

**IMPLEMENTASI DAN PERANCANGAN KUNCI PINTU HOTEL
DENGAN RADIO FREQUENCY IDENTIFICATION (RFID)**

Skripsi S-1
Untuk memenuhi persyaratan
Mencapai derajat sarjana S-1

Program Studi S1-Teknik Elektro
Jurusan Teknik Elektro



Diajukan oleh
Riki Astono
NIM. 5350402014

Kepada
PROGRAM STRATA 1
UNIVERSITAS NEGERI SEMARANG
2006

SKRIPSI

**IMPLEMENTASI DAN PERANCANGAN KUNCI PINTU HOTEL
DENGAN RADIO FREQUENCY IDENTIFICATION (RFID)**

Dipersiapkan dan disusun oleh:

Riki Astono

5350402014

Telah dipertahankan di depan Dewan Penguji

Pada Tanggal 8 Juli 2006

Susunan Dewan Penguji

Pembimbing Utama

Anggota Dewan Penguji

Selo, S.T, M.Sc

Drs. Suryono, M.T

Pembimbing Pendamping

Tatvantoro A. S.T, M.T

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh derajat pendidikan Sarjana Teknik

Tanggal : 11 September 2006

Drs. Djoko Adi Widodo, M.T

Pengelola Jurusan Teknik Elektro

Universitas Negeri Semarang

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang sepengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Semarang, 8 Juli 2006

Riki Astono

INTISARI

Kunci pintu sudah diciptakan sejak lama, untuk menjaga keamanan ruangan. Namun seiring dengan perkembangan teknologi, terdapat beberapa alternative kunci pintu antara lain kunci manual, kunci digital, kunci magnetic dan kunci dengan RFID yang akan dibahas dalam Skripsi ini.

Radio Frequency Identification (RFID) merupakan teknologi baru, salah satunya adalah untuk aplikasi sistem keamanan. Kunci pintu dengan RFID pada dasarnya sama dengan kunci pintu yang lain, biasanya terdapat sensor, unit prosesor dan relay magnetic. Yang membedakan adalah input yang digunakan yaitu menggunakan radio frequency identification (RFID).

Kunci pintu dengan RFID membutuhkan hardware yaitu: tag RFID, reader RFID, mikrokontroler, unit keypad, LCD dan magnetic relay, serta software downloader AT89C51. Sebagai pusat kendali adalah mikrokontroler AT89C51 yang memproses data masukan dari reader RFID dan keypad dengan keluaran untuk mengendalikan LCD dan magnetic relay.

Kerja dari kunci pintu dengan RFID adalah tag RFID dibaca oleh reader, kemudian data serial (ID) dikirim reader ke mikrokontroler. Dalam mikrokontroler data yang diterima digabungkan, kemudian dibandingkan dengan data yang tersimpan dalam memori mikrokontroler. Jika ID tag RFID sesuai dengan ID yang tersimpan dalam memori maka mikrokontroler akan mengaktifkan relay solenoid. LCD digunakan untuk memonitoring kerja sistem. Solenoid digunakan sebagai pengganti kunci pintu manual. Sedangkan keypad digunakan untuk mengganti tag RFID (kunci) dengan memasukkan ID tag yang baru yang tertulis di badan tag RFID.

RFID reader (ID-10) mampu membaca pada jarak maksimum 6 cm, jika ada penghalang maka kemampuan baca akan berkurang sesuai dengan bahan penghalang. Jika penghalang berupa plat besi, tag RFID tidak mampu dibaca karena tidak terjadi medan magnet yang digunakan tag sebagai catu daya. Sedangkan relay akan bekerja selama 6 sekon untuk membuka pintu.

ABSTRACT

A long time ago, keys had been invented to secure rooms. However, There are some alternative keys along with the technology development such as manual keys, digital keys, magnetic keys, and keys with RFID which will be discussed in this report.

Radio Frequency Identification (RFID), a new technology, is used in security system application as one of its functions. Door keys with RFID, basically same with other door keys, normally have a sensor, a processor unit, and a magnetic relay. The different is the used input which uses Radio Frequency Identification (RFID).

Door keys with RFID need hardware such as a RFID tag, a microcontroller, a keypad unit, LCD and magnetic relays, and also AT89C51 downloader software. The central controller is AT89C51 microcontroller. It processes the input data from the RFID reader and the keypad. Whereas, the output controls the LCD and relay magnetic.

There are some steps in the work procedure of RFID door keys. First, the RFID tag is read by the reader. Second, the identification number (ID) is sent by the reader to the microcontroller. In the microcontroller, the accepted data will blend. Third, it will be compared with saved data in the microcontroller memory. If the ID of the RFID tag is match with the saved ID of the memory, the microcontroller will activate the solenoid relay. LCD is used for monitoring the system process. Solenoid is used as a substitute of the manual key. Whereas, the keypad is used for exchanging the RFID tag (the key) by typing the new ID tag which is written in the RFID tag body.

RFID reader (ID-10) is able to read in maximum range of 6 cm. If there is an obstacle, the reading ability will reduce according to the obstacle material. If the obstacle is an iron plat, the RFID tag can't be read because there is no magnetic field found which is used as power supply by the tag. Whereas, the relay will work at 6 second to open the door.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji hanya bagi Allah SWT, atas segala limpahan rahmat dan karunia-Nya, sehingga penyusun dapat menyelesaikan pembuatan Skripsi ini. Dalam pelaksanaan dan pembuatan Skripsi, penyusun juga banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Atas kelancaran selama pembuatan Skripsi, maka penyusun juga mengucapkan banyak terima kasih dan penghargaan yang sebesar-besarnya kepada :

1. Ayah dan Ibu tercinta serta adik-adik atas segala bentuk dukungannya.
2. Bapak Drs. Djoko Adi Widodo, MT, selaku Ketua jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.
3. Bapak Selo, ST. MSc. selaku dosen pembimbing I, Bapak Tatyantoro Andrasto, ST, MT, selaku dosen pembimbing II atas segala bentuk bimbingan dan arahnya dalam pembuatan skripsi ini.
4. Bapak Drs. Ngadirin, MT, selaku dosen wali.
5. Staf dan Karyawan jurusan teknik elektro UNNES.
6. Otto, Dian, Andre, Kundono, WidosariNet Café, Cimot, Kingkong, Sugi', Unclank, Jemblink, Bayu, Desy, Ronggo, dan semuanya.

Penyusun menyadari bahwa masih terdapat beberapa kekurangan dalam laporan Skripsi ini. Untuk itu, kritik dan saran yang bersifat membangun sangat penyusun harapkan dari semua pihak. Semoga laporan ini dapat bermanfaat sebagaimana mestinya.

Semarang, Juni 2006

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN.....	iii
INTISARI.....	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Pembatasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat	3
1.6 Sistematika Skripsi	3
BAB II DASAR TEORI.....	5
2.1 Teknologi Radio Frequency Identification.....	5
2.1.1 Pengenalan Radio Frequency Identification	5
2.1.2 Sejarah RFID.....	7
2.1.3 Jenis- Jenis RFID.....	8

2.1.4 Bagian-bagian dari teknologi RFID	11
2.1.5 Frekuensi Kerja RFID.....	14
2.1.6 Tingkat Akurasi RFID	15
2.1.7 Cara Kerja RFID.....	17
2.1.8 Aplikasi RFID	18
2.2 Sistem Minimal Mikrokontroler AT89C51	23
2.2.1 Konfigurasi Mikrokontroler AT89C51.....	23
2.2.2 SFR (Special Function Register)	27
2.2.3 Sistem Interupsi	28
2.2.4 Timer/Counter	30
2.2.5 Decoding System.....	31
2.3 Liquid Crystal Display (LCD)	31
2.3.1 Deskripsi M1632	32
2.3.2 Kaki-kaki Modul M1632	33
2.3.3 Struktur Memori LCD	35
2.3.4 Register LCD.....	37
2.3.5 Perintah-perintah M1632	39
2.4 Relay Solenoid	45
BAB III IMPLEMENTASI DAN PERANCANGAN.....	47
3.1 Perancangan Perangkat Keras	47
3.1.1 Gambaran Plant	48
3.1.2 Rancangan Rangkaian.....	49
3.1.3 Sistem Minimal AT89C51	50

3.1.4 ID-10 Sebagai RFID reader	52
3.1.5 Unit Display	53
3.1.6 Memori Eksternal	54
3.1.7 Unit Masukan	56
3.1.8 Relay Magnetik	57
3.1.9 Catu Daya.....	57
3.2 Perancangan Perangkat Lunak	58
3.2.1 Inisialisasi Program	60
3.2.2 Start Program.....	62
3.2.3 Pengambilan Data Serial.....	63
3.2.4 Penggabungan Data	64
3.2.5 Membandingkan Data.....	66
3.2.6 Setting Nomer Baru	67
3.2.7 Scaning Keypad.....	68
3.2.8 Sub Rutin Tampilan LCD	70
BAB IV ANALISIS DAN PEMBAHASAN	74
4.1 Pengujian Alat.....	74
4.1.1 Pengujian Perangkat Keras	74
4.1.1.1 Pengujian Sistem Minimal AT89C51	74
4.1.1.2 Pengujian RFID reader	75
4.1.1.3 Pengujian LCD	77
4.1.1.4 Pengujian Keypad.....	77
4.1.1.5 Pengujian Relai Solenoid	78

4.1.2 Pengujian Perangkat Lunak.....	79
4.1.2.1 Seting Nomer ID Tag RFID	80
4.2 Pembahasan Alat	80
4.2.1 Penggunaan Port pada Mikrokontroler	80
4.2.2 Bagian Radio Frequency Identification (RFID).....	81
4.2.3 Bagian Relay Solenoid (Kunci).....	82
4.2.4 Bagian LCD.....	82
4.2.5 Bagian Keypad	83
4.3 Perbandingan RFID dengan Magnetik Card	84
4.4 Pemasangan Kunci Pintu	86
BAB V PENUTUP	87
5.1 Simpulan	87
5.2 Saran	87
DAFTAR PUSTAKA	89
LAMPIRAN	90

DAFTAR GAMBAR

Gambar 2.1 RFID <i>tag</i> dengan silicon chip dan antena eksternal	7
Gambar 2.2 Sistem kerja RFID	12
Gambar 2.3. Cara kerja <i>reader</i> RFID	17
Gambar 2.4 Blok diagram internal mikrokontroler AT89C51	25
Gambar 2.5 Konfigurasi pin AT89C51.....	26
Gambar 2.6 Hubungan HD44780 dengan layar LCD.....	32
Gambar 2.7 Modul M1632.....	32
Gambar 2.8 Susunan data status HD44780	38
Gambar 2.9 Karakter 'D' dikolom karakter kedua baris karakter pertama.....	41
Gambar 2.10 Kode perintah untuk menunjuk alamat 01h DDRAM.....	42
Gambar 2.11 Relay elektromekanis.....	46
Gambar 3.1 <i>Plant</i> (rancangan) sistem kunci pintu	48
Gambar 3.2 Rancangan rangkaian listrik.....	49
Gambar 3.3 Sistem minimal AT89C51.....	51
Gambar 3.4 Rangkaian <i>Radio Frequency Identification reader</i>	52
Gambar 3.5 Rangkaian unit display dan <i>external memory</i>	53
Gambar 3.6 Pemasangan unit keypad pada mikrokontroler	56
Gambar 3.7 Rangkaian output kunci pintu.....	57
Gambar 3.7 Rangkaian catu daya kunci pintu.....	58
Gambar 3.8 Diagram alir program utama	59
Gambar 3.9 Diagram alur inisialisasi mikrokontroler	60
Gambar 4.1 Pengujian kaki mikrokontroler pada port 1.....	75

Gambar 4.2 Pengujian RFID	76
Gambar 4.3 Pemasangan Kunci.....	86

DAFTAR TABEL

Tabel 2.1. Perbandingan antara teknologi <i>barcode</i> dengan RFID	14
Gambar 2.2 Sistem kerja RFID	12
Tabel 2.2 Fungsi alternatif port 3	26
Tabel 2.3. Register pada SFR.....	28
Tabel 2.4. Alamat layanan rutin interupsi.....	29
Tabel 2.5 Konfigurasi kaki M1632.....	33
Tabel 2.6 Perintah-perintah M1632	38
Tabel 3.1 Kondisi setelah reset.....	52
Tabel 4.1 Pengujian RFID tanpa penghalang.....	76
Tabel 4.2 Pengujian RFID dengan penghalang	77
Tabel 4.2 Perbandingan antara RFID dengan <i>magnetic Card</i>	84

DAFTAR LAMPIRAN

Lampiran A Listing Program	90
Lampiran B Skema Rangkaian	110
Lampiran C Foto Alat	111

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi sangat cepat seiring dengan waktu untuk membantu kepentingan manusia. Berbagai penelitian telah dilakukan oleh berbagai institusi dari seluruh penjuru dunia untuk menemukan teknologi baru. Penemuan baru tersebut sebagai modal awal untuk menciptakan teknologi yang lebih mutakhir dari teknologi sebelumnya. Berbagai upaya dilakukan untuk menciptakan teknologi baru, misalnya dengan membangun laboratorium yang mendukung penelitian. Dunia industri memiliki peran yang sangat penting dalam perkembangan teknologi, di satu sisi sebagai produsen teknologi baru dan di sisi lain sebagai konsumen yang membutuhkan teknologi dalam proses produksi. Penelitian terus dilakukan untuk menghasilkan teknologi baru dengan tujuan meningkatkan kesejahteraan manusia.

Semakin modern teknologi ternyata diikuti oleh semakin semakin tinggi tingkat kriminal disuatu daerah. Tingkat kejahatan di Indonesia mengalami peningkatan dari tahun ke tahun. Jenis kejahatan yang ditemukan juga semakin bertambah, dari pembunuhan, perampokan dan pencurian dan lain sebagainya. Misalnya tingkat pencurian dari tahun ke tahun terus mengalami peningkatan. Jenisnya pun semakin beragam, ada spesialis pencuri kendaraan, pencuri toko, pencuri rumah dan lain-lain. Masing-masing

mempunyai keahlian tertentu. Oleh karena itu dibutuhkan teknologi untuk menghindari pencuri, misalnya pencuri rumah, diperlukan teknologi pengaman rumah salah satunya adalah kunci rumah itu sendiri.

Teknologi kunci pintu sudah ada sejak lama dan terus berkembang dari tahun ke tahun. Mulai dari kunci yang sering kita temukan di toko-toko bangunan samapai kunci modern yang mempunyai teknologi yang lebih mutakhir. Salah satu teknologi yang membantu perancangan kunci rumah yang modern adalah radio frequency identification (RFID).

Teknologi RFID tergolong teknologi baru yang berkembang pesat mengikuti teknologi yang lain. Teknologi yang digunakan oleh RFID sendiri sebenarnya sudah ada sejak tahun 1920-an. Suatu teknologi yang lebih dekat dengan RFID, yang dinamakan IFF transponder, beroperasi pada tahun 1939 dan digunakan oleh Inggris pada Perang Dunia II untuk mengenali pesawat udara musuh atau teman.

Untuk menghadapi semakin meningkatnya kejahatan, terutama pencurian di ruangan misalnya hotel, maka diperlukan instrument yang dapat mencegah pencurian. Kunci pintu dengan teknologi *radio frequency identification* (RFID) ini akan membantu mencegah terjadinya pencurian. Dengan teknologi RFID akan lebih sulit untuk dibajak atau digandakan kuncinya, karena teknologi ini masih jarang digunakan.

1.2 Permasalahan

Bagaimana merencanakan sebuah sistem kunci pintu ruangan hotel yang bertujuan untuk menjaga *privacy* dan keamanan suatu ruangan dengan teknologi *Radio Frequency Identification*?

1.3 Pembatasan Masalah

Sistem kunci pintu ruangan hotel ini terdiri dari sebuah RFID tag, sebuah RFID reader (ID-10), sebuah sistem minimal AT89C51, unit keypad, unit display dan sebuah relay magnetic yang terhubung dengan solenoid..

1.4 Tujuan

1. Tujuan dari penelitian skripsi ini adalah mengimplementasikan *Radio Frequency Identification* (RFID) pada mikrokontroler AT89C51 untuk aplikasi kunci pintu ruangan hotel.

1.5 Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk :

1. Menjaga *privacy* dan meningkatkan keamanan ruangan.
2. Meningkatkan pelayanan dan kenyamanan pengguna ruangan.

1.6 Sistematika Skripsi

Laporan skripsi yang disusun memiliki sistematika sebagai berikut:

1. Bagian awal

Bagian pendahuluan ini berisi halaman judul, abstrak, halaman pengesahan, motto dan persembahan, kata pengantar, daftar isi, daftar lampiran.

2. Bagian isi

BAB I. Judul pendahuluan dapat dijelaskan tentang latar belakang masalah, perumusan masalah, tujuan penelitian, manfaat penelitian serta sistematika .

BAB II. Judul landasan teori yang dipakai dalam penelitian

BAB III. Perancangan dan Implementasi

BAB IV. Analisis dan pengukuran yang berisikan data atau hasil pengukuran yang disertai pembahasan.

BAB V. Judul di akhir bab ini ditarik kesimpulan dari penelitian yang dilakukan dan diberikan saran untuk penelitian selanjutnya

3. Bagian Akhir

Bagian akhir ini berisikan daftar pustaka dan lampiran-lampiran yang menunjang penelitian.

BAB II

LANDASAN TEORI

2.1 Teknologi *Radio Frequency Identification*

2.1.1 Pengenalan *Radio Frequency Identification*

Radio Frequency Identification (RFID) merupakan teknologi baru yang mampu mengirimkan identitas berupa digit tertentu dengan menggunakan gelombang radio. RFID sudah banyak digunakan pada pabrik dan sangat bermanfaat untuk mendukung manajemen persediaan barang. RFID dapat mengidentifikasi objek secara otomatis dan diprediksi dapat menggantikan *barcode* yang telah lebih dahulu kenal.

Kartu RFID terdiri dari sebuah microchip yang mempunyai sebuah antena. Di dalam kartu RFID tersebut dapat disimpan data yang ukurannya 2 kilobyte. Informasi ini bisa berisi data dari sebuah objek, identifikasi unik untuk sebuah objek dan informasi tambahan dari sebuah objek (tanggal pembuatan, tanggal pengiriman barang dan kasus *Supply chain*). Untuk membaca data dari kartu RFID ini diperlukan sebuah piranti pembaca yang akan memancarkan gelombang radio dan menangkap sinyal yang dipancarkan oleh kartu RFID. *Tag reader* meminta isi yang dipancarkan oleh signal *Radio Frekuensi* (RF). *Tag* merespon dengan memancarkan kembali data *resident* secara lengkap meliputi serial nomor urut yang unik. RFID mempunyai beberapa keuntungan yang utama melebihi sistem *barcode* yaitu kemungkinan data dapat dibaca secara otomatis tanpa memperhatikan garis arah pembacaan, melewati bahan non-konduktor seperti kartun kertas dengan kecepatan akses beberapa ratus *tag* per detik pada jarak

beberapa (sekitar 100) meter. *Tag* RFID terbuat dari microchip dengan dasar bahan dari silikon yang mempunyai kemampuan fungsi identitas sederhana yang disatukan dalam satu desain. Kemampuan *tag* RFID untuk membaca dan menulis (*read/write*), menyimpan data *storage* untuk mendukung enkripsi dan kontrol akses.

RFID yang didesain dipadukan pada sistem identifikasi pada semua tingkat rantai persediaan semua lini dilibatkan akan dapat mempunyai manfaat tidak hanya untuk pabrik tetapi juga untuk konsumen, pengawas obat dan makanan bahkan untuk pengelolaan limbah ruangan.

Sekarang ini RFID *tag* standard biasanya mampu menyimpan tidak lebih dari 128 byte. Sebagian besar memori tersebut dipakai untuk kode produk elektronik yang berisi informasi produsen, jenis produk, dan nomor serial. Karena setiap RFID *tag* adalah unik, maka dua buah kaleng minuman ringan dengan jenis yang sama akan memiliki kode yang berbeda, dimana sebaliknya jika menggunakan *barcode* semua produk sejenis akan menggunakan kode yang sama. Perbedaan lain antara *barcode* dan RFID adalah RFID *tag* memerlukan sumber tenaga listrik untuk menggerakkan sirkuit rangkaian terpadu di dalam tag tersebut, dan biasanya, tentunya, RFID *tag* tidak bisa menggunakan baterai yang membuat biayanya menjadi mahal. Pemecahannya adalah dengan cara mengirimkan energi listrik melalui medan elektromagnet dari *reader* ke RFID *tag*. Sebaliknya *reader* dapat membaca banyak RFID *tag* dalam waktu bersamaan dalam jarak antara beberapa cm sampai 10 meter atau lebih.



Gambar 2.1. RFID tag dengan silicon chip dan antena eksternal

2.1.2 Sejarah RFID

Teknologi untuk memancarkan gelombang radio yang merupakan cikal bakal teknologi RFID sudah berkembang sejak Perang Dunia II. Jepang, Amerika, dan Jerman. Pada masa itu sudah manusia menggunakan teknologi radar untuk alat bantu perang. Watson-Watt yang memimpin proyek rahasia di Inggris membangun *active Identity Friend or Foe* (IFF) yang diletakkan pada setiap pesawat di Inggris. Teknologi yang digunakan oleh RFID sendiri sebenarnya sudah ada sejak tahun 1920-an. Suatu teknologi yang lebih dekat dengan RFID, yang dinamakan IFF transponder, beroperasi pada tahun 1939 dan digunakan oleh Inggris pada Perang Dunia II untuk mengenali pesawat udara musuh atau teman. Ketika pesawat Inggris tertangkap oleh radar, maka pesawat itu akan memancarkan sinyal ke radar yang mengidentifikasi bahwa pesawat tersebut adalah teman. IFF ini sudah menggunakan prinsip dasar teknologi RFID. Pada tahun 1945, Leon Theremin yang ditemukan oleh suatu alat spionase untuk pemerintah Soviet yang memancarkan lagi gelombang radio peristiwa dengan informasi audio.

Frekuensi radio sebagai pembangkit dan pengirim identitas telah diteliti oleh para ilmuwan sekitar tahun 1950-1960. Hasil penelitian tersebut yang masih dipakai hingga kini adalah *anti-thelf systm* yang digunakan untuk mendeteksi apakah barang sudah dibayar atau belum. Pada tahun 1973, RFID yang bias diisi dengan data secara berulang mulai ditemukan di Amerika. Penggunaan RFID untuk maksud *tracking* pertama kali digunakan sekitar tahun 1980-an. RFID dengan cepat mendapat perhatian karena kemampuannya dalam men-*tracking* atau melacak objek yang bergerak. Seiring dengan perkembangan teknologi, maka teknologi RFID sendiri pun juga berkembang sehingga nantinya penggunaan RFID bisa digunakan untuk kehidupan sehari-hari. Kemudian pada tahun 1990 IBM membuat RFID yang menggunakan gelombang UHF sehingga mampu memancarkan sinyal lebih jauh dan lebih cepat. Meskipun demikian baru pada periode 1999-2003 RFID diadopsi oleh banyak perusahaan.

2.1.3 Jenis- Jenis RFID

Pada awalnya RFID terdiri dari dua jenis yaitu menggunakan baterai (aktif) dan tidak menggunakan baterai (pasif), yang tidak menggunakan baterai hanya dapat dibaca, sedang yang menggunakan baterai dapat dibaca dan ditulis.

