

(0806221309)

**APLIKASI SISTEM KONTROL DENGAN KOMUNIKASI
SERIAL RS – 485 MENGGUNAKAN MICROCONTROLLER
PADA BANGUNAN BERTINGKAT**

TUGAS AKHIR

diajukan sebagai persyaratan untuk
menyelesaikan Pendidikan Sarjana program Instrumentasi Elektronika

Iwan Prastyawan

0305220399



**Program Sarjana Ekstensi Fisika Instrumentasi
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Indonesia
Depok
2008**

LEMBAR PENGESAHAN

Nama : Iwan Prastyawan
NPM : 0305220399
Jurusan : Ekstensi Fisika
Peminatan : Fisika Instrumentasi
Tanggal Sidang : 09 Juni 2008
Judul Skripsi : Aplikasi Sistem Kontrol Dengan Komunikasi Serial RS – 485
Menggunakan Microcontroller Pada Bangunan Bertingkat

Skripsi ini telah diperiksa dan disetujui oleh :

PEMBIMBING I



(Drs. Arief Sudarmaji, M.T)

PEMBIMBING II



(Djati Handoko, M.Si)

PENGUJI I



(Dr. Prawito)

PENGUJI II



(Lingga Hermanto, M.Si)

KATA PENGANTAR

Alhamdulillah penulis panjatkan ke hadirat Allah SWT karena atas ridho-Nya penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Sistem Kontrol Dengan Komunikasi Serial RS 485 Menggunakan *Microcontroller* Pada Bangunan Bertingkat” bertujuan untuk memenuhi syarat dalam menyelesaikan Program Sarjana Fisika Instrumentasi Elektronika, Departemen Fisika, FMIPA, Universitas Indonesia.

Dengan terselesaikannya penelitian dan laporan skripsi ini, penulis mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang telah membantu selama proses pelaksanaan dan penyelesaiannya. Untuk itu pada kesempatan ini penulis ingin menyampaikan terima kasih dan rasa hormat kepada :

1. Allah SWT atas segala kuasa-Nya di langit dan bumi.
2. Bpk. Drs. Arief Sudarmadji, M.T., selaku dosen pembimbing 1, atas semua waktu, motivasi dan solusi alternative terbaik yang sangat membantu selama pembuatan dan penyelesaian penelitian ini.
3. Bpk Djati Handoko, M.Si, selaku dosen pembimbing 2, yang telah memberikan waktu luangnya selama proses penyelesaian alat, juga bimbingan, koreksi dan diskusi yang bermanfaat selama penyelesaian penelitian ini.
4. Seluruh Dosen Fisika FMIPA yang telah memberikan ilmunya serta seluruh staf dan karyawan yang telah membantu kepada penulis.
5. Bapak dan bundaku tercinta, dengan segala keikhlasan, kesabaran, semangat juga doa yang tak henti-hentinya selalu dipanjatkan dalam setiap kata dan doa. Ya

Allah SWT yang Maha Mulia, berikanlah ridho-Mu disetiap sudut jalan hidup mereka juga untuk kedua saudaraku Yoyo dan Hary semoga keberhasilan selalu menyertai kalian.

6. Seluruh teman-teman seperjuangan, Novan, Johan, Bondan, Angga, Firidy, Fajrul, Andri, Ikhsan, dan yang lainnya yang selalu memberikan dan menguatkan semangat dalam menyelesaikan penelitian ini.

7. Seluruh Rekan-rekan Fisika Ekstensi angkatan 2005.

8. Seluruh Rekan-rekan Instrumentasi angkatan 2000.

9. Seluruh Rekan-rekan Instrumentasi Mountenering Club.

10. Semua pihak yang secara tidak langsung ikut terlibat dalam pembuatan tugas akhir ini yang tidak saya sebutkan satu persatu, semoga amal baik yang telah dilakukan dibalas di kemudian hari.

Penulis menyadari masih banyak kekurangan, kesalahan dan jauh dari kata sempurna dalam pembuatan dan penyusunan skripsi ini. Besar harapan penulis kepada semua pihak untuk dapat memberikan saran dan kritik yang bersifat membangun sebagai dasar kuat agar selanjutnya dapat membuat skripsi yang lebih baik. Terima kasih, semoga laporan ini dapat bermanfaat bagi kita semua.

Depok, Juni 2008

Penulis

ABSTRAK

APLIKASI SISTEM KONTROL DENGAN KOMUNIKASI SERIAL RS 485 MENGGUNAKAN MICROCONTROLLER PADA BANGUNAN BERTINGKAT

Oleh :

Iwan Prastyawan

Aplikasi sistem kontrol banyak digunakan dalam berbagai bidang aplikasi, seperti pada bangunan bertingkat. Sistem yang dibuat adalah suatu sistem yang terintegrasi dengan *microcontroller* serta sensor-sensor sebagai input dari suatu besaran analog. Aplikasi *microcontroller* pada penelitian ini terdiri dari beberapa parameter seperti identifikasi *password*, pendeteksian dengan sensor arus (*current transformer*) serta pengaturan kecerahan cahaya lampu 220 VAC. Identifikasi *password* merupakan sistem *security* yang mengidentifikasi setiap orang yang akan masuk ke dalam bangunan, *current transformer* merupakan suatu sensor arus yang dapat mendeteksi arus yang mengalir pada suatu rangkaian listrik dengan begitu kita dapat mengetahui ada atau tidaknya arus sehingga kita dapat mengetahui kondisi beban / peralatan listrik pada suatu jaringan listrik. Pada sistem kecerahan mengatur intensitas cahaya lampu 220 VAC dengan mengirim paket data 8 bit melalui PC. Sensor OPT101 berfungsi sebagai indikator yang membaca intensitas cahaya pada lantai tersebut. Sistem kontrol ini menggunakan sistem komunikasi serial RS 485. Prinsip kerja dari sistem ini adalah data-data yang diterima oleh *microcontroller* nantinya akan dikirim melalui PC melalui *RS 485 to RS 232 Converter*. Sistem ini mengadopsi sebuah pengalaman sehingga antara besaran-besaran pada sistem tidak bentrok baik pada saat pembacaan maupun pengiriman datanya.

Kata kunci— *Password, Current transformer, Msicrocontroller, sensor, Converter*

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan.....	ii
Kata Pengantar	iii
Abstrak.....	v
Daftar Isi.....	vi
Daftar Gambar.....	x
Daftar Tabel	xiv
Daftar Pustaka.....	xv
Lampiran.....	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Batasan Masalah	2
1.4 Deskripsi Singkat.....	3
1.5 Metode Penelitian	5
1.6 Sistematika Penulisan	6

BAB 2 TEORI DASAR.....	8
2.1 Arus Listrik	8
2.2 Cahaya.....	9
2.2.1 Sifat dan Penjalaran Cahaya	10
2.2.2 Sumber-sumber Cahaya	11
2.3 <i>Microcontroller</i> AVR ATMEGA32	12
2.3.1 Fitur Dan Arsitektur AVR ATMEGA32	12
2.3.2 Pengorganisasian Memori.....	14
2.3.2.1 Program Memory	14
2.3.2.2 Data Memory	15
2.3.2.3 EEPROM	15
2.3.3 Konfigurasi Pin	16
2.3.4 USART	19
2.3.5 Interrupt.....	21
2.4 Komunikasi Serial.....	22
2.4.1 Komunikasi Asynchronuos	23
2.4.1.1 Start Bit Dan Stop Bit	23
2.4.1.2 Parity Bit	24
2.4.1.3 Baud Rate	24
2.4.2 Standar Serial RS-232.....	24
2.4.3 Konfigurasi Pin DB-9	25
2.4.4 Komunikasi RS-485.....	26

BAB 3 PERANCANGAN & CARA KERJA SISTEM	28
3.1 Perancangan Hardware	28
3.1.1 Personal Computer (PC)	29
3.1.2 Rangkaian RS-485 To RS-232 Converter	31
3.1.3 Rangkaian Komunikasi Serial.....	32
3.1.4 Rangkaian Regulator.....	34
3.1.5 Rangkaian Sensor OPT 101	35
3.1.6 Rangkaian keypad.....	37
3.1.7 Rangkaian Pendeteksi Kondisi Peralatan Listrik.....	40
3.1.8 Rangkaian Pengendali Kecerahan Lampu AC 220 V.....	42
3.1.9 Rangkaian Minimum Sistem.....	47
3.2 Perancangan Software.....	52
3.2.1 Protokol Pengiriman Data Serial	52
3.2.2 Flowchart Program.....	56
3.2.2.1 Flowchart Program Micro Master (ATMEGA32).....	57
3.2.2.2 Flowchart Program Interrupt	
Eksternal INT0 (Keypad).....	60
3.2.2.3 Flowchart Program	
Interrupt Serial	61
3.2.2.4 Flowchart Program Micro	
Slave (AT90S2313)	87
3.2.2.5 Flowchart Program Micro	
Interrupt Eksternal INT0 (AT90S2313).....	89

BAB 4 HASIL EKSPERIMEN DAN PEMBAHASAN.....	90
4.1 Pengujian Sistem Keamanan (Security).....	90
4.2 Pengujian Pendeteksi Kondisi Peralatan Listrik.....	99
4.3 Pengujian Pengaturan Tegangan AC.....	106
4.4 Pengujian Komunikasi Serial.....	107
4.5 Pengujian Protokol Komunikasi Serial.....	107
4.6 Pengujian Panjang Kabel Komunikasi Serial.....	120
BAB 5 KESIMPULAN DAN SARAN.....	121
5.1 Kesimpulan.....	121
5.2 Saran.....	122

DAFTAR GAMBAR

Gambar 1.1	Blok Diagram Sistem	4
Gambar 2.1	Blok diagram mikrokontroler AVR ATMEGA32	13
Gambar 2.2	Memori program dan memori data AVR ATMEGA32	14
Gambar 2.3	Pin-pin pada ATMEGA32	18
Gambar 2.4	Blok diagram USART	20
Gambar 2.5	RS-232 DB 9 pinout	25
Gambar 3.1	Blok diagram sistem keseluruhan	28
Gambar 3.2	Konfigurasi slot DB-9	29
Gambar 3.3	Rangkaian RS-485 to RS-232 Converter	31
Gambar 3.4	Rangkaian komunikasi serial RS-232 & RS-485	32
Gambar 3.5	Rangkaian regulator	34
Gambar 3.6	Rangkaian Internal OPT101	35
Gambar 3.7	OPT101 tampak atas	35
Gambar 3.8	Rangkaian sensor menggunakan OPT101	36
Gambar 3.9	Skematik keypad 4 x 4	37
Gambar 3.10	Skematik rangkaian interrupt keypad 4 x 4	39
Gambar 3.11	Rangkaian pendeteksi peralatan listrik	40
Gambar 3.12	Rangkaian pengatur intensitas cahaya	42
Gambar 3.13	Gelombang sinus pada tegangan AC	43
Gambar 3.14	Gelombang setelah melewati bridge	43

Gambar 3.15	Setengah periode gelombang	45
Gambar 3.16	Pengaturan gelombang	46
Gambar 3.17	Skematik rangkaian minimum sistem	47
Gambar 3.18	Flowchart program utama system microcontroller (ATMEGA32)	58
Gambar 3.19	Flowchart program interrupt keypad.....	60
Gambar 3.20	Flowchart program Interrupt Serial.....	65
Gambar 3.21	Flowchart program utama sistem microcontroller (AT90S2313).....	87
Gambar 3.22	Flowchart program interrupt eksternal INT0 (AT90S2313).....	89
Gambar 4.1	Bit number baris pembacaan angka 1	91
Gambar 4.2	Bit number kolom pembacaan angka 1	92
Gambar 4.3	Bit number baris pembacaan angka 2.....	93
Gambar 4.4	Bit number kolom pembacaan angka 2	94
Gambar 4.5	Bit number baris pembacaan angka 3.....	95
Gambar 4.6	Bit number kolom pembacaan angka 3	96
Gambar 4.7	Bit number baris pembacaan angka 4	97
Gambar 4.8	Bit number kolom pembacaan angka 4	98
Gambar 4.9	Rangkaian <i>comparator</i> pada rangkaian pendeteksi.....	100
Gambar 4.10	Rangkaian <i>non-inverting amplifier</i> pada rangkaian pendeteksi.....	101
Gambar 4.11	Rangkaian <i>precision rectifier</i> pada rangkaian pendeteksi	102
Gambar 4.12	Gelombang sinus setelah diatur.....	107
Gambar 4.13	Tampilan awal ketika program pertama kali dijalankan	108
Gambar 4.14	Tampilan awal pada saat ada permintaan penggantian alamat pada lantai 1	109

