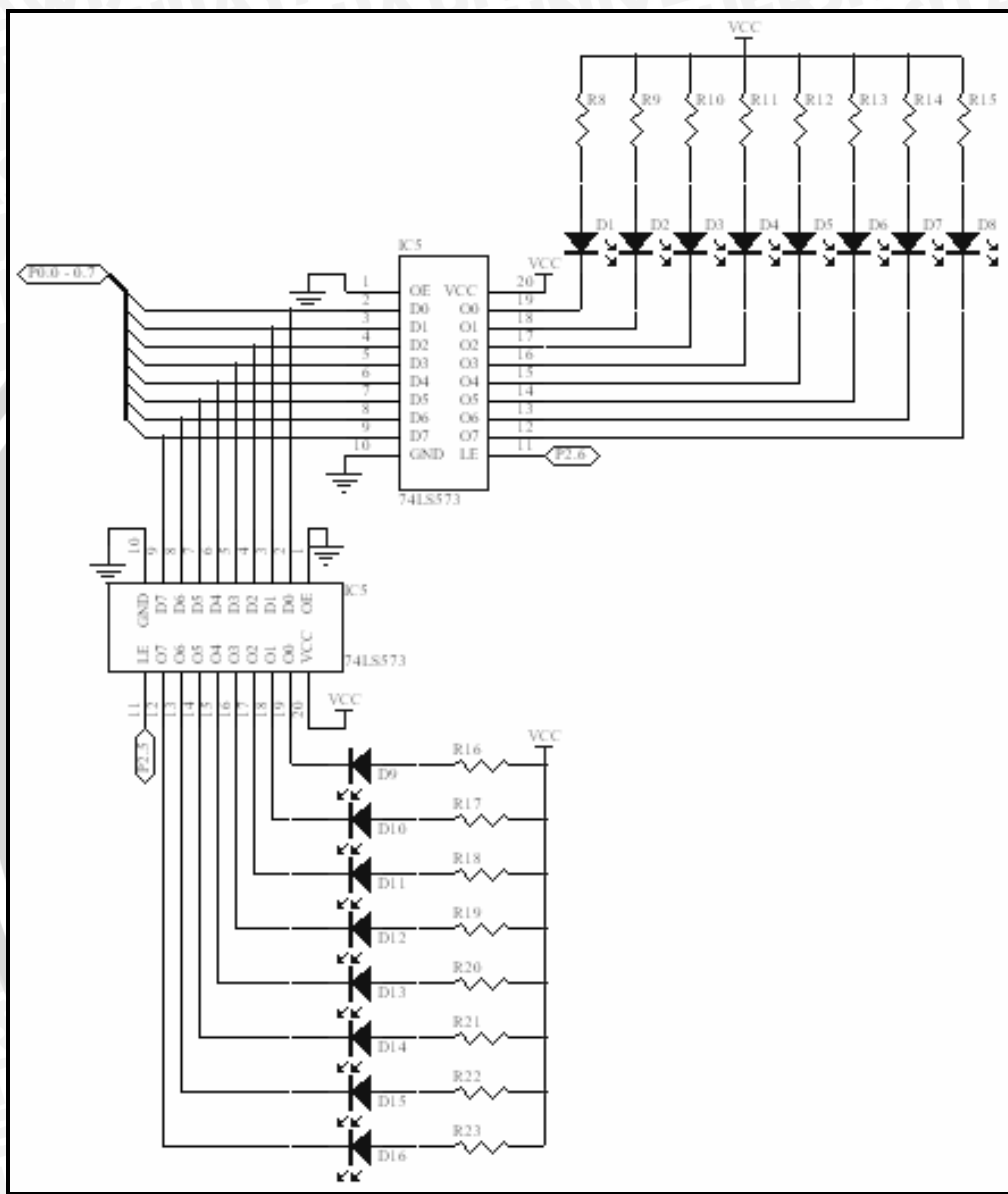
Namun port 0 ini hanya dapat digunakan untuk 8 buah LED indikator. Agar alat ini dapat bekerja dengan efektif dengan tetap mengutamakan efisiensi alat, maka digunakanlah *D – Latch* yang digunakan untuk mengatur kerja 16 buah LED indikator. Gambar rangkaian *D – Latch* untuk LED indikator ditunjukkan dalam Gambar 4.15.

Dalam perancangan ini menggunakan dua buah IC *D – Latch* tipe 74LS573. Pin masukan kedua IC 74LS573 (D0 – D7) dihubungkan ke mikrokontroler port 0.0 – 0.7. Pin keluarannya (O0 – O7) dihubungkan ke masing – masing LED indikator. Pin  $\overline{OE}$  merupakan *3-STATE output enable input* (aktif *low*) dihubungkan ke *ground*. Sedangkan pin *LE* merupakan *Latch Enable* (aktif *high*) dihubungkan ke port 2.5 – 2.6.

Dengan rangkaian LED indikator yang telah ditunjukkan dalam Gambar 4.18, masing - masing keterangan LED indikator direncanakan seperti ditunjukkan dalam Tabel 4.3.

Tabel 4.3 Perencanaan LED Indikator

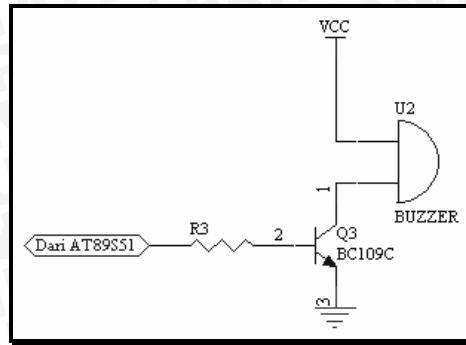
Keadaan LED		Keadaan Pengunci Pintu	Keadaan Pintu
ONyala	Merah	Tertutup	Tertutup
	Kuning		
Padam	Merah	Terbuka	Terbuka
	Kuning		



Gambar 4.18 Rangkaian D – Latch Untuk LED Indikator

#### 4.3.10 Rangkaian Driver Alarm

Rangkaian ini berfungsi untuk mengaktifkan buzzer bila sensor pintu memberikan sinyal ke mikrokontroler yang menunjukkan terdapat pintu yang masih dalam keadaan terbuka. Sehingga pengguna dapat mengetahui keadaan tersebut. Gambar rangkaian driver alarm ditunjukkan dalam Gambar 4.19.



Gambar 4.19 Rangkaian Driver Alarm

Dari *datasheet* mikrokontroler AT89S51 diketahui bahwa arus keluaran mikrokontroler pada saat logika *high* adalah  $-60 \mu\text{A}$ . Sedangkan dari *datasheet buzzer* yang mempunyai spesifikasi hampir sama dengan *buzzer* yang digunakan diketahui bahwa arus kerja *buzzer* sebesar  $25 \text{ mA}$ . Sehingga dari data-data tersebut sesuai dengan Persamaan (4-11) nilai  $\beta_{\min}$  yang diperlukan adalah:

$$\beta_{\min} = \frac{I_{\text{Buzzer}}}{I_{OH}}$$

$$\begin{aligned} \beta_{\min} &= \frac{25 \cdot 10^{-3}}{60 \cdot 10^{-6}} \\ &= 416,67 \end{aligned}$$

Sehingga transistor yang digunakan harus memiliki  $\beta$  lebih besar dari 416,67 dan arus  $I_C$  yang lebih besar dari  $25 \text{ mA}$ . Transistor BC109C memenuhi spesifikasi di atas karena transistor ini memiliki  $\beta_{\min}$  sebesar 420 dan arus kolektor maksimum  $100 \text{ mA}$ .

Dari *datasheet* transistor BC109C diketahui bahwa:

$$I_{C(\max)} = 100 \text{ mA}$$

$$V_{BE(\max)} = 0,7 \text{ V}$$

$$\beta_{(\min)} = 420 \text{ dan } \beta_{(\max)} = 800$$

Transistor BC109C akan memerlukan arus basis untuk memastikan transistor akan tetap saturasi pada level arus *buzzer*, sehingga arus basis transistor harus lebih besar daripada arus *buzzer* dibagi dengan penguatan minimum transistor. Dengan Persamaan (4-12) diketahui:

$$I_B \geq \frac{I_{\text{Buzzer}}}{\beta_{\min}}$$

$$I_B \geq \frac{25 \cdot 10^{-3}}{420}$$

$$I_B \geq 59,52 \mu A$$

Sehingga sesuai dengan Persamaan (4-3) nilai resistor pembatas arus pada basis adalah:

$$\begin{aligned} V_{OH(\min)} - I_B \cdot R_B - V_{BE(\max)} &= 0 \\ R_{B_{\max}} &= \frac{V_{OH(\min)} - V_{BE(\max)}}{I_B} \\ &= \frac{(2,4 - 0,7) V}{0,0595 \text{ mA}} \\ &= 28,57 k\Omega \end{aligned}$$

Dalam perancangan ini digunakan nilai  $R_B = 27 k\Omega$  yang akan memberikan arus  $I_B = 62,96 \mu A$ .

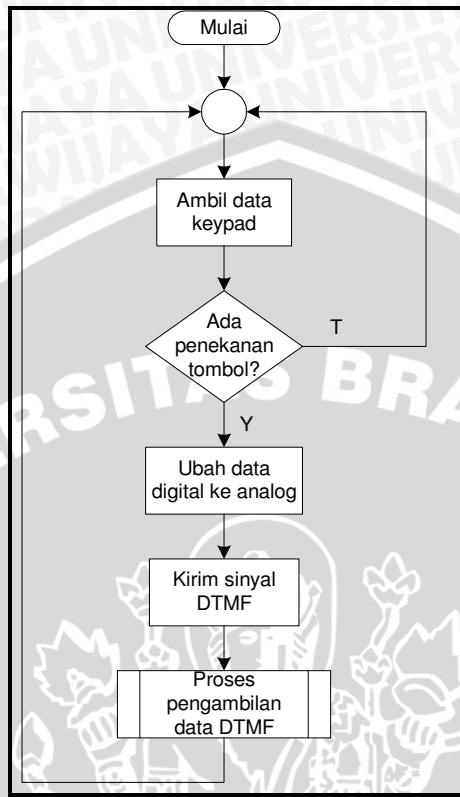
#### 4.4 Perencanaan Perangkat Lunak (*Software*)

Bahasa pemrograman yang akan digunakan dalam pembuatan perangkat lunak dari perancangan ini adalah bahasa pemrograman *assembly*. Sistem kerja dari pengendali kunci pintu jarak jauh dengan transmisi gelombang radio ini terbagi menjadi dua bagian, yaitu sistem kerja pada unit pengendali dan sistem kerja pada unit pengunci pintu. Dalam sistem ini akan terjadi komunikasi dua arah antara unit pengendali dan unit pengunci pintu. Perangkat lunak yang digunakan untuk mengendalikan sistem ini antara lain: proses pengendalian kunci pintu secara manual oleh *user*, proses pengontrolan keadaan pintu dengan *RTC* dan proses pengendalian kunci pintu secara otomatis dengan *RTC*.

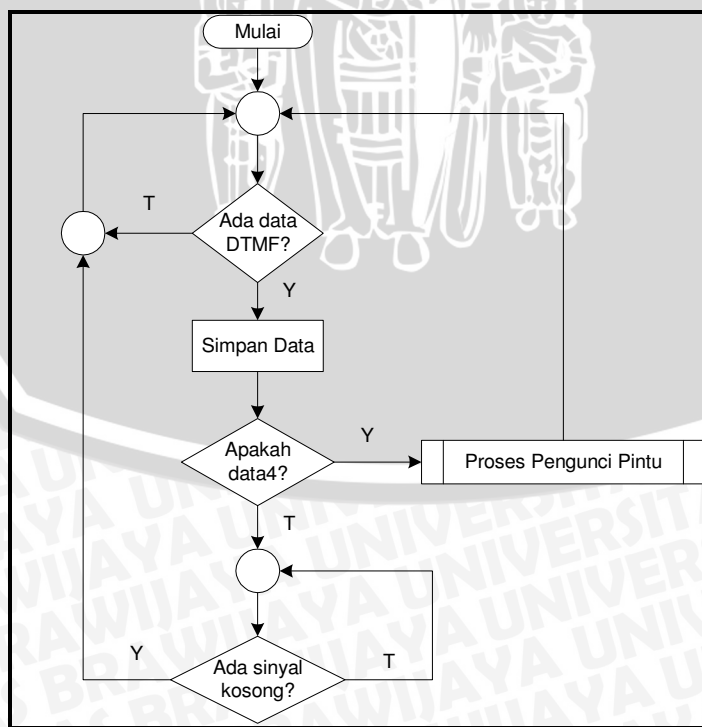
##### 4.4.1 Perencanaan Perangkat Lunak Sistem Pengendalian Kunci Pintu Manual oleh *User*

Untuk dapat membuka pengunci pintu atau mengunci pintu dari jarak jauh maka *user* harus mengendalikannya dari sebuah unit pengendali yang telah dirancang untuk memudahkan *user*. Cukup dengan menekan tombol yang sesuai dengan nomor pintu yang diinginkan maka sistem ini akan memberikan perintah kepada unit pengunci pintu untuk membuka atau menutup pengunci pintu. Pengendali kunci pintu ini dilengkapi dengan indikator, sehingga *user* dapat mengetahui keberhasilan proses penguncian yang baru saja dilakukan.

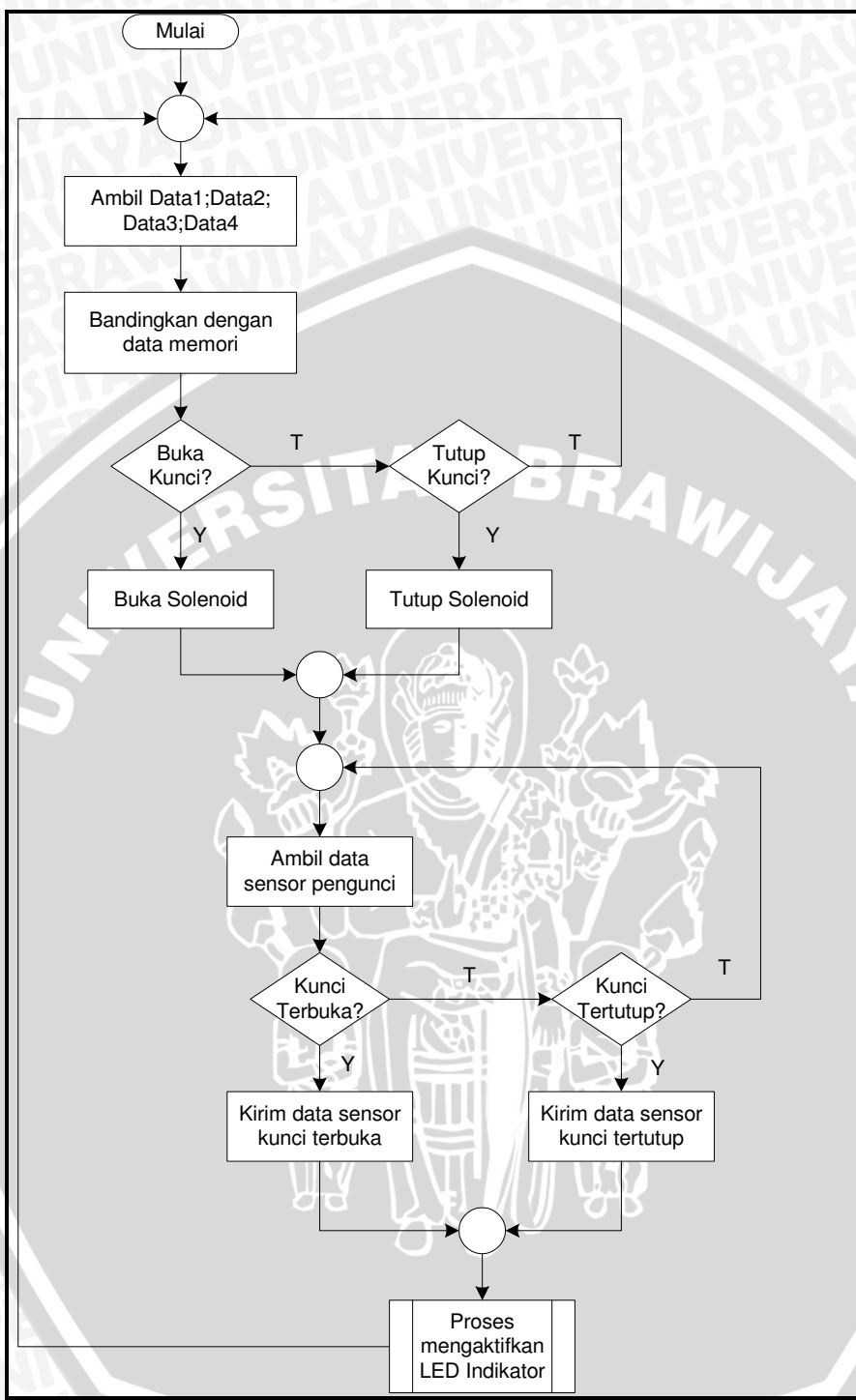
Proses awal terjadi pada unit pengendali. Hasil penekanan tombol keypad oleh user akan dikirimkan ke unit pengunci pintu. Diagram alir utama dari proses penguncian manual ini ditunjukkan dalam Gambar 4.20.