1. RFID Aktif

Pada sistem RFID aktif ini kartu RFID mempunyai sumber daya sendiri dan mempunyai *transmitter*. Sumber daya yang digunakan bisa berasal dari baterai atau tenaga surya. Karena *mempunyai* sumber daya sendiri, RFID jenis ini mempunyai jangkauan yang lebih luas, yaitu antara 20 meter sampai 100 meter. Kartu ini akan melakukan *broadcast* sinyal untuk mengirimkan data dengan

menggunakan *transmitter* yang dimilikinya. RFID jenis ini biasanya beroperasi pada frekuensi 455 MHz, 2,45 GHz, atau 5,8 GHz. Kartu jenis ini digunakan pada aset bernilai besar (kargo, kontainer atau mobil) karena kartu jenis ini berharga relatif mahal. Kartu RFID aktif ini dapat dibagi lagi menjadi 2 jenis: *transponder* dan *beacon*. *Transponder* hanya akan melakukan *broadcast* ketika mereka menerima sinyal dari piranti pembaca. Contoh umum dari sistem ini adalah pada sistem pembayaran di gerbang jalan tol. Pada saat mobil memasuki pintu keluar, maka piranti pembaca pada gerbang akan mengirim sinyal yang akan membangunkan *transponder* di kaca depan. *Transponder* kemudian akan melakukan *broadcast* data yang berisi identitas mobil tersebut. *Beacon* banyak digunakan pada *Real-Time Locating System* (RTLS), yaitu sistem untuk mengetahui lokasi suatu objek dengan cepat. Pada *beacon*, sinyal dikirimkan secara periodik pada selang interval tertentu. Frekuensi pengiriman sinyal bergantung pada tingkat kepentingan untuk mengetahui letak aset. Sinyal yang dipancarkan oleh *beacon* ditangkap dengan menggunakan minimal 3 buah piranti pembaca. Harga dari sistem RFID aktif ini berkisar antara \$10 - \$50 (95.000 s.d. 475.000). Harga tersebut dipengaruhi oleh besar memory yang digunakan dan daya tahan sumber daya. Kartu RFID aktif ini juga dapat ditambah dengan alat pembaca temperatur udara atau kelembaban udara, ada tidaknya sensor ini juga akan mempengaruhi harga dari kartu.

2. RFID Pasif

Pada sistem RFID pasif, kartu tidak mempunyai *transmitter* maupun sumber daya. Harga dari kartu dengan sistem ini biasanya lebih murah (harga

kartu RFID pasif sekitar 20 sen s.d. 40 sen) dari kartu RFID aktif. Kartu jenis ini juga tidak membutuhkan perawatan. *Transponder* RFID terdiri dari microchip yang menempel pada antena. Karena ukurannya yang kecil, transponder bisa saja dibungkus dalam berbagai macam bentuk, seperti di dalam lipatan kertas, di dalam kertas berlabel *barcode*, atau di dalam kartu plastik. Bentuk pembungkus yang digunakan tergantung pada jenis karakteristik aplikasi yang menggunakan RFID ini. Kartu RFID pasif ini dapat menggunakan *low frequency* (124 kHz, 125 kHz, atau 135 kHz), *high frequency* (13,56MHz), atau UHF (860 MHz-960 MHz). Jenis frekuensi yang digunakan juga sangat bergantung pada karakteristik aplikasi karena tiap rentang frekuensi mempunyai karakteristik tertentu. Pada rentang frekuensi tertentu gelombang radio tidak dapat menembus benda logam atau air, rentang frekuensi juga mempunyai karakteristik jarak maksimum pancaran gelombang radio yang berbeda-beda. Perusahaan pengguna RFID umumnya banyak menggunakan RFID pasif berfrekuensi UHF dibandingkan dengan *low frequency* atau *high frequency*. Hal ini karena kartu RFID pasif yang menggunakan UHF berharga lebih murah dan jangkauannya lebih luas (jangkauannya sampai dengan 3,33 meter). Banyak aplikasi biasanya membutuhkan kartu RFID yang dapat dibaca pada jarak minimal 3 meter dari piranti pembaca. Aplikasi jenis ini misalnya aplikasi pengelolaan barang di gudang yang memerlukan kartu dapat dibaca ketika masuk pintu, dan jangkauan kartu tentu saja minimal 3 meter. Sedangkan kartu RFID yang menggunakan *low frequency* hanya dapat dibaca pada jarak maksimal 0,3 meter dari piranti pembaca, sedang untuk *high frequency* dapat dibaca pada jarak 1 meter. Metode

pengiriman data kartu RFID pasif ke piranti pembaca dapat dibagi menjadi 2 macam, yaitu:

1. *Inductive Coupling*

Gulungan tembaga pada piranti pembaca membangkitkan medan elektromagnetik,

kemudian gulungan yang ada di kartu RFID terinduksi oleh medan ini, hasil induksi inilah yang menjadi sumber tenaga bagi kartu RFID untuk mengirimkan kembali sinyal yang berisi data ke piranti pembaca. Karena menggunakan prinsip induksi ini, maka jarak antara kartu RFID dengan piranti pembaca juga harus pendek agar induksi dapat ditangkap. *Inductive coupling* ini digunakan pada kartu RFID dengan *low frequency* dan *high frequency*.

2. *Propagation Coupling*

Pada sistem ini, energi yang digunakan berasal dari energi elektromagnetik (gelombang radio) yang dipancarkan oleh piranti pembaca. Kartu RFID kemudian akan mengumpulkan energi elektromagnetik ini untuk digunakan sebagai sumber daya mengirimkan data yang dimilikinya ke piranti pembaca. Mekanisme ini disebut dengan *backscatter*. Modulasi bit data ke frekuensi bisa menggunakan *amplitude shift keying*, *phase shift keying*, atau *frequency shift keying*.

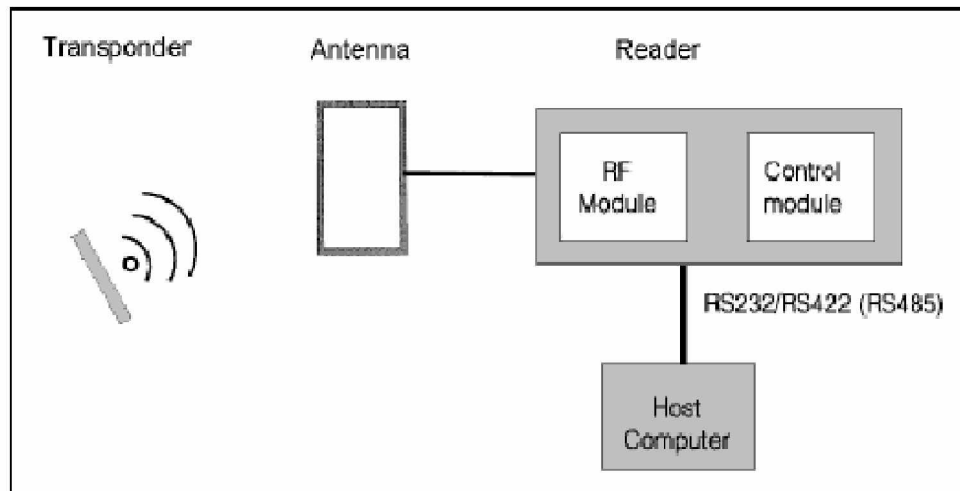
2.1.4 Bagian-bagian dari teknologi RFID

2.1.4.1 Pembaca RFID (RFID Reader)

Sebuah pembaca RFID harus menyelesaikan dua buah tugas, yaitu:

- § Menerima perintah dari *software* aplikasi
- § Berkomunikasi dengan *tag* RFID

Pembaca RFID adalah merupakan penghubung antara *software* aplikasi dengan antena yang akan meradiasikan gelombang radio ke *tag* RFID. Gelombang radio yang diemisikan oleh antena berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara *wireless* ke *tag* RFID yang berada berdekatan dengan antena.



Gambar 2.2 Sistem kerja RFID

2.1.4.2 Tag RFID (Kartu RFID/Transponder)

Tag RFID adalah devais yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari *tag* RFID umumnya memiliki memori sehingga *tag* ini mempunyai kemampuan untuk menyimpan data. Memori pada *tag* dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read-Only*, misalnya *serial number* yang unik yang disimpan pada saat *tag* tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan catu daya *tag*, *tag* RFID dapat digolongkan menjadi:

1. *Tag Aktif*: yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe *tag* ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh *tag* RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.
2. *Tag Pasif*: yaitu *tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah *tag* hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk *tag* RFID.

Setiap bagian *Tag* terdiri dari :

1. *Silicon Mikroprosesor*

Ini adalah sebuah *chip* yang terletak dalam sebuah *tag* yang berfungsi sebagai penyimpan data.

2. *Metal Coil*

Sebuah komponen yang terbuat dari kawat aluminium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Jika sebuah *tag* masuk ke dalam jangkauan *reader* maka antena ini akan mengirimkan data yang ada pada *tag* kepada *reader* terdekat.

3. *Encapsulating Material*

Encapsulating Material adalah bahan yang membungkus *tag* yang terbuat dari bahan kaca.

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai *barcode* pada masa yang akan datang. Pembacaan informasi pada *tag* RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka *tag* RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan *barcode*. Pada table 2.1 ditunjukkan perbedaan utama antara *barcode* dan RFID.

Tabel 2.1. Perbandingan antara teknologi *barcode* dengan RFID

Sistem	Barcode	RFID
Transmisi data	Optik	Elektromagnetik
Ukuran data	1 – 100 byte	128 – 8096 byte
Modifikasi data	Tidak bisa	Bisa
Posisi pembawa data	Kontak cahaya	Tanpa kontak
Jarak Komunikasi	Beberapa meter	Dari cm sampai meter
Supseptibilitas Lingkungan	Debu	Dapat diabaikan
Pembacaan jamak	Tidak bisa	Bisa

2.1.5 Frekuensi Kerja RFID

Faktor penting yang harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID. Ini adalah frekuensi yang digunakan untuk komunikasi *wireless* antara pembaca RFID dengan *tag* RFID. Ada beberapa band frekuensi yang digunakan untuk sistem RFID. Pemilihan dari frekuensi kerja sistem RFID akan mempengaruhi jarak komunikasi, interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data, dan ukuran antena. Untuk frekuensi yang rendah umumnya digunakan *tag* pasif, dan untuk frekuensi tinggi digunakan *tag* aktif. Pada frekuensi rendah, *tag* pasif tidak dapat mentransmisikan data dengan jarak

yang jauh, karena keterbatasan daya yang diperoleh dari medan elektromagnetik. Akan tetapi komunikasi tetap dapat dilakukan tanpa kontak langsung. Pada kasus ini hal yang perlu mendapatkan perhatian adalah *tag* pasif harus terletak jauh dari objek logam, karena logam secara signifikan mengurangi fluks dari medan magnet. Akibatnya *tag* RFID tidak bekerja dengan baik, karena *tag* tidak menerima daya minimum untuk dapat bekerja. Pada frekuensi tinggi, jarak komunikasi antara *tag* aktif dengan pembaca RFID dapat lebih jauh, tetapi masih terbatas oleh daya yang ada. Sinyal elektromagnetik pada frekuensi tinggi juga mendapatkan pelemahan (atenuasi) ketika *tag* tertutupi oleh es atau air. Pada kondisi terburuk, *tag* yang tertutup oleh logam tidak terdeteksi oleh pembaca RFID. Ukuran antena yang harus digunakan untuk transmisi data bergantung dari panjang gelombang elektromagnetik. Untuk frekuensi yang rendah, maka antena harus dibuat dengan ukuran yang lebih besar dibandingkan dengan RFID dengan frekuensi tinggi.

2.1.6 Tingkat Akurasi RFID

Akurasi RFID dapat didefinisikan sebagai tingkat keberhasilan pembaca RFID melakukan identifikasi sebuah *tag* yang berada pada area kerjanya. Keberhasilan dari proses identifikasi sangat dipengaruhi oleh beberapa batasan fisik, yaitu:

- § Posisi antena pada pembaca RFID
- § Karakteristik dari material lingkungan yang mencakup sistem RFID
- § Batasan catu daya
- § Frekuensi kerja sistem RFID

a. Akurasi Sistem RFID Frekuensi Rendah

Pada frekuensi rendah, contohnya pada frekuensi 13,56 MHz, komunikasi frekuensi radio antara *tag* dengan pembaca RFID sangat bergantung pada daya yang diterima *tag* dari antena yang terhubung dengan pembaca RFID. Pada ruang bebas, intensitas dari medan magnet yang diemisikan oleh antena berkurang terhadap jarak, maka terdapat batas jarak dimana *tag* tidak aktif, dan komunikasi frekuensi radio tidak dapat terjadi. Pengurangan ukuran *tag* akan mengurangi juga batas jarak. Komunikasi radio berkurang jika medan magnet harus menembus material yang mengurangi daya elektromagnetik, contohnya pada kasus objek dengan bahan logam. *Tag* RFID tidak akan terdeteksi ketika diletakkan di dalam logam, karena material logam akan meredam fluks magnet yang melalui *tag* secara drastis. Orientasi dari *tag* sangat penting dan dapat menyebabkan medan magnet bervariasi. Jika orientasi *tag* RFID sejajar dengan arah propagasi energi, maka fluks adalah nol dan komunikasi radio frekuensi tidak akan terjadi walaupun jarak antara antena dan *tag* sangat dekat.

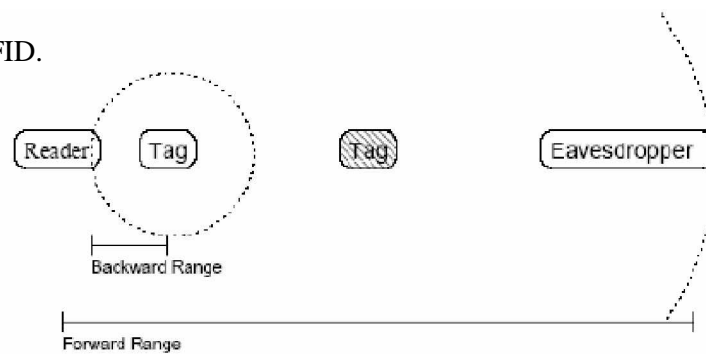
b. Akurasi Sistem RFID Frekuensi Tinggi

Pada frekuensi tinggi, performansi dari sistem RFID sangat bergantung pada lingkungan dimana komunikasi di antara *tag* dan pembaca RFID terjadi. Pada jarak tanpa hambatan proses identifikasi dapat terjadi pada jarak pada orde 10 meter. Tetapi bila ada hambatan maka jarak ini akan berkurang secara drastis. Pada frekuensi tinggi, *tag* RFID bekerja secara aktif dengan daya dari baterai. Akurasi dari *tag* RFID dapat berkurang karena kekurangan daya. Akurasi dari sistem RFID pada umumnya sangat bergantung dari lingkungan dimana sistem

RFID dioperasikan. Tantangan desain sistem RFID adalah melakukan desain infrastruktur RFID di antara lingkungan yang kurang bersahabat yang telah dijelaskan sebelumnya.

2.1.7 Cara Kerja RFID

Label *tag* RFID yang tidak memiliki baterai antenanya berfungsi sebagai pencatu sumber daya dengan memanfaatkan medan magnet dari pembaca (*reader*) dan memodulasi medan magnet, yang kemudian digunakan kembali untuk mengirimkan data yang ada dalam *tag* label RFID. Data yang diterima *reader* diteruskan ke *database host* komputer. Pada Gambar 2.3, skema proses kerja RFID.



Gambar 2.3. Cara kerja *reader* RFID

Kerugian penyebaran penggunaan RFID yang universal akan memudahkan terbukanya *privasi*, *sepionase*, dan menimbulkan ancaman keamanan baru pada suatu lingkungan pabrik yang tertutup sekalipun. Penjualan eceran yang diberi label RFID dengan *tag* yang tidak dilindungi akan dapat dimonitor dan di-*tracked* oleh pesaing lain. Pabrik mengeluarkan biaya pembuatan RFID lebih tinggi supaya dapat pendukung kriptografi seperti disampaikan Stephen A. Weis “*Most manufacturing processes currently deploying RFID systems are for higher value*

items, allowing tag costs to be in the US\$0.50-US\$1.00 range. Tags priced in this range could support basic cryptographic primitives or tamper-resistant packaging,”. Tag yang menghabiskan biaya besar ini diharapkan dapat mendukung sistem keamanan dengan kriptografi.

2.1.8 Aplikasi RFID

RFID telah banyak digunakan untuk aplikasi bisnis. Perusahaan penyedia solusi aplikasi berbasis RFID ini juga telah banyak berkembang. *Oracle* juga telah memberi dukungan terhadap aplikasi berbasis RFID ini dari segi basis data. *Xpaseo* menyediakan solusi berupa integrasi data yang dikirimkan oleh RFID dengan aplikasi bisnis. *Zebra Technologies* menyediakan aplikasi penunjang *supply chain management* dengan menggunakan bantuan RFID. Beberapa contoh aplikasi bisnis yang telah menggunakan bantuan RFID adalah:

1. *Asset tracking*, RFID dapat membantu mengetahui aset yang hilang, dicuri, atau untuk mengetahui letak aset. *Air Canada* menggunakan aplikasi ini untuk memonitor kereta makanan mereka yang tersebar di seluruh dunia.
2. *Supply chain management*, RFID dapat digunakan untuk melakukan otomasi di beberapa proses pada *supply chain management* sehingga akan mengurangi biaya pekerja, mempercepat waktu, dan meningkatkan ketepatan data.
3. *Payment sistem*, RFID dapat mempermudah dan mempercepat sistem pembayaran. Contoh dari sistem ini adalah pembayaran di pintu tol, pengendara dapat membayar biaya tol tanpa harus berhenti.

4. *Manufacturing*, RFID pada industri perakitan telah digunakan sejak satu dekade yang lalu. Dengan menggunakan RFID ini maka bagian yang akan dirakit dapat dilacak dan pekerjaan yang sedang berjalan dapat dimonitor.

Untuk sebuah produk hasil pertanian yang dijual di supermarket, jika selama ini dengan menggunakan *barcode* hanya data jenis produk yang mampu tersimpan, di masa datang diharapkan RFID *tag* mampu menyimpan tidak hanya data jenis produk namun juga misalnya untuk sebuah produk beras dapat diketahui daerah asal produksi beras, kapan beras itu pertama kali ditanam dan dipanen, metode penanaman dan pembuatannya, bahkan nama dan data petaninya secara otomatis. Keuntungan lain adalah kasir maupun pembeli dapat mengetahui total harga barang yang ada di keranjang belanja dalam waktu sekejap, atau bahkan kasir bisa mengetahui barang-barang yang mungkin saja dikutil oleh pembeli yang tidak diletakkan di keranjang belanja.

Aplikasi lain penggunaan RFID misalnya dalam pengiriman barang yang selalu dapat diawasi secara *real time* (waktu sebenarnya) dalam waktu yang tak lama lagi dapat terwujud. Orang yang tinggal di Tokyo akan mengirimkan paket kepada rekan bisnisnya di London, dimana paket tersebut dilengkapi RFID *tag* sehingga bisa selalu diamati rute perjalanannya. Orang Tokyo dapat mengetahui lokasi-lokasi paket tersebut pada waktu tertentu dengan mengaksesnya melalui internet saat paket itu mulai dikirim dari rumah orang Jepang tersebut di Shibuya sampai ke Bandara Narita, dia juga tahu kapan paketnya diangkut ke dalam pesawat JAL di Narita dan kapan paketnya diturunkan dari pesawat di Bandara Heathrow, lalu akhirnya paket itu ada dalam perjalanan dari bandara sampai di

kediaman orang London. Hal ini semua bisa dilakukan karena paket yang dilengkapi RFID *tag* itu teridentifikasi oleh *reader-reader* yang terpasang pada *gate-gate* yang dilaluinya, yang tak mungkin dilakukan jika proses identifikasi itu tidak secara otomatis dan tidak menggunakan gelombang elektromagnet.

Tak lama lagi, lingkungan kita akan mengenal diri kita bahkan tanpa kita sadari berkat teknologi-teknologi IPv6, RFID atau teknologi sensor lainnya. Laporan menunjukkan bahwa di banyak negara jumlah telepon seluler melampaui telepon biasa (non-seluler), bahkan di negara-negara tertentu perbandingan antara jumlah saluran telepon seluler dan total saluran telepon sudah melampaui angka 90%. Masyarakat di Tokyo misalnya, sebagian besar dari mereka selalu terhubung dengan internet dengan membawa telepon seluler di saat bepergian. Nantinya, informasi-informasi yang ada di RFID *tag* di pesawat telepon seluler kita, dan *reader-reader* yang tersebar di seluruh pelosok kota misalnya di setiap *ticket gate* di stasiun-stasiun kereta, akan diperbaharui dengan adanya komunikasi antara RFID *tag* dan RFID *reader*, saat kita melintas di dekat *gate-gate* tersebut. Dengan cara ini, seorang boss di kantor dapat mengecek apakah salesman-salesman di kantornya bekerja baik menawarkan produk-produk perusahaan itu kepada pelanggan atau tidak. Sebuah keuntungan bagi perusahaan namun pelanggaran privasi bagi salesman yang merasa selalu diamati langkahnya.

Sebelum kita menilai apakah teknologi seperti ini melanggar privasi atau tidak, mari kita melihat kasus lain. Sebuah SD di Propinsi Wakayama di Jepang akan mencoba penggunaan RFID *tag* yang akan dipasang di tas sekolah dan *tag* nama di seragam siswanya. Sementara RFID *reader* akan dipasang di pintu

gerbang sekolah, dan berbagai lokasi di dalam sekolah. Dengan cara ini *reader* akan mencatat apakah ada murid yang membolos atau tidak dan mengirim *e-mail* secara otomatis kepada orang tua murid yang membolos itu. Cara ini juga dapat mencegah jika ada orang yang tidak dikenal masuk ke dalam lingkungan sekolah atau terjadi tindakan *ijime* atau penindasan/kenakalan di antara sesama murid yang marak terjadi di sekolah-sekolah di Jepang. Dengan tambahan instalisasi RFID *reader* di jalur-jalur yang dilalui murid-murid diharapkan dapat mencegah kasus penculikan dan menjamin keselamatan murid-murid.

Persoalan yang tersisa adalah distribusi informasi yang berhubungan dengan privasi seseorang. Karena penyalahgunaan wewenang akses informasi ini akan melebihi dari penyadapan suara atau apa yang bisa diamati oleh seorang admin terhadap *user*nya pada sebuah internet *network*. Seseorang akan tercatat semua gaya hidupnya dengan terinstalisasinya RFID *reader* di berbagai pelosok kota, dimulai dari pagi hari saat keluar rumah sampai pulang saat malam hari, karena dalam sehari dia menggunakan kendaraan umum sebagai alat transportasinya, juga karena dia harus belanja di supermarket atau *convenient store* untuk kebutuhannya, dan lain sebagainya. Pemerintah dalam hal ini harus menjadi pelopor dengan menetapkan peraturan yang dapat mencegah terjadinya pelanggaran privasi oleh pengguna maupun penyelenggara sistem identifikasi ini. Juga sektor industri pembuat sistem RFID ini tentunya juga harus mampu menyediakan teknologi yang menggunakan teknologi nirkabel (*wireless*) ini mengakomodasi *bandwidth* yang cukup untuk kebutuhan dan dapat diakses

dengan cepat dan aman. Hanya *reader* yang terotorisasi sajalah yang dapat mengakses *tag*.

Perusahaan Hitachi tahun lalu mengeluarkan produk baru untuk RFID *tag* yang disebut μ -chip yang tak lebih besar dari sebongkah garam. Berbeda dengan RFID *tag* yang ada sebelumnya yang menggunakan external antena, μ -chip yang 0,4x0,4 mm² ini menggunakan internal antena yang dibuat di dalam silicon chip. Dengan ditambah kemasan yang baik, μ -chip dapat dipasang tidak hanya di produk yang dijual di supermarket, namun juga di uang kertas untuk mencegah pemalsuan mata uang mengingat terbatasnya *foundry* di dunia ini yang bisa memproduksi silikon chip. Atau bisa juga RFID-*tag* itu diselipkan di *tag* merek-merek pakaian atau langsung ke tekstil itu sendiri sehingga kita bahkan bisa tahu misalnya waktu terakhir kita mencuci pakaian itu di mesin cuci. Yang tentu menjadi berbahaya kalau RFID-*tag* di pakaian kita bisa terbaca oleh suatu reader yang tidak berhak sehingga semua jenis pakaian yang kita kenakan termasuk pakaian dalam tentunya. Kasus ini bukan mengada-ada, karena pelanggaran privasi serupa ini menyebabkan kekhawatiran di kalangan masyarakat yang menyebabkan terjadinya demonstrasi menentang penggunaan RFID di sebuah kota di Jerman beberapa waktu yang lampau.