Gambar 4.15	Tampilan awal pada saat ada permintaan penggantian alamat pada lantai 2.....	110
Gambar 4.16	Tampilan awal pada saat ada permintaan alamat pada lantai 1	111
Gambar 4.17	Tampilan awal pada saat ada permintaan alamat pada lantai 2	112
Gambar 4.18	Tampilan awal pada saat ada permintaan data pada lantai 1.....	113
Gambar 4.19	Tampilan awal pada saat ada permintaan data pada lantai 2.....	113
Gambar 4.20	Tampilan awal pada saat ada permintaan data nama & password pada lantai 1	114
Gambar 4.21	Tampilan awal pada saat ada permintaan data nama & password pada lantai 2	114
Gambar 4.22	Tampilan awal pada saat ada penggantian password pada lantai 1	115
Gambar 4.23	Tampilan awal pada saat ada penggantian password pada lantai 2	116
Gambar 4.24	Tampilan awal pada saat melakukan registrasi nama & password pada lantai 1	117
Gambar 4.25	Tampilan awal pada saat melakukan registrasi nama & password pada lantai 2	117
Gambar 4.26	Tampilan awal pada saat ada penggantian nama pada lantai 1	118
Gambar 4.27	Tampilan awal pada saat ada penggantian nama pada lantai 2	118
Gambar 4.28	Tampilan awal pada saat melakukan pengontrolan	

lampu AC 220 V pada lantai 1	119
Gambar 4.29 Tampilan awal pada saat melakukan pengontrolan	
lampu AC 220 V pada lantai 2	119



DAFTAR TABEL

Tabel 2.1	Fungsi alternatif PORT A	16
Tabel 2.2	Fungsi alternatif PORT B	16
Tabel 2.3	Fungsi alternatif PORT C	17
Tabel 2.4	Fungsi alternatif PORT D	17
Tabel 2.5	Reset dan interrupt vector	22
Tabel 2.6	Konfigurasi pin DB-9	26
Tabel 3.1	Susunan pin port serial (COM)	30
Tabel 3.2	Baris dan kolom keypad ke port microcontroller	38
Tabel 3.3	Konfigurasi I / O Port ATMEGA32 (Master).....	49
Tabel 3.4	Konfigurasi I / O Port AT90S2313 (Slave)	50
Tabel 3.1	Format protokol komunikasi serial PC-microcontroller dan microcontroller-PC.....	53

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi yang semakin pesat sangat diperlukan dalam kebutuhan manusia. Dengan mengembangkan teknologi tersebut, kita akan mendapatkan suatu sistem yang dapat mengkoordinasi sistem yang satu dengan sistem yang lain seperti yang digunakan pada sistem kontrol pada bangunan bertingkat. Seperti yang kita ketahui sangatlah sulit apabila kita harus mengontrol serta memantau bangunan bertingkat secara manual, dimana hal tersebut tidaklah efisien.

Oleh karena itu untuk mendapatkan suatu sistem kontrol yang baik dibutuhkan adanya pengembangan dasar pengetahuan dan dilakukannya berbagai macam riset atau penelitian yang bersifat eksperimental. Dengan pertumbuhan pembangunan khususnya bangunan – bangunan bertingkat akan sangat sulit sekali apabila sistem kontrolnya dilakukan secara manual hal itu didasarkan banyaknya lantai, ruang serta peralatan-peralatan yang harus dipantau dan dikontrol pada bangunan tersebut. Untuk memudahkan hal tersebut maka kita memerlukan suatu piranti elektronika yang dapat melakukan tugas tersebut. Piranti tersebut terdiri dari *microcontroller* yang terintegrasi dengan sensor – sensor.

Sistem kontrol ini dimaksudkan dikarenakan pada saat-saat ini banyak terjadi hal-hal yang tidak diinginkan pada gedung seperti tindak-tindak kriminal seperti pencurian, pemboman dan hal lain diluar itu seperti kebakaran yang biasanya banyak disebabkan oleh hubungan arus

pendek (konsreleting) atau juga akibat kelainan manusia seperti peralatan listrik yang lupa dimatikan yang dapat menyebabkan terjadinya kebakaran.

1.2 TUJUAN PENELITIAN

Dalam penelitian ini penulis akan membuat suatu sistem pada bangunan bertingkat. Adapun tujuan penelitian membuat sistem ini yaitu :

- Membuat sebuah sistem aplikasi yang mempermudah dalam mengetahui kondisi bangunan dengan menyediakan data-data yang diperlukan.
- Membuat sebuah sistem yang dapat di-alamatkan melalui program, yang dilakukan dari komputer sentral sehingga pengontrolan dari komputer sentral tidak akan salah rumah.
- Membuat sistem yang dapat mengetahui kondisi peralatan listrik yang ada pada bangunan tersebut dan langsung menginformasikannya ke komputer sentral.
- Membuat sistem keamanan dari orang asing yang akan masuk kedalam rumah dengan membuat sistem identifikasi.

1.3 BATASAN MASALAH

Secara keseluruhan, sistem ini dibagi kedalam dua blok bagian yaitu blok lantai 1 & lantai 2. Pada dasarnya setiap sistem memiliki blok rangkaian yang sama yang membedakannya hanya di pengalamatannya saja. Blok-blok tersebut adalah: blok rangkaian keypad matrix 4 x 4 yang berfungsi sebagai sistem keamanan (*security*) dengan mengidentifikasi setiap password dengan maksud hanya yang telah terregistrasi saja yang mempunyai akses ke lantai tersebut. Blok sensor arus (*current transformer*) yang digunakan untuk mendeteksi ada tidaknya arus yang mengalir sehingga dengan begitu kita dapat mengetahui kondisi peralatan-peralatan listrik pada setiap

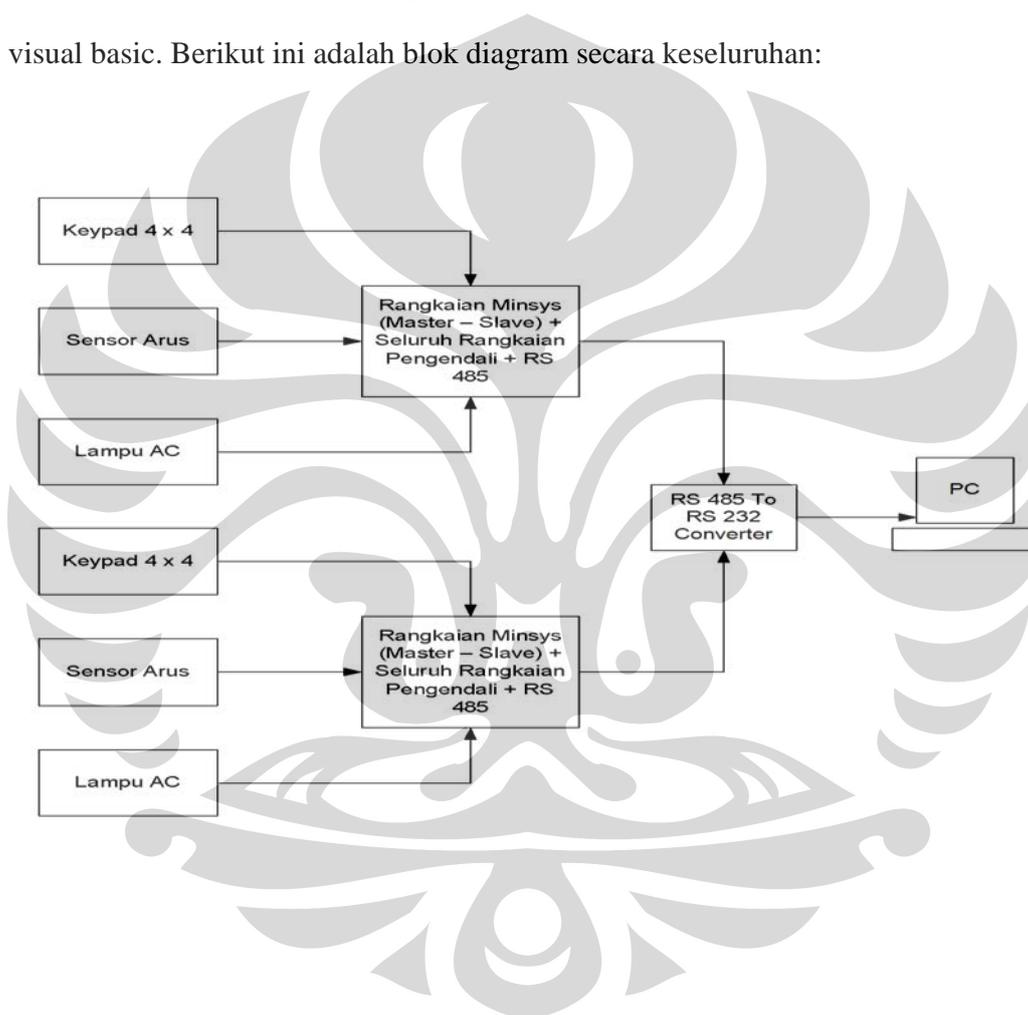
lantainya dan hal tersebut dapat mencegah terjadinya kebakaran yang biasa diakibatkan karena faktor manusia seperti lupa dalam mematikan peralatan-peralatan listrik serta blok pengatur kecerahan lampu dimana pada blok ini kecerahan lampu dapat diatur sesuai dengan pembacaan dari sensor OPT 101.

Pada tugas akhir ini, penulis mengerjakan “Aplikasi Sistem Kontrol Dengan Komunikasi Serial RS-485 Menggunakan Microcontroller Pada Bangunan Bertingkat”. Pembatasan masalah yang ditekankan adalah kemampuan sistem dari segi keamanan, untuk mengetahui kondisi perangkat listrik apa yang sedang menyala atau mati, kemampuan dalam mengatur kecerahan, dan pengalamatan sistem yang dapat diprogram dari komputer sentral sehingga pengamatan dan pengontrolan yang dilakukan tidak akan salah lantai.

1.4 DESKRIPSI SINGKAT

Seperti sistem aplikasi pintar lainnya, pada sistem aplikasi ini membutuhkan sebuah controller yang berfungsi sebagai ‘otak’ dari sistem tersebut yang mampu menerima data-data masukan, mengolahnya, dan menghasilkan outputnya. ‘Otak’ dari sistem aplikasi ini adalah sebuah microcontroller. Sebagai input dari sistem tersebut digunakan beberapa sensor seperti OPT 101 yang digunakan untuk mengatur kecerahan cahaya lampu dengan membandingkannya dengan intensitas dari luar seperti sinar matahari sehingga didapatkan kecerahan yang baik, sensor arus (*current transformer*) yang digunakan untuk mengetahui kondisi peralatan-peralatan listrik. Selain itu digunakan keypad matrix 4 x 4 yang digunakan sebagai sistem keamanan (*security*) sehingga sistem dapat mengetahui siapa-siapa saja yang telah memasuki lantai tersebut melalui komputer sentral..

Pada penelitian tugas akhir ini dari blok “Aplikasi Sistem Kontrol Dengan Komunikasi Serial RS-485 Menggunakan Microcontroller Pada Bangunan Bertingkat” dalam menerima request data-data keadaan rumah, proses pengolahan data tersebut dan pengirimannya dapat dilakukan atau disimulasikan, selain menggunakan program hyperterminal yang sudah disediakan oleh windows kita juga dapat mensimulasikannya melalui perangkat lunak lainnya seperti visual basic. Berikut ini adalah blok diagram secara keseluruhan:



Gambar 1.1 Blok diagram sistem

1.5. Metode Penulisan

Metode penelitian dalam pembuatan sistem kontrol gedung terbagi atas beberapa hal sebagai berikut :

1 Studi Literatur

Dalam studi ini penulis memperoleh informasi-informasi yang berkaitan dan erat kaitannya dengan penelitian yang dilakukan seperti melakukan diskusi dengan dosen pembimbing mengenai hal-hal yang berhubungan dengan perancangan serta pembuatan sistem selain itu informasi lain penulis dapatkan dengan mengacu kepada buku-buku, *data sheet*, informasi dari internet, serta rekan – rekan mahasiswa.