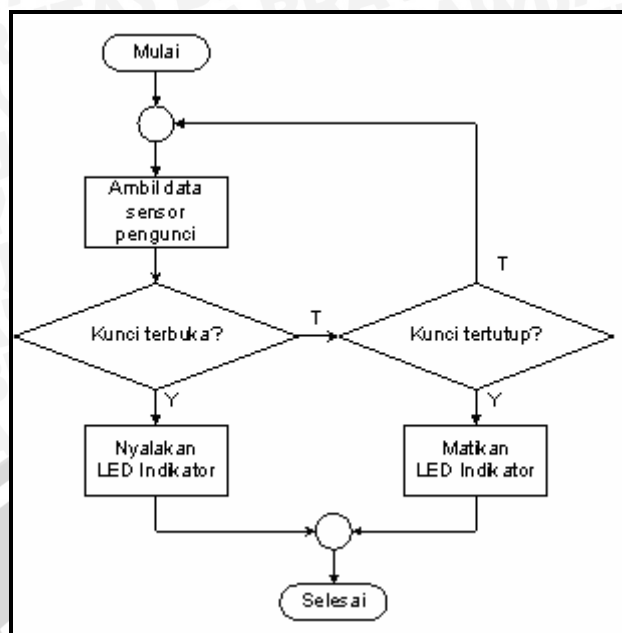
Gambar 4.20 Diagram Alir Utama Pengendali Penguncian Pintu Secara Manual oleh User



Gambar 4.21 Diagram Alir Proses Pengambilan Data DTMF



Gambar 4.22 Diagram Alir Proses Pengunci Pintu



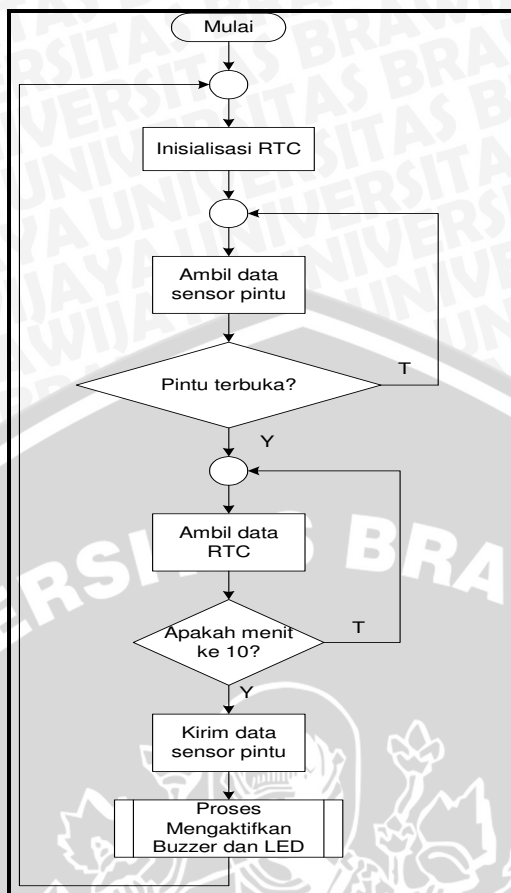
Gambar 4.23 Diagram Alir Proses Mengaktifkan LED Indikator

#### 4.4.2 Perancangan Perangkat Lunak Sistem Pengontrol Keadaan Pintu

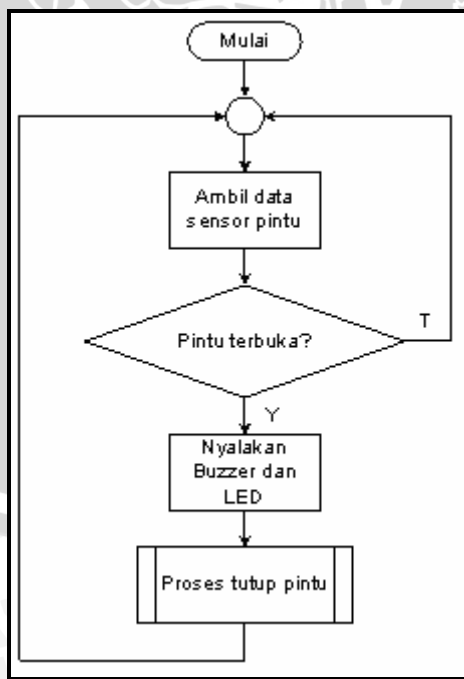
Pada sistem pengontrol keadaan pintu ini, sensor pintu akan memberikan sinyal kepada mikrokontroler jika sampai pada batas waktu yang ditentukan ada pintu yang masih terbuka. Sinyal tersebut akan diolah oleh mikrokontroler dan akan dikirimkan kepada *user* sebagai alarm pengingat. Diagram alir utama dari pengontrolan sistem ini ditunjukkan dalam Gambar 4.24.



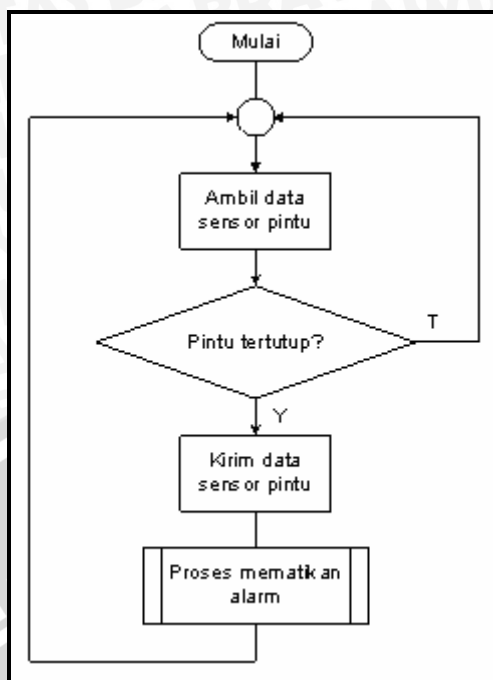




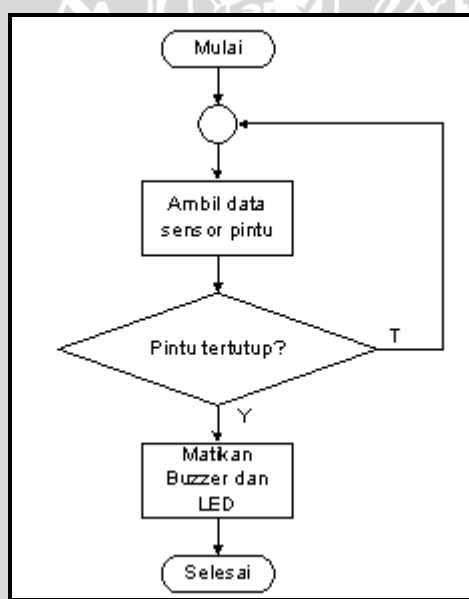
Gambar 4.24 Diagram Alir Utama Sistem Pengontrol Keadaan Pintu



Gambar 4.25 Diagram Alir Proses Mengaktifkan Buzzer dan LED



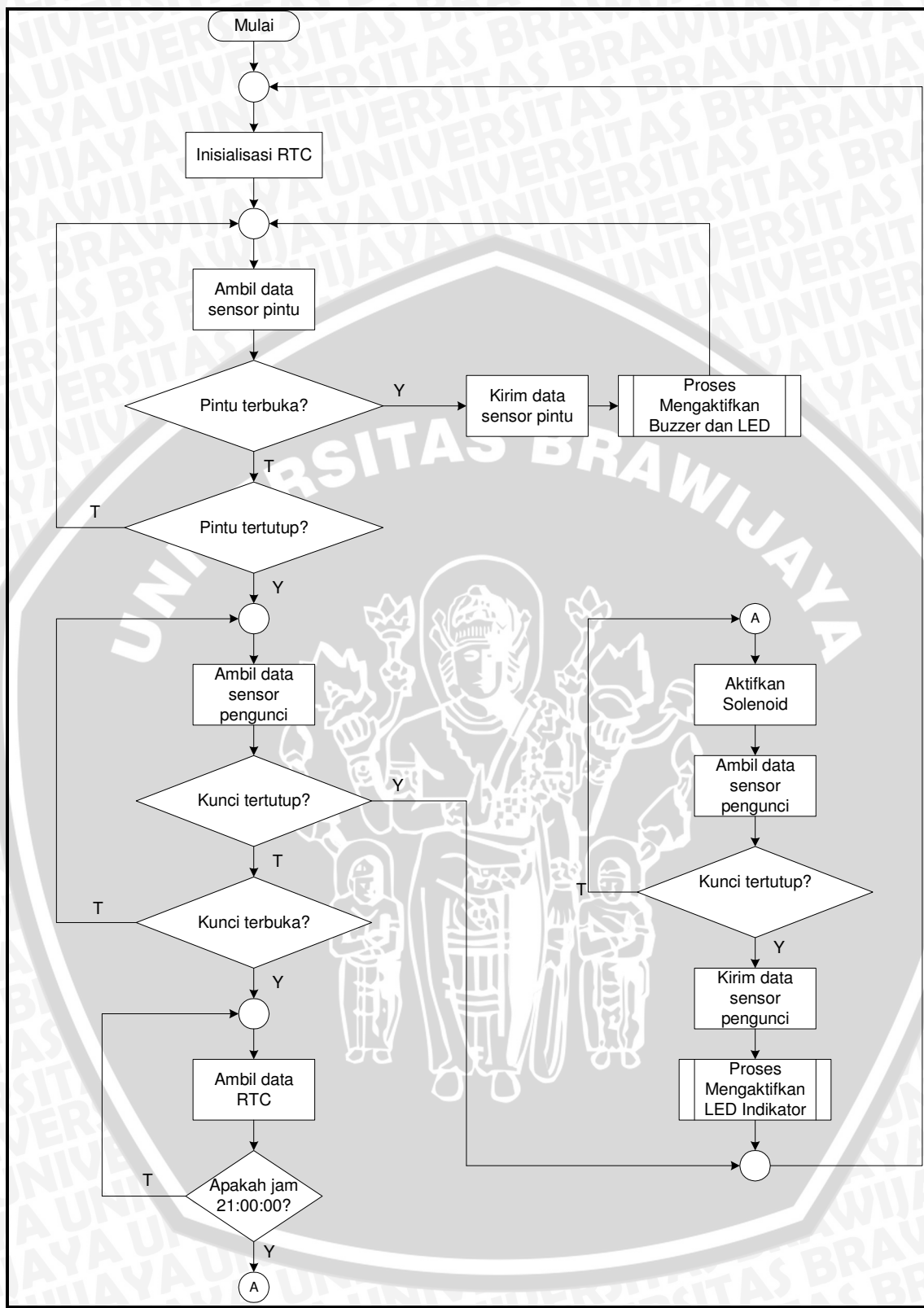
Gambar 4.26 Diagram Alir Proses Tutup Pintu



Gambar 4.27 Diagram Alir Proses Mematikan Alarm

#### 4.4.3 Perancangan Perangkat Lunak Sistem Penguncian Pintu Secara Otomatis

Pengaturan proses penguncian pintu secara otomatis ini menggunakan *Real Time Clock* (RTC). Pada satu waktu yang telah ditentukan maka semua pengunci pintu yang masih terbuka melakukan penguncian secara otomatis. Proses berhasil atau tidaknya penguncian ini juga dapat diketahui melalui indikator yang ada. Diagram alir dari sistem penguncian pintu otomatis ini ditunjukkan dalam Gambar 4.28.



Gambar 4.28 Diagram Alir Utama Sistem Penguncian Pintu Secara Otomatis

## BAB V

### PENGUJIAN DAN ANALISIS DATA

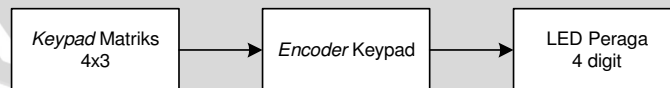
Untuk mengetahui hasil perancangan, dilakukan pengujian dan analisis data terhadap alat yang telah dibuat. Pengujian dilakukan terhadap masing-masing blok rangkaian dengan tujuan untuk mengetahui apakah blok-blok rangkaian tersebut bekerja sesuai dengan yang diharapkan. Setelah dilakukan pengujian untuk masing-masing blok, kemudian dilakukan pengujian sistem secara keseluruhan.

Pengujian yang dilakukan antara lain:

1. Pengujian Rangkaian *Keypad*
2. Pengujian Rangkaian *Encoder* DTMF TP5088
3. Pengujian Rangkaian *Decoder* DTMF MT8870
4. Pengujian Rangkaian Pemancar dan Penerima FM
5. Pengujian Rangkaian *Real Time Clock* (RTC)
6. Pengujian Rangkaian *Driver Relay* dan *Solenoid*
7. Pengujian Sistem Secara Keseluruhan

#### 5.1 Pengujian Rangkaian *Keypad*

Pengujian rangkaian *keypad* ini dilakukan untuk menunjukkan bahwa setiap penekanan tombol *keypad* menghasilkan kode tertentu yang mewakili karakter yang tertera pada setiap tombol *keypad* sesuai dengan perencanaan. Peralatan yang akan digunakan antara lain: sebuah modul *keypad* matriks 3x4, sebuah modul *encoder keypad* 74C922 dan peraga LED 4 digit. Prosedur pengujian rangkaian *keypad* ini adalah mula-mula menghubungkan modul *keypad*, modul *encoder* dan peraga LED 4 digit. Kemudian memberikan catu daya +5V dan mengamati keluaran LED peraga. Blok diagram rangkaian pengujian *keypad* ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Blok Diagram Pengujian *Keypad*

Hasil pengujian yang diperoleh adalah semua karakter hasil penekanan tombol dapat ditampilkan dengan benar di LED peraga 4 digit. Hasil penekanan tersebut sesuai dengan tabel kebenaran yang tersedia pada databook *encoder keypad*. Dari hasil yang diperoleh dalam pengujian ini dapat disimpulkan bahwa rangkaian *keypad* dapat

berfungsi dengan baik sesuai dengan perencanaan. Hasil pengujian *keypad* dapat ditunjukkan dalam Tabel 5.1

Tabel 5.1 Hasil Pengujian *Keypad*

Posisi Tombol		1	2	3	4	5	6	7	8	9	*	0	#
		R1,C1	R1,C2	R1,C3	R2,C1	R2,C2	R2,C3	R3,C1	R3,C2	R3,C3	R4,C1	R4,C2	R4,C3
Data Output	A	0	1	0	0	1	0	0	1	0	0	1	0
	B	0	0	1	0	0	1	0	0	1	0	0	1
	C	0	0	0	1	1	1	0	0	0	1	1	1
	D	0	0	0	0	0	0	1	1	1	1	1	1

## 5.2 Pengujian Rangkaian *Encoder DTMF TP5088*

Pengujian rangkaian *encoder DTMF* berguna untuk mengetahui apakah rangkaian *encoder DTMF* dapat mengkodekan bit – bit biner menjadi frekuensi untuk dapat dipancarkan melalui gelombang radio. Peralatan yang digunakan antara lain: sebuah modul *keypad* matriks 3x4, sebuah modul *encoder keypad 74C922*, rangkaian *Encoder DTMF TP5088*, *Oscilloscope*. Prosedur pengujian yang dilakukan adalah dengan menghubungkan rangkaian seperti dalam blok diagram pengujian yang ditunjukkan dalam Gambar 5.2. Pin *Tone Output* dihubungkan pada osiloskop kanal 1. Pin *Tone Enable* dihubungkan ke *ground*. Kemudian memberikan catu daya +5V dan mengamati keluaran yang ditunjukkan dalam osiloskop.



Gambar 5.2 Blok Diagram Pengujian *Encoder DTMF*

Hasil pengujian dari *encoder DTMF TP5088* ditunjukkan dalam Tabel 5.2 berikut.

Tabel 5.2 Hasil Pengujian *Encoder DTMF TP5088*

Tombol	Masukan Biner				Frekuensi Keluaran (Hz)				Kesalahan dari TP5088 (%)	
	D3	D2	D1	D0	Standar DTMF		Pengukuran		F - Low	F - High
					F - Low	F - High	F - Low	F - High		
1	0	0	0	0	941	1633	945	1641	-0,42	-0,49
2	0	0	0	1	697	1209	696	1206	+0,14	+0,25
3	0	0	1	0	697	1336	695	1331	+0,28	+0,37
4	0	1	0	0	770	1209	770	1206	0	+0,25
5	0	1	0	1	770	1336	770	1332	0	+0,29
6	0	1	1	0	770	1477	771	1486	-0,13	-0,60
7	1	0	0	0	852	1336	852	1332	0	+0,29
8	1	0	0	1	852	1477	852	1486	0	-0,60
9	1	0	1	0	941	1336	940	1331	+0,10	+0,37
*	1	1	0	0	941	1477	940	1486	+0,10	-0,60
0	1	1	0	1	697	1633	695	1640	+0,28	-0,43
#	1	1	1	0	770	1633	770	1640	0	-0,43

Persentase kesalahan pada frekuensi kesalahan TP5088 adalah sebagai berikut:

$$\frac{f.s \text{ tan darDTMF (Hz)} - f.pengukuranDTMF (Hz)}{f.s \text{ tan darDTMF (Hz)}} \times 100\%$$

Diketahui:

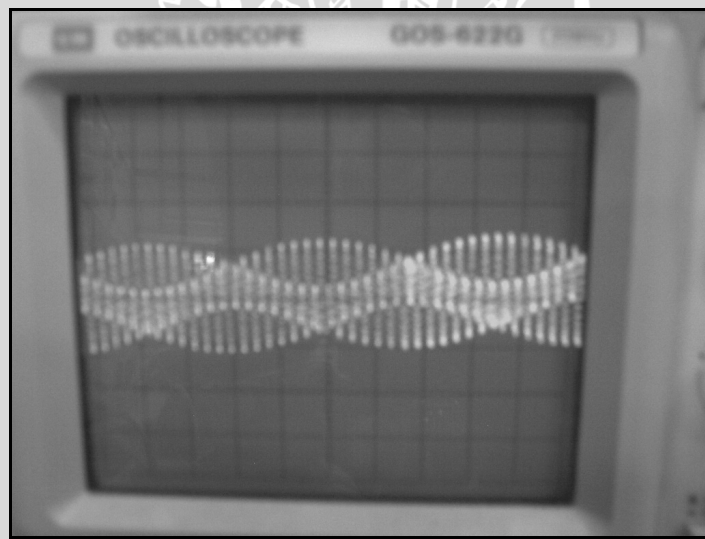
Frekuensi standar DTMF = 941 Hz

Frekuensi pengukuran DTMF = 945 Hz

$$\begin{aligned} \text{Persentase kesalahan} &= \frac{941\text{Hz} - 945\text{Hz}}{941\text{Hz}} \times 100\% \\ &= 0,4\% \end{aligned}$$

Persentase kesalahan terbesar pada frekuensi pada frekuensi keluaran DTMF terhadap standar frekuensi DTMF adalah -0,60% sehingga dapat disimpulkan bahwa pembangkit DTMF telah memenuhi syarat, karena tingkat kesalahan yang diperbolehkan adalah  $\pm 0.64\%$ .