Kasus penerapan teknologi RFID inilah yang mungkin bisa disebut sebagai contoh dibutuhkan kode etik dalam dunia teknologi dan *engineering*, seperti juga adanya kode etik dalam lingkungan kedokteran, hukum, maupun sastra.

Mungkin dalam beberapa waktu terakhir kita pernah mendengar istilah *ubiquitous computing*, atau *ubiquitous network*. Kata ‘*Ubiquitous*’ menurut kamus Merriam-Webster bisa diartikan sebagai ‘ada di berbagai tempat dalam waktu yang sama’. Sehingga konsep *ubiquitous computing*, atau *ubiquitous network* itu mungkin bisa diterjemahkan secara sempit misalnya sebagai kemampuan akses ke sebuah *network* (internet) di mana saja. Konsep *ubiquitous network* diharapkan akan menjadi semakin luas di masa depan berkat hadirnya teknologi RFID. Teknologi yang ada saat ini hanya mampu mengenal dan mengidentifikasi divais-divais elektronik yang terhubung dengan internet dengan *IP address* saja.

Di masa depan, dengan berkembangnya pemanfaatan teknologi RFID ini, tidak hanya divais-divais elektronik seperti komputer, PDA atau telepon seluler tetapi juga bahkan diharapkan semua barang-barang non-elektronik yang ada di sekitar kita dapat diidentifikasi secara otomatis. Perkembangan ini juga seiring dengan lahirnya teknologi internet protokol baru yang disebut IPv6 yang menggunakan 128 bit *address* yang berarti mampu mengakomodasi lebih dari 3x1.038 alamat. Sementara IPv4 yang ada saat ini hanya memiliki 32 bit *address* sehingga alamat-alamat yang tersedia terasa sudah sangat terbatas. Teknologi RFID ini diharapkan dapat mewujudkan suatu infrastruktur baru yang mengubah gaya hidup dan peradaban suatu kelompok masyarakat di masa depan seperti juga perubahan-perubahan yang terjadi pada gaya hidup masyarakat sejak lahirnya komputer dan internet.

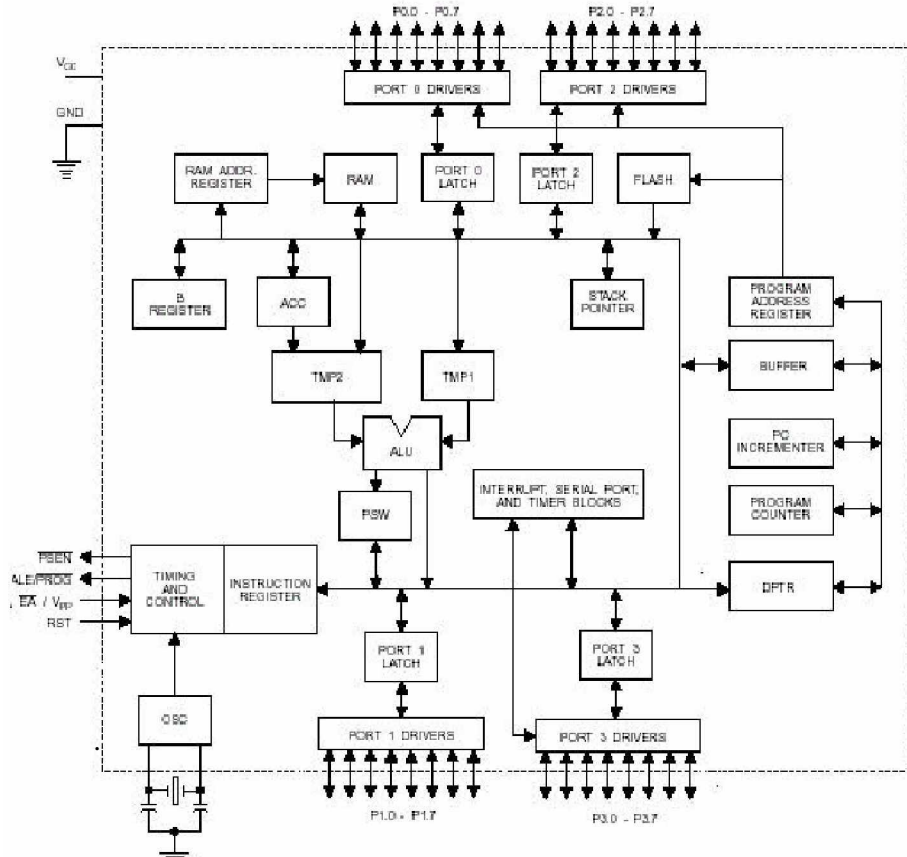
2.2 Sistem Minimal Mikrokontroler AT89C51

2.2.1 Konfigurasi Mikrokontroler AT89C51

Untuk mengimplementasikan pengendali pada kunci pintu hotel ini maka digunakan mikrokontroler AT89C51 buatan Atmel. Mikrokontroler AT89C51 adalah sebuah sistem mikrokontroler 8 bit dan memiliki 4 Kbyte *Flash Programmable and Erasable Read Only Memory* (FPEROM), AT89C51 merupakan memori dengan teknologi *nonvolatile memory*, isi memori dapat diisi ulang atau dihapus berkali-kali. Instruksi-instruksi maupun pin-nya kompatibel dengan standar MCS 51 code sehingga memungkinkan mikrokontroler ini untuk bekerja dengan mode *single chip operation* (mode *keeping* operasi tunggal) yang tidak memerlukan *memory external* (memori luar) untuk menyimpan source kode tersebut. Dengan jenis memori *flash* memudahkan memori program untuk diprogram ulang sistem. AT89C51 memiliki beberapa kelebihan antara lain: 4 Kbyte flash memory, RAM 128 byte, 32 jalur input-output, dua timer 16 bit, lima sumber interupsi, port serial dua arah, rangkaian detak (clock), dan osilator internal. AT89C51 mempunyai struktur memori terdiri atas:

- RAM internal, memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara.
- *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.
- Flash PEROM, memori yang digunakan untuk menyimpan instruksi-instruksi MCS51.

Blok diagram internal mikrokontroler AT89C51 diperlihatkan pada Gambar 2.4.



Gambar 2.4 Blok diagram internal mikrokontroler AT89C51

Konfigurasi pin-pin mikrokontroler AT89C51 diperlihatkan pada Gambar 2.5. Penjelasan dari masing-masing pin adalah sebagai berikut:

- a. Pin 1 sampai 8 (Port 1) merupakan port paralel dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan.
- b. Pin 9 (Reset) adalah masukan reset. Pin ini dihubungkan dengan rangkaian *power on reset*.
- c. Pin 10 sampai 17 (Port 3) adalah port paralel 8 bit dua arah yang memiliki fungsi alternatif. Fungsi alternatif dari tiap Port 3 dicantumkan pada Tabel 2.2

Tabel 2.2 Fungsi alternatif port 3

Port pin	Fungsi alternatif
P3.0	<i>RXD</i> (Serial Input Port)
P3.1	<i>TXD</i> (Serial Output Port)
P3.2	$\overline{INT0}$ (External Interrupt 0)
P3.3	$\overline{INT1}$ (External Interrupt 1)
P3.4	T0 (Timer 0 Eksternal Input)
P3.5	T1 (Timer 0 Eksternal Input)
P3.6	\overline{WR} (Eksternal data memori write strobe)
P3.7	\overline{RD} (Eksternal data memori read strobe)

1	P1.0	P0.0(AD0)	39
2	P1.1	P0.1(AD1)	38
3	P1.2	P0.2(AD2)	37
4	P1.3	P0.3(AD3)	36
5	P1.4	P0.4(AD4)	35
6	P1.5	P0.5(AD5)	34
7	P1.6	P0.6(AD6)	33
8	P1.7	P0.7(AD7)	32
15	P3.5(T1)	P2.0 (A8)	21
14	P3.4(T0)	P2.1 (A9)	22
19	XTAL1	P2.2(A10)	23
18	XTAL2	P2.3(A11)	24
9	RST	P2.4(A12)	25
31	$\overline{EA}/\overline{VP}$	P2.5(A13)	26
12	P3.3($\overline{INT1}$)	P2.6(A14)	27
12	P3.2($\overline{INT0}$)	P2.7(A15)	28
17	P3.7(\overline{RD})	ALE/ \overline{P}	30
16	P3.6(\overline{WR})	PSEN	29
		P3.1(TXD)	11
		P3.0(RXD)	10

Gambar 2.5 Konfigurasi pin AT89C51

- d. Pin 18 (XTAL 1) adalah pin masukan ke rangkaian osilator internal. Sebuah osilator kristal atau sumber osilator luar dapat digunakan.
- e. Pin 19 (XTAL 2) adalah pin keluaran ke rangkaian osilator internal. Pin ini digunakan bila memakai osilator kristal.

- f. Pin 20 (Ground) dihubungkan ke Vss atau ground
- g. Pin 21 sampai 28 (Port 2) adalah Port paralel 2 (P2) selebar 8 bit dua arah (bidirectional). Port 2 ini mengirimkan byte alamat bila dilakukan pengaksesan memori eksternal.
- h. Pin 29 adalah pin *PSEN (Program Store Enable)* yang merupakan sinyal pengontrol untuk membaca memori program eksternal.
- i. Pin 30 adalah pin *ALE/PROG (Address Latch Enable)* yang digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi. Pin ini juga sebagai input pulsa program selama pemrograman *flash*.
- j. Pin 31 (*EA/Vpp*). Pin *EA* harus dihubungkan ke GND agar piranti bias mengambil kode dari memori program eksternal dengan lokasi awal 0000H sampai FFFFH. *EA* harus dihubungkan ke VCC untuk eksekusi program internal. Pin ini juga menerima tegangan 12 volt selama pemrograman flash.
- k. Pin 32 sampai 39 (Port 0) merupakan port paralel 8 bit open drain dua arah. Bila digunakan untuk mengakses memori luar, port ini akan memultipleks alamat memori dengan data.
- l. Pin 40 (Vcc) dihubungkan ke Vcc (+5 volt).

2.2.2 SFR (Special Function Register)

SFR atau register fungsi khusus merupakan suatu daerah RAM dalam IC keluarga MCS51 yang digunakan untuk mengatur perilaku MCS51 dalam hal-hal khusus, misalnya tempat untuk berhubungan dengan port paralel P1 atau P3, dan sarana input/output lainnya, tapi tidak umum dipakai untuk menyimpan data seperti layaknya memori-data. SFR dalam RAM internal menempati lokasi alamat

80h sampai 7Fh. Masing-masing register pada SFR ditunjukkan dalam Tabel 2.3, yang meliputi simbol, nama dan alamatnya.

Tabel 2.3. Register pada SFR

Simbol	Nama	Alamat
Acc	Akumulator	E0h
B	B register	F0h
PSW	Program Status Word	D0h
SP	Stack Pointer	81h
DPTR (DPH) (DPL)	Data Pointer 16 Bit DPL Byte rendah DPH Byte tinggi	82h 83h
P0	Port 0	80h
P1	Port 1	90h
P2	Port 2	A0h
P3	Port 3	B0h
IP	Interupt Priority Control	B8h
IE	Interupt Enable Control	A8h
TMOD	Timer/Counter Mode Control	89h
TCON	Timer/Counter Control	88h
TH0	Timer/Counter 0 High byte	8Ch
TL0	Timer/Counter 0 Low byte	8Ah
TH1	Timer/Counter 1 High byte	8Dh
TL1	Timer/Counter 0 Low byte	8Bh
SCON	Serial Control	98h
SBUF	Serial Data Buffer	99h
PCON	Power Control	87h

2.2.3 Sistem Interupsi

Apabila CPU pada mikrokontroler AT89C51 sedang melaksanakan suatu program, maka pelaksanaan program dapat dihentikan sementara dengan meminta interupsi. Maka interupsi adalah suatu kejadian atau peristiwa yang menyebabkan

mikrokontroler berhenti sejenak untuk melayani iterupsi tersebut, apabila CPU mendapat permintaan iterupsi, program counter akan diisi alamat dari vektor iterupsi. CPU kemudian melaksanakan rutin pelayanan iterupsi mulai dari alamat tersebut. Bila rutin pelayanan iterupsi selesai dilaksanakan, CPU AT89C51 kembali ke pelaksanaan program utama yang ditinggalkan. Instruksi RETI (*return from interrupt routine*) harus digunakan untuk kembali dari layanan rutin iterupsi. Sebuah program yang seharusnya berjalan terus lurus, tiba-tiba terjadi iterupsi dan harus melayani iterupsi tersebut hingga selesai sebelum ia akan meneruskan pekerjaannya. Instruksi ini digunakan agar saluran iterupsi kembali dapat dipakai. Mikrokontroler AT89C51 menyediakan 5 sumber iterupsi yaitu: iterupsi eksternal (*External interrupt*) yang berasal dari pin 0 *INT* dan 1 *INT* , iterupsi timer (*Timer Interrupt*) yang berasal dari timer 0 maupun timer 1, dan yang terakhir adalah iterupsi port seri (*Serial Port Interrupt*) yang berasal dari bagian penerima dan pengirim port seri. Alamat awal layanan rutin iterupsi dari setiap sumber iterupsi diperlihatkan pada Tabel 2.4

Tabel 2.4. Alamat layanan rutin iterupsi

Iterupsi	Alamat	Prioritas Iterupsi
$\overline{INT0}$	03h	1
Iterupsi Timer 0	0Bh	2
$\overline{INT1}$	13h	3
Iterupsi Timer 1	1Bh	4
Iterupsi Port Serial	23h	5

Ada dua buah register yang mengontrol iterupsi, yaitu IE (*Interrupt Enable*) dan IP (*Interrupt Priority*). Register IE berfungsi untuk mengaktifkan atau

menonaktifkan sumber interupsi, sedangkan register IP digunakan untuk menentukan prioritas suatu sumber interupsi terhadap sumber interupsi lainnya, yaitu apabila ada dua atau lebih interupsi secara bersamaan. Jika register IP tidak didefinisikan, maka prioritas interupsi menggunakan urutan prioritas seperti dicantumkan pada Tabel 2.4.

2.2.4 Timer/Counter

Pada mikrokontroler AT89C51 terdapat dua buah timer/counter 16 Bit yang dapat diatur melalui perangkat lunak, yaitu *timer/counter* 0 dan *timer/counter* 1. Secara fisik sebetulnya timer juga merupakan rangkaian T flip-flop yang dapat diaktifkan dan dinonaktifkan setiap saat. Perbedaan terletak pada sumber *clock* dan aplikasinya. Jika timer mempunyai sumber *clock* dengan frekuensi tertentu yang sudah pasti sedangkan *counter* mendapat sumber *clock* dari pulsa yang hendak dihitung jumlahnya. Aplikasi dari *counter* atau penghitung biasa digunakan untuk aplikasi menghitung jumlah kejadian yang terjadi dalam periode tertentu sedangkan *timer* atau pewaktu biasa digunakan untuk menghitung aplikasi lamanya suatu kejadian yang terjadi. Apabila *timer/counter* diaktifkan pada frekuensi kerja mikrokontroler 12 MHz, *timer/counter* akan melakukan perhitungan waktu sekali setiap 1 μ S secara independen, tidak tergantung pada pelaksanaan suatu instruksi. Apabila periode waktu tertentu telah dilampaui, *timer/counter* segera menginterupsi mikrokontroler untuk memberitahukan bahwa perhitungan periode waktu telah selesai dilaksanakan. *Timer/counter* AT89C51 dapat dipilih beroperasi dalam 4 mode operasi yaitu sebagai *timer/counter* 13 bit, *timer /counter* 16 bit, *timer/counter* 8 bit dengan isi

ulang (*auto reload*), dan gabungan *timer/counter* 16 bit dan 8 bit. Register *timer mode* (TMOD) dan register *timer control* (TCON) merupakan register pembantu untuk mengatur kerja *timer 0* dan *timer 1*. Register TMOD digunakan sebagai pengontrol pemilih mode operasi *timer/counter* sedangkan register TCON digunakan sebagai pengontrol kerja *timer/counter*.

2.2.5 *Decoding System*

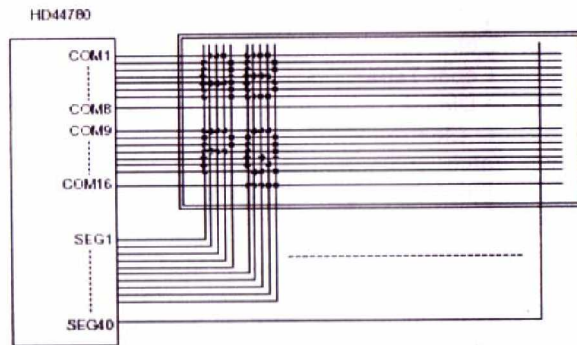
Decoding system dibutuhkan pada sebuah sistem mikrokontroler yang menggunakan *memory external* dan komponen yang mempunyai akses jalur data bus. *Decoder* ini berfungsi untuk membagi lokasi-lokasi dari *memory external* ataupun komponen yang lain. *Decoding system* ini menggunakan IC74HC138. Dengan menggunakan *decoding system*, setiap *memory external* atau komponen lain yang berhubungan dengan data bus akan menempati wilayah-wilayah yang ditentukan oleh *decoder*.

2.3 *Liquid Crystal Display (LCD)*

M1632 merupakan modul LCD matrik dengan konfigurasi 16 karakter dan 2 varis dengan setiap karakternya dibentuk oleh 8 baris pixel dan 5 kolom pixel (1 baris pixel terakhir adalah kursor). Gambar 2.6 menunjukkan hubungan antara layar LCD dengan HD44780 yang merupakan mikrokontroler pengendali LCD.

Pada Gambar 2.6 tampak karakter "A" di baris karakter 1 yang terbentuk oleh COM1 (pixel baris 1) hingga COM8 (pixel baris 8) dan SEG1 (pixel kolom 1) hingga SEG5 (pixel kolom 5). Oleh karena itu kombinasi 16 karakter akan terbentuk oleh SEG1 (pixel kolom 1) hingga SEG 40 (pixel kolom 40) dengan

setiap karakternya terdiri dari 5 kolom pixel. Kombinasi 2 baris karakter akan terbentuk oleh COM1 (pixel baris 1) hingga COM16 (pixel baris 16).

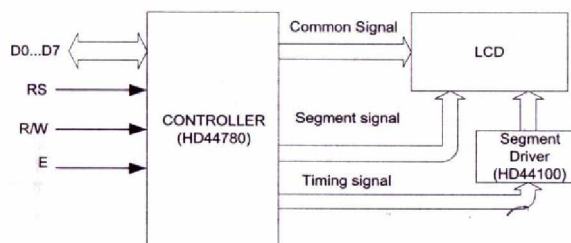


Gambar 2.6 Hubungan HD44780 dengan layar LCD

2.3.1 Deskripsi M1632

HD44780 sebetulnya merupakan mikrokontroler yang dirancang khusus untuk mengendalikan LCD dan mempunyai kemampuan untuk mengatur proses *scanning* pada layar LCD. Mikrokontroler atau perangkat tersebut hanya mengirimkan data-data yang merupakan karakter yang akan ditampilkan pada LCD atau perintah yang mengatur proses tampilan pada LCD saja.

Gambar 2.7 menunjukkan blok diagram bagian internal HD44780 yang terdiri dari bagian penguat sinyal (Signal Driver), memori, register dan antar muka mikrokontroler.



Gambar 2.7 Modul M1632

2.3.2 Kaki-kaki Modul M1632

Untuk keperluan antarmuka suatu komponen elektronik dengan mikrokontroler, perlu diketahui fungsi dari setiap kaki dari komponen tersebut. Berikut adalah table konfigurasi kaki dari LCD,

Tabel 2.5 Konfigurasi kaki M1632

No	Symbol	Level	Fungsi
1	Vss	-	0 Volt
2	Vcc	-	5 ± 10 % Volt
3	Vee	-	Penggerak LCD
4	RS	H / L	H = memasukkan data L=memasukkan instruksi
5	R / W	H / L	H = baca L = tulis
6	E	-	Enable Signal
7	DB0	H / L	Data Bus
8	DB1	H / L	
9	DB2	H / L	
10	DB3	H / L	
11	DB4	H / L	
12	DB5	H / L	

13	DB6	H / L	Kecerahan LCD
14	DB7	H / L	
15	V+BL	-	
16	V-BL	-	

Berikut merupakan penjelasan fungsi dari tiap-tiap kaki pada modul LCD M1632 sebagai berikut :

1. Kaki 1 (GND) : Kaki ini berhubungan dengan tegangan +5 volt yang merupakan tegangan untuk sumber daya dari HD44780 (khusus untuk modul M1632 keluaran Hitachi, kaki ini adalah Vcc).
2. Kaki 2 (Vcc) : Kaki ini berhubungan dengan tegangan 0 volt (Ground) dari modul LCD (khusus untuk modul M1632 keluaran Hitachi, kaki ini adalah GND).
3. Kaki 3 (Vee/Vlcd) : Tegangan pengatur kontras LCD, kontras mencapai nilai maksimum pada saat kondisi kaki ini pada tegangan 0 volt.
4. Kaki 4 (RS) : Register Select, kaki pemilih register yang akan diakses. Untuk akses ke register data, logika dari kaki ini adalah 1 dan untuk akses ke register perintah, logika dari kaki ini adalah 0.
5. Kaki 5 (R/W) : Logika 1 pada kaki ini menunjukkan bahwa modul LCD sedang pada mode pembacaan dan logika 0 menunjukkan bahwa modul LCD sedang pada mode penulisan. Untuk aplikasi yang tidak

membutuhkan pembacaan pada modul LCD, kaki ini dapat langsung dihubungkan ke ground.

6. Kaki 6 (E) : *Enable Clock LCD*, kaki mengaktifkan clock LCD. Logika 1 pada kaki ini diberikan pada saat penulisan atau pembacaan data.
7. Kaki 7-14 (D0-D7) : Data Bus, kedelapan kaki modul LCD ini adalah bagian dimana aliran data sebanyak 4 bit ataupun 8 bit mengalir saat proses penulisan maupun pembacaan data.
8. Kaki 15 (Anoda) : Berfungsi untuk tegangan positif dari *Backlight* modul LCD sekitar 4,5 volt (hanya terdapat untuk M1632 yang memiliki *backlight*).
9. Kaki 16 (Katoda) : Tegangan negatif *Backlight* modul LCD sebesar 0 volt (hanya terdapat untuk M1632 yang memiliki *Backlight*).

2.3.3 Struktur Memori LCD

Modul LCD M1632 memiliki beberapa jenis memori yang digunakan untuk menyimpan atau memproses data-data yang akan ditampilkan pada layar LCD. Setiap jenis memori mempunyai fungsi-fungsi sendiri. Jenis-jenis tersebut meliputi :

1. *DDRAM*

DDRAM merupakan memori tempat karakter yang ditampilkan berada. Contohnya, karakter "A" atau 41h yang ditulis pada alamat 00 akan tampil pada baris pertama dan kolom pertama dari LCD.

1. *CGRAM*

CGRAM adalah memori untuk menggambarkan pola sebuah karakter dan bentuk karakter dapat diubah-ubah sesuai keinginan. Akan tetapi isi memori ini akan langsung hilang saat *power supply* tidak aktif sehingga pola karakter akan hilang.