2 Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat yang penulis lakukan adalah suatu bentuk manifestasi dari studi literatur yang penulis lakukan selama ini. Dari studi literatur penulis dapat mengembangkan suatu sistem kontrol yang pernah ada sebelumnya, menambahkan serta melengkapi sistem sebelumnya sehingga menjadi suatu sistem yang lebih baik dan bermanfaat.

3 Pengujian Sistem dan Komponen Pendukung

Pada metode ini penulis melakukan pengujian terhadap alat yang dibuat, pengujian tersebut meliputi pengambilan data, testing *hardware*, *software* serta sistem komunikasi yang membangun sistem ini.

4 Pengumpulan Data

Data–data yang diperoleh penulis dapatkan dari pengujian alat, pengujian data sensor yang terintegrasi dengan suatu unit pengontrol.

5 Penulisan Hasil Penelitian

Langkah berikutnya adalah menulis serta menganalisa dari pengujian serta pengumpulan data yang telah dilakukan sebelumnya. Hal ini penting dilakukan untuk mengetahui seberapa jauh tingkat keberhasilan kita dalam merancang suatu sistem serta melengkapi perancangan sistem yang sama dimasa mendatang.

1.6. Sistematika Penulisan

Laporan tugas akhir ini terdiri dari beberapa bab dan memuat beberapa sub bab. Agar lebih memper-mudah pemahaman dan pembacaan, maka laporan tugas akhir ini disusun menjadi seperti di bawah ini :

BAB I. PENDAHULUAN

Pada bab ini berisi tentang latar belakang, tujuan penelitian, pembatasan masalah, metode penelitian dan sistematika penulisan.

BAB 2. TEORI DASAR

Pada Bab ini berisi tentang konsep sebagai hasil pembelajaran studi literatur mengenai cara kerja dari perancangan sistem kontrol gedung baik hardware maupun software.

BAB 3. PERANCANGAN SISTEM

Pada bab ini meliputi penjelasan mengenai cara kerja dari semua sistem yang dibuat. Menjelaskan secara detail dari perancangan sistem secara hardware, software yang dibuat secara keseluruhan.

BAB 4 PENGUJIAN SISTEM DAN PENGAMBILAN DATA

Pada bab ini terdiri atas pengujian dari parameter-parameter dari system yang dibuat. Baik itu pengujian secara *hardware* maupun *software*. Pada bab ini juga dijelaskan metode pengambilan data yang dilakukan, menjelaskan sejauh mana keberhasilan dari perancangan yang telah dibuat. Selanjutnya data - data yang didapat dari pengetesan hardware dan software dapat kita analisa sebagai hasil perbandingan dengan nilai perhitungan yang sebenarnya.

BAB 5. KESIMPULAN

Pada Bab ini akan dikemukakan point – point penting yang penulis dapatkan selama penelitian yang telah dilakukan yang berkaitan dengan sistem kontrol gedung.

BAB 2

TEORI DASAR

Pada penelitian ini diperlukan teori-teori yang melandasi penelitian ini antara lain :

2.1 Arus Listrik

Kita semua tentu paham bahwa arus listrik terjadi karena adanya aliran elektron dimana setiap elektron mempunyai muatan yang besarnya sama. Jika kita mempunyai benda bermuatan negatif berarti benda tersebut mempunyai kelebihan elektron. Derajat termuatinya benda tersebut diukur dengan jumlah kelebihan elektron yang ada. Muatan sebuah elektron, sering dinyatakan dengan simbol q atau e , dinyatakan dengan satuan coulomb, yaitu sebesar :

$$q = 1.6 \times 10^{-19} \text{ coulomb}$$

Misalkan kita mempunyai sepotong kawat tembaga yang biasanya digunakan sebagai penghantar listrik dengan alasan harganya relatif murah, kuat dan tahan terhadap korosi. Besarnya hantaran pada kawat tersebut hanya tergantung pada adanya elektron bebas (dari elektron valensi), karena muatan inti dan elektron pada lintasan dalam terikat erat pada struktur kristal.

Pada dasarnya dalam kawat penghantar terdapat aliran elektron dalam jumlah yang sangat besar, jika jumlah elektron yang bergerak ke kanan dan ke kiri sama besar maka seolah-olah tidak terjadi apa-apa. Namun jika ujung sebelah kanan kawat menarik elektron sedangkan ujung sebelah kiri melepaskannya maka akan terjadi aliran elektron ke kanan (tapi ingat, dalam

hal ini disepakati bahwa arah arus ke kiri). Aliran elektron inilah yang selanjutnya disebut arus listrik. Besarnya arus listrik diukur dengan satuan banyaknya elektron per detik, namun demikian ini bukan satuan yang praktis karena harganya terlalu kecil. Satuan yang dipakai adalah ampere, dimana :

$$i = \frac{dq}{dt}$$

1 ampere = 1 coulomb / det

2.2 Cahaya

Cahaya merupakan energi berbentuk gelombang dan membantu kita melihat. Cahaya juga merupakan dasar ukuran meter dimana 1 meter bersamaan dengan jarak dilalui cahaya melalui vakum pada $1/299.792.458$ second. Kecepatan cahaya adalah 299.792.458 meter per second. Cahaya diperlukan dalam kehidupan sehari-hari. Sifat-sifat cahaya adalah cahaya bergerak lurus ke semua arah. Buktinya adalah kita dapat melihat sebuah lampu yang menyala dari segala penjuru dalam sebuah ruang gelap. Apabila cahaya terhalang, bayangan yang dihasilkan disebabkan cahaya yang bergerak lurus tidak dapat berbelok. Cahaya tampak merupakan bagian dari spektrum elektromagnetik. Terdapat beberapa hal yang perlu diingat bahwa bagian yang tampak dari spektrum elektromagnetik tersebut akan berlaku juga pada dasarnya untuk semua bagian lain dari spektrum elektromagnetik tersebut. Perbedaan yang utama adalah di dalam cara menghasilkan dan mendeteksi di dalam berbagai jangkauan spektrum.

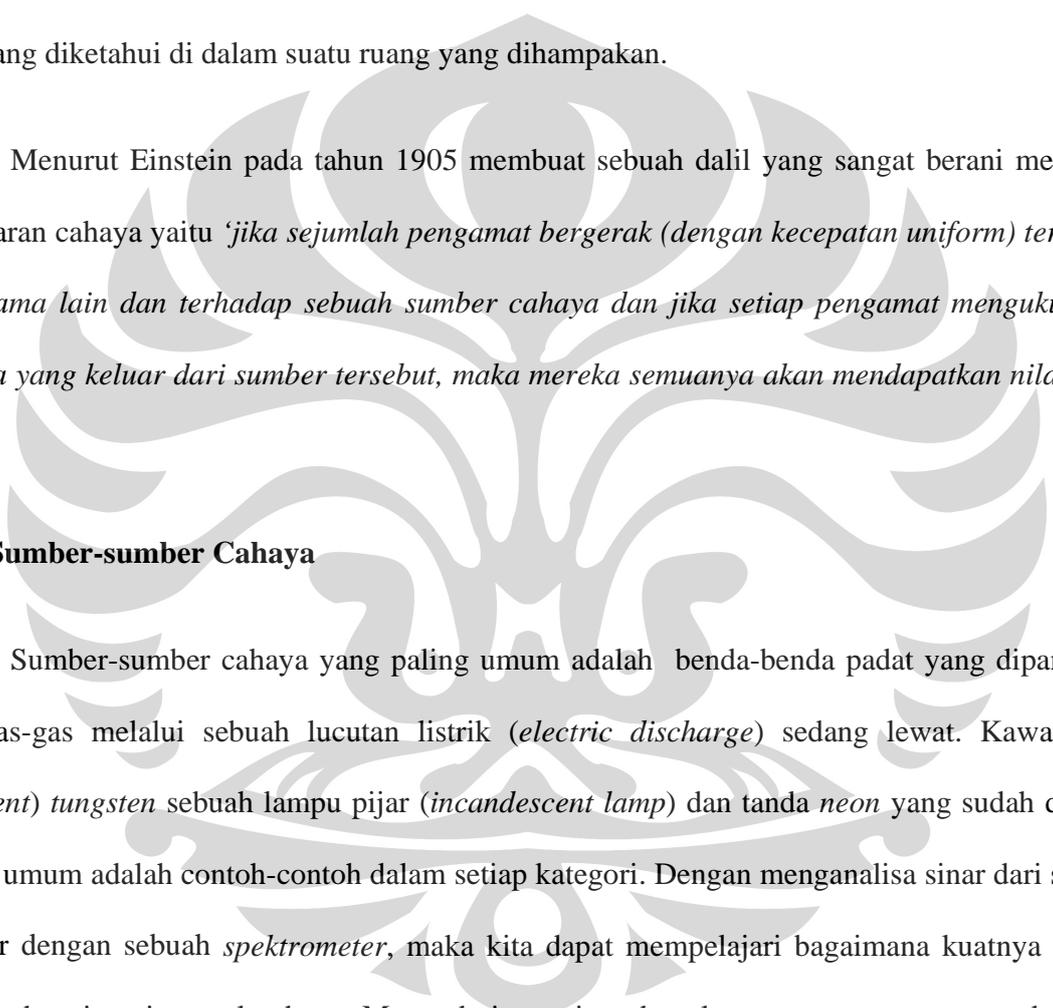
2.2.1 Sifat dan Penjalaran Cahaya

Dalam hal ini cahaya (*light*) dapat didefinisikan sebagai radiasi yang dapat mempengaruhi mata. Batas spektrum tampak tidak didefinisikan dengan baik karena kurva sensitivitas mata mendekati sumbu tersebut secara asimptotis pada kedua panjang gelombang yang panjang dan yang pendek. Jika kita mengambil batas-batas tersebut secara sebarang, maka karena panjang gelombang pada sensitivitas mata telah menurun sampai menjadi 1% dari nilai maksimumnya, batas-batas ini adalah kira-kira 430 dan 690 nm., yang lebih kecil daripada sebuah faktor sebesar dua di dalam panjang gelombang. Mata dapat mendeteksi radiasi di luar batas-batas ini jika radiasi tersebut cukup kuat. Di dalam eksperimen fisika dapat menggunakan plat-plat fotografis atau detektor elektronik yang sensitif terhadap cahaya sebagai pengganti mata manusia.

Cahaya berjalan (merambat) begitu cepat sehingga tidak ada sesuatu di dalam pengalaman kita sehari-hari yang menganjurkan bahwa lajunya tidak ada berhingga. Untuk mengukur kecepatan besar secara langsung, maka kita harus mengukur suatu interval waktu yang kecil maupun harus menggunakan sebuah garis basis yang panjang. Pulsa-pulsa gelombang mikro direfleksikan dari bulan secara teratur. Laju cahaya (dan laju gelombang mikro) sekarang ini sudah diketahui besarnya dengan baik dari eksperimen-eksperimen lain sehingga kita akan menggunakan pengukuran-pengukuran ini untuk mengukur jarak bulan secara teliti.

Bila kita mengatakan bahwa laju cahaya di dalam ruang bebas adalah 3×10^8 m/s, maka kerangka referensi tidak dapat menyatakan medium melalui gelombang cahaya yang berjalan karena cahaya tidak memerlukan medium perambatan. Konsep sebuah gelombang yang tidak memerlukan medium tidaklah disukai oleh para fisikawan dari abad ke-19, karena mereka

dipengaruhi oleh analogi palsu di antara gelombang cahaya dan gelombang bunyi atau gangguan mekanis murni yang lain. Para fisikawan mendalilkan adanya *eter (ether)*, yang merupakan suatu zat yang renggang yang mengisi seluruh ruang dan bertindak sebagai medium transmisi untuk cahaya. *Eter* tersebut diharuskan mempunyai kerapatan yang sangat kecil sekali untuk menerangkan kenyataan bahwa *eter* tersebut tidak dapat diamati dengan menggunakan setiap cara yang diketahui di dalam suatu ruang yang dihampakan.