Gambar sinyal DTMF dengan penekanan tombol '6' ditunjukkan dalam Gambar 5.3.

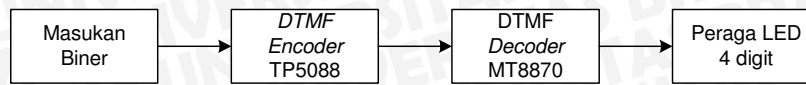


Gambar 5.3 Sinyal DTMF Dengan Penekanan Tombol '6'

### 5.3 Pengujian Rangkaian Decoder DTMF MT8870

Pengujian rangkaian DTMF untuk mengetahui apakah rangkaian penerima DTMF dapat menerima nada-nada dari rangkaian pengirim DTMF dan menerjemahkannya dalam data biner 4 bit. Peralatan yang digunakan antara lain: sebuah modul keypad matriks 3x4, sebuah modul encoder keypad 74C922, rangkaian decoder DTMF MT8870, rangkaian encoder DTMF TP5088, peraga LED 4 digit. Prosedur

pengujian *decoder* DTMF adalah menghubungkan rangkaian seperti dalam blok diagram yang ditunjukkan dalam Gambar 5.4.



**Gambar 5.4** Blok Diagram Pengujian DTMF

Hasil pengujian DTMF dapat dilihat dalam Tabel 5.3

**Tabel 5.3** Hasil Pengujian *Decoder* DTMF

Masukan Biner	Keluaran DTMF				
	Q4	Q3	Q2	Q1	StD
0 0 0 0	0	0	0	0	1
0 0 0 1	0	0	0	1	1
0 0 1 0	0	0	1	0	1
0 1 0 0	0	1	0	0	1
0 1 0 1	0	1	0	1	1
0 1 1 0	0	1	1	0	1
1 0 0 0	1	0	0	0	1
1 0 0 1	1	0	0	1	1
1 0 1 0	1	0	1	0	1
1 1 0 0	1	1	0	0	1
1 1 0 1	1	1	0	1	1
1 1 1 0	1	1	1	0	1

Hasil pengujian tersebut dapat dilihat dari nyala tidaknya peraga LED 4 digit yang akan bergantian apabila ada masukan dari pengindra. Sehingga untuk setiap pengaktifan pengindra akan diubah oleh rangkaian *encoder* DTMF menjadi sinyal analog berupa kombinasi frekuensi, kemudian oleh rangkaian *decoder* DTMF akan diubah menjadi data digital 4 bit. Apabila pengindra pertama yang aktif maka LED akan menyala pada Q1. Jika pengindra kedua yang aktif maka LED akan menyala pada Q2.

Contoh:

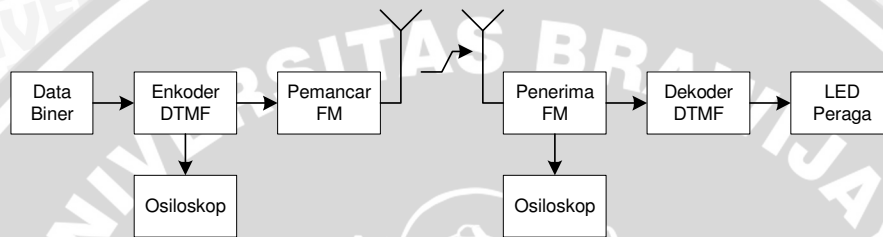
Jika kode masukan biner 0 0 0 1 maka keluaran *decoder* DTMF adalah 0 0 0 1 dengan pin StD dalam kondisi '1' (*high*).

Dari pengujian tersebut didapatkan hasil bahwa rangkaian *encoder* dan *decoder* DTMF dapat menunjukkan data biner 4 bit sesuai dengan pengindra yang aktif, sehingga dapat disimpulkan bahwa rangkaian dapat bekerja sesuai dengan perencanaan. Dimana kondisi *encoder* dan *decoder* DTMF dalam keadaan tanpa *noise*.

#### 5.4 Pengujian Pemancar dan Penerima FM

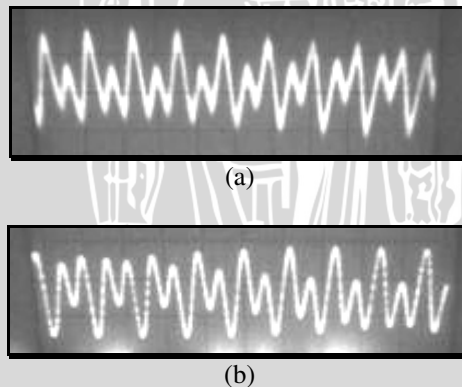
Pengujian pemancar dan penerima ini untuk mengetahui kemampuan pemancar dan penerima FM yang digunakan dapat mengirimkan dan menerima frekuensi sinyal radio. Peralatan yang digunakan adalah: sebuah modul *keypad* matriks 3x4, sebuah

modul *encoder keypad* 74C922, rangkaian *encoder* DTMF TP5088, rangkaian pemancar FM, rangkaian penerima FM, rangkaian *decoder* DTMF MT8870, LED peraga dan osiloskop. Prosedur pengujian rangkaian ini adalah mula-mula dengan mengatur frekuensi pemancar dan penerima FM sehingga beroperasi pada frekuensi kerja yang sama, kemudian menghubungkan seluruh rangkaian seperti ditunjukkan dalam blok diagram rangkaian Gambar 5.5. Pin *tone output* pada *encoder* DTMF dihubungkan ke osiloskop kanal 1 dan pin keluaran dari penerima FM dihubungkan ke osiloskop kanal 2. Data biner diganti sesuai dengan kode-kode DTMF, mengamati sinyal yang ditunjukkan oleh osiloskop.



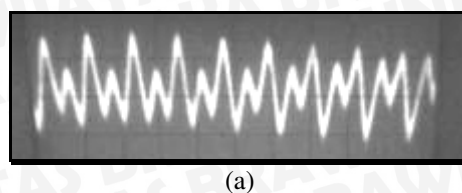
**Gambar 5.5** Blok Diagram Pengujian Pemancar dan Penerima FM

Pengujian pemancar dan penerima FM ini diulang selama 3x dengan perubahan jarak antara pemancar dan penerima FM. Hal ini untuk mengetahui perbedaan sinyal yang dapat diterima oleh penerima FM. Jarak yang diberikan pada pengujian pertama adalah 5 meter dan hasil pengujiannya ditunjukkan dalam Gambar 5.6 berikut.



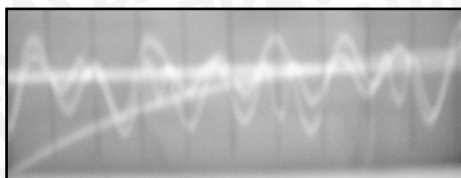
**Gambar 5.6** Hasil Pengujian Rangkaian Pada Jarak 5 meter (a). Pemancar FM (b). Penerima FM

Pengujian yang kedua dengan memberikan jarak sejauh 10 meter. Hasil pengujiannya ditunjukkan dalam Gambar 5.7.



(a)





(b)

**Gambar 5.7** Hasil Pengujian Rangkaian Pada Jarak 10 meter (a). Pemancar FM (b). Penerima FM

Pengujian yang ketiga pada jarak jangkauan 15 meter. Hasil pengujiannya ditunjukkan dalam Gambar 5.8.



(a)



(b)

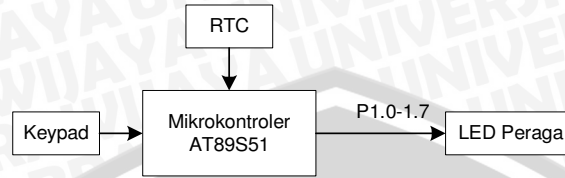
**Gambar 5.8** Hasil Pengujian Rangkaian Pada Jarak 15 meter (a). Pemancar FM (b). Penerima FM

Hasil pengujian menunjukkan bahwa pemancar dan penerima FM telah bekerja pada frekuensi yang sama untuk menerima sinyal dan saling berkomunikasi. Penerima FM dapat menerima sinyal masukan dengan sangat baik pada jarak jangkauan 5 meter karena sinyal tersebut dapat diterima dengan jelas dan bersih hampir tidak terdapat *noise*. Pada jarak jangkauan 10 meter sinyal DTMF tidak dapat diterima dengan baik oleh penerima FM, karena sudah dipengaruhi oleh *noise* sehingga sinyal yang diterima tidak sempurna. Sedangkan pada jarak jangkauan 15 meter, sinyal tersebut benar-benar tidak dapat diterima penerima FM. Sehingga, dari hasil pengujian dapat disimpulkan bahwa pemancar dan penerima FM dapat berfungsi dengan baik sesuai perencanaan dengan jarak jangkauan kurang lebih 10 meter.

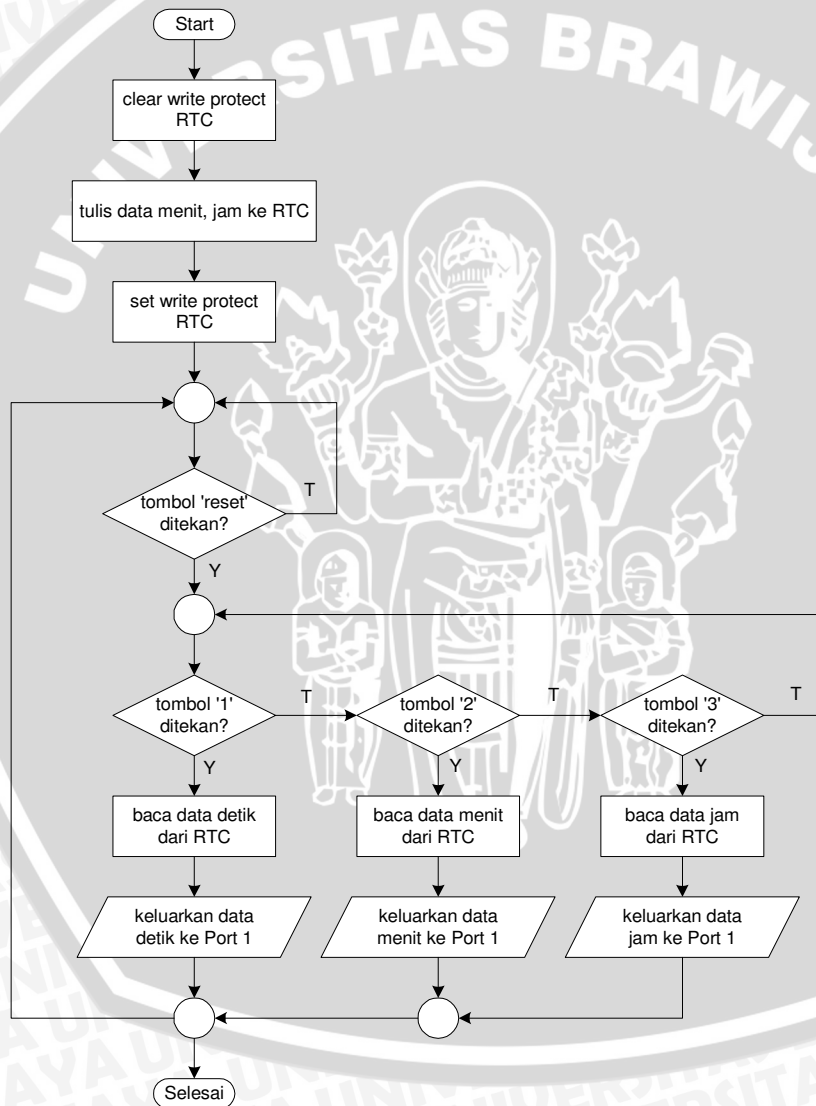
### 5.5 Pengujian Rangkaian *Real Time Clock* (RTC)

Pengujian rangkaian *Real Time Clock* ini adalah untuk mengetahui apakah RTC DS1302 dapat bekerja dengan baik dan mampu memberikan informasi yang diinginkan berupa informasi detik, menit, jam, tanggal, bulan maupun tahun. Peralatan yang digunakan antara lain: PC (*Personal Computer*), modul rangkaian emulator

mikrokontroler AT89S51, keypad, 8 buah LED, rangkaian DS1302, dan multimeter. Prosedur pengujian yang dilakukan adalah mula-mula menghubungkan rangkaian seperti dalam blok diagram rangkaian yang ditunjukkan dalam Gambar 5.9. Kemudian mengamati perubahan tiap waktu melalui nyala LED peraga.



**Gambar 5.9** Diagram Blok Rangkaian Pengujian RTC DS1302



**Gambar 5.10** Diagram Alir Pengujian RTC DS1302

Hasil pengujian rangkaian RTC DS1302 untuk data menit dan jam ditunjukkan dalam Tabel 5.4 dan Tabel 5.5.

**Tabel 5.4** Hasil Pengujian RTC DS1302 Untuk Mengeluarkan Data Menit Pada Port 1

Langkah	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1	Data (Menit)
0	ON	ON	ON	OFF	ON	ON	ON	ON	10
1	ON	ON	ON	OFF	ON	OFF	ON	OFF	15
2	ON	ON	OFF	ON	ON	ON	ON	ON	20

**Tabel 5.5** Hasil Pengujian RTC DS1302 Untuk Mengeluarkan Data Jam Pada Port 1

Langkah	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1	Data (Jam)
0	ON	ON	ON	ON	OFF	ON	ON	OFF	09

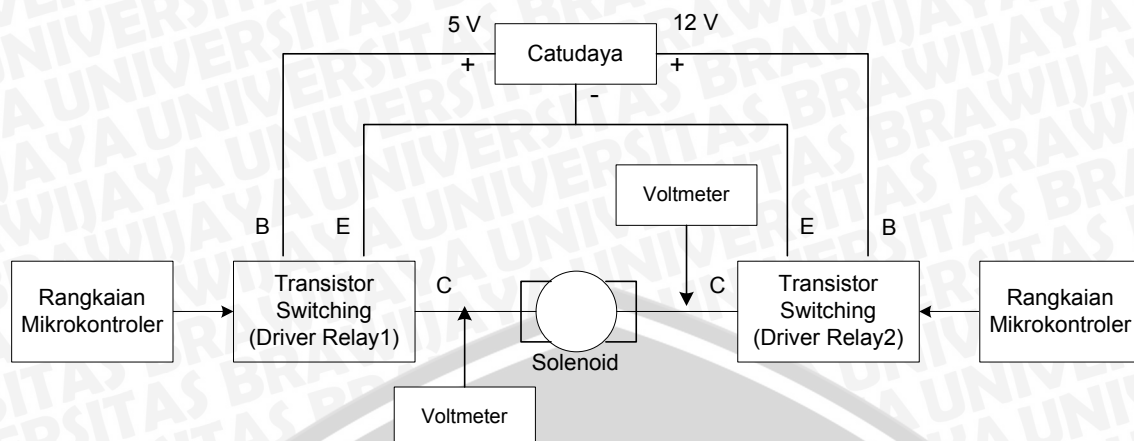
Keterangan:

- Tegangan keluaran port saat LED ON = 0,5 V
- Tegangan keluaran port saat LED OFF = 4,8 V

Untuk mendapatkan hasil yang akurat pengujian ini dilakukan dengan mengaktifkan RTC secara terus-menerus selama 5 jam dan pengujian masing-masing langkah berulang hingga 10 kali. Hasil pengujian menunjukkan bahwa perbedaan penunjukan menit dan jam antara RTC dan arloji tidak begitu signifikan, rata-rata sekitar 5-10 detik. Dengan demikian, selisih waktu tersebut dapat diabaikan dan dari hasil pengujian diperoleh kesimpulan bahwa rangkaian RTC DS1302 dapat berfungsi dengan baik. Data menit dan jam dapat dikeluarkan pada port 1 sesuai dengan waktu awal yang telah ditentukan terlebih dahulu pada program pengujian. Perubahan setiap menit dan jamnya sesuai dengan perhitungan waktu pada arloji.

## 5.6 Pengujian *Driver Relay* dan *Solenoid*

Tujuan dari pengujian *driver solenoid* ini untuk mengetahui apakah *driver relay* dapat berfungsi dengan baik dan *solenoid* dapat bergerak sesuai dengan perencanaan. Peralatan yang digunakan antara lain: rangkaian *driver relay*, *solenoid* dan *multimeter*. Prosedur pengujian yang dilakukan adalah mula-mula menyusun rangkaian seperti dalam blok diagram yang ditunjukkan dalam Gambar 5.11, yaitu dengan menghubungkan keluaran rangkaian mikrokontroler ke rangkaian *driver relay*. Tegangan keluaran mikrokontroler meliputi tegangan pada level *low* dan *high*. Tegangan keluaran dengan level *low* diwakilkan dengan tegangan 0 V dan tegangan keluaran dengan level *high* diwakilkan dengan tegangan 5 V. Untuk dapat mengetahui kondisi kontak *relay*, maka digunakan *multimeter* yang dipasang pada kontak *relay*. Kemudian setelah catu daya diaktifkan, mengamati perubahan gerak *solenoid* dan juga mengamati perubahan tegangan pada *multimeter*.



Gambar 5.11 Rangkaian Pengujian Driver Relay dan Solenoid

Tabel 5.6 menunjukkan bahwa pada saat output 1 mikrokontroler pada level *high* dan output 2 mikrokontroler pada level *low* menyebabkan kontak relay 1 tertutup dan kontak relay 2 terbuka maka menyebabkan *plunger solenoid* aktif. Sedangkan pada saat output 1 mikrokontroler pada level *low* maka output 2 mikrokontroler juga harus pada level logika *low* agar *plunger solenoid* non aktif atau kembali pada keadaan normal. Maka dari hasil pengujian dapat disimpulkan bahwa rangkaian *driver relay* dan *solenoid* dapat bekerja sesuai dengan perencanaan.