3. CGROM

CGROM adalah memori untuk menggambarkan pola sebuah karakter dan pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Oleh karena ROM bersifat permanen, pola karakter tersebut tidak akan hilang walaupun *power supply* tidak aktif. Saat HD44780 akan menampilkan data 41h yang tersimpan pada DDRAM, HD44780 akan mengambil data di alamat 41h(010 00B) yang ada pada CGROM, yaitu pola karakter A.

Seperti telah dijelaskan sebelumnya, CGRAM maupun CGROM merupakan tempat menyimpan data berupa pola-pola karakter yang akan ditampilkan pada LCD. Pola-pola karakter sebagian besar tersimpan di memori CGROM kecuali pola karakter yang ada pada lokasi yang tersimpan di memori CGRAM. Oleh karena itu, apabila data yang dikirimkan ke DDRAM adalah 00h hingga 08h, tampilan pada layar LCD adalah pola yang tersimpan pada CGRAM dan berupa pola-pola yang dapat diubah dengan mengedit isi CGRAM. Akan tetapi untuk data 11h hingga 0FFh, tampilan pada layar LCD merupakan pola yang tersimpan pada CGROM yang berupa pola-pola permanen yang sudah ditentukan oleh IC HD44780.

Oleh karena itu dapat disimpulkan bahwa pola karakter tersimpan di memori CGRAM (untuk pola karakter yang dapat diedit) dan CGROM (untuk pola karakter yang tidak dapat diedit), sedangkan data pada DDRAM berfungsi untuk menunjukkan lokasi pola karakter yang akan ditampilkan pada layar LCD.

Sebagai contoh, apabila karakter “B” akan ditampilkan pada kolom karakter kedua baris karakter pertama pada M1632, alamat DDRAM harus diatur pada lokasi 01 yang merupakan lokasi kolom karakter kedua dari M1632, kemudian data DDRAM di alamat tersebut diisi dengan alamat dari pola karakter “B”, yaitu 42h.

2.3.4 Register LCD

HD44780 yang terdapat pada modul M1632 mempunyai dua buah register yang aksesnya diatur dengan kaki RS. Saat RS berlogika 0, register yang akan diakses adalah Register Perintah dan saat RS berlogika 1, register yang akan diakses adalah Register Data.

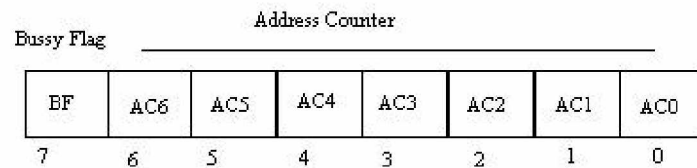
1. Register Data

Register ini adalah register dimana mikrokontroler dapat menuliskan atau membaca data ke atau dari DDRAM maupun CGRAM. Akses data ke DDRAM, baik penulisan maupun pembacaan, merupakan akses ke bagian memori tampilan pada layar LCD, sedangkan akses ke CGRAM merupakan proses untuk mengedit pola karakter yang ada pada lokasi CGRAM tersebut.

2. Register Perintah

Register ini adalah register dimana perintah-perintah mikrokontroler ke HD44780 selaku pengendali modul M1632 diberikan. Perintah-perintah tersebut

berfungsi untuk mengatur tampilan pada layar LCD atau alamat dari DDRAM dan CGRAM. Selain itu, register ini juga merupakan tempat dimana status HD44780 dapat dibaca. Bit ke-7 data status yang terbaca adalah *Busy Flag* (tanda sibuk), yaitu tanda yang mengindikasikan bahwa HD4780 masih dalam kondisi sibuk sehingga proses akses data lebih lanjut dari mikrokontroler yang terhubung pada modul M1632 harus menunggu hingga tanda sibuk ini selesai. Bit ke-6 hingga bit ke-0 adalah Address Counter (Penghitung Alamat) dari DDRAM. Address Counter ini menunjukkan lokasi dari DDRAM yang sedang ditunjuk saat itu.



Gambar 2.8 Susunan data status HD44780

2.3.5 Perintah-perintah M1632

Untuk mengatur tampilan pada layar LCD, alamat DDRAM atau CGRAM mikrokontroler yang terhubung dengan M1632 harus mengirimkan data-data tertentu ke Register Perintah sesuai Tabel 2.6 (Perintah-perintah M1632). Register-register perintah dapat dilihat pada table di bawah ini.

Tabel 2.6 Perintah-perintah M1632

Perintah	D7	D6	D5	D4	D3	D2	D1	D0	Deskripsi
Hapus Display	0	0	0	0	0	0	0	1	Hapus display dan DDRAM
Posisi	0	0	0	0	0	0	1	X	Set alamat

Awal									DDRAM di 0
Set Mode	0	0	0	0	0	1	I/D	S	Atur arah pergeseran kursor dan display
Display On/Off	0	0	0	0	1	D	C	B	Atur display (D) On/Off, kursor (C) On/OFF, Blinking (B)
Geser Kursor/Dis play	0	0	0	1	S/C	R/L	X	X	Geser kursor atau display tanpa mengubah alamat DDRAM

Set Fungsi	0	0	1	DL	N	F	X	X	Atur panjang data, jumlah baris yang tampil dan font karakter
Set Alamat CGRAM	0	1	AC G	AC G	AC G	AC G	AC G	AC G	
Set Alamat DDRAM	1	AD D	AD D	AD D	AD D	AD D	AD D	AD D	

Keterangan:

X = diabaikan

I/D , 1 = Increment, 0 = Decrement

S 0 = Display tidak geser

S/C, 1 = Display Shift, 0 = Geser Kursor

R/L, 1 = Geser Kiri, 0 = Geser Kanan

DL, 1= 8 bit, 0 = 4 bit

N, 1 = 2 baris, 0 = 1 baris

F, 1 = 5x10, 0 = 5x8

D, 0 = Display Off, 1 = Display On

C, 0 = Cursor Off, 1 = Cursor On

B, 0 = Blinking Off, 1 = Blinking On

1. Menghapus Display DDRAM

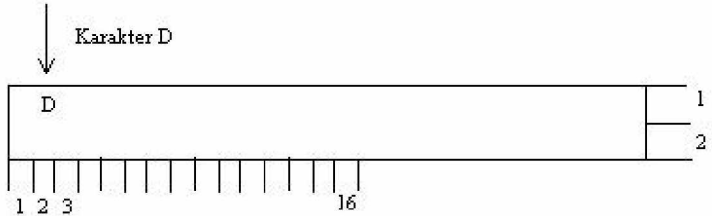
Perintah ini merupakan perintah untuk menghapus isi DDRAM sehingga layar LCD tidak akan menampilkan pola karakter apapun. Perintah ini berfungsi untuk membersihkan layar LCD dan sekaligus menghapus isi DDRAM. Kode perintah ini adalah 01h atau 0000 0001B.

2. Mengatur Alamat DDRAM

Perintah ini merupakan perintah untuk menunjuk lokasi alamat DDRAM yang diakses. Kode 02h (0000 0010B) akan memerintahkan pointer (penunjuk) agar menunjuk lokasi alamat awal, yaitu alamat 0 DDRAM. Perintah dengan logika 1 pada bit ke-7 akan memerintahkan pointer agar menunjuk lokasi sesuai konfigurasi bit ke-6 hingga bit ke-0.

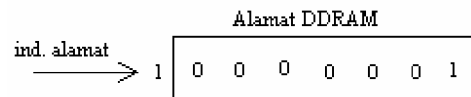
Contoh :

Menunjuk lokasi karakter 'D' ke baris karakter pertama kolom karakter kedua (Gambar 2.9)



Gambar 2.9 Karakter 'D' dikolom karakter kedua baris karakter pertama

Karakter 'D' pada Gambar 2.9 menempati alamat 01h dari DDRAM, karena itu pointer harus ditujukan ke alamat tersebut. Agar pointer dapat menunjuk ke alamat 01h DDRAM, data 81h merupakan kode yang harus dikirim ke Register Perintah.



Gambar 2.10 Kode perintah untuk menunjuk alamat 01h
DDRAM

3. Mengatur Mode

Bagian ini adalah bagian pengatur pergeseran kursor atau tampilan dengan atau tanpa mengubah alamat DDRAM. Pada proses pergeseran kursor atau tampilan dengan mengubah alamat DDRAM, logika bit ke-2 adalah logika 1 dan logika bit ke-3 hingga ke-7 adalah logika 0, sedangkan pada proses pergeseran kursor atau tampilan tanpa mengubah alamat DDRAM maka logika bit ke-4 adalah logika 1 dan bit ke-5 hingga ke-7 adalah logika 0.

a. Pergeseran Kursor atau Tampilan dengan Mengubah Format Alamat DDRAM

Proses pergeseran ini dilakukan dengan mempengaruhi alamat DDRAM, baik pada proses pergeseran kursor maupun tampilan.

Kode 04h atau 000 0100b

Pada kode ini kondisi bit I/D adalah berlogika 0 sehingga proses pergeseran adalah *Decrement* (berkurang), sedangkan logika bit S adalah logika 0 sehingga pereseran hanya terjadi pada kursor. Proses pergeseran ini juga mempengaruhi pointer alamat DDRAM (Address Counter) sehingga Address Counter juga akan mengikuti pergeseran tersebut.

Kode 05h atau 0000 0101b

Pada kode ini kondisi bit I/D adalah logika 1 sehingga proses pergeseran adalah *Increment* (bertambah), sedangkan logika bit S adalah logika 0 sehingga pergeseran hanya terjadi pada kursor. Seperti pada kode 04h, proses pergeseran ini juga mempengaruhi pointer alamat DDRAM (Address Counter) sehingga Address Counter juga akan mengikuti pergeseran tersebut, namun pada arah yang berlawanan, yaitu ke kanan

b. Pergeseran Kursor atau Tampilan Tanpa Mengubah Alamat DDRAM

Kode 06h atau 0000 0110b

Pada kode ini kondisi bit I/D adalah berlogika 0 sehingga proses pergeseran adalah *Decrement* (berkurang), sedangkan logika bit S adalah logika 1 sehingga pergeseran terjadi pada seluruh tampilan. Proses pergeseran ini juga mempengaruhi isi DDRAM sehingga data-data di alamat DDRAM dipindah ke

alamat sebelumnya. Pergeseran ini akan menyebabkan tampilan pada layar LCD bergeser ke kiri.

Kode 07h atau 0000 0111b

Pada kode ini kondisi bit I/D adalah berlogika 1 sehingga proses pergeseran adalah *Increment* (bertambah), sedangkan logika bit S adalah logika 1 sehingga pergeseran terjadi pada seluruh tampilan. Proses pergeseran ini juga mempengaruhi isi DDRAM sehingga data-data di alamat DDRAM dipindah ke alamat sesudahnya. Pergeseran ini akan menyebabkan tampilan pada layar LCD bergeser ke kanan.

4. Mengatur Keaktifan Tampilan

Proses pergeseran ini, baik pada kursor maupun seluruh tampilan, dilakukan tanpa mempengaruhi DDRAM sehingga nilai Address Counter maupun isi data pada DDRAM tetap pada kondisi sebelumnya. Proses pergeseran hanya tampak pada tampilan layar LCD saja. Hal ini dilakukan oleh HD44780 dengan mengubah urutan data yang ditampilkan.

5. Mengatur Fungsi

Proses ini dilakukan saat kondisi bit ke-5 berlogika 1, sedangkan bit ke-6 dan ke-7 berlogika 0. Bit ke-4 (DL), bit ke-3 (N) dan bit ke-2 (F) berfungsi sebagai berikut :

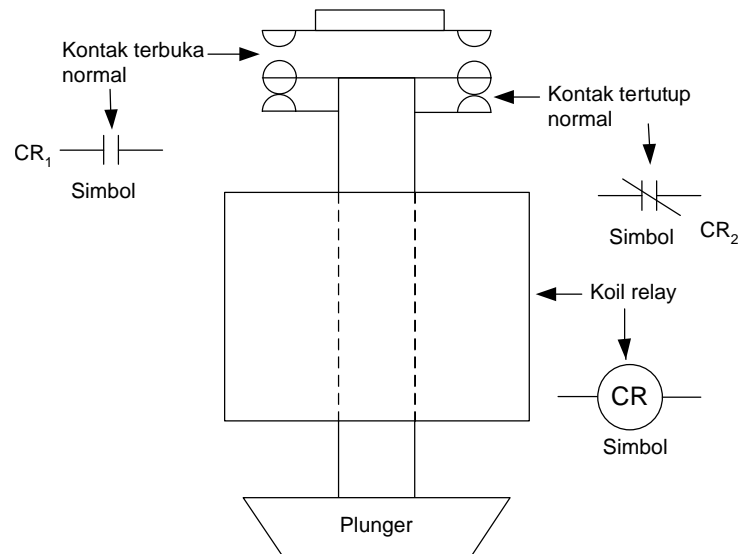
- a. DL : Bit untuk mengatur panjang data pada proses antarmuka. Logika 1 pada bit ini bertujuan untuk proses antarmuka 8 bit dan logika 0 bertujuan untuk proses antarmuka 4 bit.
- b. N : Bit untuk mengatur jumlah baris yang digunakan. Logika 1 pada bit ini adalah untuk menggunakan 2 baris dan logika 0 adalah untuk menggunakan 1 baris.
- c. F : Bit untuk mengatur *font* karakter. Logika 1 pada bit ini adalah untuk font 5x10 dan logika 0 adalah untuk font 5x8.

6. Mengatur Alamat CGRAM

Proses ini dilakukan saat kondisi bit ke-6 berlogika 1 dan bit ke-7 berlogika 0. Bit ke-5 hingga bit ke-0 merupakan Address Counter CGRAM sebanyak 6 bit atau 64 byte. Setiap pola karakter dibentuk oleh 8 byte data CGRAM sehingga total keseluruhan CGRAM mampu menyimpan 8 buah pola karakter.

2.4 Relay Solenoid

Relay pengendali elektromekanis (*an electromechanical relay = EMR*) adalah saklar magnetis. Relai ini menghubungkan rangkaian beban ON atau OFF dengan pemberian energi elektromagnetis, yang membuka atau menutup kontak pada rangkaian. EMR mempunyai variasi aplikasi yang luas baik pada rangkaian listrik maupun elektro magnetis.



Gambar 2.11 Relay elektromekanis

Relay biasanya hanya mempunyai satu kumparan, tetapi relay dapat mempunyai beberapa kontak. Jenis EMR diperlihatkan pada Gambar 2.11. Relay elektro mekanis berisi kontak diam dan kontak bergerak. Kontak yang bergerak dipasangkan pada plunger. Kontak ditunjuk sebagai normally open (NO) dan normally close (NC). Apabila kumparan diberi tenaga, terjadi medan elektromagnetis. Aksi dari medan pada gilirannya menyebabkan plunger bergerak pada kumparan menutup kontak NO dan membuka kontak NC. Jarak gerak plunger biasanya pendek – sekitar $\frac{1}{4}$ in atau kurang.

BAB III

IMPLEMENTASI DAN PERANCANGAN ALAT

3.1. Perencanaan Perangkat Keras

Dalam perancangan kunci pintu dengan teknologi RFID ini dibutuhkan hardware dan software. Hardware terdiri dari perangkat keras (yang terlihat oleh mata) dan software yang terdiri dari perangkat lunak (tidak terlihat oleh mata). Perangkat keras yang dibutuhkan meliputi alat dan bahan, sedang perangkat lunak yang dibutuhkan adalah sebuah software downloader AT89C51. Alat yang dibutuhkan antara lain:

1. *Personal computer (PC)*
2. *Downloader AT89C51*
3. *catu daya 5 Volt*
4. *Multimeter digital*
5. *Solder*
6. *Soldering Atractor*
7. *Toolset*

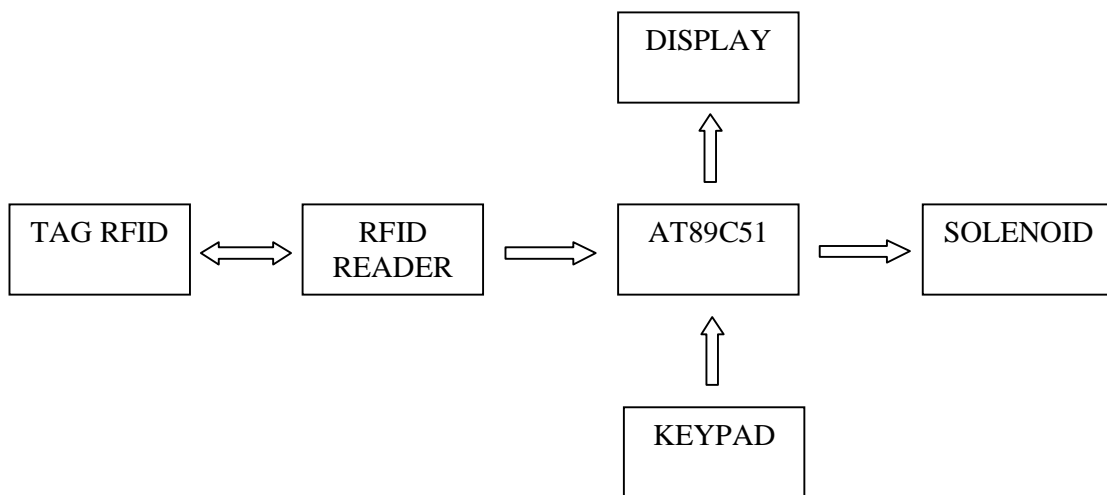
Sedangkan bahan yang digunakan pada pembuatan kunci pintu hotel diantaranya:

1. *sistem minimal AT89C51*
2. *ID-10 reader dan tag*
3. *Magnetig solenoid*
4. *external memory*
5. *Liquid Crystal Display (LCD)*

6. Catu daya 5V dan 12V

Untuk software yang digunakan adalah Tasm 301 yang berfungsi untuk mendownload rancangan perangkat lunak dari *personal computer* ke mikrokontroler.

3.1.1. Gambaran Plant

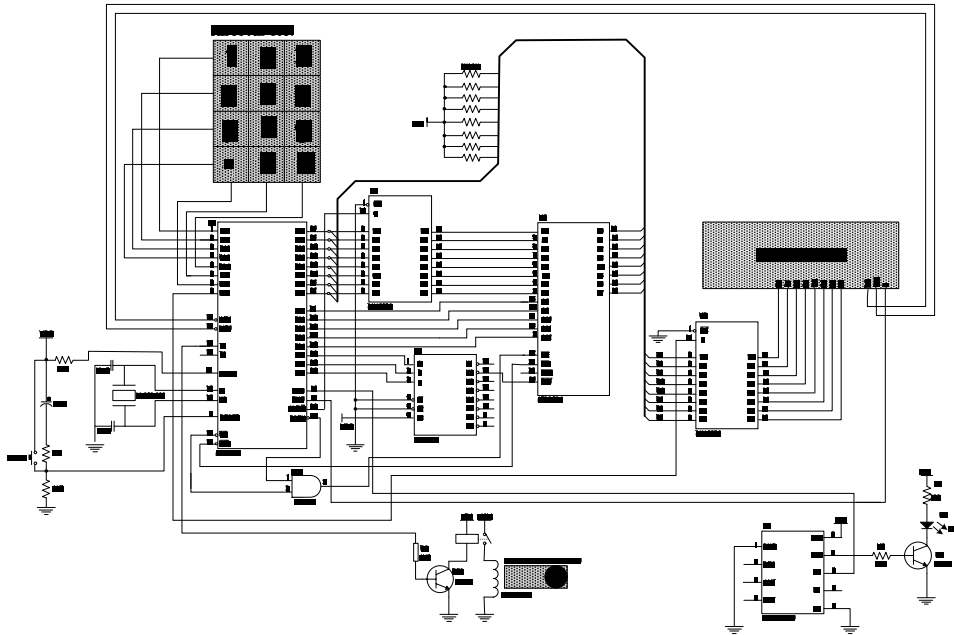


Gambar 3.1 *Plant* (rancangan) sistem kunci pintu

Dari gambar 3.1 dapat dilihat beberapa hubungan antar bagian dalam sistem aplikasi *radio frequency identification* pada kunci pintu hotel. Dari gambar 3.1, sistem minimal AT89C51 merupakan bagian utama yang berfungsi sebagai pengendali utama (*central processing unit*). Sebagai masukan atau input yaitu RFID reader (pembaca tag RFID) dan sebuah keypad sebagai masukan numeris yang dibutuhkan pada saat mengganti kunci. Sedangkan keluaran atau output sistem yaitu tampilan *liquid crystal display* (LCD) dan sebuah solenoid sebagai

kunci. Dalam tugas akhir ini akan dijelaskan satu per satu bagian sistem kunci pintu hotel dengan teknologi *radio frequency identification* (RFID).

3.1.2. Rancangan Rangkaian Listrik



Gambar 3.2 Rancangan rangkaian listrik

Gambar 3.2 diatas merupakan rangkaian yang akan digunakan dalam pengunci rumah dengan teknologi RFID. Perangkat keras ini terdiri dari beberapa bagian, yaitu bagian utama berupa sistem minimal AT89C51, dan beberapa bagian pendukung berupa unit keypad, unit RFID, unit solenoid, unit display dan unit memori eksternal. Setiap unit mempunyai mempunyai fungsi-fungsi khusus yang akan dijelaskan satu per satu pada subbab berikutnya.

sistem minimal AT89C51 ini membutuhkan external memory karena memori yang tersedia dalam internal memori mikrokontroler tidak memenuhi kebutuhan. Port 0 digunakan sebagai bus data dan bus alamat. Bus alamat ini terhubung dengan IC 74HC573 sebagai input (1D...6D).

Port 1.1 sampai port 1.6 digunakan sebagai masukan dari keypad, sedangkan

Port 1.7 digunakan sebagai enable data yang masuk ke LCD.

Port 2 digunakan sebagai pengalamatan byte tinggi yaitu port 2.0 sampai port 2.4 sedang port 2.5, port 2.6, port 2.7 digunakan untuk mengaktifkan chip select pada *external memori* melalui decoder 74HC138.

Port 3.0/RXD digunakan untuk memberikan perintah latch enable 8 bit data yang akan masuk ke LCD.

Port 3.1/TXD terhubung dengan enable pada kaki LCD

Port 3.2/INT0 mikrokontroler terhubung dengan kaki R/W pada LCD.

Port 3.3/INT1 mikrokontroler digunakan untuk mengendalikan kaki RS pada LCD.

Port 3.5 T1 digunakan sebagai output yang terhubung dengan magnetic relay yang mengendalikan solenoid pada kunci pintu

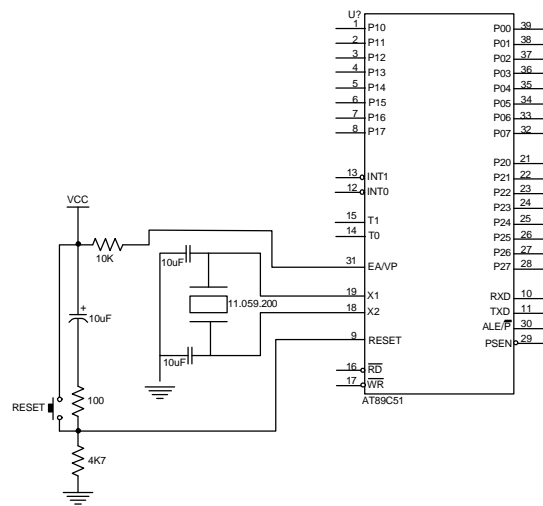
Port 3.6/WD dihubungkan dengan kaki WE (Write Enable) pada AT28C64 yang berfungsi untuk memberikan perintah tulis pada *external memory*.

Port 3.7/RD port ini bekerja bersamaan dengan kaki PSEN, kaki ini digunakan sebagai write strobe pada *external memory*. Sedangkan PSEN sendiri digunakan saat eksekusi program yang terletak pada memori eksternal.