Menurut Einstein pada tahun 1905 membuat sebuah dalil yang sangat berani mengenai penjalaran cahaya yaitu '*jika sejumlah pengamat bergerak (dengan kecepatan uniform) terhadap satu sama lain dan terhadap sebuah sumber cahaya dan jika setiap pengamat mengukur laju cahaya yang keluar dari sumber tersebut, maka mereka semuanya akan mendapatkan nilai yang sama*'.


2.2.2 Sumber-sumber Cahaya

Sumber-sumber cahaya yang paling umum adalah benda-benda padat yang dipanaskan dan gas-gas melalui sebuah lucutan listrik (*electric discharge*) sedang lewat. Kawat pijar (*filament tungsten*) sebuah lampu pijar (*incandescent lamp*) dan tanda *neon* yang sudah dikenal secara umum adalah contoh-contoh dalam setiap kategori. Dengan menganalisa sinar dari sebuah sumber dengan sebuah *spektrometer*, maka kita dapat mempelajari bagaimana kuatnya radiasi pada berbagai panjang gelombang. Massa dari uap air pada volume tertentu campuran udara atau gas, dan umumnya dilaporkan dalam gram per meter kubik (g/m^3).

2.3 Microcontroller AVR ATMEGA32

ATMEGA32 adalah *microcontroller* dengan *single chip computer* yang memiliki kemampuan untuk diprogram dan digunakan sebagai pengontrolan. *Microcontroller* ini termasuk dalam keluarga CMOS tegangan rendah 8-bit berdasarkan AVR arsitektur RISC (*Reduced Instruction Set Computer*), setiap instruksi dengan menggunakan *microcontroller* Atmega32 dapat dieksekusi dalam satu siklus *clock oscillator*. *Microcontroller* ini menghasilkan 1MIPS (*Million Instruction Per Second*) per MHz sehingga konsumsi daya bisa optimal dan kecepatan proses eksekusi menjadi maksimal.

2.3.1 Fitur Dan Arsitektur AVR ATMEGA32

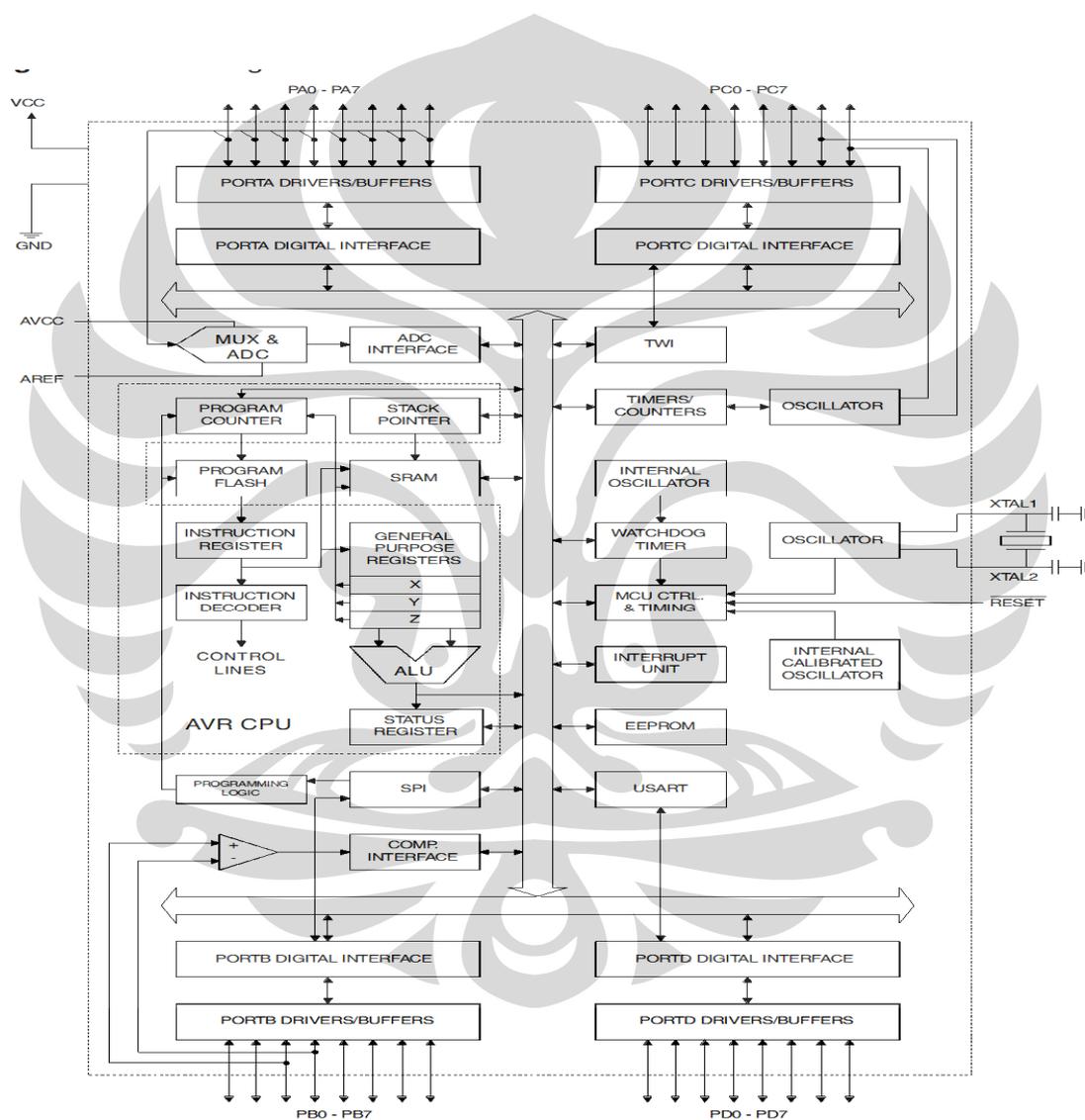
Berikut ini adalah fitur-fitur utama yang dimiliki oleh *microcontroller* Atmega8535:

Microcontroller Atmega32 memiliki fitur :

- a. 32 KByte untuk *System Programmable Flash*
- b. Memori data EEPROM 1024 Byte
- c. Memori data SRAM 2 K Byte
- d. 32 Jalur Input-output
- e. 32 Register umum
- f. 3 *Flexible Timer / counter* dengan *Compare-Modes*
- g. Internal dan eksternal *Interrupt*
- h. Pemrograman serial dengan USART
- i. Interface serial Two-wire sebesar 1Byte
- j. 8 Single Ended Channels, 10 bit untuk ADC dan 4 kanal PWM
- k. Watchdog timer yang dapat deprogram dengan Internal Oscillator

- l. Sebuah serial port SPI
- m. 6 pilihan software dengan power save mode yaitu Idle, ADC Noise Reduction, Power save, Power down, Standby dan Extended Standby
- n. Memiliki kecepatan 0-8 MHz, dan beroperasi pada tegangan 2.7-5.5

Berikut adalah gambar arsitektur dari microcontroller ATMEGA32 :



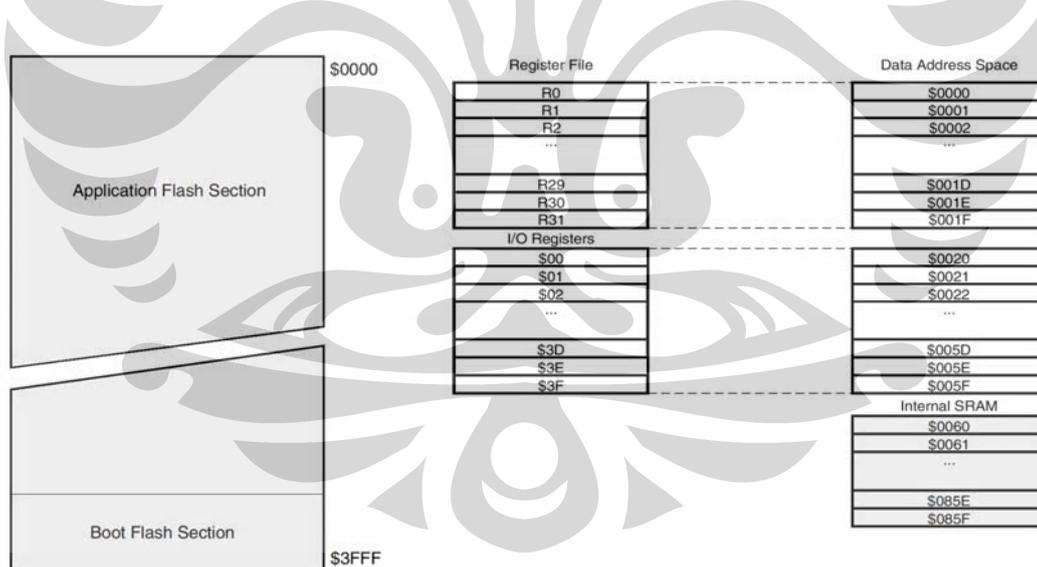
Gambar 2.1 Blok diagram microcontroller AVR ATMEGA32

2.3.2 Pengorganisasian Memori

Microcontroller AVR ATMEGA32 memiliki dua ruang utama memori yaitu memori data dan memori program. Sebagai tambahan terdapat memori EEPROM untuk menyimpan data. Berikut ini akan dijelaskan satu-per satu mengenai ketiga media penyimpanan tersebut:

2.3.2.1 Program Memory

ATMEGA32 terdiri dari 32 Kbyte On-Chip In-system Reprogrammable Flash Memory untuk penyimpanan program. Selama besar instruksinya adalah 16 sampai 32 bit, Flash akan terorganisasi sebesar 16K x 16 [1]. Untuk keamanan software, memori program *Flash* dibagi menjadi dua bagian yaitu Program Boot dan Program Aplikasi.



Gambar 2.2 Memori program dan memori data AVR ATMEGA32

Memori Flash memiliki kemampuan paling tidak 1000 kali dihapus dan ditulis. Program Counter-nya (PC) sebesar 14 bit, terletak pada alamat 16K pada memori program [1].

2.3.2.2 Data Memory

Pada gambar 2.2 ditunjukkan bagaimana SRAM memori pada ATMEGA32 diorganisasikan. 2144 lokasi alamat data memori berisi register file, I/O memori, dan internal data SRAM. 96 lokasi alamat perlama berisi register file, dan I/O memori dan selanjutnya 2048 lokasi alamat dari internal data SRAM. Terdapat lima mode pengalamatan yang berbeda pada data memori, yaitu: mode pengalamatan langsung, tidak langsung dengan displacement, tidak langsung, tidak langsung dengan pre-decrement, dan tidak langsung dengan post-increment. 32 register fungsi umum, 64 I/O register, dan 2048 byte internal data SRAM pada ATMEGA8535 secara keseluruhan dapat diakses menggunakan semua mode pengalamatan [1].

2.3.2.3 EEPROM

ATMEGA32 memiliki EEPROM internal sebesar 1024 byte. EEPROM memiliki kemampuan sampai 100.000 tulis/hapus. Untuk melakukan pengaksesan EEPROM dan CPU dideskripsikan dalam dalam EEPROM Address Register, EEPROM Data Register, dan EEPROM Control Register [1]. Memori EEPROM bersifat non volatile yang artinya saat supply tegangan mati maka data yang tersimpan didalamnya tidak akan hilang.

2.3.3 Konfigurasi Pin

Microcontroller AYMEGA32 yang diproduksi oleh ATMEL, memiliki beberapa fasilitas yang mendukung pembuatan *system controller* ini. Berikut akan diterangkan mengenai fungsi dari masing-masing port dari IC AVR yaitu:

- a. *PORT A (A0-A7)* : Port A selain sebagai *port I/O 8-bit bi-directional*, juga melayani *input analog* sebagai *A/D converter*.