Tabel 5.6 Hasil Pengujian Rangkaian Driver Relay dan Solenoid

Output 1 Mikrokontroler	Output 2 Mikrokontroler	Tegangan Relay 1 (Volt)	Tegangan Relay 2 (Volt)	Kontak Relay 1	Kontak Relay 2	Gerak Solenoid
<i>High</i>	<i>Low</i>	12	0	Tertutup	Terbuka	Aktif
<i>Low</i>	<i>Low</i>	0	0	Terbuka	Terbuka	Non Aktif

### 5.7. Pengujian Sistem Secara Keseluruhan

Tujuan dari pengujian sistem secara keseluruhan ini untuk mengetahui apakah sistem dapat berfungsi dengan baik sesuai dengan perencanaan. Pengujian ini merupakan penggabungan dari seluruh sistem, sehingga dapat diketahui apakah seluruh sistem dapat bekerja dengan baik sesuai dengan perancangan.

Pengujian yang pertama adalah pengujian sistem pengendali kunci pintu secara manual. Pintu 10 sebagai *prototype* pengujian. Perlu diketahui bahwa, LED warna merah merupakan LED indikator keadaan pintu dan LED warna kuning merupakan LED indikator pengunci pintu. Keadaan awal pengunci pintu 10 adalah terkunci dan keadaan pintu dalam keadaan tertutup, sehingga LED indikator pintu dan pengunci pintu 10 dalam keadaan 'nyala'. Pengujian ini dilakukan untuk membuka pengunci

repository.ub.ac.id

pintu 10, yaitu dengan memasukkan kode pintu '0\*' (pintu 10 "Buka") melalui *keypad* pada pengendali. Kemudian proses pengiriman data dilakukan. Setelah kode pintu terkirim maka data '0\*' akan diterima oleh penerima kemudian data '0\*' akan didekodekan kembali oleh *decoder* DTMF dan dikirimkan ke mikrokontroler sebagai pusat pengendali sistem. Kemudian data '0\*' diterjemahkan oleh mikrokontroler. Jika data tersebut benar, maka digunakan untuk menggerakkan *solenoid* sebagai elektro mekanik pengunci pintu. Gerak *solenoid* tersebut akan mempengaruhi sensor pengunci pintu dan selanjutnya data sensor tersebut akan diolah oleh mikrokontroler dan dikirimkan ke bagian pengendali untuk menyalakan atau mematikan LED indikator penguncian pintu. Jika proses membuka pengunci pintu ini berhasil maka LED indikator penguncian pintu akan padam, jika proses membuka pengunci pintu ini tidak berhasil maka buzzer akan memberikan alarm. Selama pintu masih dalam keadaan tertutup maka LED indikator keadaan pintu akan tetap menyala. Setelah pintu dibuka, maka LED indikator tersebut akan padam.

Dari hasil pengujian pertama dapat dilihat bahwa kode perintah dapat diterima dengan baik oleh penerima dan laporan keberhasilan proses membuka pengunci pintu dapat ditunjukkan melalui indikator LED pengunci pintu. Demikian juga untuk LED indikator keadaan pintu dapat mengindikasikan keadaan pintu dengan tepat. Sehingga perangkat keras dan perangkat lunak yang telah diuji dalam pengujian pertama telah beroperasi dengan baik. Gambar 5.12 menunjukkan kondisi LED indikator sebelum proses membuka pengunci pintu dilakukan dan Gambar 5.13 menunjukkan kondisi LED indikator setelah proses membuka pengunci pintu.



**Gambar 5.12** Kondisi LED Indikator Sebelum Proses Membuka Pengunci Pintu



**Gambar 5.13** Kondisi LED Indikator Sesudah Proses Membuka Pengunci Pintu

Pengujian yang kedua adalah pengujian sistem pengontrolan pintu. Pengujian ini dilakukan dengan mengatur sensor pintu dimana pintu dalam keadaan terbuka. Setelah 10 menit (batas waktu maksimum pintu terbuka) maka mikrokontroler mengirimkan kode data kepada pengendali untuk mengaktifkan *buzzer* dan menyalakan LED indikator keadaan pintu untuk mengingatkan *user*.

Dari hasil pengujian kedua ini dapat dilihat bahwa data sensor dari penerima dapat diterima dengan baik oleh pengendali, sehingga dapat memberikan informasi kepada *user* dengan baik.

Pengujian yang ketiga adalah pengujian sistem penguncian pintu secara otomatis. Pengujian yang dilakukan adalah mengamati sistem pengendali dan penerima tepat pada waktu yang telah ditentukan. Pada pengujian ini pengunci pintu akan aktif secara otomatis pada pukul 21.00 (jam 9 malam). Jika pada pukul 21.00 ini ada pengunci pintu yang belum terkunci, maka secara otomatis mikrokontroler akan mengecek sensor pengunci secara bergantian. Jika data sensor menunjukkan pengunci pintu masih terbuka maka mikrokontroler akan mengirimkan perintah untuk menggerakkan *solenoid*. Seperti pada pengujian sistem pengunci pintu secara manual, gerak solenoid tersebut akan mengakibatkan sensor pengunci memberikan data ke mikrokontroler yang selanjutnya akan dikirimkan ke bagian pengendali sebagai laporan keberhasilan penguncian pintu. Demikian juga untuk pengunci pintu lain yang masih terbuka, sehingga semua pintu dapat terkunci. *User* cukup mengecek proses penguncian pintu melalui LED indikator.

Dari hasil pengujian sistem penguncian pintu secara otomatis ini dapat dilihat juga bahwa penguncian pintu dapat berhasil sesuai dengan waktu yang telah ditentukan,

sehingga perangkat keras dan perangkat lunak yang telah diuji dalam ketiga pengujian ini telah beroperasi dengan baik.

Perlu diketahui bahwa pengujian ini dilakukan dalam 5 hari dengan mengaktifkan sistem secara terus-menerus. Hal ini mempunyai tujuan untuk dapat mengetahui keandalan dari sistem yang dirancang. Adapun hasil operasi yang dilakukan dalam 5 hari ditunjukkan dalam Tabel 5.7.

**Tabel 5.7** Hasil Pengujian Sistem Secara Keseluruhan

Hari ke-	Tanggal	Waktu	Kegiatan	Kondisi Pintu	Alarm	Kondisi Pengunci
1	23 April 2007	08.00		Tertutup	Tidak Aktif	Terbuka
		08.15	Menutup Pengunci Pintu	Tertutup	Tidak Aktif	Tertutup
		09.00	Membuka Pengunci Pintu	Tertutup	Tidak Aktif	Terbuka
		10.00	Alarm Keadaan Pintu	Terbuka	Aktif	Terbuka
		21.00	Penguncian Keseluruhan Otomatis	Tertutup	Tidak Aktif	Tertutup
2	24 April 2007	08.00	Membuka Pengunci Pintu	Tertutup	Tidak Aktif	Terbuka
		09.00	Alarm Keadaan Pintu	Terbuka	Aktif	Terbuka
		10.00	Menutup Pengunci Pintu	Tertutup	Tidak Aktif	Tertutup
		21.00	Penguncian Keseluruhan Otomatis	Tertutup	Tidak Aktif	Tertutup
3	25 April 2007	13.00	Membuka Pengunci Pintu	Tertutup	Tidak Aktif	Terbuka
		14.30	Alarm Keadaan Pintu	Terbuka	Aktif	Terbuka
		14.45	Menutup Pengunci Pintu	Tertutup	Tidak Aktif	Tertutup
		21.00	Penguncian Keseluruhan Otomatis	Tertutup	Tidak Aktif	Tertutup
4	26 April 2007	08.00	Membuka Pengunci Pintu	Tertutup	Tidak Aktif	Terbuka
		09.00	Alarm Keadaan Pintu	Terbuka	Aktif	Terbuka
		10.00	Menutup Pengunci Pintu	Tertutup	Tidak Aktif	Tertutup
		21.00	Penguncian Keseluruhan Otomatis	Tertutup	Tidak Aktif	Tertutup
5	27 April 2007	13.00	Membuka Pengunci Pintu	Tertutup	Tidak Aktif	Terbuka
		14.00	Alarm Keadaan Pintu	Terbuka	Aktif	Terbuka
		15.00	Menutup Pengunci Pintu	Tertutup	Tidak Aktif	Tertutup
		21.00	Penguncian Keseluruhan Otomatis	Tertutup	Tidak Aktif	Tertutup

Ada empat jenis kegiatan pengujian yang dilakukan pada sistem secara langsung selama 5 hari. Hasil setiap pengujian menunjukkan bahwa sensor pengunci, sensor pintu, buzzer dan elektronik pengunci pintu dapat bekerja sesuai dengan perintah yang diberikan. Sehingga seluruh sistem dapat bekerja dengan baik sesuai dengan perencanaan.

## BAB VI PENUTUP

### 6.1 Kesimpulan

Dari perancangan dan pengujian Sistem Pengendali Kunci Pintu Jarak jauh Dengan Transmisi Gelombang Radio ini diperoleh kesimpulan sebagai berikut:

1. Pengujian perangkat keras menunjukkan bahwa perangkat keras tersebut dapat berjalan sesuai dengan perancangan.
2. Data masukan yang dikirimkan dari bagian pengendali dapat diterima dengan baik pada bagian penerima dengan jarak jangkauan maksimum 10 meter.
3. Data DTMF dapat didekodekan dengan tepat.
4. Elektronik pengunci pintu dapat berfungsi dengan baik.
5. Data sensor mengaktifkan LED indikator sesuai dengan keadaan sebenarnya.

### 6.2 Saran

Saran-saran yang sangat diperlukan untuk pengembangan alat ini antara lain:

1. Perbaiki sistem pemancar dan penerima FM sehingga dapat mengirimkan data yang tepat dan mempunyai jarak jangkauan yang lebih jauh.
2. Perbaiki catu daya sehingga suatu saat jika listrik PLN mati maka sistem pengendali kunci pintu jarak jauh ini tetap berfungsi.



## DAFTAR PUSTAKA

- Atmel. 2003. *AT89S51-8 bit Microcontroller with 4K Bytes In System Programmable Flash Datasheet*. Atmel Corporation.
- Budiharto, Widodo. 2005. *Perancangan Sistem dan Aplikasi Mikrokontroler*. Jakarta: PT Elex Media Komputindo.
- Chattopadhyay, Rakshit, Saha dan Purkait. 1989. *Dasar Elektronika*. Jakarta: Penerbit Universitas Indonesia.
- Dallas Semiconductor. 2001. *DS1302 Trickle Charge Timekeeping Chip*. [www.datasheetcatalog.com](http://www.datasheetcatalog.com). Diakses tanggal 11 Januari 2006. Pukul 09.00
- Eko Putra, Agfianto. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta: Penerbit Gaya Media.
- Hioki, Warren. 1996. *Telecommunications*. New York: Prentice Hall International.
- Hissen, Helmut. 2007. *DTMF Tones*. [Zeebar Technology Services, Inc. www.dialabc.com/sound/dtmf](http://Zeebar Technology Services, Inc. www.dialabc.com/sound/dtmf). Diakses tanggal 5 Mei 2006. Pukul 11.00
- Krauss dan Bostian. 1990. *Teknik Radio Benda Padat*. Jakarta: Penerbit Universitas Indonesia.
- Malvino, Albert Paul. 1994. *Prinsip-Prinsip dan Penerapan Digital*. Jakarta: Penerbit Erlangga.
- Malvino, Albert Paul. 1987. *Prinsip-Prinsip Elektronika*. Jakarta: Penerbit Erlangga.
- Mitel. 1995. *Integrated DTMF Receiver*. [www.datasheetcatalog.com](http://www.datasheetcatalog.com). Diakses tanggal 11 Januari 2006. Pukul 09.00
- National Semiconductor. 1991. *TP5088 DTMF Generator for Binary Data*. Natinal Semiconductor Corporation. [www.datasheetcatalog.com](http://www.datasheetcatalog.com). Diakses tanggal 11 Januari 2006. Pukul 09.00
- Roody, Dennis dan Coolen, John. 1984. *Komunikasi Elektronika*. Jakarta: Penerbit Erlangga.
- Simanjuntak, Tiur. LH. 1993. *Dasar-Dasar Telekomunikasi*. Bandung: Penerbit Alumni.
- Suhana dan Shoji. 2002. *Buku Pegangan Teknik Telekomunikasi*. Jakarta: PT. Pradnya Paramita.
- Tocci, Ronald. 1988. *Digital System Principles and Applications*. New Jersey: Prentice Hall.

repository.ub.ac.id

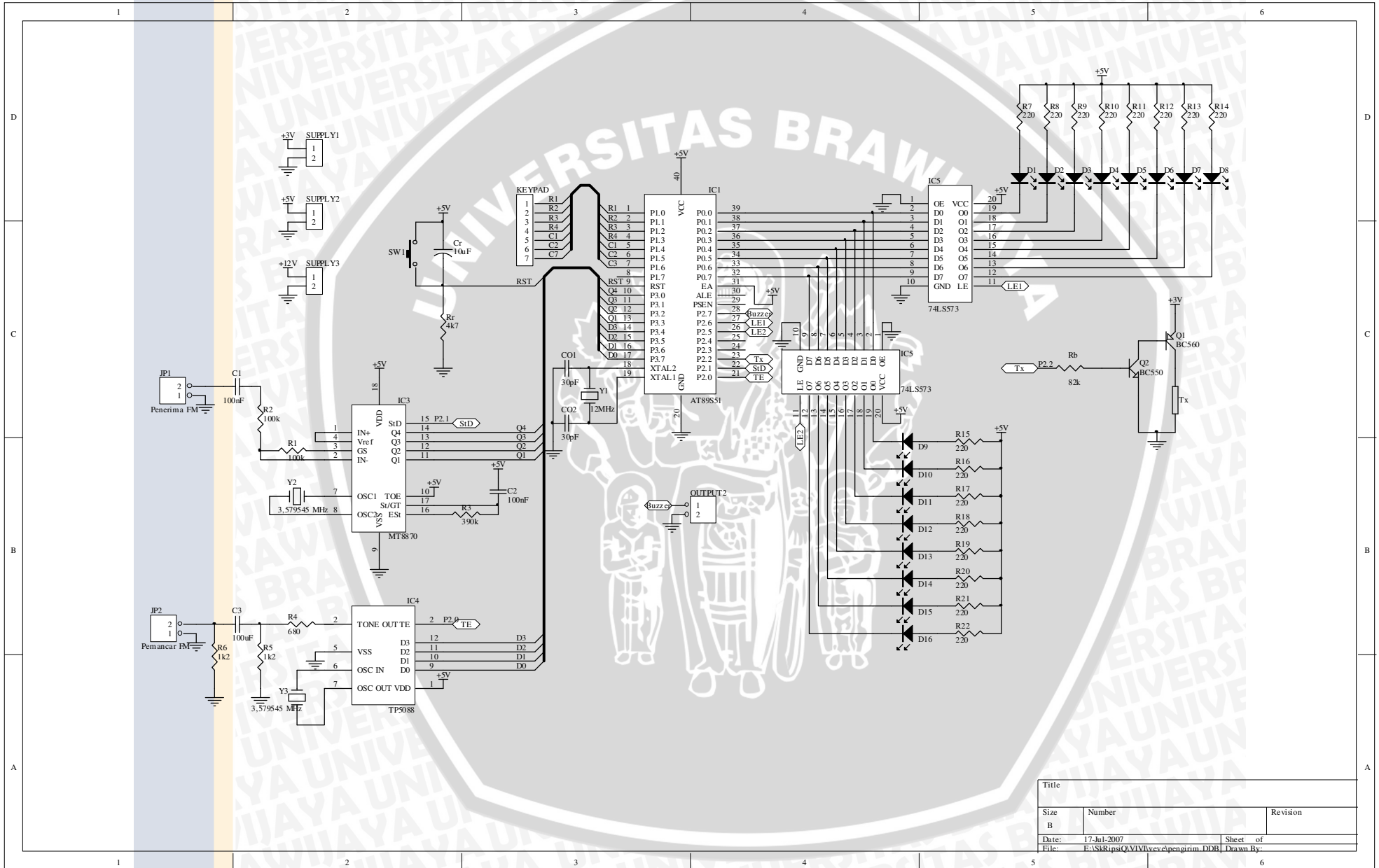
Wilde, Robert. 2007. *Drawings of Hydraulic Systems of the Silver Shadow models*.  
Eindhoven, Holland. <http://www.rrbew.co.uk>. Diakses tanggal 7 Mei 2006.  
Pukul 08.30