3.1.3 Sistem minimal AT89C51

Gambar 3.3 menunjukkan rancangan dari sistem sistem minimal AT89C51. Pada sistem minimal AT89C51 ini, mikrokontroler AT89C51 sebagai komponen utama didukung oleh beberapa komponen lain diantaranya yaitu

penahan alamat (*address latch*) 74HC138, penyandi alamat (*address decoder*) 74HC138, dan memori eksternal AT28C64.



Gambar 3.3 Sistem minimal AT89C51

Unit sistem minimal AT89C51 merupakan tempat pengolahan data yang akan mengendalikan sistem pengunci rumah dengan RFID. sistem minimal AT89C51 adalah sebuah rangkaian mikrokontroler dengan komponen pendukungnya dengan tujuan dapat melakukan fungsi pemrosesan data dengan menerima input (masukan) dan mengeluarkan output dengan mengendalikan suatu instrument. Sistem minimal AT89C51 ini layaknya sebuah sebuah central processing unit (CPU) dari sebuah komputer. Sistem minimal AT89C51 membutuhkan sebuah mikrokontroler AT89C51, osilator, rangkaian reset yang terdiri dari sebuah resistor dan kapasitor dan sebuah power supplay 5 V.

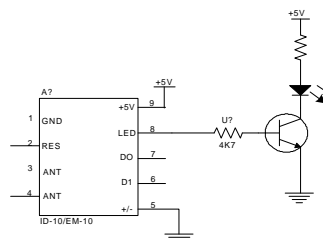
Dalam sistem minimal AT89C51 terdapat rangkaian reset yang berfungsi untuk mengembalikan mikrokontroler ke kondisi awal seperti table berikut:

Tabel 3.1 Kondisi setelah reset

Register	Isi Register
Program counter	000H
Akumulator	00H
Register B	00H
PSW	00H
Stack Pointer	07H
DPTR	0000H
Port 0-3	FFH
Interrupt Priority (IP)	XXX00000B
Interrupt Enable (IE)	0XX00000B
Register Timer	00H
SCON	00H
SBUF	00H
PCON (HMOS)	0XXXXXXXXB
PCON (CMOS)	0XXX0000B

Reset terjadi dengan adanya logika 1 selama minimal 2 cycle pada kaki RST. Setelah kondisi pin RST kembali low, mikrokontroler akan mulai menjalankan program pada alamat 0000H. Kondisi pada internal RAM tidak terjadi perubahan selama reset.

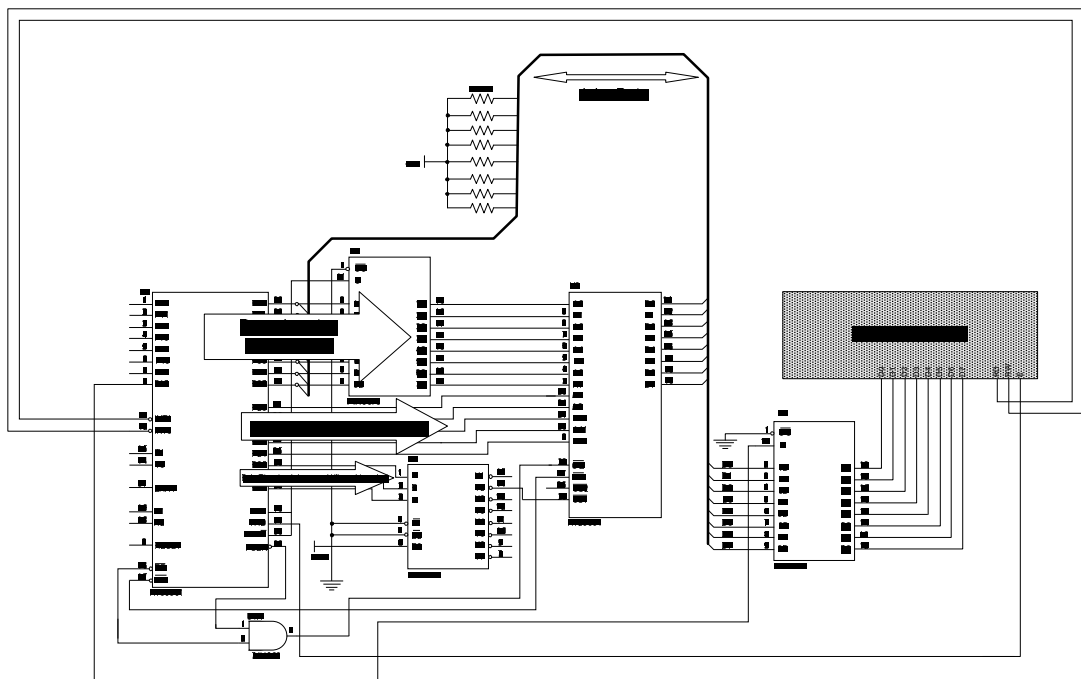
3.1.4. ID-10 Sebagai *Radio Frequency Identification (RFID)* reader



Gambar 3.4 Rangkaian *Radio Frequency Identification* reader

Bagian ini adalah rangkaian RFID reader yang berfungsi untuk membaca tag rfid. Rangkaian ini terdiri dari sebuah ID10 sebagai reader (pembaca), dan beberapa komponen pendukung yaitu resistor 4K7 Ohm dan 330 Ohm, transistor BC547 sebagai pemicu dan sebuah LED. Resistor ini digunakan sebagai hambatan agar arus yang masuk sesuai dengan kebutuhan, baik pada transistor maupun pada LED. LED ini digunakan sebagai indikator bahwa ada sebuah tag rfid yang terdeteksi oleh reader. Jika ada maka LED akan menyala (berkedip). Pin 1 pada ID-10 berfungsi sebagai ground, pin 9 sebagai jalur data yang dikirim ke mikrokontroler, pin 3 dan pin 4 digunakan sebagai antenna, pin 2 sebagai reset, pin 9 sebagai sumber arus dan pin 6 dan pin 7 sebagai jalur data, namun hanya pin 7 yang digunakan.

3.1.5. Unit Display



Gambar 3.5 Rangkaian unit display dan *external memory*

Pada perangkat keras unit display ini tidak dapat dipisahkan dengan *external memory*. Port 0 memiliki dua fungsi yaitu sebagai jalur data dan sebagai jalur alamat (address). Jalur data digunakan untuk mengirim data menuju external memory dan LCD. Sedangkan jalur alamat digunakan sebagai digunakan untuk mengirim alamat byte rendah menuju ke eksternal memori. Untuk memisahkan jalur data dan jalur alamat byte rendah agar tidak tabrakan maka digunakan Latch (penahan) yaitu 74HC573. Data yang dikirim ke LCD adalah data 8 byte. Data tersebut akan menuju ke LCD setelah kaki port 1.7 bernilai 1 (high) yang terhubung dengan kaki 11 (C) yang digunakan sebagai *latch enable*. Data yang menuju ke LCD akan ditahan atau di-*latch* oleh 74HC573 karena data tidak dalam mode external

3.1.6 Memori eksternal

Aplikasi kunci pintu ini membutuhkan memory external karena memory internal dalam mikrokontroler tidak mencukupi kebutuhan. Sistem pengalamatan memory dari AT89C51 menggunakan sistem multiplex addressing (pengalamatan bergantian) yaitu memultiplex data dan low byte addressing (byte alamat rendah) menggunakan *Octal D latch* yaitu IC 74HC573. Dengan sistem pengalamatan ini port 2 dan port 0 dari mikrokontroler AT89C51 dapat melakukan pengalamatan untuk 64 Kbyte alamat memory.

Pembacaan data dari memory eksternal dilakukan dengan cara mikrokontroler akan mengeluarkan alamat dari data yang akan dibaca dengan melalui port 0 dan port 2, karena port 0 harus dimultiplex dengan alamat byte rendah, maka diperlukan IC Octal D Latch yaitu 74HC573 untuk menahan alamat

byte rendah tersebut di saat data muncul di jalur port) agar tidak terjadi bentrok antara data maupun alamat byte rendah. IC 74HC573 hanya akan mengirim masukan (D0...D7) ke bagian keluaran (Q0...Q7) setelah kaki nomor 11 (C) yang terhubung ke ALE/P berlogika 1.

Alamat byte tinggi langsung dikirim melalui port 2, dan sebagian dari alamat tersebut digunakan untuk mengalami memory eksternal dan yang sebagian lagi digunakan untuk mengaktifkan chip select (pemilihan keping) dari memory external melalui decoder.

Kemudian perintah untuk mengambil data dilakukan dengan mengeluarkan sinyal read pada pin RD. Setelah dikeluarkan maka data akan muncul pada jalur data bus (port 0) dan mikrokontroler akan langsung membaca data tersebut. Data tersebut mengalir tanpa mengganggu alamat byte rendah yang ada pada keluaran 74HC573 karena kaki 11 © yang terhubung dengan ALE/P telah berubah menjadi logika 0.

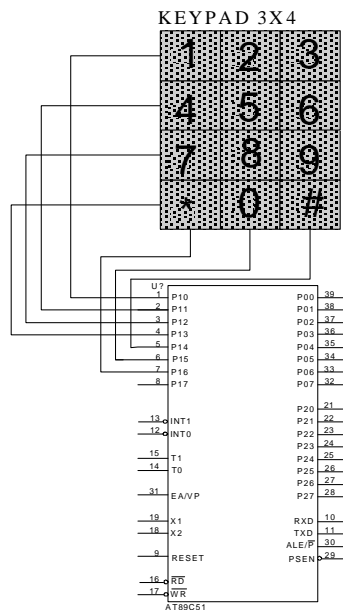
Penulisan data ke memory external dilakukan dengan cara Mikrokontroler mengeluarkan alamat tujuan pada port 0 dan port 2. alamat pada port 0 (low byte address) langsung di *latch* (tahan) ke output *octal D Latch* 74HC573 dengan menggunakan sinyal ALE (Address Latch Enable) yang terhubung pada input CLK dari 74HC573. Proses yang yang terjadi sama dengan yang dilakukan pada bagian pengiriman alamat diproses pembacaan data ataupun kode yang telah dijelaskan sebelumnya.

Kemudian mikrokontroler mengeluarkan data yang akan ditulis pada port 0. Walaupun Low Byte Address berasal dari port 0, tetapi oleh karena *Low*

Byte Address ini sudah ditahan oleh 74HC573 maka tidak terganggu dengan munculnya data pada port 0. Kemudian mikrokontroler akan memberikan perintah kepada memory untuk menuliskan data yang telah disiapkan dan alamat yang telah disiapkan pula dengan cara mengeluarkan sinyal write pada pin WR yang terhubung dengan pin WR dari *memory*.

3.1.7 Unit masukan

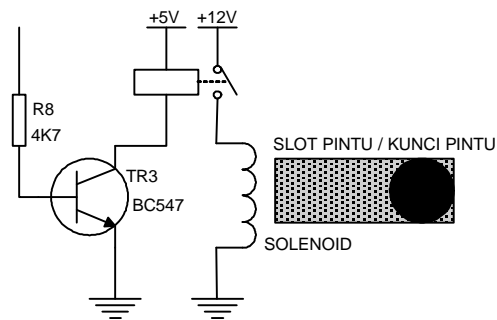
Aplikasi kunci pintu ini menggunakan keypad sebagai masukan (input). Keypad yang digunakan adalah keypad dengan 4 baris dan 3 kolom. Seperti terlihat pada Gambar 3.6. Keypad ini digunakan untuk memasukkan nomor digit yang tertulis dalam badan tag RFID. Nomor tersebut akan dicatat dalam memori mikrokontroler, dimana akan digunakan sebagai pembandingan antara nomor asli yang terdapat dalam tag RFID. Cara memasukkan nomor ini dengan menekan tombol # kemudian diikuti nomor yang dimaksud di depan.



Gambar 3.6 Pemasangan unit keypad pada mikrokontroler

Keypad digunakan sebagai media untuk memberikan instruksi kepada alat dengan melalui penekanan pada tombol-tombolnya. Sebelum sebuah keypad dapat digunakan harus dilakukan Scanning Keypad terlebih dahulu. Keypad yang digunakan di sini adalah keypad 4x3 (4 baris dan 3 kolom).

3.1.8 Relay magnetic

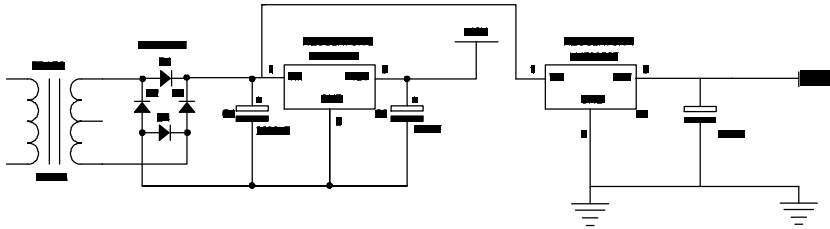


Gambar 3.7 Rangkaian output kunci pintu

Pada bagian rangkaian ini terdapat sebuah resistor 4K7 Ohm, sebagai penahan arus yang masuk pada transistor BC547 yang berperan sebagai pemicu tegangan yang masuk dalam relay dan yang akan memindahkan tegangan 5 V ke tegangan 12 V. Tegangan 12 V ini digunakan untuk menggerakkan solenoid yang berfungsi sebagai kunci pada pintu.

3.1.9 Catu Daya

Catu daya yang dibutuhkan kunci pintu rumah adalah 5V DC dan 12V DC. Daya 5V digunakan untuk mensuplai mikrokontroler, ID-10 dan LCD, sedangkan 12 Volt digunakan untuk menggerakkan solenoid (kunci). Skema rangkaian power supply seperti Gambar 3.7 di bawah.

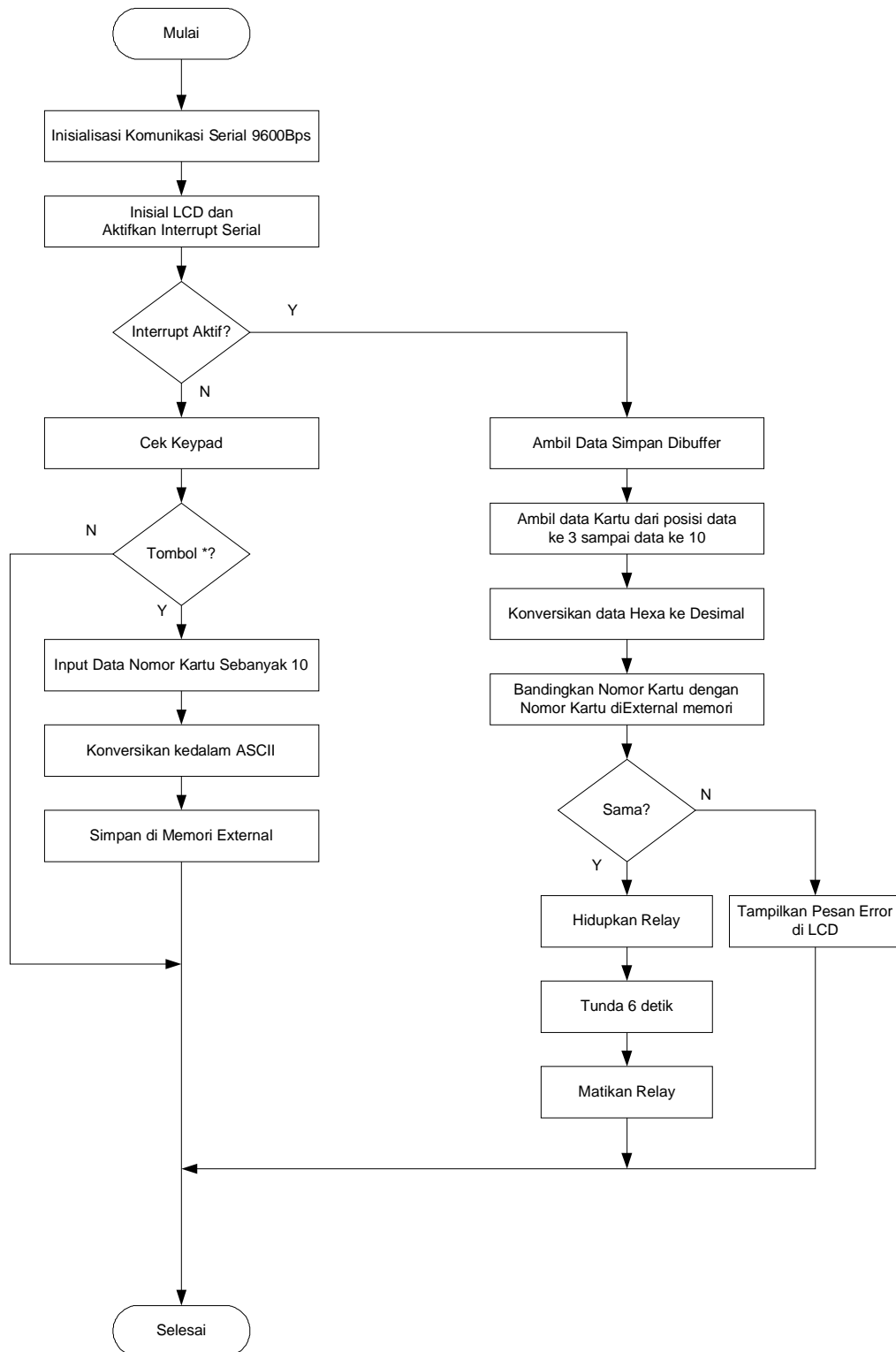


Gambar 3.7 Rangkaian catu daya kunci pintu

3.2. Perencanaan Perangkat Lunak

Pertama yang dilakukan oleh mikrokontroler adalah inisialisasi komunikasi serial dengan kecepatan 9600 byte per second. Kemudian mikrokontroler melakukan inisialisasi pada bagian *liquid crystal display* (LCD) dan mengaktifkan interup serial. Jika interrupt sudah diaktifkan maka mikrokontroler akan mengambil data simpanan yang ada dalam buffer, kemudian mikrokontroler mengambil data dalam kartu dari data ketiga sampai data kesepuluh (8 bit). Setelah itu mikrokontroler mengkonversikan data dari hexa ke data decimal. Setelah data dikonversikan ke bentuk decimal kemudian data nomor kartu yang ada terdapat di external memory dengan data yang terdapat dalam tag RFID. Jika sama maka mikrokontroler akan mengaktifkan relai selama 6 detik dan proses selesai. Namun jika tidak maka mikro akan mengeluarkan pesan error di dalam LCD.

Jika interrupt tidak aktif maka mikrokontroler akan mengecek keypad. Jika ada masukan (menekan tombol *) maka mikrokontroler akan mengambil input data nomor kartu sebanyak 10, kemudian mengkonversikan ke dalam bentuk ASCII dan menyimpan dalam memory external. Penjelasan di atas sesuai dengan Gambar 3.8 dibawah ini.



Gambar 3.8 Diagram alir program utama

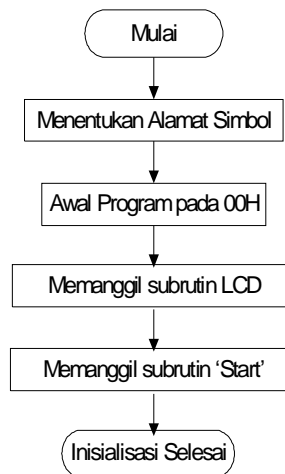
3.2.1 Inisialisasi Program

Pertama yang dilakukan oleh mikrokontroler adalah inisialisasi,

Jadi tahap yang dilakukan mikrokontroler pada saat inisialisasi adalah sebagai berikut ini :

1. Menentukan alamat untuk simbol-simbol.
2. Menetapkan alamat awal dalam penulisan program
3. Memanggil subrutin *LCD*
4. Memanggil subrutin *Start*
5. Inisialisasi selesai

Algoritma di atas dapat dijelaskan dalam diagram alir berikut ini :



Gambar 3.9 Diagram alir inisialisasi mikrokontroler

Sedangkan alur program inisialisasi mikrokontroler seperti yang telah dijelaskan di atas adalah sebagai berikut.

```

#include          "8051.H"

SELENOID         .EQU    P3.5
RS_LCD           .EQU    P3.3
E_LCD            .EQU    P3.1
RW_LCD           .EQU    P3.2
  
```

```

SLDATALCD      .EQU    P1.7
PORTLCD        .EQU    P0
;-----Control Data Magnetic-----
SWITCH         .EQU    P1
X1             .EQU    P1.4
X2             .EQU    P1.5
X3             .EQU    P1.6
Y1             .EQU    P1.0
Y2             .EQU    P1.1
Y3             .EQU    P1.2
Y4             .EQU    P1.3

                .ORG    $30
DATAKEY        .BLOCK  1
ADDRH          .BLOCK  1
ADDRL          .BLOCK  1
BUFEEPROM      .BLOCK  16          ;Buffer data Nomor dari
EEPROM

DATANOHEX1     .BLOCK  1
DATANOHEX2     .BLOCK  1
DATANOHEX3     .BLOCK  1
DATANOHEX4     .BLOCK  1
DATADECHASIL1  .BLOCK  1
DATADECHASIL2  .BLOCK  1
DATADECHASIL3  .BLOCK  1
DATADECHASIL4  .BLOCK  1
DATADECHASIL5  .BLOCK  1
DATAPENAMBAH1  .BLOCK  1
DATAPENAMBAH2  .BLOCK  1
DATAPENAMBAH3  .BLOCK  1
DATAPENAMBAH4  .BLOCK  1
DATAPENAMBAH5  .BLOCK  1
DATAS1         .BLOCK  1
DATAS2         .BLOCK  1
DATAS3         .BLOCK  1
DATAS4         .BLOCK  1
DATAS5         .BLOCK  1
DATAS6         .BLOCK  1
DATAS7         .BLOCK  1
DATAS8         .BLOCK  1
BUFDATANOMOR   .BLOCK  18
DATANOMORASCI1 .BLOCK  1
DATANOMORASCI2 .BLOCK  1
DATANOMORASCI3 .BLOCK  1
DATANOMORASCI4 .BLOCK  1
DATANOMORASCI5 .BLOCK  1
DATANOMORASCI6 .BLOCK  1
DATANOMORASCI7 .BLOCK  1
DATANOMORASCI8 .BLOCK  1
DATANOMORASCI9 .BLOCK  1
DATANOMORASCI10 .BLOCK  1

ADDREEPROM     .EQU    $2000

```

3.2.2 Start Program

Pada bagian ini yaitu start, hanya berjalan beberapa program diantaranya menampilkan TEXTAWAL1 yaitu “TUGAS AKHIR”, TEXTAWAL2 yaitu “TEKNIK ELEKTRO”, TEXTAWAL3 yaitu “RIKI ASTONO”, TEXTAWAL4 yaitu “NIM: 5350402014”. Setelah itu akan menjalankan Program 1 sampai dengan Program 4. Dalam satu isi program adalah menampilkan KATA1 yaitu “M O N I T O R I N G”, dan KATA2 yaitu “YOUR CARD”. Itu akan berjalan terus sampai kita menekan keypad, yang perintah programnya di PROGRAM3.

```

START:      MOV     SP, #20H
            MOV     PSW, #0
            MOV     R0, #BUFDATANOMOR
            CLR     SLDATALCD
            CLR     RW_LCD
            CLR     RS_LCD
            SETB    E_LCD
            CLR     SELENOID
            LCALL   DELAY
            LCALL   DELAY
            LCALL   INIT_LCD
            LCALL   DELAY
            LCALL   DELAY
            MOV     DPTR, #TEXTAWAL1
            LCALL   PROC_STRTOLCD
            LCALL   PROC_LFLCD
            MOV     DPTR, #TEXTAWAL2
            LCALL   PROC_STRTOLCD
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   PROC_HOMELCD
            MOV     DPTR, #TEXTAWAL3
            LCALL   PROC_STRTOLCD
            LCALL   PROC_LFLCD
            MOV     DPTR, #TEXTAWAL4
            LCALL   PROC_STRTOLCD
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW
            LCALL   DELAYSW