Tabel 2.1 Fungsi alternatif PORT A

Port Pin	Alternate Function
PA7	ADC7 (ADC input channel 7)
PA6	ADC6 (ADC input channel 6)
PA5	ADC5 (ADC input channel 5)
PA4	ADC4 (ADC input channel 4)
PA3	ADC3 (ADC input channel 3)
PA2	ADC2 (ADC input channel 2)
PA1	ADC1 (ADC input channel 1)
PA0	ADC0 (ADC input channel 0)

- b. *Port B (B0-B7)* : *Port B* merupakan *port I/O 8-bit bi-directional*, selain itu *port B* juga mempunyai beberapa fungsi tambahan seperti terlihat pada tabel berikut :

Tabel 2.2 Fungsi alternatif PORT B

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	\overline{SS} (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

- c. *Port C (C0-C7)*: merupakan *port I/O 8-bit bi-directional*, selain itu *port C* juga mempunyai fungsi alternatif, yaitu :

Tabel 2.3 Fungsi Alternatif PORT C

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

- d. *PORT D (D0-D7)*: merupakan *port I/O 8-bit bi-directional*, selain itu *port D* juga mempunyai fungsi alternatif, yaitu :

Tabel 2.4 Fungsi Alternatif PORT D

Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

- e. *VCC: Power Supply*
- f. *GND : Ground*
- g. *RESET*

Reset input. Kondisi *low* berlangsung lebih panjang dari panjang pulsa minimum akan membuat *microcontroller* berada pada kondisi *reset*. Pulsa yang lebih pendek tidak menjamin timbulnya kondisi *reset*.

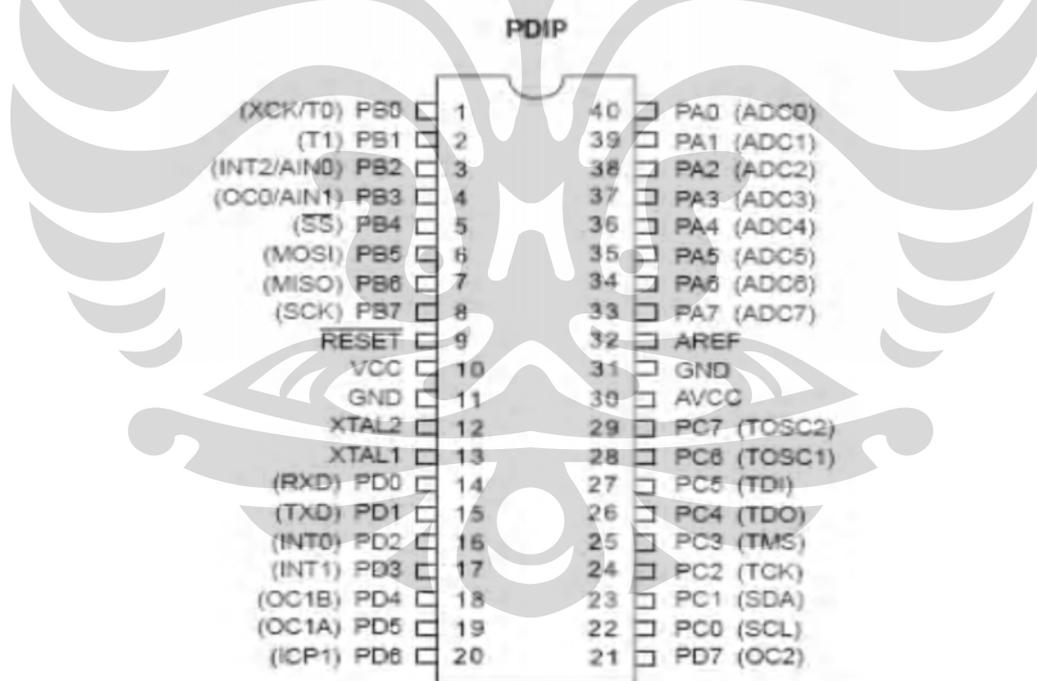
h. AVCC

AVCC adalah tegangan supply untuk input Port analog (PORT A) dan AID *converter*. Saat ADC tidak dipakai AVCC terhubung ke VCC, dan pada saat ADC dipakai AVCC terhubung ke VCC yang melalui *Low Pass filter*.

i. AREF: AREF adalah referensi *analog* untuk AID *converter*

j. XTAL1: *Input* untuk *inverting oscillator amplifier* dan *input* untuk *clock internal*.

k. XTAL2: Output *inverting oscillator amplifier*.



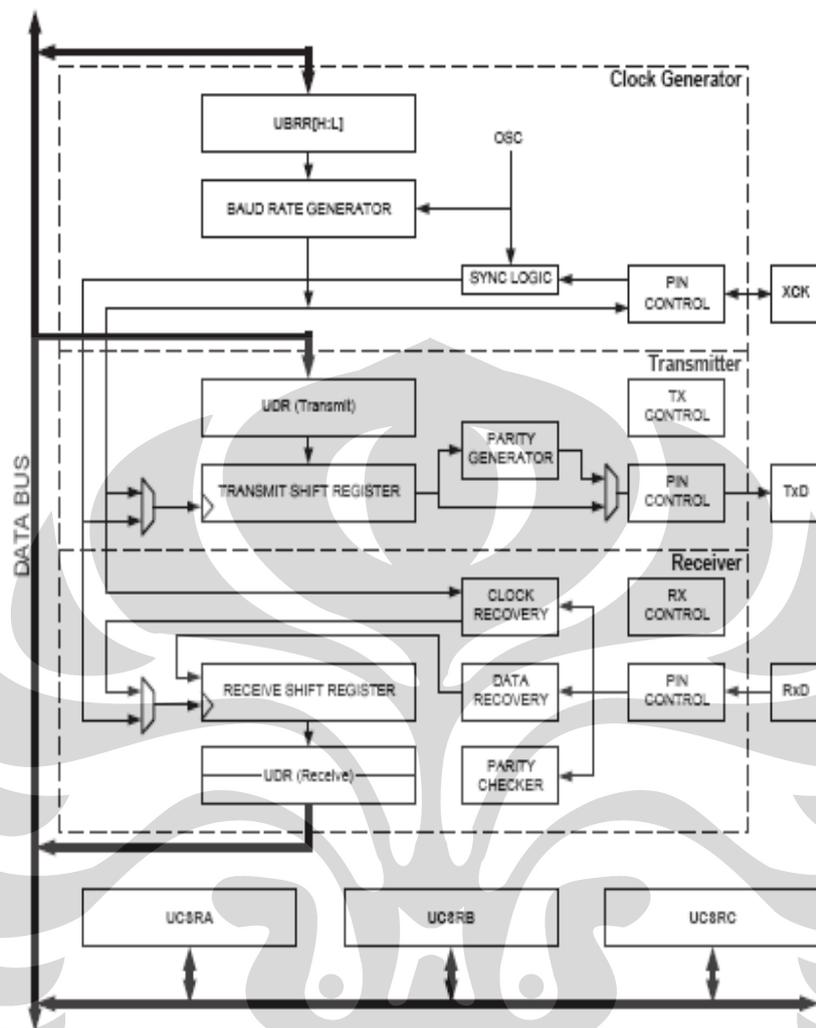
Gambar 2.3 Pin-Pin Pada ATMEGA32

Gabungan dari inti AVR yaitu set instruksi yang banyak dengan register kerja tujuan umum yang kesemuanya sebanyak 32 register secara langsung terhubung dengan *Arithmetic Logic Unit* (ALU), melewati 2 register yang tetap untuk akses instruksi untuk mengakses dalam satu Instruksi yang dieksekusi dalam satu *cycle*. Arsitektur menghasilkan kode efisiensi ketika mampu lebih cepat 10 kali daripada *Microcontroller* konvensional CISC.

2.3.4 USART

Universal Synchronous and Asynchronous Receiver Transmitter (USART) piranti Komunikasi serial dengan fleksibilitas yang tinggi. Fasilitas ini berguna agar sistem aplikasi AVR dapat berkomunikasi dengan system lainnya [1]. Fitur-fitur yang dimiliki USART adalah:

- a. Operation *Full duplex* (memiliki register serial Rx dan Tx bebas)
- b. Operasi sinkron dan asinkron
- c. Operasi *Master or Slave Clocked Synchronous*
- d. Generator berkecepatan dan beresolusi tinggi
- e. Didukung *Serial Frame* dengan 5.6.7.8. atau 9 bit data dan 1 atau 2 stop bit
- f. Generasi paritas ganjil atau genap dan pengecekan paritas yang didukung oleh *hardware*
- g. Deteksi data yang berlebih
- h. Deteksi kesalahan frame
- i. Penyaringan *noise* termasuk deteksi kesalahan bit pertama dan *digital low pass filter*
- j. Memiliki tiga bu interrupt yang terpisah. Tx *Complete*, Tx Data Register Empty, dan Rx *Complete*
- k. Mode komunikasi *Multi-Processor*
- l. Mode komunikasi Asynchronous dengan kecepatan ganda.



Gambar 2.4 Blok diagram USART

USART harus terlebih dahulu diinisialisasi, tahap penginisialisasian terdiri atas *baudrate*, *setting frame format*, meng-*enable*-kan transmitter dan receiver sesuai penggunaan. Untuk pemakaian interrupt pada USART terlebih dahulu harus mengaktifkan *global interrupt flag* terlebih dahulu [1].

Pemakaian *Transmitter* USART, dengan cara mengatur *Transmit Enable* pada register UCSRB. Setelah itu operasi normal port pada, pin TxD ditolak oleh USART dan akan

difungsikan sebagai *Transmitter's Serial Output*. Pada saat operasi sinkron terjadi, *clock* dari pin XCK tidak digunakan sebagai gantinya digunakan *clock* pengiriman (transmisi) [1].

Pemakaian *Receiver* USART, dengan cara mengatur *Receive Enable* pada register UCSRB Setelah itu operas] normal port pada pin RxD ditolak oleh USART dan akan -lifungsikan sebagai *Receiver's Serial Input*. Pada saat operasi sinkron terjadi *Clock* dari pin XCK digunakan sebagai *Transfer Clock* [1].

2.3.5 Interrupt

AVR menyediakan berbagai sumber interrupt yang berbeda. Pada interrupt tersebut dan pembagian tiap-tiap reset vector memiliki pembagian Program vector didalam program memori. Keseluruhan interrupt ditandai dengan pengaktifan sendiri tiap bit dimana harus ditulis dalam logika *high* (logika satu) secara bersama dengan mengaktifkan bit global interrupts yang terletak di dalam status register. Secara default, pengalamatan paling rendah dalam program memori didefinisikan sebagai reset dan *interrupts vector*. Daftar secara lengkap dapat dilihat seperti tabel 2.1, pada daftar tersebut juga menentukan prioritas level pada tiap-tiap interrupt yang berbeda. Alamat paling rendah, memiliki prioritas interrupt yang paling tinggi. Reset memiliki prioritas yang tertinggi, kemudian INTO – External interrupt request 0 [1].

Tabel 2.5 Reset dan Interrupt vector

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

2.4 Komunikasi Serial

Komunikasi serial merupakan salah satu cara untuk mengkomunikasikan data dari suatu peralatan ke peralatan lain dengan cara pengiriman bit-perbit. Misalnya komunikasi antara PC dengan PC, microcontroller dengan PC, dll. Pada PC, komunikasi serial RS232 dapat dilakukan melalui port serial (COM port).

Komunikasi serial terdiri dari dua jenis, yang pertama adalah komunikasi *synchronous* dan yang kedua adalah *asynchronous*. Komunikasi serial *asynchronous* mempunyai ciri khas yang terletak pada adanya start bit dan stop bit. Berbeda dengan komunikasi *synchronous* yang menggunakan *clock osilator* sebagai parameter pembeda data bit yang dikirim. Pada komunikasi serial *asynchronous*, sinyal data dikirim dengan menambahkan bit awal dan bit akhir pada data sebagai penanda, jadi antara data yang satu dengan yang lainnya mempunyai tanda pemisah sehingga alat tidak salah dalam menerjemahkan data. Sistem yang dirancang oleh penulis menggunakan jenis komunikasi *asynchronous*, sehingga yang akan dijelaskan lebih lanjut adalah komunikasi *asynchronous*.