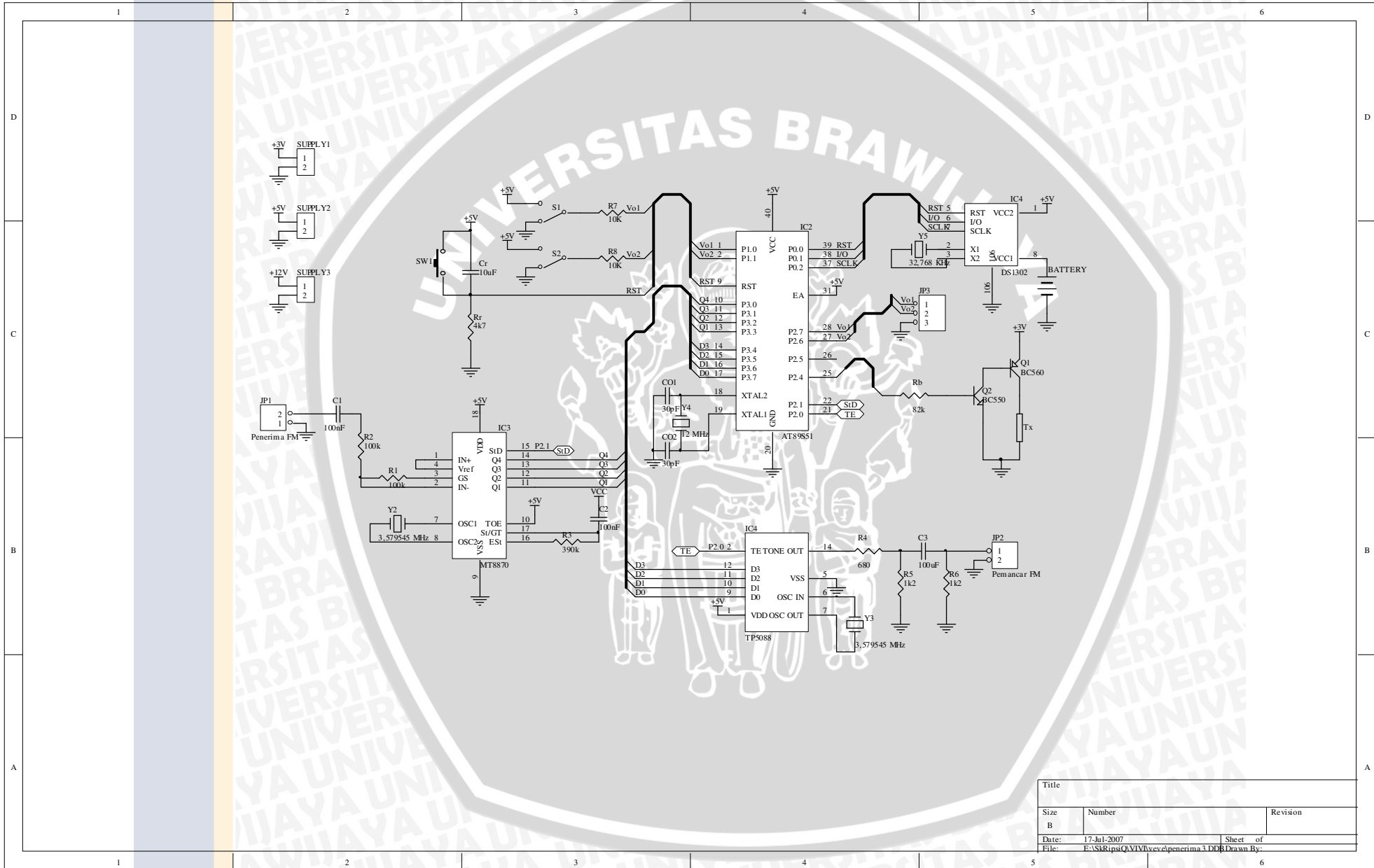
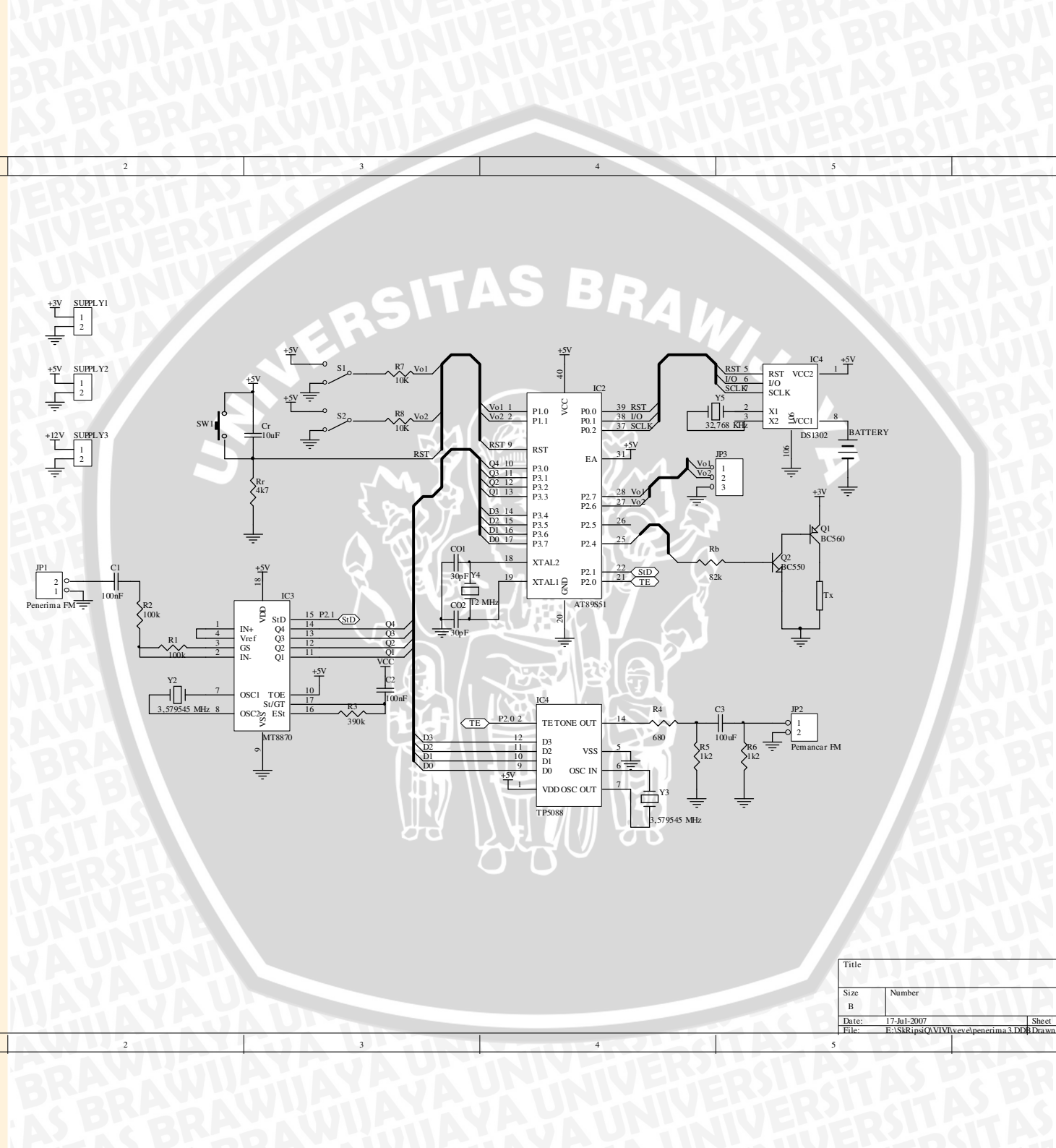
## LAMPIRAN I

### SKEMA RANGKAIAN

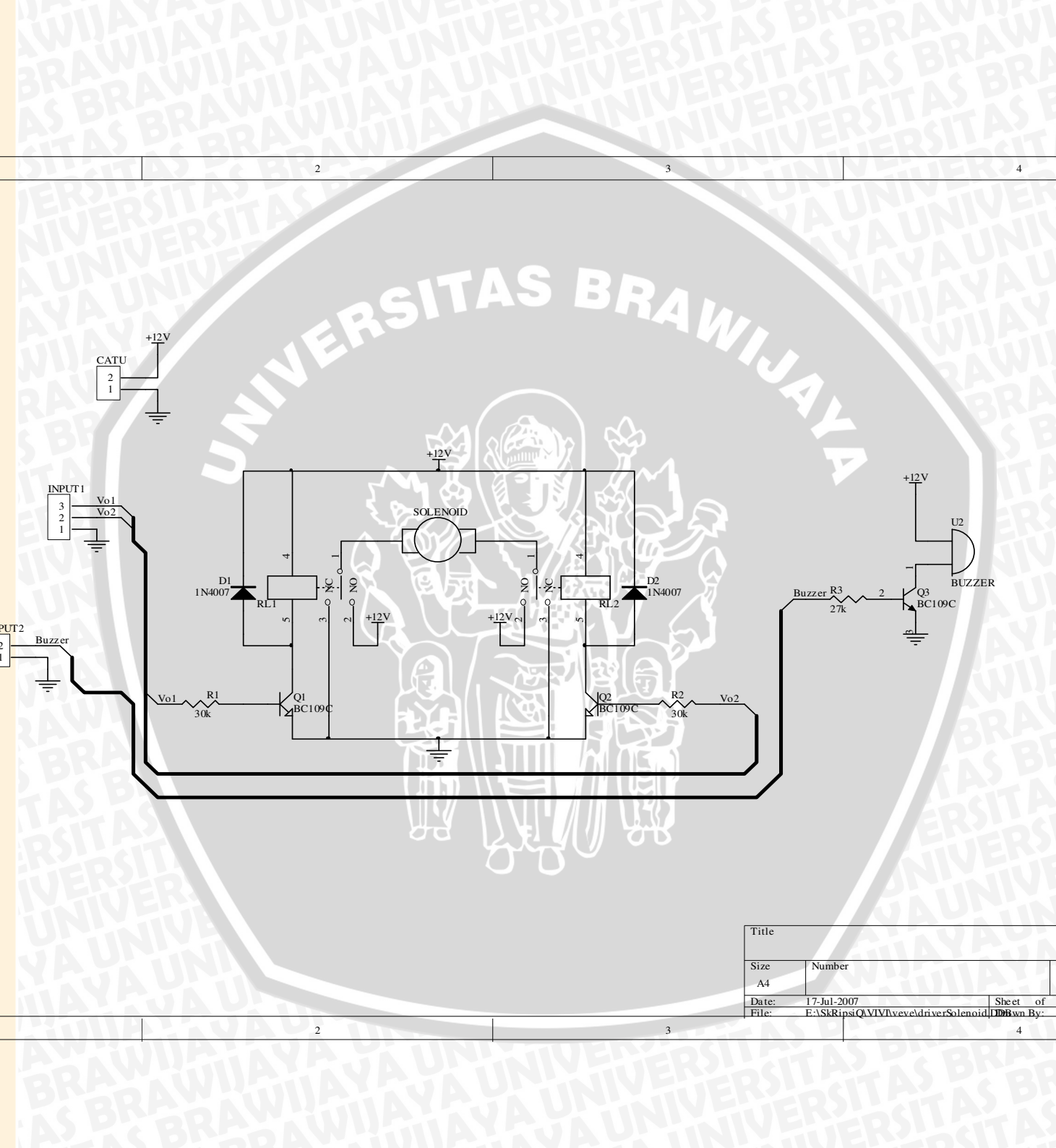




Title		
Size B	Number	Revision
Date: 17-Jul-2007	Sheet of	
File: F:\Skr\ps\QVI\VI\ve\pengirim_DDB	Drawn By:	



Title		
Size	Number	Revision
B		
Date:	17-Jul-2007	Sheet of
File:	E:\Skripsi\QVIVI\veva\penerima 3 DBR	Drawn By:

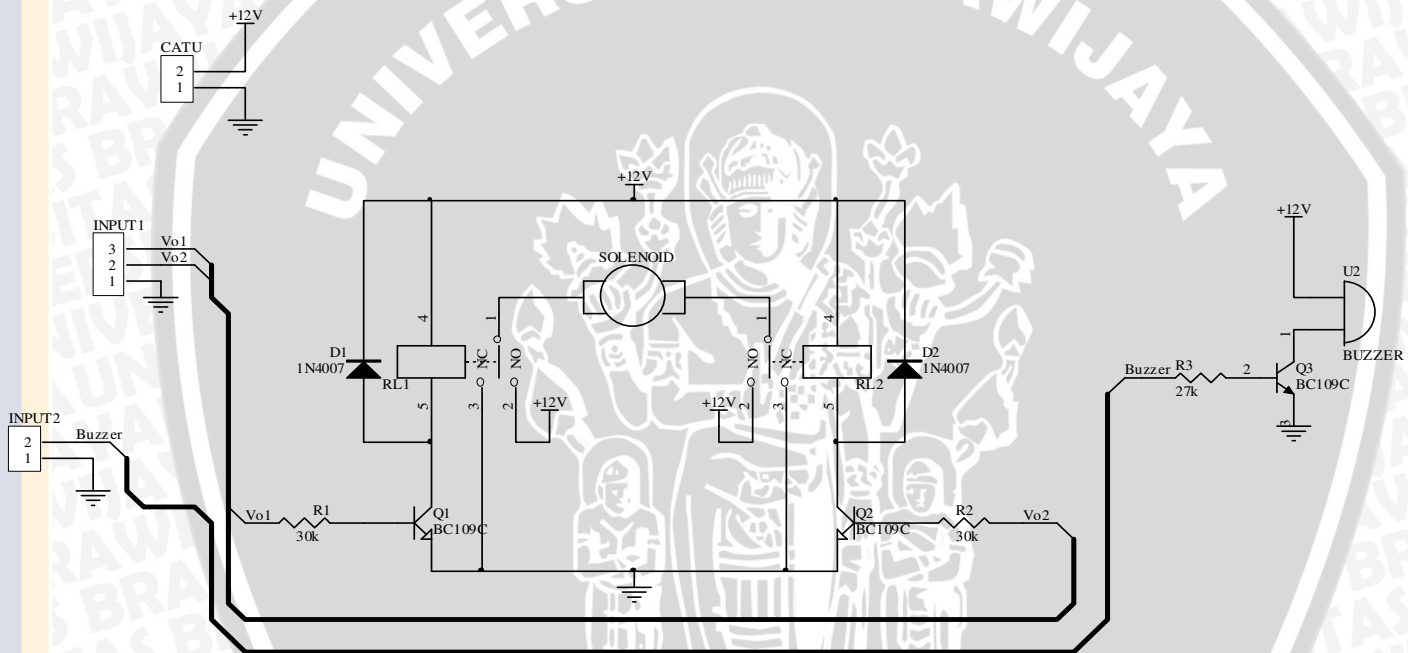


1 2 3 4

D  
C  
B  
A

D  
C  
B  
A

1 2 3 4



Title		
Size	Number	Revision
A4		
Date:	17-Jul-2007	Sheet of
File:	E:\Skripsi\QAMIV\veveldriverSolenoid\Drawn By:	

**LAMPIRAN II**

***LISTING PROGRAM***





## LISTING PROGRAM UNIT PENGENDALI

```

;*****
;Program Utama
;*****

$mod51
Led1      bit    p0.0
Led2      bit    p0.1
Led3      bit    p0.2
Led4      bit    p0.3
Led5      bit    p0.4
Led6      bit    p0.5
Led7      bit    p0.6
Led8      bit    p0.7

Tx         bit    p2.2
Buzzer    bit    p2.6
LE1       bit    p2.4
LE2       bit    p2.5
Std       bit    p2.1
TE        bit    p2.0

baris1    bit    p1.0
baris2    bit    p1.1
baris3    bit    p1.2
baris4    bit    p1.3
kolom1    bit    p1.4
kolom2    bit    p1.5
kolom3    bit    p1.6

D0_TP     bit    p3.7
D1_TP     bit    p3.6
D2_TP     bit    p3.5
D3_TP     bit    p3.4
Q0_MT     bit    p3.3
Q1_MT     bit    p3.2
Q2_MT     bit    p3.1
Q3_MT     bit    p3.0

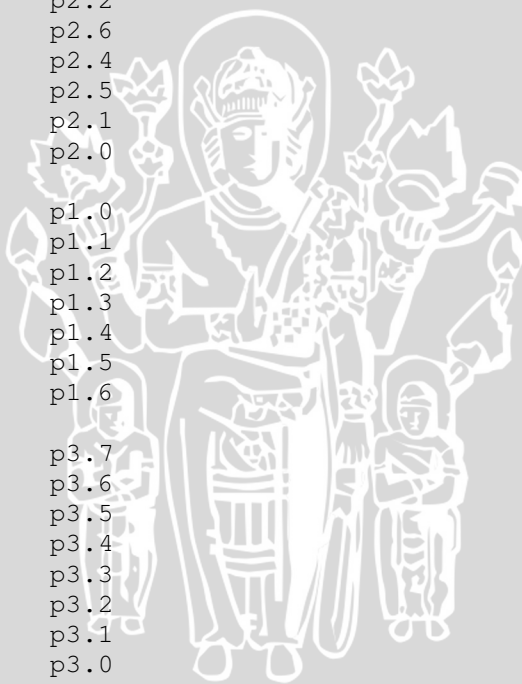
keyport   equ    P1

keybounce equ    200

KEYDATA   DATA  61H
KEYBOUNC  DATA  62H

Kondisi1  bit    20h.0
Kondisi2  bit    20h.1
Kondisi3  bit    20h.2
Kondisi4  bit    20h.3
Kondisi5  bit    20h.4
Kondisi6  bit    20h.5
Kondisi7  bit    20h.6
Kondisi8  bit    20h.7
Kondisi9  bit    21h.0
    
```

UNIVERSITAS BRAWIJAYA





Kondisi10	bit	21h.1
T500mS_oke	bit	21h.2
Dummy	data	22h
Time50	data	23h
bounceDTMF	data	26h
Tombol	data	27h
Menit	data	2ch
JumTekan	data	2dh
BufferSimpan	data	30h
DataTekan	data	3ah
Data1	data	50h
Data2	data	51h
Data3	data	52h
Data4	data	53h

```

Org      0h
ljmp     mulai

org      000bh
acall    time0
reti

mulai:   org      50h
         lcall    MatikanLed
         lcall    Reset_time0
         clr     TE
         clr     Tx
         mov     20h,#0
         mov     21h,#0
         mov     sp,#07h
         setb   Buzzer
         lcall   Awal
         clr     Tx

mulai1:  Mov      JumTekan,#0

MulaiA1: mov     keybounc,#keybounce
ulang:  call    keypad4x4
         mov     A,keydata
         cjne   A,#0FFh,tekan
         mov     keybounc,#keybounce
         ljmp   BacaDTMF

tekan:  cjne   a,#01,tekan2
         Mov    Data1,#0
         Mov    Data4,#2
         ljmp   Proses

tekan2: cjne   a,#02,tekan2c
         Mov    Data1,#1
         Mov    Data4,#3
         ljmp   Proses
    
```

```

tekan2c:  cjne    a,#03,tekan2d
           Mov     Data1,#2
           Mov     Data4,#4
           ljmp    Proses

tekan2d:  cjne    a,#04,tekan2e
           Mov     Data1,#3
           Mov     Data4,#5
           ljmp    Proses

tekan2e:  cjne    a,#05,tekan2f
           Mov     Data1,#4
           Mov     Data4,#6
           ljmp    Proses

tekan2f:  cjne    a,#06,tekan2g
           Mov     Data1,#5
           Mov     Data4,#7
           ljmp    Proses

tekan2g:  cjne    a,#07,tekan2h
           Mov     Data1,#6
           Mov     Data4,#8
           ljmp    Proses

tekan2h:  cjne    a,#08,tekan2i
           Mov     Data1,#7
           Mov     Data4,#9
           ljmp    Proses

tekan2i:  cjne    a,#09,tekan2j
           Mov     Data1,#8
           Mov     Data4,#0
           ljmp    Proses

tekan2j:  cjne    a,#0,tekan2k
           Mov     Data1,#9
           Mov     Data4,#1
           ljmp    Proses

tekan2k:  cjne    a,#0eh,tekan2l
           ljmp    Proses1

tekan2l:  cjne    a,#0fh,mulai1ax
           ljmp    Proses1