```

```

                LCALL  INITSERIAL
                SETB   ES
                SETB   EA

PROGRAM1 :     LCALL  PROC_HOMELCD
                MOV    DPTR, #KATA1
                LCALL  PROC_STRTOLCD
                LCALL  PROC_LFLCD
                MOV    DPTR, #KATA2
                LCALL  PROC_STRTOLCD

PROGRAM2 :     MOV    SP, # $20
                LCALL  SCANNINGKEYPAD
                MOV    A, DATAKEY
                CJNE   A, #0, PROGRAM3
                LJMP   PROGRAM1

PROGRAM3 :     CJNE   A, # $0B, PROGRAM4
                LJMP   SETTING

PROGRAM4 :     LJMP   PROGRAM1

```

3.2.3 Pengambilan Data Serial

Pengambilan data serial ini merupakan bagian yang digunakan untuk mengambil data serial dari ID-10 (RFID reader) menuju ke mikrokontroler. Data serial yang telah diambil ditampung dalam buffer.

```

SERINT :       JBC    RI, GETDATASERIAL
                RETI

GETDATASERIAL :
                PUSH  ACC

                MOV   A, SBUF
                CLR   RI
                CJNE  A, # $03, ISIKEBUFFER
                LCALL PROSES
                MOV   R0, #BUFDATANOMOR
                POP   ACC
                RETI

ISIKEBUFFER :  MOV   @R0, A
                INC   R0

                POP   ACC
                RETI

PROSES :       MOV   R0, #BUFDATANOMOR
                INC   R0
                INC   R0

```

```

INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS1,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS2,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS3,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS4,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS5,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS6,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS7,A
INC     R0
MOV     A,@R0
LCALL  CEKA_F
ANL     A,#$0F
MOV     DATAS8,A

```

3.2.4 Penggabungan Data

Data yang telah diambil tadi dan disimpan dalam buffer digabungkan menjadi satu bagian.

```

MOV     A,DATAS1
SWAP   A
ANL     A,#$F0
MOV     DATANOHEX1,A
MOV     A,DATAS2
ANL     A,#$0F
ORL     A,DATANOHEX1
MOV     DATANOHEX1,A

```

```
MOV     A,DATAS3
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX2,A
MOV    A,DATAS4
ANL    A,#$0F
ORL    A,DATANOHEX2
MOV    DATANOHEX2,A

MOV    A,DATAS5
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX3,A
MOV    A,DATAS6
ANL    A,#$0F
ORL    A,DATANOHEX3
MOV    DATANOHEX3,A

MOV    A,DATAS7
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX4,A
MOV    A,DATAS8
ANL    A,#$0F
ORL    A,DATANOHEX4
MOV    DATANOHEX4,A

LCALL  HEXTODES
LCALL  KOSONGKAN

MOV    A,DATADECHASIL1
LCALL  ANDF0
MOV    DATANOMORASCII1,A
MOV    A,DATADECHASIL1
LCALL  AND0F
MOV    DATANOMORASCII2,A

MOV    A,DATADECHASIL2
LCALL  ANDF0
MOV    DATANOMORASCII3,A
MOV    A,DATADECHASIL2
LCALL  AND0F
MOV    DATANOMORASCII4,A

MOV    A,DATADECHASIL3
LCALL  ANDF0
MOV    DATANOMORASCII5,A
MOV    A,DATADECHASIL3
LCALL  AND0F
MOV    DATANOMORASCII6,A

MOV    A,DATADECHASIL4
LCALL  ANDF0
MOV    DATANOMORASCII7,A
MOV    A,DATADECHASIL4
LCALL  AND0F
```

```

MOV     DATANOMORASCII8 ,A

MOV     A ,DATADECHASIL5
LCALL  ANDF0
MOV     DATANOMORASCII9 ,A
MOV     A ,DATADECHASIL5
LCALL  AND0F
MOV     DATANOMORASCII10 ,A

```

3.2.5 Membandingkan Data

Data yang telah digabungkan dibandingkan dengan data yang telah tertulis dalam memori. Jika data benar maka akan menjalankan KATA4 yaitu “==KUNCI DIBUKA==” dan KATA5 “[_____]” dan membukan kunci pintu selama 6 detik, Jika data salah maka LCD akan menampilkan KATAERROR1 dan KATAERROR2 yaitu “M A A F ! ! ! “ , “C A R D T D K D I K E N A L” dan mengosongkan seluru data nomor ascii.

```

LCALL  BACADATAEEPROM
MOV     R0 ,#DATANOMORASCII1
MOV     R1 ,#BUFEEPROM
MOV     R2 ,#10
ULANGCEK :
CLR     A
MOV     B ,@R0
MOV     A ,@R1
CJNE   A ,B ,NOMORTIDAKCOCOK
INC     R0
INC     R1
DJNZ   R2 ,ULANGCEK
DATABENAR :
LCALL  PROC_HOME_LCD
MOV     DPTR ,#KATA4
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD
MOV     DPTR ,#KATA5
LCALL  PROC_STRTOLCD
UNTUK_SELENOID SETB   SELENOID           ; AKTIFKAN RELAY
MOV     R3 ,#20
LCALL  DELAYLONG
LCALL  PROC_HOME_LCD
MOV     DPTR ,#KATA6
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD

```



```

MOV     DPTR, #KATA5
LCALL  PROC_STRTOLCD
LCALL  DELAYSW
CLR     SELENOID                ;MATIKAN KEMBALI
RELAY SELENOID
LCALL  DELAY
RET

NOMORTIDAKCOCOK:
LCALL  PROC_HOMELCD
MOV     DPTR, #KATAERROR1
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD
MOV     DPTR, #KATAERROR2
LCALL  PROC_STRTOLCD
MOV     R3, #2
LCALL  DELAYLONG
RET

KOSONGKAN:
MOV     DATANOMORASCII1, #0
MOV     DATANOMORASCII2, #0
MOV     DATANOMORASCII3, #0
MOV     DATANOMORASCII4, #0
MOV     DATANOMORASCII5, #0
MOV     DATANOMORASCII6, #0
MOV     DATANOMORASCII7, #0
MOV     DATANOMORASCII8, #0
MOV     DATANOMORASCII9, #0
MOV     DATANOMORASCII10, #0
RET

```

3.2.6 Setting Nomor Baru

Setting nomer baru digunakan apabila kita ingin mengganti kunci pintu

(tag rfid) denga kunci (tag rfid) yang lain.

```

INITSERIAL:
MOV     TMOD, #20H
MOV     TCON, #41H
MOV     TH1, #0FDH
MOV     SCON, #50H
RET

SETTING:
LCALL  BACADATAEEPROM
LCALL  PROC_CLEARLCD
MOV     DPTR, #TEXTSETTING1
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD
LCALL  DELAYSW
LCALL  DELAYSW
LCALL  DELAYSW
LCALL  DELAYSW
LCALL  PROC_CLEARLCD
MOV     DPTR, #TEXTSETTING1
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD

```

```

MOV      R4, #10
MOV      DPTR, #ADDREEPROM

SETLOOP :
        LCALL SCANNINGKEYPAD
        MOV   A, DATAKEY
        CJNE A, #0, MASUKKANNOMOR
        LJMP SETLOOP
MASUKKANNOMOR : CJNE A, #$0C, MASUKKANNOMOR1
                LJMP PROGRAM1
MASUKKANNOMOR1 : CJNE A, #$0A, MASUKKANNOMOR2
                MOV   A, #$0
MASUKKANNOMOR2 :
        ADD   A, #$30
        MOVX  @DPTR, A
        MOV   ADDRH, DPH
        MOV   ADDR L, DPL

        LCALL KEYPADTOLCD
        MOV   DPH, ADDRH
        MOV   DPL, ADDR L
        INC   DPTR
        LCALL DELAY
        DJNZ  R4, SETLOOP
        LCALL PROC_CLEARLCD
        LCALL BACADATAAEEPROM
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LJMP PROGRAM1

BACADATAAEEPROM :
        MOV   R4, #10
        MOV   DPTR, #ADDREEPROM
        MOV   R0, #BUFEEPROM

BACAEEMPLAGI :
        MOVX  A, @DPTR
        MOV   @R0, A
        INC   R0
        INC   DPTR
        DJNZ  R4, BACAEEMPLAGI
        RET

```

3.2.7 Scaning Keypad

Keypad digunakan sebagai media untuk memberikan instruksi kepada alat dengan melalui penekanan pada tombol-tombolnya. Sebelum sebuah keypad dapat digunakan harus dilakukan Scanning Keypad terlebih dahulu. Keypad yang

digunakan di sini adalah keypad 4x3 (4 baris dan 3 kolom). Scanning dilakukan untuk mengenalkan tombol-tombol yang ada, listing program dari scanning keypad ini adalah sebagai berikut

```

SCANNINGKEYPAD:
KOLOM1:      MOV      DATAKEY, #0
              CLR      X1
              SETB     X2
              SETB     X3
              MOV      A, SWITCH
              ANL      A, #0FH
CTOMBOL1:    CJNE     A, #0EH, CTOMBOL4
              MOV      DATAKEY, #01H
              LJMPL   TOLCD
CTOMBOL4:    CJNE     A, #0DH, CTOMBOL7
              MOV      DATAKEY, #04H
              LJMPL   TOLCD
CTOMBOL7:    CJNE     A, #0BH, CTOMBOLB
              MOV      DATAKEY, #07H
              LJMPL   TOLCD
CTOMBOLB:    CJNE     A, #07H, KOLOM2
              MOV      DATAKEY, #0BH
              LJMPL   TOLCD

KOLOM2:      SETB     X1
              CLR      X2
              SETB     X3
              MOV      A, SWITCH
              ANL      A, #0FH
CTOMBOL2:    CJNE     A, #0EH, CTOMBOL5
              MOV      DATAKEY, #02H
              LJMPL   TOLCD
CTOMBOL5:    CJNE     A, #0DH, CTOMBOL8
              MOV      DATAKEY, #05H
              LJMPL   TOLCD
CTOMBOL8:    CJNE     A, #0BH, CTOMBOL0
              MOV      DATAKEY, #08H
              LJMPL   TOLCD
CTOMBOL0:    CJNE     A, #07H, KOLOM3
              MOV      DATAKEY, #0AH
              LJMPL   TOLCD

KOLOM3:      SETB     X1
              SETB     X2
              CLR      X3
              MOV      A, SWITCH
              ANL      A, #0FH
CTOMBOL3:    CJNE     A, #0EH, CTOMBOL6
              MOV      DATAKEY, #03H
              LJMPL   TOLCD
CTOMBOL6:    CJNE     A, #0DH, CTOMBOL9
              MOV      DATAKEY, #06H
              LJMPL   TOLCD

```

```

CTOMBOL9 : CJNE    A, #0BH, CTOMBOLP
            MOV     DATAKEY, #09H
            LJMP   TOLCD
CTOMBOLP : CJNE    A, #07H, KEYPADRET
            MOV     DATAKEY, #0CH
TOLCD :
            LCALL  DELAYSW
            MOV     A, DATAKEY
KEYPADRET :
            RET

KEYPADTOLCD :
            MOV     DPTR, #KEY
            MOV     A, DATAKEY
            MOVC   A, @A+DPTR
            LCALL  WRITE_DATA_LCD
            RET

```

3.2.8 Sub rutin tampilan LCD

Penulisan karakter pada LCD dilakukan oleh mikrokontroler dengan mengirimkan kode ASCII dari karakter yang akan ditulis dan mengakses CGROM dari LCD karena disanalah LCD menyimpan karakter-karakter yang sudah ditetapkan. Penulisan karakter pada LCD memiliki pola tertentu seperti halnya pemberian instruksi kerja modul LCD., yaitu sesuai dengan algoritma program berikut ini.

1. Mulai pengiriman karakter pada LCD
2. Memberikan logika 1 (high) pada pin RD (Port 3.3) pada LCD
3. Memberikan logika 0 (low) pada pin R/W (Port 3.2) pada LCD
4. Mengirimkan nibble atas (D7 – D4) data karakter dalam bentuk ASCII pada mikrokontroler.
5. Mengirimkan nibble bawah (D3 – D0) data karakter dalam bentuk ASCII pada mikrokontroler.
6. Mengaktifkan pin enable dengan memberi logika 1 pada Port 3.1
7. Mengpasifkan pin enable dengan memberi logika 0 pada Port 3.1

8. Pengiriman data karakter ke LCD selesai dilakukan.

Listing program pengiriman dan penampilan data ke LCD adalah sebagai dapat diuraikan sebagai berikut.

```

PROC_HOMELCD:
    MOV     A, #02H
    LCALL  WRITE_CTRLLCD
    RET

PROC_LFLCD:
    MOV     A, #0C0H
    LCALL  WRITE_CTRLLCD
    RET

PROC_CLEARLCD:
    MOV     A, #01H
    LCALL  WRITE_CTRLLCD
    RET

WRITE_DATA_LCD:
    MOV     PORTLCD, A
    SETB   SLDATA_LCD
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    CLR    SLDATA_LCD
    SETB   RS_LCD
    CLR    E_LCD
    LCALL  DELAY_LCD

    SETB   RS_LCD
    SETB   E_LCD
    LCALL  DELAY_LCD

    SETB   RS_LCD
    CLR    E_LCD
    NOP
    CLR    RS_LCD
    CLR    E_LCD
    LCALL  DELAY_LCD
    RET

WRITE_CTRLLCD:
    MOV     PORTLCD, A
    SETB   SLDATA_LCD
    NOP
    NOP

```

```

NOP
NOP
NOP
NOP
NOP
NOP
CLR          SLDATA_LCD

CLR          RS_LCD
CLR          E_LCD
LCALL       DELAY_LCD
CLR          RS_LCD
SETB        E_LCD
LCALL       DELAY_LCD
CLR          RS_LCD
CLR          E_LCD
NOP
CLR          RS_LCD
CLR          E_LCD
LCALL       DELAY_LCD

RET

;-----
;          Routine Pengiriman String ke Display LCD
;-----
PROC_STRTOLCD:
CLR          A
MOVC        A,@A+DPTR
CJNE        A,#00H,STRTOLCD
RET

STRTOLCD:
LCALL       WRITE_DATA_LCD
INC         DPTR
SJMP        PROC_STRTOLCD

;-----
;          INISIAL LCD
;-----
INIT_LCD
MOV          A,#38H
LCALL       WRITE_CTRL_LCD
MOV          A,#0EH
LCALL       WRITE_CTRL_LCD
MOV          A,#0CH
LCALL       WRITE_CTRL_LCD
MOV          A,#06H
LCALL       WRITE_CTRL_LCD
MOV          A,#01H
LCALL       WRITE_CTRL_LCD
RET

DELAY_LCD:
MOV          R7,#04H

```

```

D_LCD1:      MOV      R6,#3FH          ;4F
D_LCD2:      DJNZ    R6,D_LCD2
              DJNZ    R7,D_LCD1
              RET

DELAWSW:
              MOV     R5,#02H
DELAWSW0:    MOV     R6,#0FFH
DELAWSW1:    MOV     R7,#0FFH
DELAWSW2:    DJNZ    R7,DELAWSW2
              DJNZ    R6,DELAWSW1
              DJNZ    R5,DELAWSW0
              RET

DELAYLONG:
              LCALL   DELAWSW
              DJNZ    R3,DELAYLONG
              RET

DELAY:
              MOV     R7,#$ff
DLD1:        MOV     R6,#$FF
DLD2:        DJNZ    R6,DLD2
              DJNZ    R7,DLD1
              RET

KEY:         .BYTE           " 1234567890*#"
TEXTAWAL1:   .BYTE           " TUGAS AKHIR ",0
TEXTAWAL2:   .BYTE           " TEKNIK ELEKTRO ",0
TEXTAWAL3:   .BYTE           " RIKI ASTONO ",0
TEXTAWAL4:   .BYTE           " NIM:5350402014",0

KATA1:       .BYTE           " MONITORING ",0
KATA2:       .BYTE           "Y O U R C A R D",0
KATA3:       .BYTE           " KARTU ANDA ",0
KATA4:       .BYTE           "==KUNCI DIBUKA==",0
KATA5:       .BYTE           "[ _____ ]",0
KATA6:       .BYTE           "=PINTU DIKUNCI=",0
KATAERROR1:  .BYTE           " M A A F !!! ",0
KATAERROR2:  .BYTE           "CARD TDK DIKENAL",0
TEXTSETTING1: .BYTE           "SET NOMOR KARTU ",0

CODEPARITY:  .BYTE
$80,$00,$00,$80,$00,$80,$80,$00,$00,$80,$80,$00,$80,$00,$00,$80

```

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Pengujian Alat

4.1.1 Pengujian Perangkat Keras

Pengujian ini dilakukan untuk mengetahui apakah perangkat yang telah direncanakan bekerja dengan baik. Pengujian dilakukan dengan berbagai cara sesuai dengan kebutuhan. Ada beberapa bagian blok rangkaian yang perlu diuji secara khusus antara lain sebagai berikut:

4.1.1.1 Pengujian Minimal Sistem Mikrokontroler AT89C51

Pengujian minimum mikrokontroler dilakukan untuk mengetahui apakah mikrokontroler dapat bekerja sesuai dengan fungsi semestinya. Pengujian dilakukan dengan mendownloadkan sebuah logika dari komputer ke AT89C51 dan diujicobakan di system minimal tersebut. Setiap port dihubungkan dengan LED untuk mengetahui apakah masing-masing kaki berfungsi dengan baik. Program yang didownload misalnya untuk membuat led ping pong sebagai berikut:

```

Orh    0h
Mov    A, #0FEH                ;simpan data ke akumulator
                                ;port 1 menyala duluan

MULAI:
MOV    P1,A                    ;kirim data ke port 1
ACALL  DELAY                   ;tunda sebentar
RL     A                       ; putar isi akumulator ke kiri 1 bit
CJNE  A,#7FH, MULAI           ; apakah A=01111111B?
                                ; Tidak, ulangi lagi
                                ; Ya, lanjutkan ke proses berikut...
MULAI1:
        MOV    P1,A            ; kirim data ke port 1
        ACALL  DELAY           ; tunda sebentar
        RR     A                ; putar isi akumulator ke kanan 1
bit
        CJNE  A,#0FEH, MULAI1  ; apakah A=11111110B?
                                ; Tidak, ulangi lagi

```

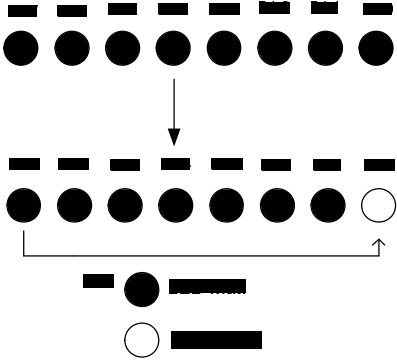


```

      SJMP MULAI           ; Ya, lanjutkan ke proses paling
awal..
      DELAY
      MOV R0,#0           ; isi register R0 dengan 0 (256x
ulang)
      DELAY1: MOV R1,#0   ; isi register R1 dengan 0 (256x
ulang)
      Djnz R1,$
      Djnz R0, DELAY1
      Ret
      End

```

Program ini digunakan untuk menguji apakah mikrokontroler bekerja sesuai dengan perintah yang dirancang. Outputnya berupa LED yang terhubung pada kaki-kaki mikrokontroler seperti gambar 4.1



Gambar 4.1 Pengujian kaki mikrokontroler pada port 1

4.1.1.2.Pengujian RFID reader

Pengujian pada reader pertama kali dilakukan dengan menghubungkan jalur salah satu jalur data (D0 atau D1) dengan serial port komputer (DB9). Karena data yang dikirim RFID reader merupakan data serial maka dapat langsung dibaca oleh komputer.

Pengukuran jarak baca ini untuk mengetahui jarak yang paling efektif kemampuan baca RDIF reader ID-10. Setiap jenis RFID reader mempunyai kempuan baca yang berbeda. Kemampuan daya baca ini tergantung pada catu daya dan frekuensi yang digunakan oleh RFID. Untuk tag RFID pasif biasanya

mempunyai daya baca yang lebih pendek dari pada RFID aktif. Sampai saat ini teknologi RFID sudah mampu membaca *tag* RFID dari jarak ratusan meter. Hasil pengukuran kemampuan baca RFID reader ID-10 adalah sebagai berikut:

a. Pengujian Tanpa Halangan

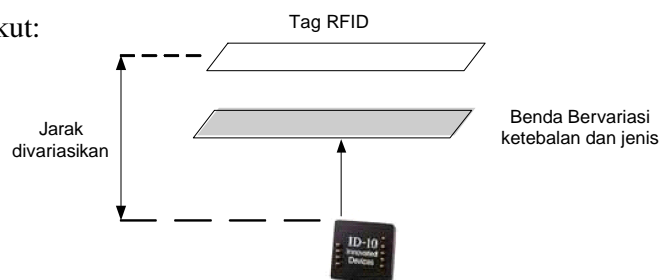
Pengujian ini dilakukan dengan cara menghitung jarak yang mampu dibaca oleh RFID reader tanpa ada penghalang. Pengujian ini digunakan untuk menentukan jarak yang efektif terhadap penggunaan *tag* RFID.

Tabel 4.1 Pengujian RFID tanpa penghalang

Posisi <i>Tag</i> RFID	Kemampuan Baca
Posisi <i>tag</i> RFID horisontal	6 cm
Posisi <i>tag</i> miring 45 derajat	3,8 cm
Posisi <i>tag</i> RFID vertical	1 cm

b. Pengujian dengan Halangan

Pengujian dengan halangan ini dilakukan dengan berbagai percobaan. Penghalang yang digunakan antara lain dengan kertas, tripleks, kaca dan plat besi. Pengujian ini dilakukan untuk menentukan kemampuan RFID reader dalam membaca *tag* dengan penghalang tertentu. Penghalang ini dapat diibaratkan sebagai wadah atau casing yang paling cocok digunakan untuk perangkat kunci pintu. Hasil pengukuran kemampuan baca RFID reader dengan penghalang adalah sebagai berikut:



Gambar 4.2 Pengujian RFID

Tabel 4.2 Pengujian RFID dengan penghalang

No	Jenis Penghalang	Jarak Baca
1	Kertas 2,6 cm	4,3 cm
2	Kayu 3,7 cm	6 cm
3	Kaca 1,9cm	5,8 cm
4	Plat Besi 1 mm	Tidak terdeteksi
5	Karet 1,3 cm	5,8 cm

4.1.1.3. Pengujian LCD

Pengujian LCD ini untuk mengetahui apakah LCD yang dipakai rusak atau bisa dipakai semestinya. LCD memiliki 14 kaki yang terdiri dari 8 pin jalur data, 2 pin power suplai, 1 pin untuk mengatur kontras, dan 3 pin control. Pengujian pertama yang dilakukan dengan memberi tegangan pada kaki power supply. Mekan LCD akan menyala, namun demikian tidak berarti LCD akan bekerja dengan baik jika dapat menyala.