2.4.1 Komunikasi Asynchronous

Kedatangan data pada bagian akhir jalur penerimaan data dalam pemindahan data serial seluruhnya adalah 0 dan 1, hal ini sulit untuk membuat pengertian dari sebuah data kecuali kalau pengirim dan penerima menyetujui pada sebuah kumpulan aturan *protocol*, bagaimana data tersebut dipaketkan, berapa banyak bit merupakan karakter, dan kapan data berawal dan berakhir [5]. Beberapa parameter penting yang terdapat pada komunikasi serial *asynchronous* akan dijelaskan sebagai berikut :

2.4.1.1 Start Bit Dan Stop Bit

Komunikasi data serial *asynchronous* secara luas digunakan untuk transmisi berorientasi karakter; dan pemindahan orientasi blok data menggunakan metoda *synchronous* [5]. Dalam metoda *asynchronous*, setiap karakter terletak antara *bit start* dan *bit stop*. Hal ini disebut *Framing*. Dalam *framing* data untuk komunikasi *asynchronous*, data, seperti karakter ASCII,

dipaketkan berada antara sebuah bit *start* dan bit *stop*. Bit *start* selalu satu bit tetapi bit *stop* dapat satu atau dua bit. Bit *start* selalu 0 (*low*) dan bit *stop* adalah 1 (*high*) [5].

2.4.1.2 Parity Bit

Dalam beberapa sistem untuk memelihara integritas data, *parity* bit karakter *byte* terdapat dalam *frame* data. Hal ini berarti untuk tiap-tiap karakter (7 atau 8 bit, bergantung pada sistem) memiliki bit *parity* tunggal dalam penambahan pada bit *start* dan bit *stop* [5]. Bit *parity* adalah *odd* atau *even*. Dalam kasus ini banyaknya bit data sebuah bit *odd-parity*, termasuk bit *parity*, memiliki sebuah nilai ganjil dari banyaknya 1. Dengan cara yang sama, banyaknya bit keseluruhan dalam sebuah bit *even-parity*, termasuk bit *parity*, adalah genap [5].

2.4.1.3 Kecepatan Pemindahan Data (*Baud Rate*)

Kecepatan pemindahan data dalam komunikasi data serial ditetapkan dalam *bps* (*bits per second*) [5]. Hal yang lebih luas lainnya lagi digunakan terminologi untuk *bps* adalah *baud rate*. *Baud rate* adalah terminologi pada modem dan didefinisikan sebagai banyaknya perubahan sinyal tiap detik. Kecepatan pemindahan data yang diberikan sistem komputer bergantung pada port komunikasi yang tergabung dalam sistem.

2.4.2 Standar Serial RS-232

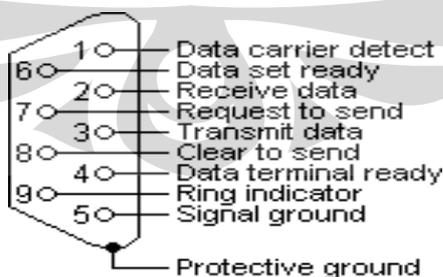
Untuk menyediakan kompatibilitas diantara peralatan komunikasi data yang dibuat oleh berbagai manufaktur, sebuah standar penghubung yang disebut RS232 dibuat oleh EIA (*Electronic Industries Associates*) pada tahun 1960. Saat ini RS232 digunakan secara luas sebagai standar penghubung I/O serial.

Standar yang dibuat sebelum kedatangan keluarga logika TTL, level tegangan *input* dan *output* tidak cocok dengan TTL. Pada RS232 sebuah logika 1 direpresentasikan oleh -3 hingga -25 V, sementara itu bit 0 adalah $+3$ hingga $+25$ V, sehingga membuat -3 hingga $+3$ tidak terdefinisi. Karena alasan ini, untuk menghubungkan RS232 apapun pada system berbasis mikroprosesor, kita harus mengubah level logika TTL ke level tegangan RS232. Fungsi untuk menterjemahkan dari TTL ke level tegangan RS232 disebut *line drivers* dan fungsi untuk mengkonversi dari RS232 ke level tegangan TTL disebut *line receivers*.

Pada tugas akhir ini, untuk mengkonversi beda level tegangan tersebut penulis menggunakan IC Max-232. Di dalam IC ini terdapat *Charge Pump* yang akan membangkitkan $+10$ Volt dari sumber dan $+5$ Volt tunggal yang dikemas dalam IC DIO (*Dual In Line Package*) 26 pin (8 pinx 2 baris) ini terdapat dua buah *transmitter* dan dua buah *receiver*.

2.4.3 Konfigurasi Pin DB-9

Berikut ini adalah konfigurasi dari pin DB-9 yang akan menghubungkan antara microcontroller dengan PC. Pada table 2.2 akan dijelaskan tiap-tiap fungsi dari pin DB-9.



Gambar 2.5 RS-232 DB 9 pinout

Tabel 2.6 Konfigurasi pin DB-9

Pin	Direction of signal
1	Carrier Deterc (CD) (from DCE) incoming signal from a modem
2	Received Data (RD) incoming data from a DCE
3	Transmitted Data (TD) Outgoing data to a DCE
4	Data Terminal Ready (DTR) Out going Handshaking signal
5	Signal ground commond reference voltage
6	Data Set Ready (DSR) incoming handshaking signal
7	Request To Send (RTS) outgoing flow control signal
8	Clear to send (CTS) incoming flow control signal
9	Ring Indicator (RI) (from DCE) incoming signal from a modem

2.4.4 Komunikasi RS-485

RS-485 memiliki banyak keuntungan jika dibandingkan dengan RS-232. Dengan menggunakan RS-485 dapat dipasang 32 driver/receiver perangkat pada sebuah jalur. Jarak jangkauannya bisa sampai sejauh 4000 kaki atau $\frac{3}{4}$ mil [2]. Dengan jarak yang lebih pendek, kecepatan transmisi data dapat sebesar 10M bit /detik [2]. Dan driver dan receiver RS-485 tidak terlalu mahal dan hanya memerlukan suplai tegangan 5 volt, berbeda dengan RS-232 yang memerlukan minimum suplai sebesar 5 volt output.

RS-485 menggunakan jalur yang seimbang, dimana hal tersebut berarti tiap-tiap sinyal memiliki dua kabel, dan sinyal pada kabel kedua sama dengan negative dari sinyal pada kabel pertama. Receiver RS-485 merespon perbedaan tegangan antara dua buah kabel tersebut. Istilah lain dari pengukuran tersebut adalah differential measurement [2].

Berbeda dengan hal tersebut, RS-232 menggunakan jalur yang tidak seimbang dimana tiap sinyal hanya memiliki satu kabel, dan receiver merespon perbedaan tegangan antara tiap kabel terhadap ground, yang akan digunakan oleh kedua buah sinyal tersebut. Istilah lain dari pengukuran tersebut adalah single-ended [2].

Penggunaan jalur yang seimbang memiliki banyak keuntungan bila dibandingkan dengan jalur yang tidak seimbang. Salah satunya adalah noise (tegangan yang tidak stabil, osilasi, atau interferensi secara umum). Pada jalur yang tidak seimbang, sinyal dari arus balik (return current) dalam kabel ground dapat menghasilkan noise pada receiver. Pada jalur yang seimbang, perbedaan sinyal menghasilkan dua persamaan tetapi berlawanan dengan arus balik (return current). Keuntungan lainnya pada jalur yang seimbang adalah tahan terhadap perbedaan potensial ground antara tiap titik [2]. Pada jaringan yang memiliki jalur yang panjang, potensial ground dapat berubah sebanyak sejumlah volt dari satu titik dengan lainnya. Tetapi differential measurement tidak akan peduli dengan perbedaan tersebut, dikarenakan pengukuran hanya pada tegangan antara dua buah sinyal pada kabel masing-masing [2].

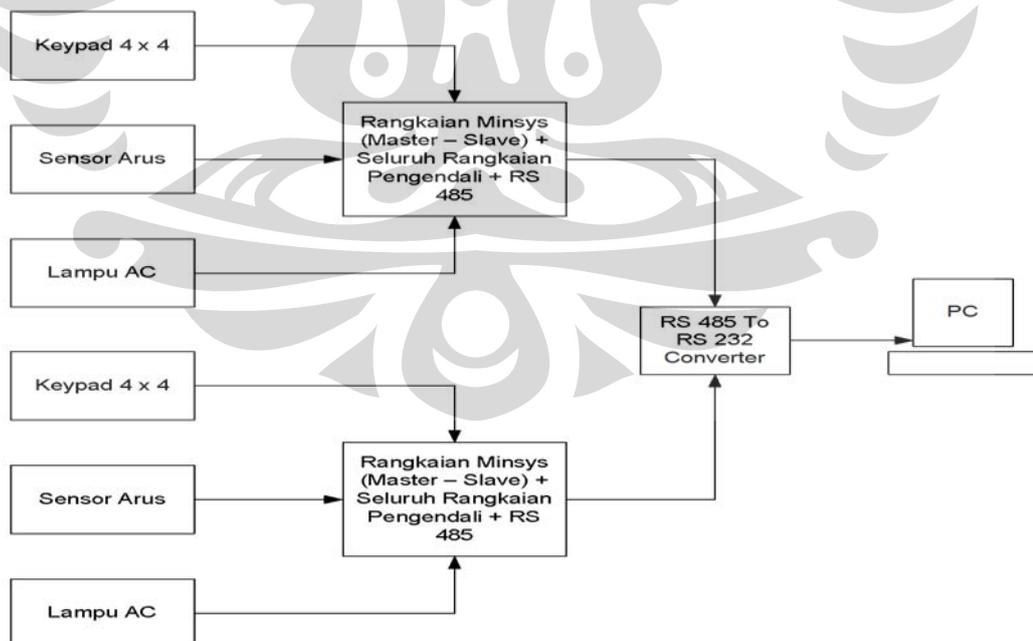
BAB 3

PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing *hardware* serta *software* yang digunakan penulis dalam pembuatan alat “Aplikasi Sistem Kontrol Dengan Komunikasi Serial RS 485 Menggunakan *Microcontroller* Pada Bangunan Bertingkat”.

3.1 Perancangan *Hardware*

Dalam bab ini selain perancangan alat, juga akan dibahas mengenai cara kerja alat. Blok diagram dari sistem kontrol dapat dilihat pada gambar di bawah ini.

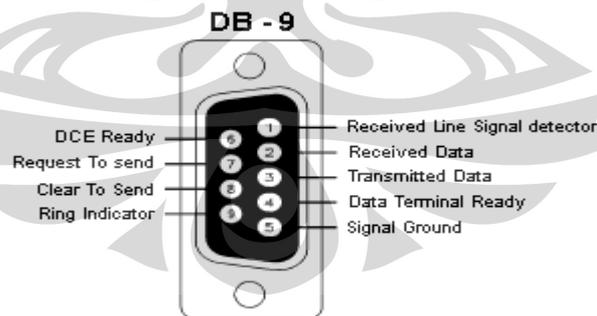


Gambar 3.1. Blok diagram sistem keseluruhan.

3.1.1 *Personal Computer (PC)*

Perancangan sistem kontrol ini menggunakan PC sebagai Visualisasi dari pengamatan serta pengontrolan terhadap parameter-parameter yang ada. Penggunaan PC dimaksudkan untuk mempermudah dalam melakukan pengamatan serta pengontrolan. Visualisasi tersebut menampilkan identifikasi *password*, pendeteksian menggunakan sensor arus (*current transformer*) serta kecerahan cahaya lampu 220 VAC melalui PC. PC tersebut akan melakukan komunikasi ke *microcontroller* baik yang berada pada lantai 1 maupun *microcontroller* yang terdapat pada lantai 2. Sistem komunikasi penerimaan dan pengiriman datanya melalui *port serial (COM)*.

Port serial (COM) pada komputer menggunakan slot DB-9 *male* yang terdapat di belakang komputer. Adapun konfigurasi slot DB-9 *male* adalah seperti gambar di bawah ini.