```

```

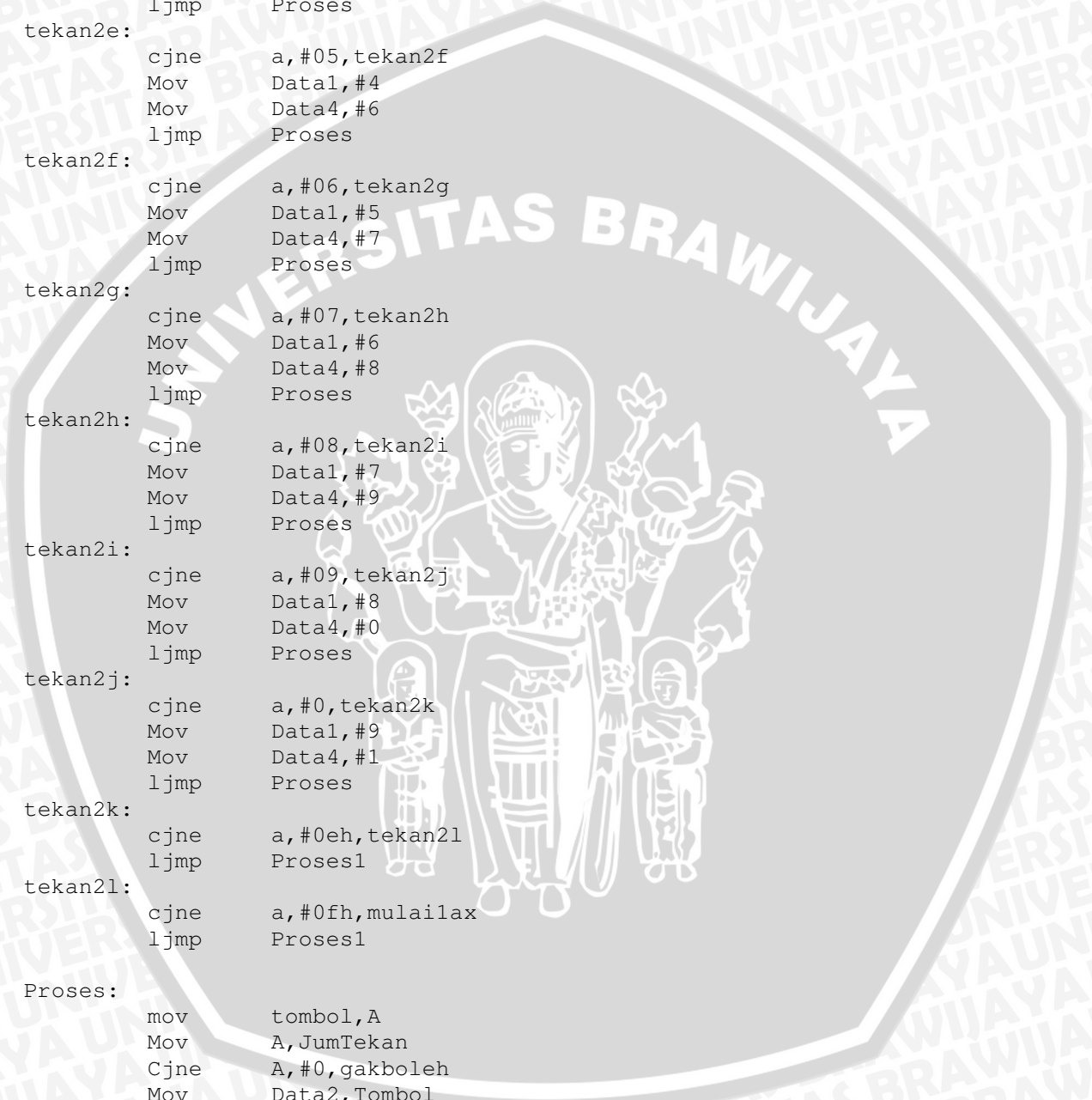
Proses:   mov     tombol,A
           Mov     A,JumTekan
           Cjne   A,#0,gakboleh
           Mov     Data2,Tombol
           Inc    JumTekan
           ljmp   MulaiA1

```

```

Proses1:  mov     tombol,A
           Mov     A,JumTekan
           Cjne   A,#1,gakboleh
           Mov     Data3,Tombol
           sjmp   ProsesTombol

```



```

gakboleh:
    lcall    Tiit_panjang
    ljmp     mulai1
mulailax:
    ljmp     mulaial1
ProsesTombol:
    Setb    Tx
    lcall   delay2
    mov     r1,#4
    mov     r0,#Data1

LoopKirim:
    mov     a,@r0
    swap   a
    anl    a,#0f0h
    lcall   convertTP
    mov     p3,a
    setb   TE
    lcall   tiit_pendek
    clr    TE
    lcall   delay1
    lcall   delay1
    lcall   delay1
    inc    R0
    djnz   R1,loopKirim
    lcall   delay1
    clr    Tx
    ljmp   mulai1

TidakAdaDTMF1:
    ljmp   TidakAdaDTMF

BacaDTMF:
    mov    BounceDTMF,#100
bd1:
    jnb   Std,TidakAdaDTMF1
    mov   r0,#2
    djnz  r0,$
    djnz  bounceDTMF,BD1
;ADA DTMF
    mov   a,p3
    anl  a,#0fh
    mov  data1,a
bd2:
;
    clr   buzzer
    jb   Std,$
    mov  r0,#255
    djnz r0,$
    jb   std,bd2
    setb buzzer

;DTMF pertama sudah terbaca maka timer diaktifkan

    setb  EA
    setb  ET0
    acall reset_time0
    acall timer50ms

;
;
;
    mov   p0,a
    setb  le2
    lcall delay

```

```

;      clr      Le2
      Mov      R1,#Data1+1
      Mov      R2,#7

bd3:   jb      t500ms_oke,ResetSaja
      mov      BounceDTMF,#100

bd1a:  jnb     Std,BD3
      mov      r0,#10
      djnz    r0,$
      djnz    bouncedTMF,BD1a

;ADA DTMF
      mov      a,p3
      anl     a,#0fh
      mov     @r1,a
      inc     r1

bd2a:  ;
      clr     buzzer
      jb     Std,$
      mov     r0,#255
      djnz   r0,$
      jb     std,bd2a
      setb   buzzer

;      mov     p0,a
;      setb   Le2
;      lcall  delay
;      clr     Le2

      djnz   r2,bd3
      clr    ET0
      sjmp   terjemahan

ResetSaja:
      clr     ET0
      acall  reset_time0
      ljmp   gagal

terjemahkan:
      mov     r1,#8
      mov     r0,#Data1

bd4:   mov     a,@r0
      lcall  convertMT
      mov     @r0,a
      inc    r0
      djnz   r1,bd4

;proses alarm
      mov     r0,#Data1
      mov     r1,#4
      mov     dptr,#Alarm

bd5x:  mov     a,@r0
      mov     b,a
      mov     a,#0
      movc   a,@+dptr
      cjne  a,b,dilanjut1x

```

```

inc      r0
inc      dptr
djnz    r1,bd5x
;berarti Alarm
Alm:    mov      r1,#5
        clr      Buzzer
        lcall   delay2
        setb    Buzzer
        lcall   delay2
        djnz   r1,alm

dilanjut1x:
;proses kondisi pintu tertutup
mov      r0,#Data1
mov      r1,#4
mov      dptr,#PintuTutup
bd5:    mov      a,@r0
        mov      b,a
        mov      a,#0
        movc   a,@a+dptr
        cjne  a,b,dilanjut1
        inc    r0
        inc    dptr
        djnz  r1,bd5
;berarti pintu tertutup
        clr    Led1
        setb  LE1
        lcall delay
        clr  LE1
        sjmp CekKunci
dilanjut1:
mov      r0,#Data1
mov      r1,#4
mov      dptr,#PintuBuka
bd5a:   mov      a,@r0
        mov      b,a
        mov      a,#0
        movc   a,@a+dptr
        cjne  a,b,CekKunci
        inc    r0
        inc    dptr
        djnz  r1,bd5a
;berarti pintu terbuka
        setb  Led1
        setb  LE1
        lcall delay
        clr  LE1

CekKunci:
;proses kondisi kunci
mov      r0,#Data1+4
mov      r1,#4
mov      dptr,#KunciTutup
bd6:    mov      a,@r0
        mov      b,a
        mov      a,#0
        movc   a,@a+dptr

```

```

    cjne    a,b,dilanjut2
    inc    r0
    inc    dptr
    djnz   r1,bd6
;berarti kunci tertutup
    clr    Led1
    setb   LE2
    lcall  delay
    clr    LE2
    sjmp   TidakAdaDTMF
dilanjut2:
    mov    r0,#Data1+4
    mov    r1,#4
    mov    dptr,#KunciBuka
bd6a:
    mov    a,@r0
    mov    b,a
    mov    a,#0
    movc   a,@a+dptr
    cjne   a,b,Gagal
    inc    r0
    inc    dptr
    djnz   r1,bd6a
;berarti kunci terbuka
    setb   Led1
    setb   LE2
    lcall  delay
    clr    LE2
    sjmp   TidakAdaDTMF

Gagal:
    lcall  Tiit_panjang

TidakAdaDTMF:
    ljmp   MulaiA1

AdaDTMF:
    lcall  Tiit_panjang
    ljmp   Mulai1

ConvertMT:
    Mov    Dummy,#0
    jnb   Acc.0,d5_0a
    setb  dummy.3
    sjmp  bit5a
d5_0a:
    clr   dummy.3
bit5a:
    jnb  Acc.1,d6_0a
    setb dummy.2
    sjmp bit6a
d6_0a:
    clr   dummy.2
bit6a:
    jnb  Acc.2,d7_0a
    setb dummy.1
    sjmp bit7a
d7_0a:
    clr   dummy.1

```

```

bit7a:      jnb      Acc.3,d8_0a
            setb     dummy.0
            sjmp     bit8a
d8_0a:      clr      dummy.0
bit8a:      mov      a,dummy
            ret
    
```

```

ConvertTP:
            Mov      Dummy,#0
            jnb     Acc.4,d5_0
            setb     dummy.7
            sjmp     bit5
    
```

```

d5_0:      clr      dummy.7
bit5:      jnb     Acc.5,d6_0
            setb     dummy.6
            sjmp     bit6
    
```

```

d6_0:      clr      dummy.6
bit6:      jnb     Acc.6,d7_0
            setb     dummy.5
            sjmp     bit7
    
```

```

d7_0:      clr      dummy.5
bit7:      jnb     Acc.7,d8_0
            setb     dummy.4
            sjmp     bit8
    
```

```

d8_0:      clr      dummy.4
bit8:      setb     dummy.3
            setb     dummy.2
            setb     dummy.1
            setb     dummy.0
    
```

```

            mov      a,dummy
            ret
    
```

```

;*****
;Proses Baca Keypad
;*****
Keypad4x4:
            mov      keybounc,#keybounce
            mov      keyport,#0FFh
            clr      kolom1
    
```

```

u11:      jb      baris1,key1
            djnz     keybounc,u11
            mov      keydata,#1
            lcall    Beep
            jnb     baris1,$
            ret
    
```

```

            mov      a,dummy
            ret
    
```

```

;*****
;Proses Baca Keypad
;*****
Keypad4x4:
            mov      keybounc,#keybounce
            mov      keyport,#0FFh
            clr      kolom1
    
```

```

u11:      jb      baris1,key1
            djnz     keybounc,u11
            mov      keydata,#1
            lcall    Beep
            jnb     baris1,$
            ret
    
```

```

            mov      a,dummy
            ret
    
```

```

            mov      a,dummy
            ret
    
```



```
key1:    jb      baris2,key2
         djnz   keybounc,key1
         mov    keydata,#4
         lcall  Beep
         jnb   baris2,$
         ret

key2:    jb      baris3,key3
         djnz   keybounc,key2
         jnb   baris3,$
         lcall  Beep
         mov    keydata,#7
         ret

key3:    jb      baris4,key4
         djnz   keybounc,key3
         jnb   baris4,$
         lcall  Beep
         mov    keydata,#0Eh ; (*) bintang
         ret

key4:    setb   kolom1
         clr    kolom2
         jb     baris1,key5
         djnz   keybounc,key4
         jnb   baris1,$
         lcall  Beep
         mov    keydata,#2
         ret

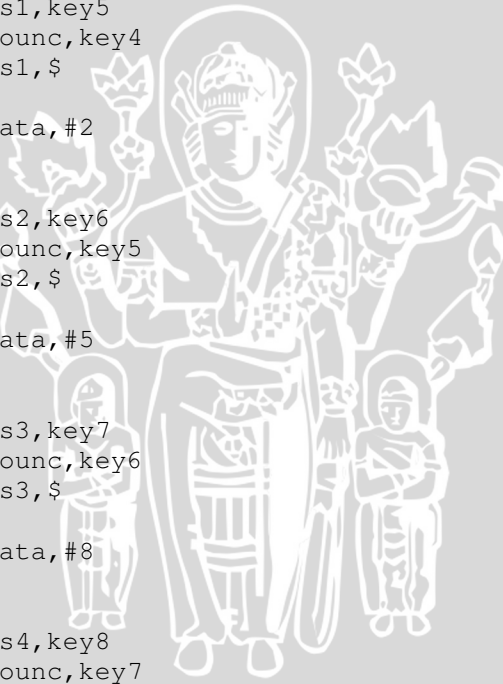
key5:    jb      baris2,key6
         djnz   keybounc,key5
         jnb   baris2,$
         lcall  Beep
         mov    keydata,#5
         ret

key6:    jb      baris3,key7
         djnz   keybounc,key6
         jnb   baris3,$
         lcall  Beep
         mov    keydata,#8
         ret

key7:    jb      baris4,key8
         djnz   keybounc,key7
         jnb   baris4,$
         lcall  Beep
         mov    keydata,#0
         ret

key8:    setb   kolom2
         clr    kolom3
         jb     baris1,key9
         djnz   keybounc,key8
         jnb   baris1,$
         lcall  Beep
         mov    keydata,#3
         ret

key9:    jb      baris2,key10
         djnz   keybounc,key9
```





```

jnb      baris2,$
lcall   Beep
mov     keydata,#6
ret

key10:   jb      baris3,key11
         djnz   keybounc,key10
         jnb   baris3,$
         mov   keydata,#9
         lcall Beep
         ret

key11:   jb      baris4,key12
         djnz   keybounc,key11
         jnb   baris4,$
         lcall Beep
         mov   keydata,#0Fh ;pagar
         ret

key12:   mov     keydata,#0FFh
         ret

delay10: lcall   delay1
         lcall   delay1
         lcall   delay1
         lcall   delay1
         lcall   delay1
         ret

delay2:   lcall   delay10
         lcall   delay10
         ret

delay1:   mov     r2,#0ffh

ld1:     lcall   delay
         djnz   r2,ld1
         ret

Beep:    clr     buzzer
         lcall   delay21
         setb   buzzer
         lcall   delay21
         ret

delay:   mov     r7,#0ffh

sini:    djnz   r7,sini
         ret

delay21: lcall   delay
         lcall   delay
         lcall   delay
         lcall   delay

```



```

lcall    delay
ret

delay6:
lcall    delay
lcall    delay
lcall    delay
lcall    delay
lcall    delay
lcall    delay
lcall    delay
lcall    delay
lcall    delay
ret

```

Awal:

```

Clr      Buzzer
lcall    delay1
Setb     Buzzer

Mov      P0,#0ffh
Clr      Cy

Mov      R1,#8
Mov      A,#0ffh

```

loopAwal:

```

RLC      A
Mov      p0,A
setb     LE1
setb     LE2
nop
nop
clr      LE1
clr      LE2
lcall    delay10
djnz     r1,loopAwal

acall    MatikanLED

ret

```

MatikanLed:

```

mov      p0,#0ffh
lcall    delay
clr      LE1
clr      LE2
nop
nop
setb     LE1
setb     LE2
nop
nop
clr      LE1
clr      LE2
ret

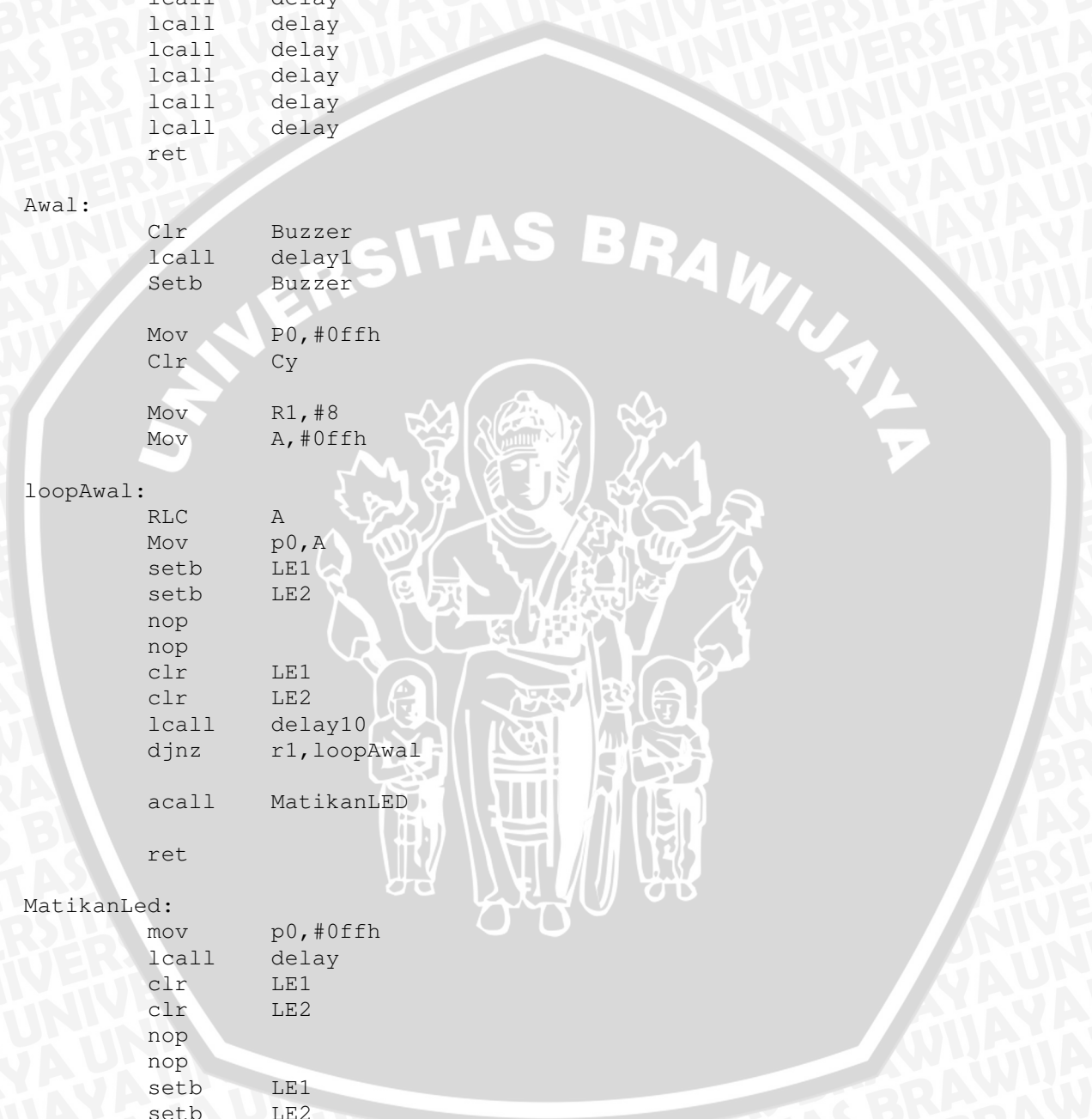
```