4.1.1.4. Pengujian Keypad

Pengujian dilakukan secara langsung apakah antara masing-masing tombol yang ditekan memberikan logika seperti yang seharusnya. Logika-logika ini yang akan digunakan sebagai masukan pada port 1.0 sampai port 1.6. Pengujian dilakukan dengan Ohmmeter untuk mengetahui adanya resistansi pada kombinasi tombol keypad 3X4. Selain itu pengujian keypad dilakukan oleh mikrokontroler dengan listing program sebagai berikut:

```
SCANNINGKEYPAD:
KOLOM1:      MOV      DATAKEY, #0
              CLR      X1
```

```

                SETB    X2
                SETB    X3
                MOV     A, SWITCH
                ANL     A, #0FH
CTOMBOL1 :    CJNE    A, #0EH, CTOMBOL4
                MOV     DATAKEY, #01H
                LJMPL  TOLCD
CTOMBOL4 :    CJNE    A, #0DH, CTOMBOL7
                MOV     DATAKEY, #04H
                LJMPL  TOLCD
CTOMBOL7 :    CJNE    A, #0BH, CTOMBOLB
                MOV     DATAKEY, #07H
                LJMPL  TOLCD
CTOMBOLB :    CJNE    A, #07H, KOLOM2
                MOV     DATAKEY, #0BH
                LJMPL  TOLCD

KOLOM2 :     SETB    X1
                CLR     X2
                SETB    X3
                MOV     A, SWITCH
                ANL     A, #0FH
CTOMBOL2 :    CJNE    A, #0EH, CTOMBOL5
                MOV     DATAKEY, #02H
                LJMPL  TOLCD
CTOMBOL5 :    CJNE    A, #0DH, CTOMBOL8
                MOV     DATAKEY, #05H
                LJMPL  TOLCD
CTOMBOL8 :    CJNE    A, #0BH, CTOMBOL0
                MOV     DATAKEY, #08H
                LJMPL  TOLCD
CTOMBOL0 :    CJNE    A, #07H, KOLOM3
                MOV     DATAKEY, #0AH
                LJMPL  TOLCD
KOLOM3 :     SETB    X1
                SETB    X2
                CLR     X3
                MOV     A, SWITCH
                ANL     A, #0FH
CTOMBOL3 :    CJNE    A, #0EH, CTOMBOL6
                MOV     DATAKEY, #03H
                LJMPL  TOLCD
CTOMBOL6 :    CJNE    A, #0DH, CTOMBOL9
                MOV     DATAKEY, #06H
                LJMPL  TOLCD
CTOMBOL9 :    CJNE    A, #0BH, CTOMBOLP
                MOV     DATAKEY, #09H
                LJMPL  TOLCD
CTOMBOLP :    CJNE    A, #07H, KEYPADRET
                MOV     DATAKEY, #0CH

```

4.1.1.5. Pengujian Relai Solenoid

Pengujian ini dilakukan untuk menyesuaikan lama seseorang membuka sampai masuk ruangan. Waktu yang dibutuhkan seseorang untuk membuka pintu

sampai masuk ruangan adalah 6 second. Waktu ini yang digunakan sebagai acuan untuk menentukan lamanya kunci pintu dalam kondisi membuka. Perintah untuk mengaktifkan dan mematikan relay pada mikrokontroler dan menampilkan pesan di LCD adalah sebagai berikut:

```

                SETB      SELENOID          ;AKTIFKAN RELAY UNTUK
SELENOID
                MOV      R3 ,#20
                LCALL   DELAYLONG

                LCALL   PROC_HOMELCD
                MOV     DPTR ,#KATA6
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
                MOV     DPTR ,#KATA5
                LCALL   PROC_STRTOLCD
                LCALL   DELAYSW
                CLR      SELENOID          ;MATIKAN KEMBALI RELAY
SELENOID
                LCALL   DELAY
                RET

```

Pengaktifan relay yang terhubung dengan solenoid dilakukan dengan mengaktifkan timer 2, yaitu dengan mengaktifkan pin 15 atau port 3.5 selama t (waktu) tertentu. Pin 15 tersebut ternyata aktif selama 6,19 detik.

Pengujian relai juga dilakukan secara terpisah yaitu dengan memberikan tegangan 5V DC pada pemicunya, sehingga akan menggerakkan relay on/off.

4.1.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak ini dilakukan pada beberapa sub rutin program. Tujuan pengujian adalah untuk memastikan sub rutin bekerja sesuai dengan yang direncanakan. Pengujian ini dilakukan anatara lain pada sub rutin LCD, keypad, get data serial (membaca *tag* RFID), dan sub rutin pengaktifan relai.

Untuk sub rutin yang menerima input dan meneruskan output dapat dicobakan secara langsung pada mikrokontroler sehingga hasilnya dapat dilihat. Misalnya sub rutin pada LCD maka akan dapat dilihat secara langsung pada LCD.

Sedang untuk sub rutin yang tidak menerima input dan output misalnya pada sub rutin untuk mengubah angka kebentuk heksa atau yang lain tidak dapat dilihat secara langsung.

4.1.2.1 Setting Nomor ID *Tag* RFID

Setting ini nomor ID RFID dilakukan jika akan mengganti kunci pintu (*tag* RFID) dengan kunci (*tag* RFID) yang lain. Penggantian dilakukan jika *tag* RFID yang digunakan sebagai kunci hilang, atau untuk kepentingan peningkatan keamanan ruangan, maka kunci dapat diganti setiap rentang waktu tertentu. Penggantian dilakukan dengan memasukkan ID yang tertulis di sisi *tag*. Dengan menekan tombol * (bintang) terlebih dahulu, diikuti dengan 10 digit ID yang tertulis di *tag* RFID. Sub rutin yang mengatur setting nomer adalah sebagai berikut:

```
SETTING:          LCALL    BACADATAEEPROM
                  LCALL    PROC_CLEARLCD
                  MOV      DPTR, #TEXTSETTING1
                  LCALL    PROC_STRTOLCD
                  LCALL    PROC_LFLCD
                  LCALL    DELAYSW
                  LCALL    DELAYSW
                  LCALL    DELAYSW
                  LCALL    DELAYSW
                  LCALL    PROC_CLEARLCD
                  MOV      DPTR, #TEXTSETTING1
                  LCALL    PROC_STRTOLCD
                  LCALL    PROC_LFLCD

                  MOV      R4, #10
                  MOV      DPTR, #ADDREEPROM
```

4.2 Pembahasan Alat

4.2.1 Penggunaan port pada mikrokontroler

Port 0 digunakan sebagai jalur data yang menuju ke LCD dan external , serta pengambilan data dari external memory. Selain itu port ini juga digunakan sebagai low byte address sebagai jalur alamat yang menuju ke external memory.

Dalam praktiknya port ini membutuhkan dua buah *latch* (penahan) yaitu 74HC573. *Latch* yang pertama digunakan sebagai penahan *low byte address*, yang terhubung dengan *external memory*. *Latch* yang kedua digunakan untuk menahan data yang dikirim ke LCD karena LCD tidak dalam *mode external*. Untuk mengaktifkan *latch* ini dengan memberikan logic 0 pada kaki 11 (C), demikian juga dengan *latch* yang terdapat pada jalur data yang menuju LCD.

Port 1.0 sampai 1.6 digunakan untuk unit input yaitu keypad, sedangkan port 1.7 digunakan untuk mengaktifkan *latch* yang terdapat pada jalur data yang menuju LCD.

Port 2.0 sampai port 2.4 digunakan untuk mengirimkan address byte tinggi yang dapat secara langsung ke memori eksternal. Untuk port 2.5 sampai port 2.7 digunakan untuk mengaktifkan chip select dengan decoder 74HC138.

Untuk mengaktifkan relai yang menuju solenoid digunakan pin T1 yaitu pin timer. Pin ini akan aktif selama 6 detik. Untuk membaca serial data yang terdapat dalam RFID reader digunakan pin RXD yaitu pin yang sering digunakan untuk menerima data.

4.2.2 Bagian *Radio Frequency Identification (RFID)*

RFID reader ID-10 memancarkan frekuensi gelombang radio, kemudian *tag* RFID menerima dengan internal antena. Pada saat itu terjadi interferensi gelombang yang menyebabkan adanya medan magnet. Medan magnet tersebut yang digunakan sebagai catu daya *tag* RFID untuk mengirimkan data yang berupa ID yang unik ke RFID reader. RFID reader mengirim data tersebut secara serial melalui pin 6 dan pin 7 yang merupakan kaki D0 dan D1. Data serial dikirim ke

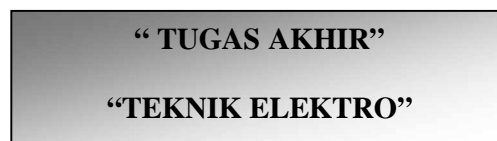
mikrokontroler melalui kaki RXD. RFID reader mampu membaca isi *tag* dengan halangan yang telah diujicobakan, namun reader mengalami kesulitan ketika mendapat halangan berupa logam (besi). Karena prinsip kerja berdasarkan medan magnet maka kemampuan reader akan menurun atau bahkan tidak bisa membaca pada penghalang dari logam (besi). Hal ini dikarenakan tidak terjadi medan magnet pada saat reader didekatkan, sehingga tidak tersedia daya pada *tag* RFID.

4.2.3 Bagian Relai Solenoid (kunci)

Kunci pintu ini menggunakan solenoid yang membutuhkan tegangan 12 V DC yang disediakan oleh relai 5VDC. Solenoid akan bekerja jika relai 5V menghubungkan dengan catu daya 12 V DC. Dia akan bekerja selama 6 detik untuk membuka kunci pintu.

4.2.4 Bagian *Liquid Crystal Display* (LCD)

Bagian ini digunakan untuk menampilkan keterangan yang berhubungan dengan program yang sedang dijalankan. Misalnya pada saat pertama kali kunci dihidupkan LCD akan menampilkan kata



“ TUGAS AKHIR”
“TEKNIK ELEKTRO”



“RIKI ASTONO”
“NIM:5350402014”

Setelah teks ditampilkan LCD akan menampilkan pesan seperti berikut:



“MONITORING”
“Y O U R C A R D”

Pesan ini ditampilkan pada saat sebelum *tag* RFID didekatkan.

Setelah *tag* didekatkan maka akan ada dua pilihan yaitu :



Jika *tag* yang dipakai benar maka akan muncul teks seperti diatas secara bersamaan dengan membukanya kunci (solenoid), setelah 6 detik pintu akan terkunci kembali dengan tampilan LCD sebagai berikut:



Jika *tag* yang digunakan tidak sesuai maka yang akan muncul di LCD adalah sebagai berikut



4.2.5 Bagian keypad

Bagian ini sebenarnya hanya akan aktif pada saat interrupt tidak aktif. Saat interrupt tidak aktif maka mikrokontroler akan mengecek keypad. Jika menekan tombol bintang (*) maka mikrokontroler akan menerima input data ID 10 digit kemudian mengkonversikandalam bentuk ascii dan menyimpan dalam eksternal memory. Jika tidak menekan tombol bintang (*) maka akan selesai.

4.3 Perbandingan teknologi RFID dengan Kartu Magnetik

Teknologi *magnetic card* sudah ada lebih dulu dibanding dengan teknologi *Radio Frequency Identification* (RFID). Teknologi *magnetic card* menggunakan media pita magnetik sebagai media penyimpanan, sedangkan *Radio Frequency Identification* sudah menggunakan chip tersendiri. Untuk lebih jelasnya dalam Tabel 4.2 dibawah ini:

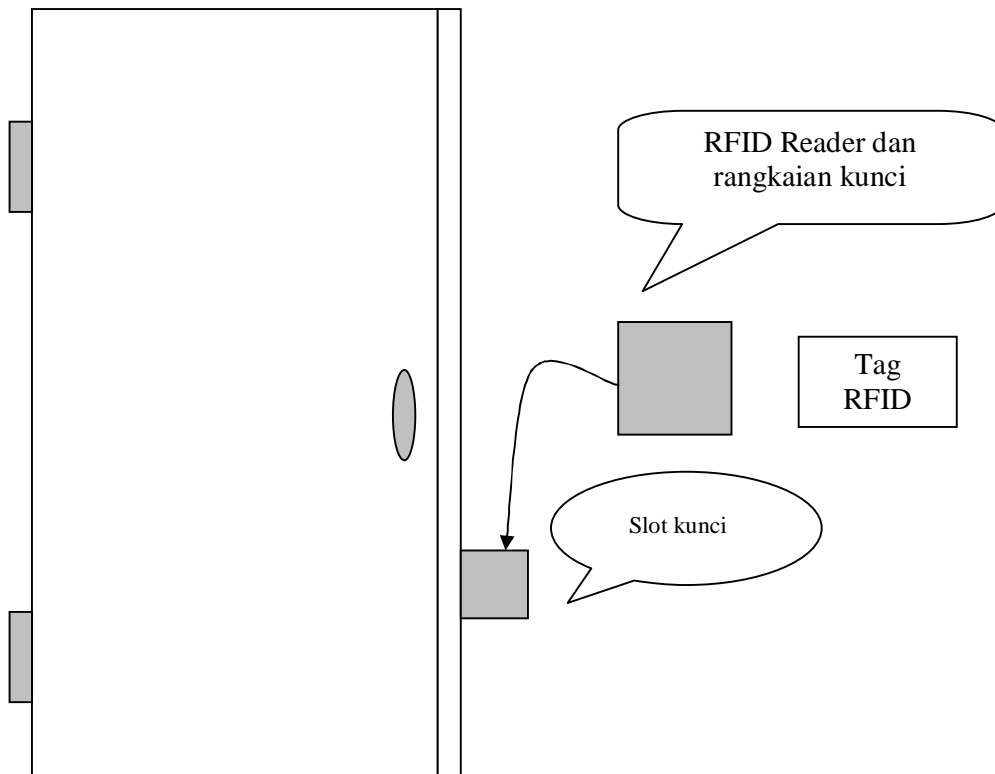
Tabel 4.2 Perbandingan antara RFID dengan *magnetic Card*.

No	Keterangan	RFID	Magnetic Card
1.	Memory	Dalam single chip sehingga dapat menyimpan data lebih banyak	Dalam pita magnetic, data yang disimpan terbatas
2.	Sifat Reader	-Mampu membaca dalam jarak tertentu (dari 8 cm sampai 100). -tidak mengenal <i>Head Life</i> , usia pakai tak terbatas.	-Hanya mampu membaca pada jarak hanya beberapa milimeter saja. -Ada batasan head Life (usia pakai), berapa ribu kali pakai.
3.	Media Transmitter	Gelombang radio	Gelombang magnetic
4.	Ukuran tag	Ukuran tag RFID berbagai macam, ada yang setebal kertas HVS (beberapa millimeter persegi),	Biasannya berukuran lebih besar.

		bahkan bisa ditanam dalam tubuh manusia.	
5.	Tingkat keamanan data	RFID memiliki tingkat keamanan lebih dibanding dengan magnetic card, misalnya dari frekuensi yang dipakai saja dari 125kHz sampai GHz, kemudian setiap tag memiliki ID yang unik dari setiap perusahaan yang membuatnya.	Tingkat keamanan kurang memadai karena magnetic card bisa digandakan dengan mengakopi isi card ke card yang lain.
6.	Multiple Access	Mampu membaca data secara acak dan cepat.	Tidak mampu membaca data secara acak.
7.	Real time System	Mampu mengirim data secara realtime.	Tidak mampu mengirim data secara rael time.
8.	Penggunaan	Access system, security system, supply chin, tracking asset, payment system, apparel, IT sensor, smart home, retail item check out.	Payment system, access system

4.4. Pemasangan Kunci Pintu

Pemasangan dapat dilakukan dengan cara memasang slot kunci pada kusen pintu. Kunci pintu disini digunakan untuk satu pintu saja jadi perlu diperhatikan jika digunakan untuk dua pintu, yaitu letak sumber arus rangkaian. Seperti gambar 4.3 dibawah ini adalah contoh pemasangan kunci pintu hotel pada satu pintu.



Gambar 4.3 Pemasangan kunci pintu RFID

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan bahwa:

1. Sistem minimal AT89C51 berfungsi sebagai *central processing unit* yang mengolah data dari keypad dan reader RFID, kemudian menampilkan ke LCD dan mengendalikan solenoid relai.
2. *Tag RFID* mampu memindahkan data (ID) ke *reader* dengan menggunakan frekuensi 125 kHz, kemudian reader meneruskan data secara serial ke system minimal mikrokontroler AT89C51.
3. Jarak baca efektif antara tag dan reader RFID (ID-10) tanpa penghalang kurang lebih 6 cm namun mengalami penurunan ketika ada penghalang, bahkan jika penghalang adalah lempengan logam (besi), *reader* RFID tidak mampu membaca karena tidak terjadi medan magnet yang digunakan *tag* RFID sebagai catu daya.
4. Pin T1 dalam kondisi high selama kurang lebih 6 detik, untuk memberikan pemicuan transistor BC547, kemudian transistor memberikan pemicuan terhadap relai 5V DC, dan relai ini yang akan menghubungkan solenoid (kunci pintu) dengan catu daya 12V DC.

5.2 Saran

1. Kunci pintu ini mempunyai kelemahan pada relai yang digunakan sehingga perlu dikembangkan kunci pintu dengan relai yang lebih kuat.

2. Kunci pintu yang dibuat belum dapat membaca dan menampilkan ID yang tersimpan dalam tag RFID, ID masih dimasukkan lewat keypad. Bisa dibuat kunci yang memiliki kemampuan membaca dan menampilkan ke LCD, sehingga tidak perlu lagi memasukkan ID tag ke mikrokontroler dengan keypad.
3. Di masa yang akan datang dapat dikembangkan tidak hanya kunci pintu saja tetapi juga kunci dan sistem keamanan yang lain.
4. Teknologi RFID masih baru sehingga masih banyak aspek pengembangan yang lebih dari sekedar kunci pintu, seperti negara-negara maju.
5. Kunci pintu ini masih mempunyai *device* yang berukuran relative besar, mungkin suatu saat dapat diciptakan dalam single chip saja.

DAFTAR PUSTAKA

1. Andi Nalwan, Paulus, *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*, PT Elex Media Komputindo, Jakarta 2003.
2. Agfianto Eko Putra, *Belajar Mikrokontroler AT89C51/52/55*, Penerbit Gava Media, Yogyakarta, 2002.
3. Budiharto, Widodo, *Interfacing Komputer dan Mikrokontroler*, PT Elex Media Komputindo, Jakarta 2004.
4. Petruzella, Frank D, *Elektronika Industri Edisi II*, Terjemahan dari *Industrial Electronics* oleh Sumanto, Drs. M.A., ANDI Yogyakarta, Yogyakarta 2001.
5. S.W Amos, *Kamus Elektronika*, PT Elex Media Komputindo, Jakarta 1996.
6. Zaks, Rodney, *Teknik Perantaraan Mikroprosesor Edisi Ketiga*, Penerbit Erlangga, Jakarta 1993.

```

#include "8051.H"

SELENOID .EQU P3.5
RS_LCD .EQU P3.3
E_LCD .EQU P3.1
RW_LCD .EQU P3.2

SLDATALCD .EQU P1.7
PORTLCD .EQU P0
;-----Control Data Magnetic-----
SWITCH .EQU P1
X1 .EQU P1.4
X2 .EQU P1.5
X3 .EQU P1.6
Y1 .EQU P1.0
Y2 .EQU P1.1
Y3 .EQU P1.2
Y4 .EQU P1.3

.ORG $30
DATAKEY .BLOCK 1
ADDRH .BLOCK 1
ADDRL .BLOCK 1
BUFEEPROM .BLOCK 16 ;Buffer data Nomor dari
EEPROM

DATANOHEX1 .BLOCK 1
DATANOHEX2 .BLOCK 1
DATANOHEX3 .BLOCK 1
DATANOHEX4 .BLOCK 1
DATADECHASIL1 .BLOCK 1
DATADECHASIL2 .BLOCK 1
DATADECHASIL3 .BLOCK 1
DATADECHASIL4 .BLOCK 1
DATADECHASIL5 .BLOCK 1
DATAPENAMBAH1 .BLOCK 1
DATAPENAMBAH2 .BLOCK 1
DATAPENAMBAH3 .BLOCK 1
DATAPENAMBAH4 .BLOCK 1
DATAPENAMBAH5 .BLOCK 1
DATAS1 .BLOCK 1
DATAS2 .BLOCK 1
DATAS3 .BLOCK 1
DATAS4 .BLOCK 1
DATAS5 .BLOCK 1
DATAS6 .BLOCK 1
DATAS7 .BLOCK 1
DATAS8 .BLOCK 1
BUFDATANOMOR .BLOCK 18
DATANOMORASCI11 .BLOCK 1
DATANOMORASCI12 .BLOCK 1
DATANOMORASCI13 .BLOCK 1
DATANOMORASCI14 .BLOCK 1
DATANOMORASCI15 .BLOCK 1
DATANOMORASCI16 .BLOCK 1

```



```

DATANOMORASCI17 .BLOCK 1
DATANOMORASCI18 .BLOCK 1
DATANOMORASCI19 .BLOCK 1
DATANOMORASCI110 .BLOCK 1

ADDREEPROM      .EQU    $2000

                .ORG    00H
                LJMP    START

                .ORG    $23
                LJMP    SERINT

START:          .ORG    100H
                MOV     SP,#20H
                MOV     PSW,#0
                MOV     R0,#BUFDATANOMOR
                CLR     SLDATA_LCD
                CLR     RW_LCD
                CLR     RS_LCD
                SETB    E_LCD
                CLR     SELENOID
                LCALL   DELAY
                LCALL   DELAY
                LCALL   INIT_LCD
                LCALL   DELAY
                LCALL   DELAY
                MOV     DPTR,#TEXTAWAL1
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
                MOV     DPTR,#TEXTAWAL2
                LCALL   PROC_STRTOLCD
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   PROC_HOMELCD
                MOV     DPTR,#TEXTAWAL3
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
                MOV     DPTR,#TEXTAWAL4
                LCALL   PROC_STRTOLCD
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   DELAYSW
                LCALL   INITSERIAL
                SETB    ES
                SETB    EA

PROGRAM1:      LCALL   PROC_HOMELCD

```

```

MOV     DPTR, #KATA1
LCALL  PROC_STRTOLCD
LCALL  PROC_LFLCD
MOV     DPTR, #KATA2
LCALL  PROC_STRTOLCD

PROGRAM2: MOV     SP, # $20
          LCALL  SCANNINGKEYPAD
          MOV     A, DATAKEY
          CJNE   A, #0, PROGRAM3
          LJMP   PROGRAM1
PROGRAM3: CJNE   A, # $0B, PROGRAM4
          LJMP   SETTING
PROGRAM4: LJMP   PROGRAM1

SERINT:  JBC     RI, GETDATASERIAL
          RETI

GETDATASERIAL:
          PUSH   ACC

          MOV     A, SBUF
          CLR     RI
          CJNE   A, # $03, ISIKEBUFFER
          LCALL  PROSES
          MOV     R0, #BUFDATANOMOR
          POP     ACC
          RETI

ISIKEBUFFER: MOV    @R0, A
             INC    R0

             POP    ACC
             RETI

PROSES:   MOV     R0, #BUFDATANOMOR
             INC    R0
             INC    R0
             INC    R0
             MOV    A, @R0
             LCALL  CEKA_F
             ANL    A, # $0F
             MOV    DATAS1, A
             INC    R0
             MOV    A, @R0
             LCALL  CEKA_F
             ANL    A, # $0F
             MOV    DATAS2, A
             INC    R0
             MOV    A, @R0
             LCALL  CEKA_F
             ANL    A, # $0F
             MOV    DATAS3, A
             INC    R0

```

```

MOV     A,@R0
LCALL  CEKA_F
ANL    A,#$0F
MOV    DATAS4,A
INC    R0
MOV    A,@R0
LCALL  CEKA_F
ANL    A,#$0F
MOV    DATAS5,A
INC    R0
MOV    A,@R0
LCALL  CEKA_F
ANL    A,#$0F
MOV    DATAS6,A
INC    R0
MOV    A,@R0
LCALL  CEKA_F
ANL    A,#$0F
MOV    DATAS7,A
INC    R0
MOV    A,@R0
LCALL  CEKA_F
ANL    A,#$0F
MOV    DATAS8,A