Gambar 3.2. Konfigurasi slot DB-9

Dibandingkan dengan menggunakan port parallel, penggunaan port serial terkesan lebih rumit. Beberapa keuntungan penggunaan port serial dibandingkan dengan port parallel adalah pada komunikasi serial masalah *cable loss* tidak akan

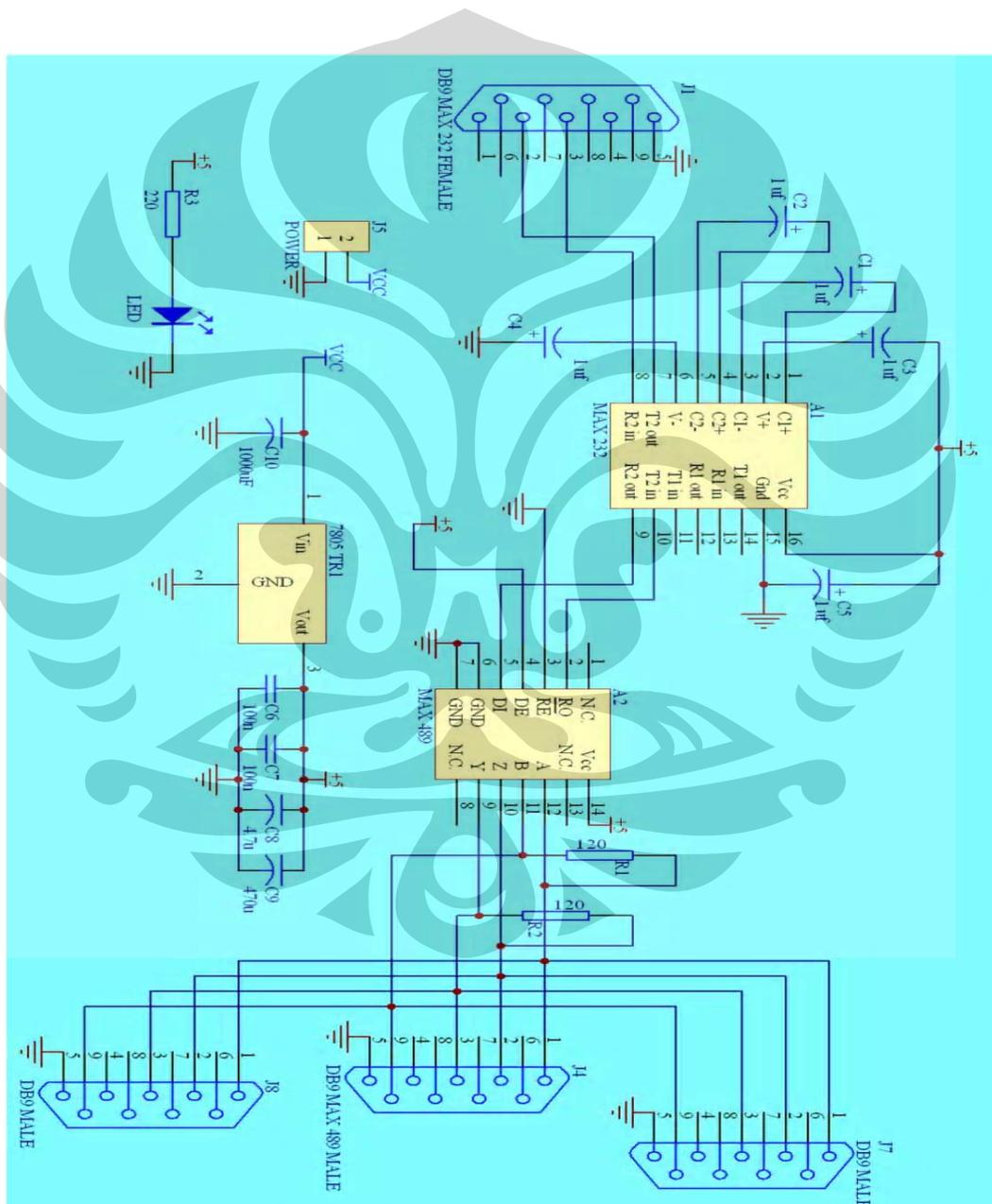
menjadi masalah besar daripada menggunakan kabel parallel. Port serial mentransmisikan “1” pada level tegangan -3 Volt sampai -25 Volt dan “0” pada level tegangan +3 Volt sampai +25 Volt, sedangkan port parallel mentransmisikan “0” pada level tegangan 0 Volt dan “1” pada level tegangan 5 Volt. Jumlah kabel yang dibutuhkan pada komunikasi serial jauh lebih sedikit, bisa hanya menggunakan tiga kabel, yaitu saluran transmit data, saluran receive data dan saluran ground selain itu lebih compatible dengan *microcontroller*. Hal itu dikarenakan *microcontroller* telah dilengkapi dengan SCI (*Serial Communication Interface*) yang dapat digunakan untuk komunikasi dengan port serial computer. Gambar di bawah ini adalah susunan konfigurasi dari pin DB 9.berikut ini :

Tabel 3.1. Susunan pin port serial (COM).

9 Pin Connector on a DTE device (PC connection)	
Male RS232 DB9	
Pin Number	Direction of signal:
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

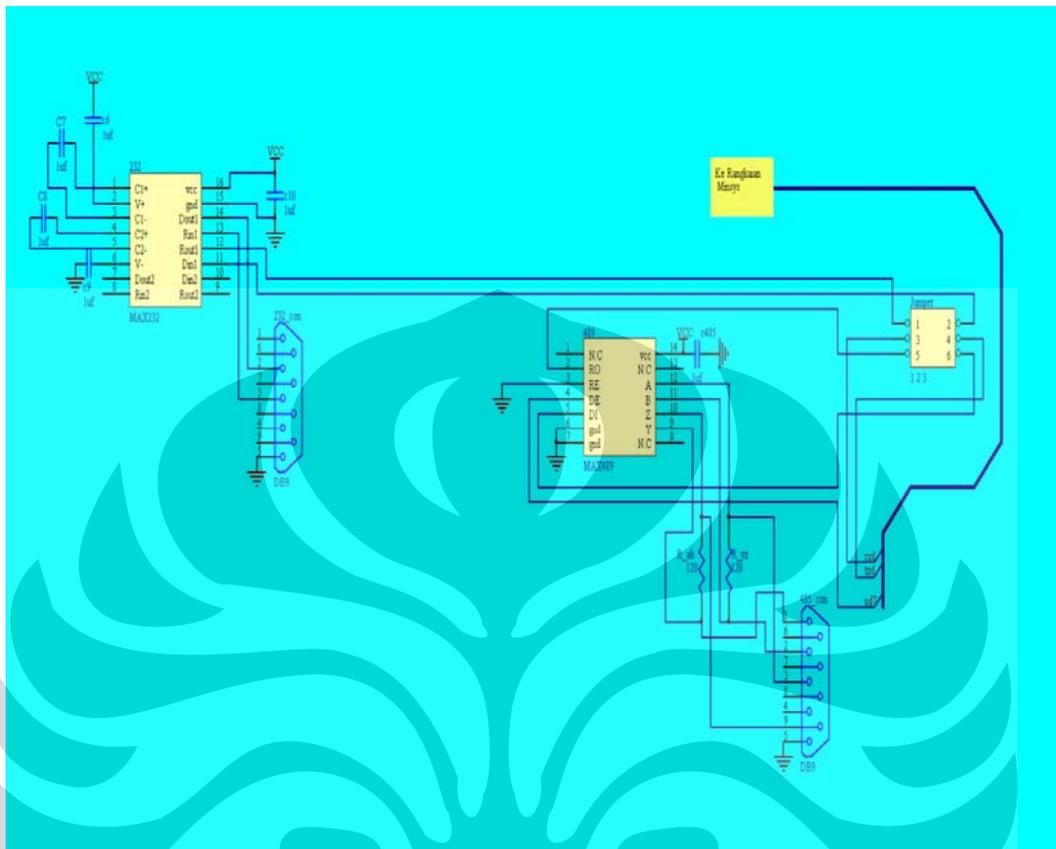
3.1.2 Rangkaian RS 485 to RS 232 Converter

Untuk mengkoneksikan antara PC dengan *microcontroller* diperlukan suatu rangkaian yang dapat mengubah tegangan PC ke tegangan *microcontroller* yang biasa kita dengan *RS 485 to RS 232 Converter*. Adapun pemilihan sistem komunikasi ini dimaksudkan data pengirimannya menjadi lebih jauh.



Gambar 3.3. Rangkaian RS-485 to RS-232 Converter

3.1.3 Rangkaian Komunikasi Serial



Gambar 3.4 Rangkaian komunikasi serial RS-232 dan RS-485

Rangkaian diatas tidak bekerja secara bersamaan. Saat jumper dihubungkan pada pin 1-3 dan 2-4 dari *header*, maka kaki TXD dan RXD mikrokontroler akan terhubung ke rangkaian *converter* RS-232, sedangkan saat jumper dihubungkan pada pin 2-5 dan 4-6 dari *header*, maka kaki TXD dan RXD mikrokontroler akan terhubung ke rangkaian RS-485.

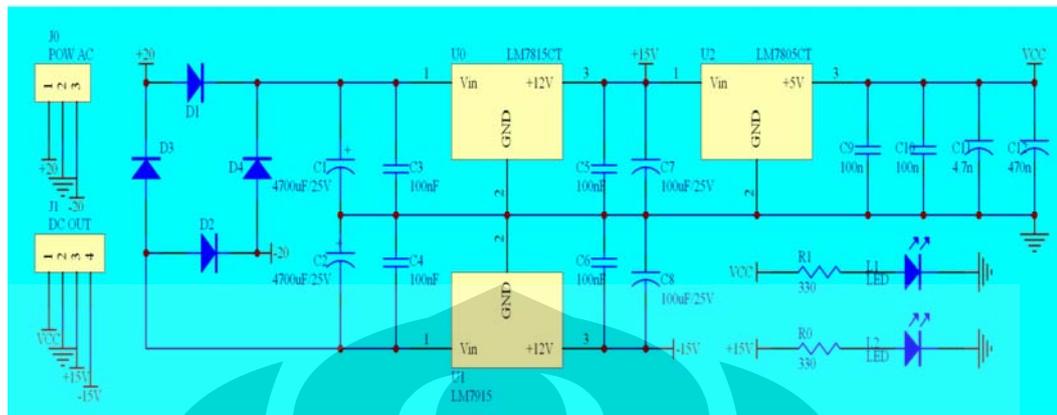
Pada rangkaian RS-485 menggunakan IC max 489 yang bersifat *full duplex* sehingga pada saat melakukan pengiriman data, mikrokontroler dapat melakukan penerimaan data yang lain. Agar dapat melakukan pengiriman dan penerimaan data serial, kaki *enable driver* dan *receiver* pada IC MAX 489 harus diaktifkan

terlebih dahulu. Untuk mengaktifkan *enable driver* pada IC MAX 489 dengan memberikan logika high atau “1” pada kaki tersebut dengan men-set pind.7 mikrokontroler, sedangkan untuk mengaktifkan *enable receiver* dengan memberikan logika low atau “0” pada kaki tersebut. Secara default, kaki *enable receiver* IC MAX 489 sudah terhubung ke ground hal ini bertujuan agar setiap saat sistem siap menerima paket data serial dari PC. Secara *software*, tidak ada perbedaan antara program komunikasi serial RS-232 maupun RS-485.

Level tegangan pada mikrokontroler adalah level tegangan TTL yang hanya mengenal logika 1 untuk 3-5 V dan logika 0 untuk 0-0.8 V, sedangkan pada PC memiliki level tegangan RS-232 yang menganggap tegangan -3 s/d -25 V sebagai logika 1 dan tegangan 3 s/d 25 V sebagai logika low (0). Saat menggunakan komunikasi RS-232 memiliki kendala yaitu masalah keterbatasan jarak dan rentan terhadap noise.

Untuk mengatasi keterbatasan koneksi, digunakan komunikasi RS-485 yang dapat digunakan sampai sejauh 4000 kaki dan lebih kecil kemungkinan terkena gangguan interferensi sinyal dari luar [2], barulah nantinya jalur komunikasi ini terhubung ke rangkaian RS-485 – RS-232 *converter* pada saat akan dihubungkan ke PC untuk mengkonversi beda level tegangan. Fungsi dari rangkaian RS-485 diatas adalah agar sistem yang ada pada rumah dapat berkomunikasi dengan PC sentral, walaupun dengan jarak yang jauh.