Tiit\_Panjang:

```

Clr      Buzzer
lcall    delay2

```



```

        lcall    delay2
        lcall    delay2
        Setb    Buzzer
        ret

Tiit_Pendek:
        Clr     Buzzer
        lcall    delay1
        lcall    delay1
        lcall    delay1
        Setb    Buzzer
        ret

reset_time0:
        mov     time50,#00
        clr     t500ms_oke
        ret

kill_T0:
        clr     t500ms_oke
        clr     tr0
        clr     ET0
        ret

Timer50ms:
        mov     tmod,#21h
        mov     th0,#3ch
        mov     t10,#0afh
        setb    tr0
        ret

time0:
        push    0e0h
        push    psw
        clr     tr0
        clr     t500ms_oke
        inc     time50
        mov     a,time50
        cjne   a,250,out_time0
        setb    t500ms_oke
        sjmp   quit_time
out_time0:
        acall   timer50ms
quit_time:
        pop     psw
        pop     0e0h
        ret

PintuTutup:
        db     1,1,1,6,
PintuBuka:
        db     2,2,2,7,
KunciTutup:
        db     3,3,3,8,
KunciBuka:
        db     4,4,4,9,
Alarm:
        db     2,2,1,2,
        end

```





## LISTING PROGRAM UNIT PENGUNCI PINTU

```

;*****
;Program Utama
;*****

$mod51
T_clk          bit    p0.2
IQ_clk         bit    p0.1
Rst_Clk        bit    p0.0

Led1           bit    p2.4
Led2           bit    p2.5
RY1            bit    p2.6
RY2            bit    p2.7

Tx             bit    p2.2
SPintu         bit    p1.1 ;norm high
SKunci         bit    p1.0 ;norm high
Std            bit    p2.0 ;kebalikan mster
TE            bit    p2.1 ;iden
D3_TP         bit    p3.0
D2_TP         bit    p3.1
D1_TP         bit    p3.2
D0_TP         bit    p3.3
Q1_MT         bit    p3.7
Q2_MT         bit    p3.6
Q3_MT         bit    p3.5
Q4_MT         bit    p3.4

SEC            EQU    00H
MIN            EQU    02H
HR             EQU    04H
DATE           EQU    06H
MONTH          EQU    08H
DAY            EQU    0AH
YEAR           EQU    0CH
CTR            EQU    0EH ;CONTROL REGISTER
TCR            EQU    10H ;TRICKLE CHARGE REGISTER

RTCBuffer     data    50h
JumMenit      data    60h
Detik         data    61h

Nyala          bit    20h.0
T500ms_oke    bit    20h.1
SudahDiperingatkan bit 20h.2
KondisiPintu  bit    20h.3
KondisiKunci  bit    20h.4

time50        data    25h

BounceDTMF    data    26h
Dummy         data    27h
    
```



```
Data1      data      28h
Data2      data      29h
Data3      data      2ah
Data4      data      2bh
Menit      data      2ch
```

```
BufferSimpan  data      30h
DataTekan    data      3ah
```

```
SetMenit     equ      10
```

```
org          0h
ljmp        mulai
```

```
org          000bh
acall       time0
reti
```

```
org          50h
mulai:
```

```
mov         sp,#07h
clr         led1
clr         led2
clr         RY1
clr         RY2
clr         TE
clr         Tx
clr         SudahDiperingatkan
jb          std,$
```

```
;Ambil Kondisi Awal
```

```
jb          SPintu,SetKondisiPintu
Clr         KondisiPintu
sjmp       CekKonKunci
```

```
SetKondisiPintu:
```

```
Setb       KondisiPintu
```

```
CekKonKunci:
```

```
jb          SKunci,SetKondisiKunci
Clr         KondisiKunci
sjmp       Lanjut12
```

```
SetKondisiKunci:
```

```
Setb       KondisiKunci
```

```
Lanjut12:
```

```
lcall      settingWaktu
lcall      SetKalender
lcall      ledkedip
mov        r0,#RTCBuffer
lcall      BurstRDClkReg
MOV        R0,#RTCBuffer+1      ;Display Menit
MOV        A,@R0
mov        Menit,A
MOV        R0,#RTCBuffer+0      ;Ambil detik
MOV        A,@R0
mov        detik,A
mov        JumMenit,#SetMenit
Clr        Nyala
```

```
mulail:
```

```
mov        BounceDTMF,#100
```

```
bd1:      jb      Std,AdaDTMF      ;cek dtmf jika tdk ada cek apakah jam 9
          ljmp     m11
```

```
AdaDTMF:  mov     r0,#2
          djnz    r0,$
          djnz    bounceDTMF,BD1
```

```
          mov     a,p3
          anl     a,#0f0h
          swap    a
          mov     data1,a
```

```
bd2:      setb    led1
          jb      Std,$
          mov     r0,#255
          djnz    r0,$
          jb      std,bd2
          clr     led1
```

```
;DTMF pertama sudah terbaca maka timer diaktifkan
```

```
          setb    EA
          setb    ET0
          acall   reset_time0
          acall   timer50ms
```

```
          Mov     R1,#Data1+1
          Mov     R2,#3
bd3:      jb      t500ms_oke,ResetSaja
          mov     BounceDTMF,#100
```

```
bd1a:     jnb     Std,BD3
          mov     r0,#10
          djnz    r0,$
          djnz    bounceDTMF,BD1a
```

```
          mov     a,p3
          anl     a,#0f0h
          swap    a
          mov     @r1,a
          inc     r1
```

```
bd2a:     setb    led1
          jb      Std,$
          mov     r0,#255
          djnz    r0,$
          jb      std,bd2a
          clr     led1
          djnz    r2,bd3
          clr     ET0
          sjmp    terjemahkan
```

```
ResetSaja:  clr     ET0
          acall   reset_time0
          ljmp    m11a
```

terjemahkan:

```
    mov     r1,#4
    mov     r0,#Data1
bd4:   mov     a,@r0
       lcall convertMT
       mov     @r0,a
       inc     r0
       djnz   r1,bd4
```

;proses perintah

```
    mov     r0,#Data1
    mov     r1,#4
    mov     dptr,#Lock
```

```
bd5:   mov     a,@r0
       mov     b,a
       mov     a,#0
       movc   a,@a+dptr
       cjne   a,b,dilanjut1
       inc     r0
       inc     dptr
       djnz   r1,bd5
```

;berarti perintah untuk mengunci

```
    setb   RY1
    sjmp   ML1b
```

dilanjut1:

```
    mov     r0,#Data1
    mov     r1,#4
    mov     dptr,#Unlock
```

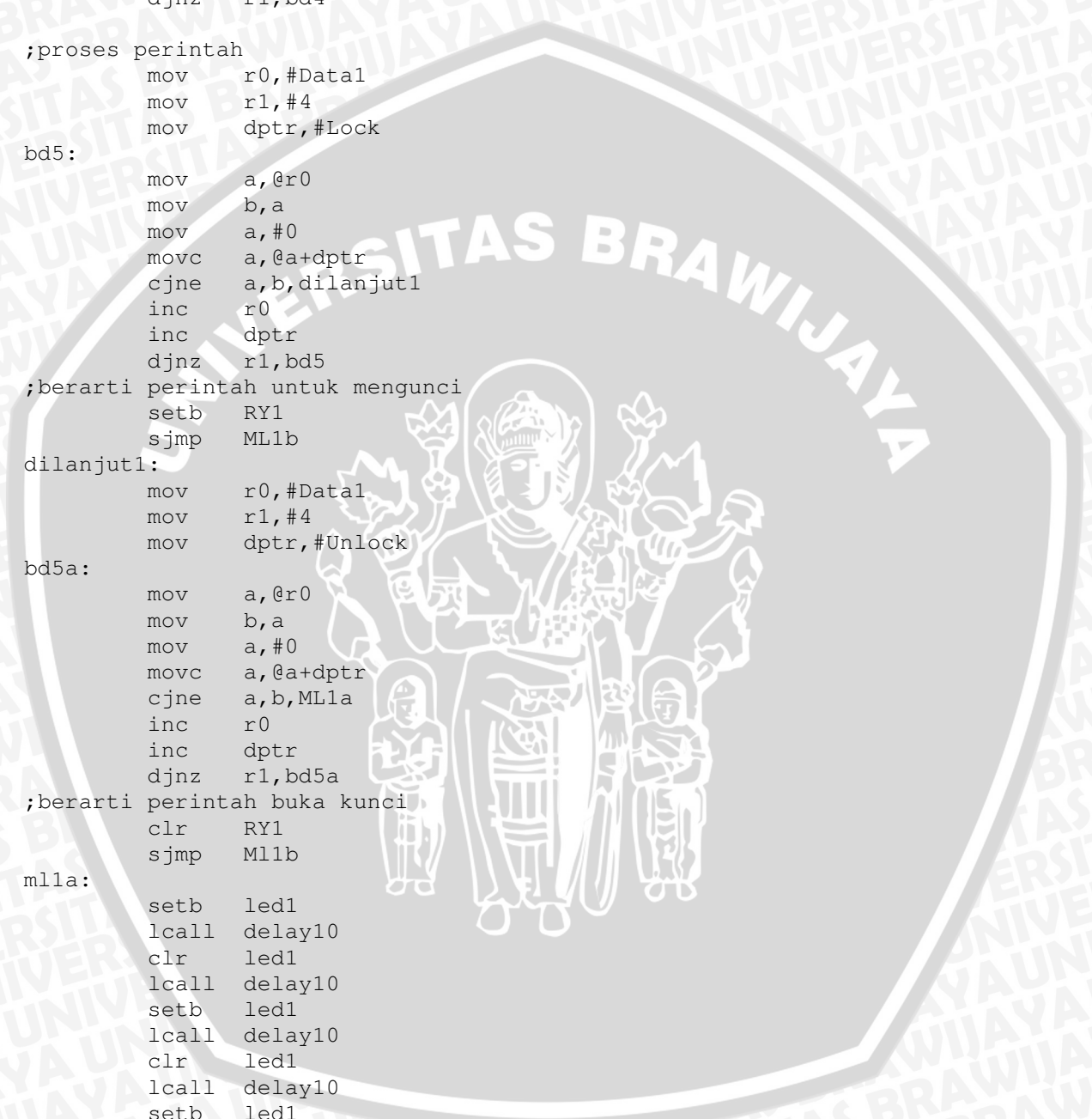
```
bd5a:  mov     a,@r0
       mov     b,a
       mov     a,#0
       movc   a,@a+dptr
       cjne   a,b,ML1a
       inc     r0
       inc     dptr
       djnz   r1,bd5a
```

;berarti perintah buka kunci

```
    clr    RY1
    sjmp   M11b
```

m11a:

```
    setb   led1
    lcall  delay10
    clr    led1
    lcall  delay10
    setb   led1
    lcall  delay10
    clr    led1
    lcall  delay10
    setb   led1
    lcall  delay10
    clr    led1
    lcall  delay10
    setb   led1
    lcall  delay10
    clr    led1
    lcall  delay10
    setb   led1
    lcall  delay10
    clr    led1
    lcall  delay10
    sjmp  m11
```



```

ml1b:
    lcall delay10
    lcall delay10
    lcall delay10
    lcall delay10
    lcall KirimKondisiPintu

ml1:
    mov     r0,#RTCBuffer
    lcall  BurstrDClkReg

MOV      R0,#RTCBuffer+2      ;Ambil jam
MOV      A,@R0
Cjne     A,#21h,AmbilDetik     ;apa jam 9???
Jb       SudahDiperingatkan,AmbilDetik1
setb     SudahDiperingatkan
Jb       SPintu,PintuMshBuka1
Jb       SKunci,KunciSaja
ljmp     mulai

PintuMshBuka1:
    lcall  KirimAlarm
    ljmp  mulai

KunciSaja:
    Setb   RY1
    ljmp  mulai

AmbilDetik:
    Clr    SudahDiperingatkan
AmbilDetik1:
    MOV    R0,#RTCBuffer+0      ;Ambil detik
    MOV    A,@R0
    Cjne   A,Detik,Kelapkelip

    sjmp  mulai2

KelapKelip:
    jb     Nyala,MatikanLED
    setb   Nyala
    clr    led2
    Mov    Detik,A
    sjmp  mulai2

MatikanLED:
    clr    Nyala
    setb   led2
    Mov    Detik,A

mulai2:
    Jb     SPintu,CekBefore1
    Jb     KondisiPintu,ClrKP      ;Pintu Skrg=0, before=1, Clr KP
    sjmp  CekKK                    ;pintu skrg=0, before=0
CekBefore1:
    Jnb    KondisiPintu,SetKP      ;Pintu Skrg=1, before=0, Set KP
    ;pintu skrg=1, before=1

CekKK:
    Jb     SKunci,CekBefore2      ;
    Jb     KondisiKunci,ClrKK     ;Kunci Skrg=0, before=1, Clr KK
    sjmp  CekLain                 ;kunci skrg=0, before=0
CekBefore2:
    Jnb    KondisiKunci,SetKK     ;Kunci Skrg=1, before=0, Set KK
    sjmp  CekLain                 ;kunci skrg=1, before=1

```



```

ClrKP:      Clr      KondisiPintu
           sjmp     KirimKon

SetKP:      Setb    KondisiPintu
           sjmp     KirimKon

ClrKK:      Clr      KondisiKunci
           sjmp     KirimKon

SetKK:      Setb    KondisiKunci

KirimKon:   lcall   KirimKondisiPintu

CekLain:    Jb      SPintu,PintuMshBuka
           Mov      JumMenit,#SetMenit
           ljmp     mulai1

PintuMshBuka:
           MOV      R0,#RTCBuffer+1           ;Display Menit
           MOV      A,@R0
           Cjne     A,Menit,TambahMenit

ljmulai:    ljmp     mulai1

TambahMenit:
           clr      led2
           Mov      Menit,A
           Djnz     JumMenit,ljmulai
           lcall   KirimAlarm
           mov      JumMenit,#SetMenit
           ljmp     mulai1

KirimAlarm:
           Mov      Dptr,#Alarm
           Mov      R1,#4
           lcall   ToneDTMF
           Mov      Dptr,#Alarm
           Mov      R1,#4
           lcall   ToneDTMF
           ret

KirimKondisiPintu:
           Jb      SPintu,PintuTerbuka
           Mov      Dptr,#PintuTutup
           Mov      R1,#4
           lcall   ToneDTMF
           sjmp     CekKunci

PintuTerbuka:
           Mov      Dptr,#PintuBuka
           Mov      R1,#4
           lcall   ToneDTMF

CekKunci:   Jb      SKunci,KunciTerbuka
           Mov      Dptr,#KunciTutup
           Mov      R1,#4
           lcall   ToneDTMF
           sjmp     KirimAman

KunciTerbuka:
           Mov      Dptr,#KunciBuka
           Mov      R1,#4

```

```

        lcall    ToneDTMF
KirimanAman:
        ret

ToneDTMF:
        Setb    Tx
        lcall   delay1
        lcall   delay1
        lcall   delay1
        Mov     a,#0
        Movc   a,@dptr
        anl    a,#0fh
        lcall   convertTP
        mov    p3,a
        setb   TE
        lcall   delay1
        lcall   delay1
        clr    TE
        lcall   delay1
        lcall   delay1
        inc    Dptr
        djnz   R1,ToneDTMF
        lcall   delay1
        lcall   delay1
        lcall   delay1
        clr    Tx
        ret

```

```

ConvertMT:
        Mov     Dummy,#0
        jnb    Acc.0,d5_0a
        setb   dummy.3
        sjmp   bit5a
d5_0a:
        clr    dummy.3
bit5a:
        jnb    Acc.1,d6_0a
        setb   dummy.2
        sjmp   bit6a
d6_0a:
        clr    dummy.2
bit6a:
        jnb    Acc.2,d7_0a
        setb   dummy.1
        sjmp   bit7a
d7_0a:
        clr    dummy.1
bit7a:
        jnb    Acc.3,d8_0a
        setb   dummy.0
        sjmp   bit8a
d8_0a:
        clr    dummy.0
bit8a:
        mov    a,dummy
        ret