```

```

;-----PENGABUNGAN

```

```

MOV    A,DATAS1
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX1,A
MOV    A,DATAS2
ANL    A,#$0F
ORL    A,DATANOHEX1
MOV    DATANOHEX1,A

MOV    A,DATAS3
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX2,A
MOV    A,DATAS4
ANL    A,#$0F
ORL    A,DATANOHEX2
MOV    DATANOHEX2,A

MOV    A,DATAS5
SWAP   A
ANL    A,#$F0
MOV    DATANOHEX3,A
MOV    A,DATAS6
ANL    A,#$0F
ORL    A,DATANOHEX3
MOV    DATANOHEX3,A

MOV    A,DATAS7
SWAP   A
ANL    A,#$F0

```

```

MOV     DATANOHEX4 , A
MOV     A , DATAS8
ANL     A , #$0F
ORL     A , DATANOHEX4
MOV     DATANOHEX4 , A

LCALL   HEXTODES
LCALL   KOSONGKAN

MOV     A , DATADECHASIL1
LCALL   ANDF0
MOV     DATANOMORASCII1 , A
MOV     A , DATADECHASIL1
LCALL   AND0F
MOV     DATANOMORASCII2 , A

MOV     A , DATADECHASIL2
LCALL   ANDF0
MOV     DATANOMORASCII3 , A
MOV     A , DATADECHASIL2
LCALL   AND0F
MOV     DATANOMORASCII4 , A

MOV     A , DATADECHASIL3
LCALL   ANDF0
MOV     DATANOMORASCII5 , A
MOV     A , DATADECHASIL3
LCALL   AND0F
MOV     DATANOMORASCII6 , A

MOV     A , DATADECHASIL4
LCALL   ANDF0
MOV     DATANOMORASCII7 , A
MOV     A , DATADECHASIL4
LCALL   AND0F
MOV     DATANOMORASCII8 , A

MOV     A , DATADECHASIL5
LCALL   ANDF0
MOV     DATANOMORASCII9 , A
MOV     A , DATADECHASIL5
LCALL   AND0F
MOV     DATANOMORASCII10 , A

;----- PROSES MEMBANDINGKAN
LCALL   BACADATAEEPROM
MOV     R0 , #DATANOMORASCII1
MOV     R1 , #BUFEEPROM
MOV     R2 , #10

ULANGCEK :
CLR     A
MOV     B , @R0
MOV     A , @R1
CJNE   A , B , NOMORTIDAKCOCOK
INC     R0

```

```

                                INC      R1
                                DJNZ    R2, ULANGCEK

DATABENAR :

                                LCALL   PROC_HOMELCD
                                MOV     DPTR, #KATA4
                                LCALL   PROC_STRTOLCD
                                LCALL   PROC_LFLCD
                                MOV     DPTR, #KATA5
                                LCALL   PROC_STRTOLCD

UNTUK SELENOID

                                SETB   SELENOID           ;AKTIFKAN RELAY
                                MOV     R3, #20
                                LCALL   DELAYLONG

                                LCALL   PROC_HOMELCD
                                MOV     DPTR, #KATA6
                                LCALL   PROC_STRTOLCD
                                LCALL   PROC_LFLCD
                                MOV     DPTR, #KATA5
                                LCALL   PROC_STRTOLCD
                                LCALL   DELAYSW
                                CLR     SELENOID           ;MATIKAN KEMBALI

RELAY SELENOID

                                LCALL   DELAY
                                RET

NOMORTIDAKCOCOK :

                                LCALL   PROC_HOMELCD
                                MOV     DPTR, #KATAERROR1
                                LCALL   PROC_STRTOLCD
                                LCALL   PROC_LFLCD
                                MOV     DPTR, #KATAERROR2
                                LCALL   PROC_STRTOLCD
                                MOV     R3, #2
                                LCALL   DELAYLONG
                                RET

KOSONGKAN :

                                MOV     DATANOMORASCII1, #0
                                MOV     DATANOMORASCII2, #0
                                MOV     DATANOMORASCII3, #0
                                MOV     DATANOMORASCII4, #0
                                MOV     DATANOMORASCII5, #0
                                MOV     DATANOMORASCII6, #0
                                MOV     DATANOMORASCII7, #0
                                MOV     DATANOMORASCII8, #0
                                MOV     DATANOMORASCII9, #0
                                MOV     DATANOMORASCII10, #0
                                RET

ANDF0 :

                                SWAP   A
                                ANL   A, #$0F
                                ADD   A, #$30
                                RET

```

```

AND0F:
        ANL     A, #\$0F
        ADD     A, #\$30
        RET

CEKA_F:
CEKA:   CJNE   A, #'A', CEKB
        MOV     A, #\$0A
        ADD     A, #\$30
        RET
CEKB:   CJNE   A, #'B', CEKC
        MOV     A, #\$0B
        ADD     A, #\$30
        RET
CEKC:   CJNE   A, #'C', CEKD
        MOV     A, #\$0C
        ADD     A, #\$30
        RET
CEKD:   CJNE   A, #'D', CEKE
        MOV     A, #\$0D
        ADD     A, #\$30
        RET
CEKE:   CJNE   A, #'E', CEKF
        MOV     A, #\$0E
        ADD     A, #\$30
        RET
CEKF:   CJNE   A, #'F', CEKPA
        MOV     A, #\$0F
        ADD     A, #\$30
CEKPA:
        RET

HEXTODES:
        MOV     DATADECHASIL1, #0
        MOV     DATADECHASIL2, #0
        MOV     DATADECHASIL3, #0
        MOV     DATADECHASIL4, #0
        MOV     DATADECHASIL5, #0

        MOV     A, DATANOHEX4
KE1:    CLR     C
        RRC     A
        JNC     KE2
        LCALL   HITBIT1
        LCALL   HITUNG
KE2:    CLR     C
        RRC     A
        JNC     KE3
        LCALL   HITBIT2
        LCALL   HITUNG
KE3:    CLR     C
        RRC     A

```

```

                JNC      KE4
                LCALL   HITBIT3
                LCALL   HITUNG
KE4:            CLR      C
                RRC     A
                JNC     KE5
                LCALL   HITBIT4
                LCALL   HITUNG
KE5:            CLR      C
                RRC     A
                JNC     KE6
                LCALL   HITBIT5
                LCALL   HITUNG
KE6:            CLR      C
                RRC     A
                JNC     KE7
                LCALL   HITBIT6
                LCALL   HITUNG
KE7:            CLR      C
                RRC     A
                JNC     KE8
                LCALL   HITBIT7
                LCALL   HITUNG
KE8:            CLR      C
                RRC     A
                JNC     KE9
                LCALL   HITBIT8
                LCALL   HITUNG
KE9:            MOV     A, DATANOHEX3
                CLR     C
                RRC     A
                JNC     KE10
                LCALL   HITBIT9
                LCALL   HITUNG
KE10:           CLR     C
                RRC     A
                JNC     KE11
                LCALL   HITBIT10
                LCALL   HITUNG
KE11:           CLR     C
                RRC     A
                JNC     KE12
                LCALL   HITBIT11
                LCALL   HITUNG
KE12:           CLR     C
                RRC     A
                JNC     KE13
                LCALL   HITBIT12
                LCALL   HITUNG
KE13:           CLR     C
                RRC     A
                JNC     KE14
                LCALL   HITBIT13
                LCALL   HITUNG
KE14:           CLR     C

```

```

RRC      A
JNC      KE15
LCALL   HITBIT14
LCALL   HITUNG
KE15:   CLR      C
RRC      A
JNC      KE16
LCALL   HITBIT15
LCALL   HITUNG
KE16:   CLR      C
RRC      A
JNC      KE17
LCALL   HITBIT16
LCALL   HITUNG
KE17:
MOV      A,DATANOHEX2
CLR      C
RRC      A
JNC      KE18
LCALL   HITBIT17
LCALL   HITUNG
KE18:   CLR      C
RRC      A
JNC      KE19
LCALL   HITBIT18
LCALL   HITUNG
KE19:   CLR      C
RRC      A
JNC      KE20
LCALL   HITBIT19
LCALL   HITUNG
KE20:   CLR      C
RRC      A
JNC      KE21
LCALL   HITBIT20
LCALL   HITUNG
KE21:   CLR      C
RRC      A
JNC      KE22
LCALL   HITBIT21
LCALL   HITUNG
KE22:   CLR      C
RRC      A
JNC      KE23
LCALL   HITBIT22
LCALL   HITUNG
KE23:   CLR      C
RRC      A
JNC      KE24
LCALL   HITBIT23
LCALL   HITUNG
KE24:   CLR      C
RRC      A
JNC      KE25
LCALL   HITBIT24
LCALL   HITUNG

```



```

KE25:      MOV     A, DATANOHEX1
           CLR     C
           RRC     A
           JNC     KE26
           LCALL  HITBIT25
           LCALL  HITUNG
KE26:      CLR     C
           RRC     A
           JNC     KE27
           LCALL  HITBIT26
           LCALL  HITUNG
KE27:      CLR     C
           RRC     A
           JNC     KE28
           LCALL  HITBIT27
           LCALL  HITUNG
KE28:      CLR     C
           RRC     A
           JNC     KE29
           LCALL  HITBIT28
           LCALL  HITUNG
KE29:      CLR     C
           RRC     A
           JNC     KE30
           LCALL  HITBIT29
           LCALL  HITUNG
KE30:      CLR     C
           RRC     A
           JNC     KE31
           LCALL  HITBIT30
           LCALL  HITUNG
KE31:      CLR     C
           RRC     A
           JNC     KE32
           LCALL  HITBIT31
           LCALL  HITUNG
KE32:      CLR     C
           RRC     A
           JNC     KE33
           LCALL  HITBIT32
           LCALL  HITUNG
KE33:

           RET

HITUNG:   PUSH    ACC
           MOV     A, DATADECHASIL5
           ADD     A, DATAPENAMBAH5
           DA      A
           MOV     DATADECHASIL5, A

           MOV     A, DATADECHASIL4

```

```

        ADDC    A, DATAPENAMBAH4
        DA     A
        MOV    DATADECHASIL4, A

        MOV    A, DATADECHASIL3
        ADDC   A, DATAPENAMBAH3
        DA     A
        MOV    DATADECHASIL3, A

        MOV    A, DATADECHASIL2
        ADDC   A, DATAPENAMBAH2
        DA     A
        MOV    DATADECHASIL2, A

        MOV    A, DATADECHASIL1
        ADDC   A, DATAPENAMBAH1
        DA     A
        MOV    DATADECHASIL1, A
        POP    ACC
        RET

;-----
HITBIT1:
        MOV    DATAPENAMBAH5, #$01
        MOV    DATAPENAMBAH4, #$00
        MOV    DATAPENAMBAH3, #$00
        MOV    DATAPENAMBAH2, #$00
        MOV    DATAPENAMBAH1, #$00
        RET

HITBIT2:
        MOV    DATAPENAMBAH5, #$02
        MOV    DATAPENAMBAH4, #$00
        MOV    DATAPENAMBAH3, #$00
        MOV    DATAPENAMBAH2, #$00
        MOV    DATAPENAMBAH1, #$00
        RET

HITBIT3:
        MOV    DATAPENAMBAH5, #$04
        MOV    DATAPENAMBAH4, #$00
        MOV    DATAPENAMBAH3, #$00
        MOV    DATAPENAMBAH2, #$00
        MOV    DATAPENAMBAH1, #$00
        RET

HITBIT4:
        MOV    DATAPENAMBAH5, #$08
        MOV    DATAPENAMBAH4, #$00
        MOV    DATAPENAMBAH3, #$00
        MOV    DATAPENAMBAH2, #$00
        MOV    DATAPENAMBAH1, #$00
        RET

HITBIT5:
        MOV    DATAPENAMBAH5, #$16
        MOV    DATAPENAMBAH4, #$00
        MOV    DATAPENAMBAH3, #$00
        MOV    DATAPENAMBAH2, #$00

```



```

MOV     DATAPENAMBAH3, # $21
MOV     DATAPENAMBAH2, # $34
MOV     DATAPENAMBAH1, # $01
RET

HITBIT29:
MOV     DATAPENAMBAH5, # $56
MOV     DATAPENAMBAH4, # $54
MOV     DATAPENAMBAH3, # $43
MOV     DATAPENAMBAH2, # $68
MOV     DATAPENAMBAH1, # $02
RET

HITBIT30:
MOV     DATAPENAMBAH5, # $12
MOV     DATAPENAMBAH4, # $09
MOV     DATAPENAMBAH3, # $87
MOV     DATAPENAMBAH2, # $36
MOV     DATAPENAMBAH1, # $05
RET

HITBIT31:
MOV     DATAPENAMBAH5, # $24
MOV     DATAPENAMBAH4, # $18
MOV     DATAPENAMBAH3, # $74
MOV     DATAPENAMBAH2, # $73
MOV     DATAPENAMBAH1, # $10
RET

HITBIT32:
MOV     DATAPENAMBAH5, # $48
MOV     DATAPENAMBAH4, # $36
MOV     DATAPENAMBAH3, # $48
MOV     DATAPENAMBAH2, # $47
MOV     DATAPENAMBAH1, # $21
RET
;-----

```

```

INITSERIAL:

```

```

MOV     TMOD, #20H
MOV     TCON, #41H
MOV     TH1, #0FDH
MOV     SCON, #50H
RET

```

```

SETTING:

```

```

LCALL   BACADATAEEPROM
LCALL   PROC_CLEARLCD
MOV     DPTR, #TEXTSETTING1
LCALL   PROC_STRTOLCD
LCALL   PROC_LFLCD
LCALL   DELAYSW
LCALL   DELAYSW
LCALL   DELAYSW
LCALL   DELAYSW

```

```

        LCALL  PROC_CLEARLCD
        MOV    DPTR,#TEXTSETTING1
        LCALL  PROC_STRTOLCD
        LCALL  PROC_LFLCD

        MOV    R4,#10
        MOV    DPTR,#ADDREEPROM

SETLOOP:
        LCALL  SCANNINGKEYPAD
        MOV    A,DATAKEY
        CJNE  A,#0,MASUKKANNOMOR
        LJMP  SETLOOP
MASUKKANNOMOR:
        CJNE  A,#$0C,MASUKKANNOMOR1
        LJMP  PROGRAM1
MASUKKANNOMOR1:
        CJNE  A,#$0A,MASUKKANNOMOR2
        MOV    A,#$0
MASUKKANNOMOR2:
        ADD   A,#$30
        MOVX  @DPTR,A
        MOV   ADDRH,DPH
        MOV   ADDR L,DPL

        LCALL  KEYPADTOLCD
        MOV   DPH,ADDRH
        MOV   DPL,ADDR L
        INC   DPTR
        LCALL  DELAY
        DJNZ  R4,SETLOOP
        LCALL  PROC_CLEARLCD
        LCALL  BACADATAAEEPROM
        LCALL  DELAYSW
        LCALL  DELAYSW
        LCALL  DELAYSW
        LJMP  PROGRAM1

BACADATAAEEPROM:
        MOV   R4,#10
        MOV   DPTR,#ADDREEPROM
        MOV   R0,#BUFEEEPROM

BACAEPLAGI:
        MOVX  A,@DPTR
        MOV   @R0,A
        INC   R0
        INC   DPTR
        DJNZ  R4,BACAEPLAGI
        RET

SCANNINGKEYPAD:
KOLOM1:  MOV    DATAKEY,#0
        CLR    X1
        SETB   X2
        SETB   X3

```

```

MOV      A, SWITCH
ANL      A, #0FH
CTOMBOL1: CJNE     A, #0EH, CTOMBOL4
MOV      DATAKEY, #01H
LJMP     TOLCD
CTOMBOL4: CJNE     A, #0DH, CTOMBOL7
MOV      DATAKEY, #04H
LJMP     TOLCD
CTOMBOL7: CJNE     A, #0BH, CTOMBOLB
MOV      DATAKEY, #07H
LJMP     TOLCD
CTOMBOLB: CJNE     A, #07H, KOLOM2
MOV      DATAKEY, #0BH
LJMP     TOLCD

KOLOM2:  SETB     X1
CLR      X2
SETB     X3
MOV      A, SWITCH
ANL      A, #0FH
CTOMBOL2: CJNE     A, #0EH, CTOMBOL5
MOV      DATAKEY, #02H
LJMP     TOLCD
CTOMBOL5: CJNE     A, #0DH, CTOMBOL8
MOV      DATAKEY, #05H
LJMP     TOLCD
CTOMBOL8: CJNE     A, #0BH, CTOMBOL0
MOV      DATAKEY, #08H
LJMP     TOLCD
CTOMBOL0: CJNE     A, #07H, KOLOM3
MOV      DATAKEY, #0AH
LJMP     TOLCD
KOLOM3:  SETB     X1
SETB     X2
CLR      X3
MOV      A, SWITCH
ANL      A, #0FH
CTOMBOL3: CJNE     A, #0EH, CTOMBOL6
MOV      DATAKEY, #03H
LJMP     TOLCD
CTOMBOL6: CJNE     A, #0DH, CTOMBOL9
MOV      DATAKEY, #06H
LJMP     TOLCD
CTOMBOL9: CJNE     A, #0BH, CTOMBOLP
MOV      DATAKEY, #09H
LJMP     TOLCD
CTOMBOLP: CJNE     A, #07H, KEYPADRET
MOV      DATAKEY, #0CH

TOLCD:   LCALL    DELAYSW
MOV      A, DATAKEY
KEYPADRET:
RET

KEYPADTOLCD:
MOV      DPTR, #KEY

```



```

MOV     A,DATAKEY
MOVC   A,@A+DPTR
LCALL  WRITE_DATAALCD
RET

PROC_HOME_LCD:
MOV     A,#02H
LCALL  WRITE_CTRL_LCD
RET

PROC_LF_LCD:
MOV     A,#0C0H
LCALL  WRITE_CTRL_LCD
RET

PROC_CLEAR_LCD:
MOV     A,#01H
LCALL  WRITE_CTRL_LCD
RET

WRITE_DATAALCD:
MOV     PORT_LCD,A
SETB   SL_DATAALCD
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
CLR     SL_DATAALCD
SETB   RS_LCD
CLR     E_LCD
LCALL  DELAY_LCD

SETB   RS_LCD
SETB   E_LCD
LCALL  DELAY_LCD

SETB   RS_LCD
CLR     E_LCD
NOP
CLR     RS_LCD
CLR     E_LCD
LCALL  DELAY_LCD
RET

WRITE_CTRL_LCD:
MOV     PORT_LCD,A
SETB   SL_DATAALCD
NOP
NOP
NOP
NOP
NOP
NOP

```

```

NOP
NOP
CLR          SLDATA_LCD

CLR          RS_LCD
CLR          E_LCD
LCALL       DELAY_LCD
CLR          RS_LCD
SETB        E_LCD
LCALL       DELAY_LCD
CLR          RS_LCD
CLR          E_LCD
NOP
CLR          RS_LCD
CLR          E_LCD
LCALL       DELAY_LCD

RET

;-----
;          Routine Pengiriman String ke Display LCD
;-----
PROC_STRTOLCD:
CLR          A
MOVC        A,@A+DPTR
CJNE        A,#00H,STRTOLCD
RET

STRTOLCD:
LCALL       WRITE_DATA_LCD
INC         DPTR
SJMP        PROC_STRTOLCD

;-----
;          INISIAL LCD
;-----
INIT_LCD
MOV          A,#38H
LCALL       WRITE_CTRL_LCD
MOV          A,#0EH
LCALL       WRITE_CTRL_LCD
MOV          A,#0CH
LCALL       WRITE_CTRL_LCD
MOV          A,#06H
LCALL       WRITE_CTRL_LCD
MOV          A,#01H
LCALL       WRITE_CTRL_LCD
RET

DELAY_LCD:
MOV          R7,#04H
D_LCD1:    MOV          R6,#3FH          ; 4F
D_LCD2:    DJNZ        R6,D_LCD2
            DJNZ        R7,D_LCD1
RET

```

```

DELAWSW:
MOV     R5,#02H
DELAWSW0:
MOV     R6,#0FFH
DELAWSW1:
MOV     R7,#0FFH
DELAWSW2:
DJNZ   R7,DELAWSW2
DJNZ   R6,DELAWSW1
DJNZ   R5,DELAWSW0
RET

DELAYLONG:
LCALL  DELAWSW
DJNZ   R3,DELAYLONG
RET

DELAY:
MOV     R7,#$ff
DLD1:  MOV     R6,#$FF
DLD2:  DJNZ   R6,DLD2
DJNZ   R7,DLD1
RET

KEY:   .BYTE   " 1234567890*#"
TEXTAWAL1: .BYTE " TUGAS AKHIR ",0
TEXTAWAL2: .BYTE " TEKNIK ELEKTRO ",0
TEXTAWAL3: .BYTE " RIKI ASTONO ",0
TEXTAWAL4: .BYTE " NIM:5350402014",0

KATA1:  .BYTE   " MONITORING ",0
KATA2:  .BYTE   "Y O U R C A R D",0
KATA3:  .BYTE   " KARTU ANDA ",0
KATA4:  .BYTE   "==KUNCI DIBUKA==",0
KATA5:  .BYTE   "[ _____ ]",0
KATA6:  .BYTE   "=PINTU DIKUNCI=",0
KATAERROR1: .BYTE " M A A F !!! ",0
KATAERROR2: .BYTE "CARD TDK DIKENAL",0
TEXTSETTING1: .BYTE "SET NOMOR KARTU ",0

CODEPARITY: .BYTE
$80,$00,$00,$80,$00,$80,$80,$00,$00,$80,$80,$00,$80,$00,$00,$80

.END

```

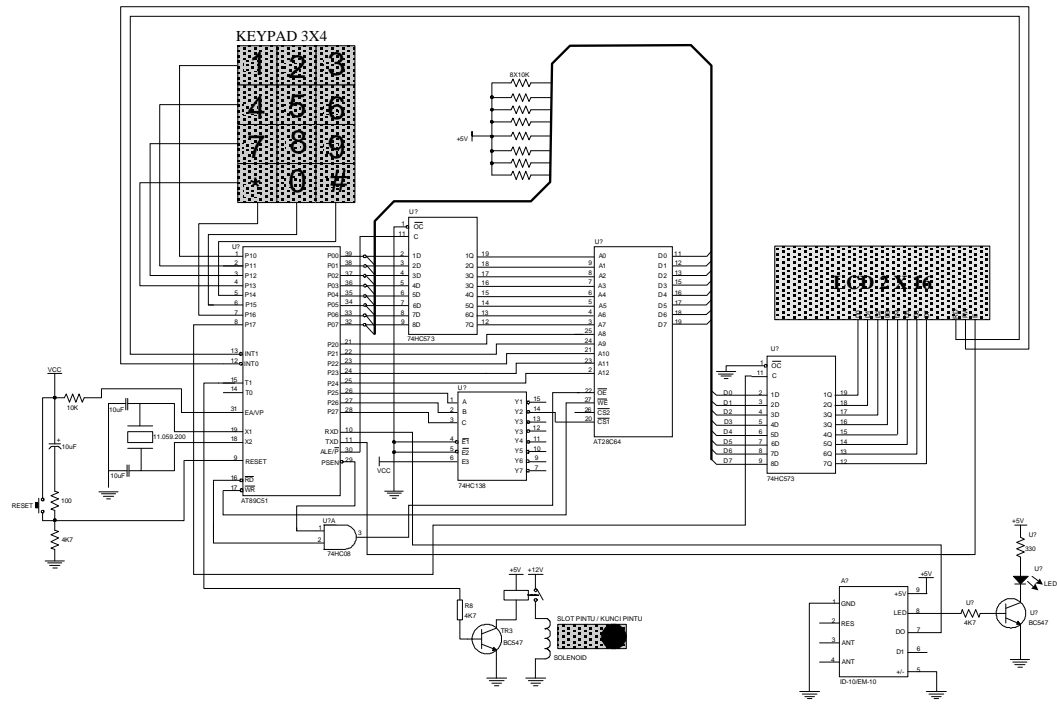


FOTO ALAT

Rangkaian kunci pintu dengan RFID tampak atas



Rangkaian catu daya tampak atas