```

```

ConvertTP:
        Mov     Dummy,#0
        jnb    Acc.0,d5_0

```



```

                setb    dummy.3
                sjmp    bit5
d5_0:
                clr     dummy.3
bit5:
                jnb    Acc.1,d6_0
                setb    dummy.2
                sjmp    bit6
d6_0:
                clr     dummy.2
bit6:
                jnb    Acc.2,d7_0
                setb    dummy.1
                sjmp    bit7
d7_0:
                clr     dummy.1
bit7:
                jnb    Acc.3,d8_0
                setb    dummy.0
                sjmp    bit8
d8_0:
                clr     dummy.0
bit8:
                setb    dummy.4
                setb    dummy.5
                setb    dummy.6
                setb    dummy.7
                mov     a,dummy

                ret

SettingWaktu:
                mov     r1,#BufferSimpan
                mov     r2,#8
                mov     dptr,#DataKalender
lsw:
                mov     a,#0
                movc   a,@a+dptr
                mov     @r1,a
                inc     r1
                inc     dptr
                djnz   r2,lsw
                ret

SetKalender:
;Pindahkan data dari data setting ke RTCBuffer
                MOV     R7,#8
                mov     r1,#BufferSimpan
                MOV     A,#0
                MOV     R0,#RTCBuffer
NxtMovDataRTC: PUSH    ACC
                Mov     a,@r1
                inc     r1
                MOV     @R0,A
                POP     ACC
                INC     A
                INC     R0
                DJNZ   R7,NxtMovDataRTC

                LCALL   RESET                ;Reset RTC

```

```
LCALL RUNCLK ;Enable RTC Clock
LCALL WRENB ;Enable write to RTC register
MOV B,#TCR ;Enable TC
MOV A,#10100101B
LCALL BYTEWRCLKREG

;Setting RTC
MOV R0,#RTCBuffer
LCALL BURSTWRCLKREG
LCALL WRDIS

; MOV Today,#0FFH
ret

delay10:
lcall delay1
lcall delay1
lcall delay1
lcall delay1
lcall delay1
ret

delay2:
lcall delay10
lcall delay10
ret

delay1:
mov r2,#0ffh
ld1:
lcall delay
djnz r2,ld1
ret

delay:
mov r7,#0ffh
sini:
djnz r7,sini
ret

delay21:
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
ret

delay6:
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
lcall delay
ret
```



```

;-----
;RESET SERIAL EEPROM & RTC
;-----
RESET:   CLR     t_clk      ;SK=0
         CLR     Rst_clk   ;CS=0
         SETB    Rst_clk   ;CS=1
         RET

;-----
;      RTC DS1302 ROUTINE
;-----
;WRITE BIT TO RTC
;IN : A = DATA/ADDRESS TO BE WRITE
;-----
WRBITRTC:  PUSH    07H
          MOV     R7,#8
ShfOutBit: RRC     A
          JNC     WrBitLow
WrBitHigh: SETB    IO_clk
          AJMP   ClkIn
WrBitLow:  CLR     IO_clk
ClkIn:     CLR     t_clk
          SETB    t_clk
          DJNZ   R7,ShfOutBit
          RRC     A
          POP     07H
          RET

;-----
;READ BIT FROM RTC
;OUT : A = READ DATA
;-----
RDBITRTC:  PUSH    07H
          CLR     A        ;Init ACC to zero
          SETB    IO_clk   ;IO_clk set as input
          MOV     R7,#8
ClkOut:    SETB    t_clk
          CLR     t_clk
          JNB     IO_clk,RdBitLow
RdBitHigh: SETB    C
          AJMP   ShfInBit
RdBitLow:  CLR     C
ShfInBit:  RRC     A
          DJNZ   R7,ClkOut
          POP     07H
          RET

;-----
;WRITE BYTE TO CLOCK REGISTER
;IN : B = REGISTER ADDRESS
;      A = DATA TO WRITE
;-----
BYTEWRCLKREG:
          PUSH    ACC
          PUSH    B
          LCALL  RESET
          XCH    A,B      ;A=Address
          ORL   A,#80H    ;Bit7=1
          ANL   A,#0FEH   ;Bit0=0=WR
          LCALL  WRBITRTC ;Write address
          XCH    A,B      ;A=Data

```

```

LCALL    WRBITRTC    ;Write Data
LCALL    RESET
POP      B
POP      ACC
RET

;-----
;READ BYTE FROM CLOCK REGISTER
;IN  : B = Register Address
;OUT : A = Data read
;-----
BYTERDCLKREG:    PUSH    B
                  LCALL   RESET
                  XCH    A,B                ;A=Address
                  ORL    A,#81H            ;Bit7,Bit0=1
                  LCALL   WRBITRTC        ;Write Address
                  LCALL   RDBITRTC        ;A=Read Data
                  LCALL   RESET
                  POP     B
                  RET

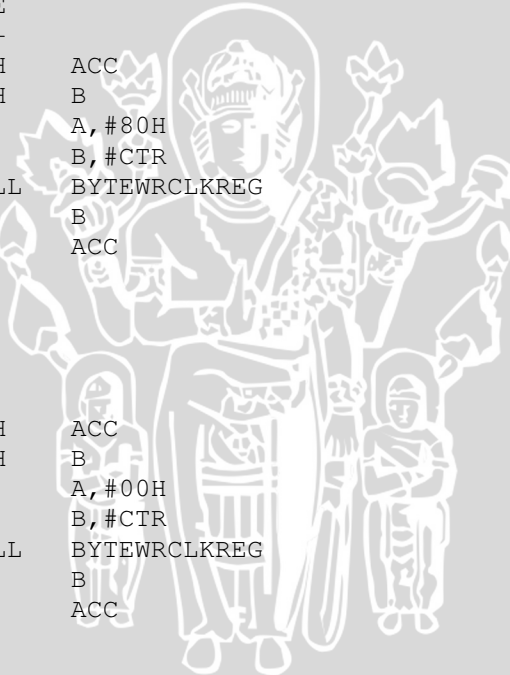
;-----
;WRITE DISABLE ROUTINE
;-----
WRDIS:          PUSH    ACC
                PUSH    B
                MOV     A,#80H
                MOV     B,#CTR
                LCALL   BYTEWRCLKREG
                POP     B
                POP     ACC
                RET

;-----
;WRITE ENABLE ROUTINE
;-----
WRENB:         PUSH    ACC
                PUSH    B
                MOV     A,#00H
                MOV     B,#CTR
                LCALL   BYTEWRCLKREG
                POP     B
                POP     ACC
                RET

;-----
;RUN CLOCK ROUTINE
;-----
RUNCLK:       PUSH    ACC
                PUSH    B
                MOV     B,#SEC
                LCALL   BYTERDCLKREG
                ANL    A,#7FH                ;0XXXXXXXB
                LCALL   BYTEWRCLKREG
                POP     B
                POP     ACC
                RET

```

UNIVERSITAS BRAWIJAYA



```

;-----
;WRITE BURST TO CLOCK REGISTER
;IN : R0 = Point to first buffer addr
;Remark : - Prepare 8 clock register data
;         start @ addr point by R0
;         - Write start @bit 0 address 0
;-----
BURSTWRCLKREG:   PUSH    ACC
                 PUSH    00H
                 PUSH    07H
                 LCALL   RESET
                 MOV     A,#0BEH           ;Clock Burst Reg. (WR)
                 LCALL   WRBITRTC
                 MOV     R7,#8
NxtByteWR:      MOV     A,@R0
                 LCALL   WRBITRTC
                 INC     R0
                 DJNZ   R7,NxtByteWR
                 LCALL   RESET
                 POP     07H
                 POP     00H
                 POP     ACC
                 RET

```

```

;-----
;READ BURST TO CLOCK REGISTER
;IN : R0 Point to first buffer addr
;OUT : 8 clock register data
;         start @ addr point by R0
;REMARK : Read start @bit 0 address 0
;-----
BURSTRDCLKREG:  PUSH    ACC
                 PUSH    00H
                 PUSH    07H
                 LCALL   RESET
                 MOV     A,#0BFH           ;Clock Burst Reg. (RD)
                 LCALL   WRBITRTC
                 MOV     R7,#8
NxtByteRD:      LCALL   RDBITRTC
                 MOV     @R0,A
                 INC     R0
                 DJNZ   R7,NxtByteRD
                 LCALL   RESET
                 POP     07H
                 POP     00H
                 POP     ACC
                 RET

```

```

;-----
;Data untuk setting RTC
;-----
DataKalender:
                DB      00H,59H,20H,18H,05H,04H,07H,00H
                 ;Sec,Min,Hr ,Dt ,Mth,Day,Yr ,CW
AmbilMenit:
                mov     r0,#RTCBuffer
                lcall   BurstRDclkReg
                MOV     R0,#RTCBuffer+1   ;Display Menit
                MOV     A,@R0
                ret

```

```

LedKedip:
    setb    p2.4
    acall   delay10
    setb    p2.5
    acall   delay10
    clr     p2.4
    acall   delay10
    clr     p2.5
    acall   delay10
    ret

```

```

reset_time0:
    mov     time50,#00
    clr     t500ms_oke
    ret

```

```

kill_T0:
    clr     t500ms_oke
    clr     tr0
    clr     ET0
    ret

```

```

Timer50ms:
    mov     tmod,#21h
    mov     th0,#3ch
    mov     t10,#0afh
    setb    tr0
    ret

```

```

time0:
    push    0e0h
    push    psw
    clr     tr0
    clr     t500ms_oke
    inc     time50
    mov     a,time50
    cjne   a,120,out_time0
    setb    t500ms_oke
    sjmp   quit_time

```

```

out_time0:
    acall   timer50ms

```

```

quit_time:
    pop     psw
    pop     0e0h
    ret

```

```

PintuTutup:
    db      1,1,1,6,

```

```

PintuBuka:
    db      2,2,2,7,

```

```

KunciTutup:
    db      3,3,3,8,

```

```

KunciBuka:
    db      4,4,4,9,

```

```

Alarm:
    db      2,2,1,2,

```

```

Lock:
    db      9,0,0eh,1,

```

```

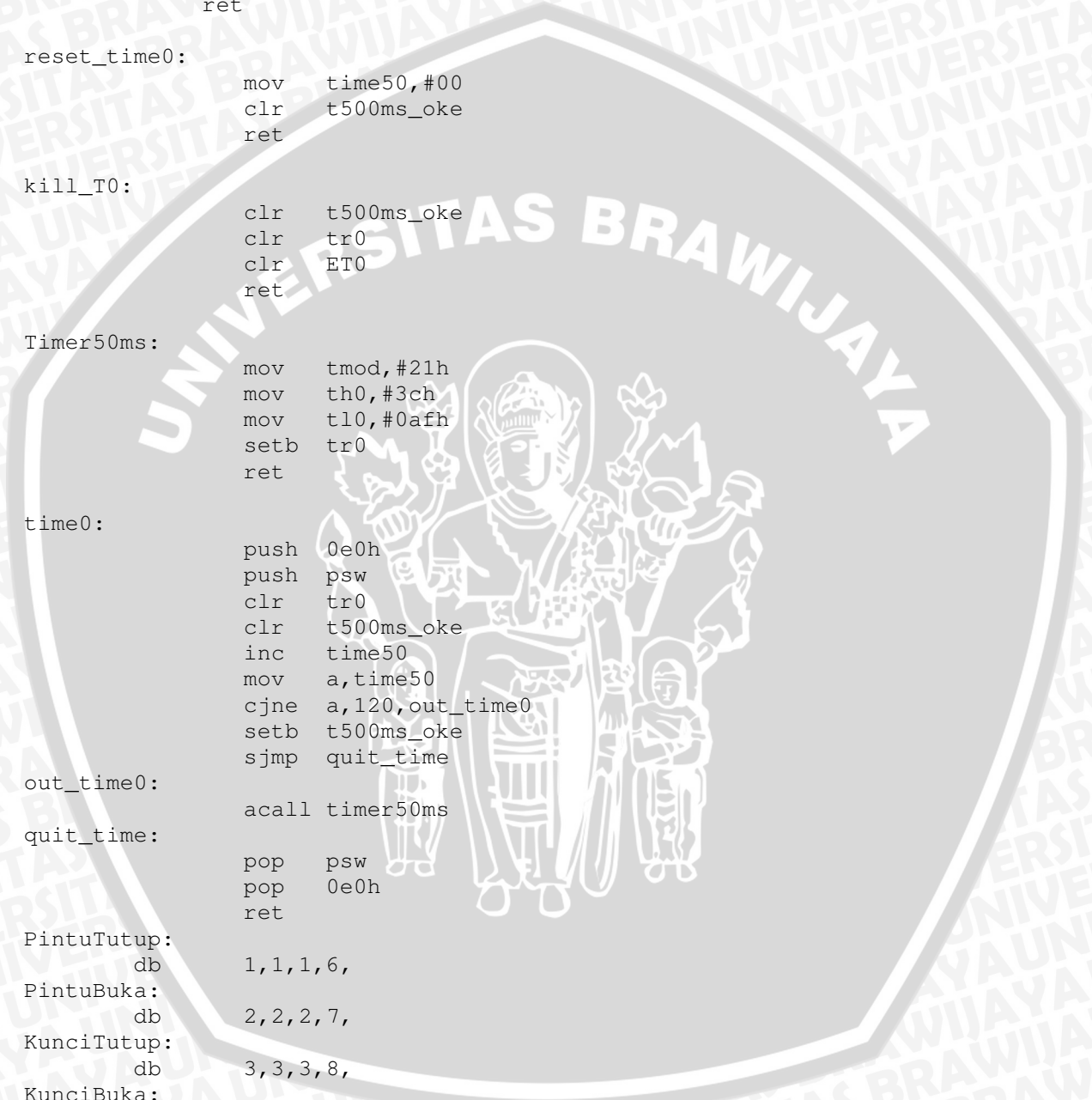
Unlock:
    db      9,0,0fh,1,

```

```

end

```





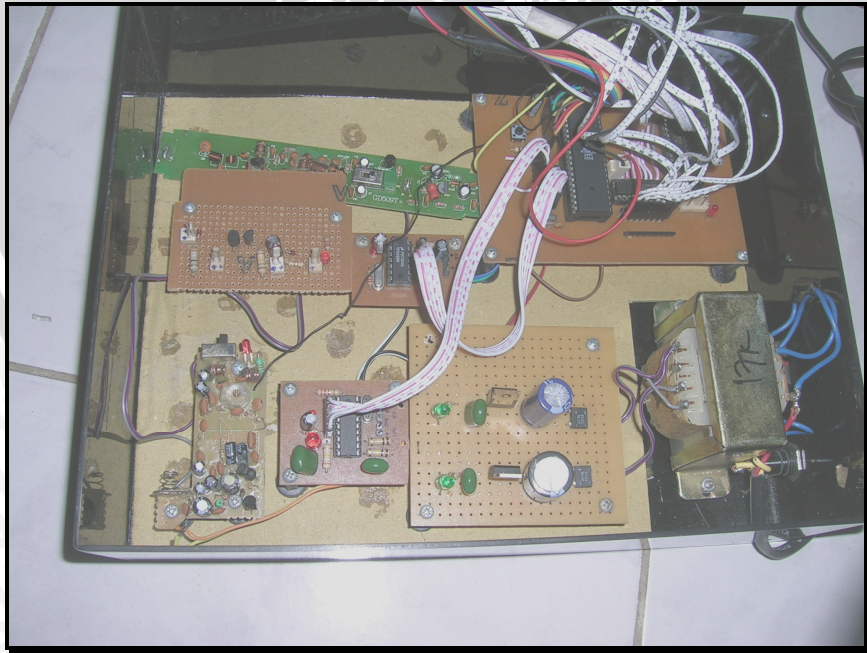
## LAMPIRAN III

### FOTO ALAT



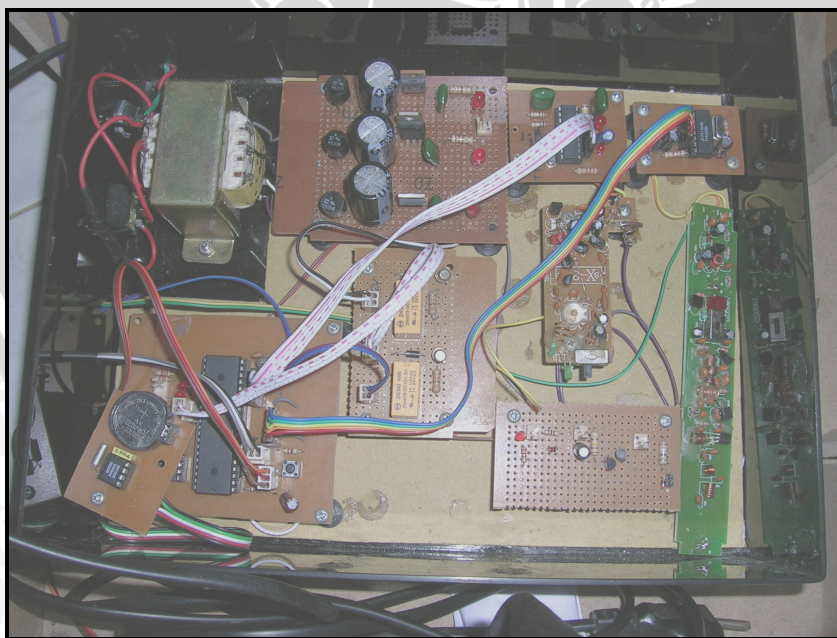
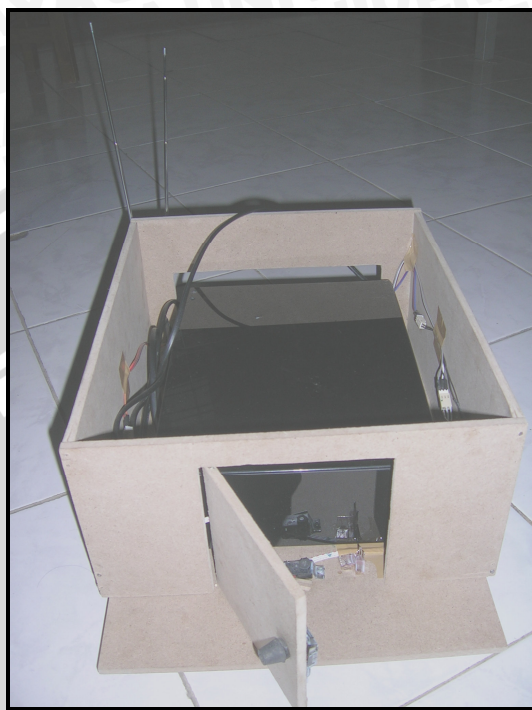


# UNIT PENGENDALI





## UNIT PENGUNCI PINTU



**LAMPIRAN IV**

***DATASHEET***