3.1.4 Rangkaian Regulator

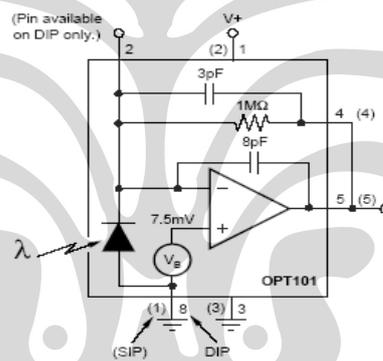


Gambar 3.5 Rangkaian regulator

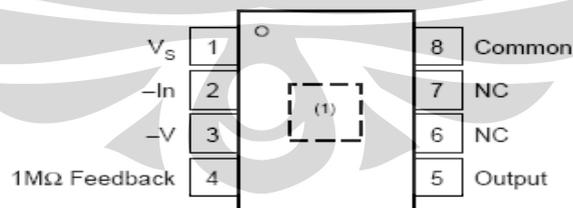
Gambar diatas adalah rangkaian regulator yang berfungsi memberi supply tegangan ke seluruh sistem sesuai dengan keluaran dan kebutuhan dari masing-masing rangkaian. Tegangan AC dari trafo disearahkan oleh diode bridge sehingga menghasilkan tegangan DC. Setelah di searahkan maka tegangan di regulasi agar menghasilkan tegangan +15 V dengan dimasukkan ke IC LM 7815. Penggunaan kapasitor elcho sebelum dan sesudah melewati IC regulator yakni untuk meminimalkam riak dari tegangan DC yang dihasilkan pada saat proses penyearahan oleh diode bridge, sehingga dihasilkan tegangan DC yang lebih stabil. Sedangkan penggunaan kapasitor mika / non polar dimaksudkan untuk menghindari adanya RF (radio frekuensi) yang dapat mengganggu kestabilan dari tegangan tersebut. Sedangkan output tegangan – 15 dihasilkan akibat pembalikan fasa gelombang pada saat terjadi penyearahan tegangan dan setelah melalui proses regulasi dari IC LM7915. Adapun output tegangan + 5 V dihasilkan dari output tegangan IC LM7815 berfungsi sebagai input tegangan bagi IC LM7805 dan setelah melalui proses regulasi didapatkan tegangan sebesar + 5 V.

3.1.5 Rangkaian Sensor OPT 101

Sensor OPT101 adalah sensor yang digunakan untuk mengukur tingkat kecerahan atau intensitas. Sensor ini menggunakan sensor *photodiode* dengan jenis OPT 101. pada umumnya *photodiode* akan menghasilkan arus apabila diberikan intensitas cahaya tertentu. Sedangkan pada sistem ini menggunakan OPT 101 karena *output*-nya sudah berupa tegangan. Karena didalam OPT 101 tersebut memiliki *internal* rangkaian pengkondisi sinyal yaitu berupa rangkaian RC serta sebuah *operational amplifier (op-amp)*. Adapun berikut ini merupakan gambar dari OPT 101.



Gambar 3.6 Rangkaian *internal* OPT 101.

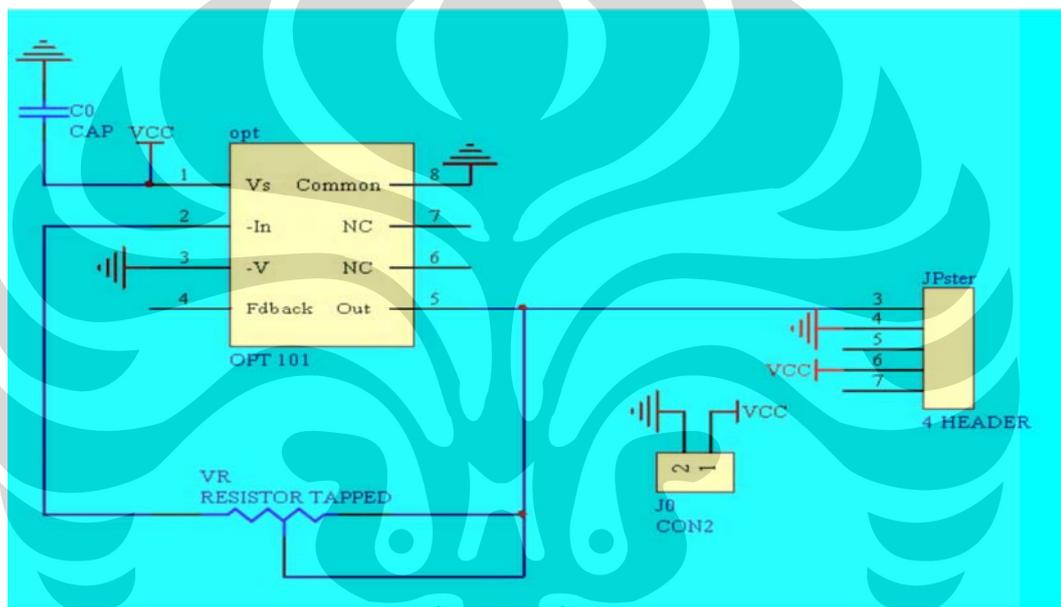


NOTE: (1) Photodiode location.

Gambar 3.7 OPT 101 tampak atas.

Pada sistem ini tidak menggunakan rangkaian RC *internal* pada OPT 101 karena pada sistem ini diinginkan suatu kondisi yang selalu berubah nilai

hambatannya sehingga sistem ini menambahkan rangkaian eksternal RC, dimana digunakan untuk menentukan kondisi *output* tegangan dari sensor sehingga dapat terbaca oleh ADC yang telah tersedia pada IC *microcontroller* yaitu ATMEGA 32 serta menentukan tingkat tegangan keluaran sensor yang disesuaikan oleh intensitas cahaya yang masuk ke dalam *photodiode*. Adapun rangkaian keseluruhan dari rangkaian sensor ini yaitu sebagai berikut :

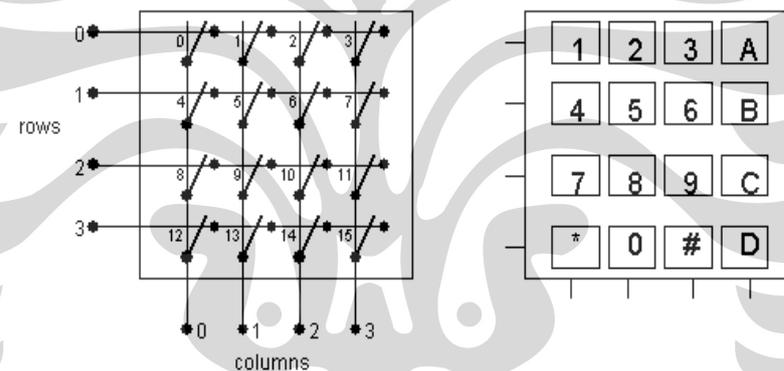


Gambar 3.8 Rangkaian sensor menggunakan OPT 101

Rangkaian ini dapat menghasilkan tegangan keluaran sesuai dengan jumlah intensitas cahaya yang masuk ke dalam *photodiode*. Dimana semakin tinggi tingkat intensitas cahaya yang menyinari *photodiode* maka semakin tinggi pula tegangan keluarannya. Akan tetapi pada sistem ini tegangan keluaran hanya dibatasi sebesar 5 V, karena sesuai dengan tegangan maksimum ADC yang digunakan. Sehingga dibutuhkan pengkalibrasian terlebih dahulu setiap kali dilakukan penyinaran agar mendapatkan tegangan keluaran yang diinginkan.

3.1.6 Rangkaian keypad

Identifikasi *password* di sini berhubungan dengan sistem keamanan (*security*). Keypad dibutuhkan untuk memasukkan data *password* yang akan masuk ke *microcontroller*. *Microcontroller* kemudian akan mengolah data dan membandingkan *password* sama atau tidaknya ID. Apabila sama maka *microcontroller* akan memberikan trigger + 5 VDC ke solenoid sehingga apabila *password* benar maka pintu akan terbuka. Adapun cara kerja dari sistem ini adalah seperti yang akan dijelaskan di bawah ini.



Gambar 3.9 Skematik keypad 4 x 4

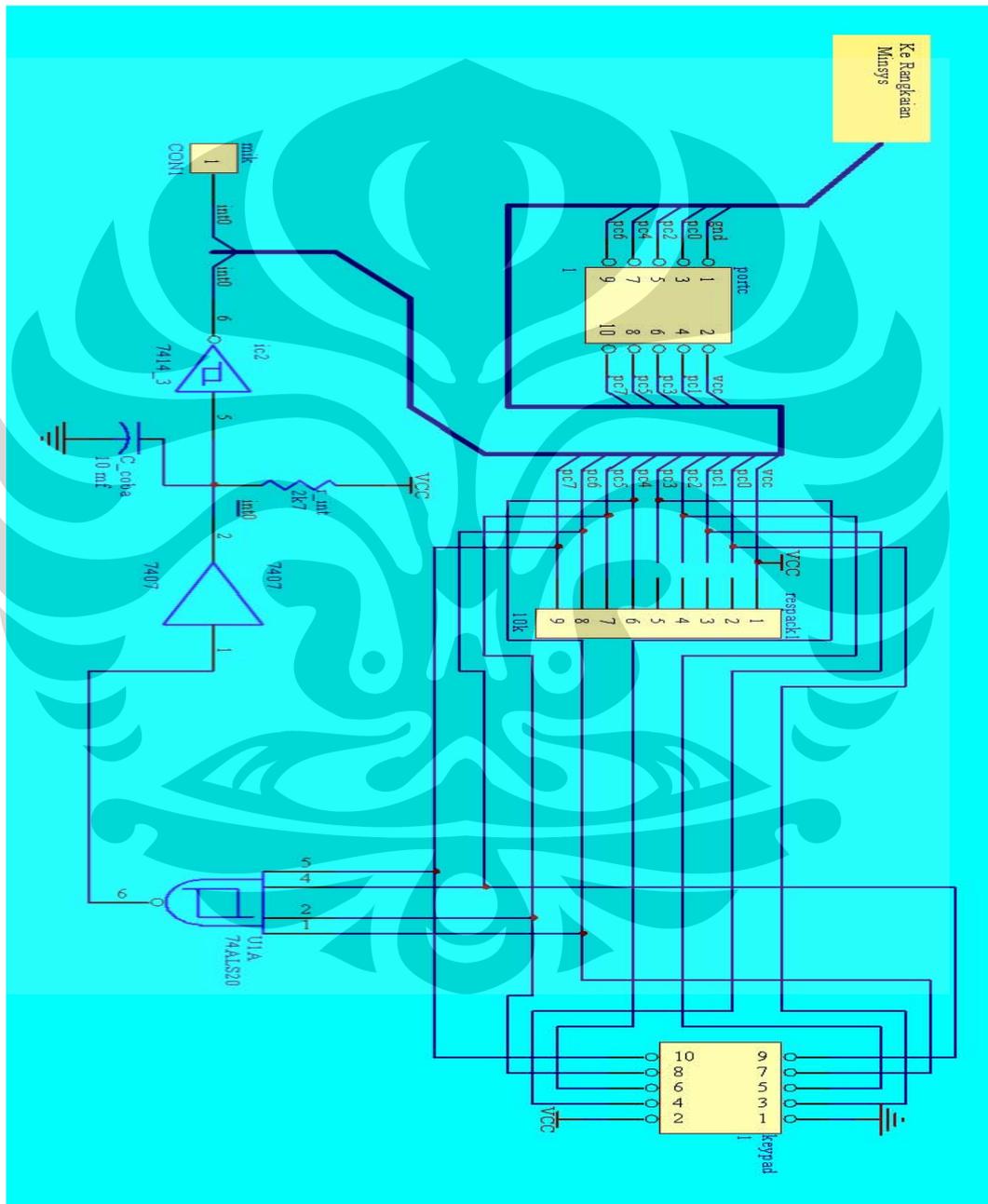
Keypad matrix 4 x 4 memiliki 16 switch, seperti terlihat pada gambar diatas 4 baris (*row*) dan 4 kolom (*column*). Adapun perincian baris dan kolom dari keypad ke *microcontroller* adalah seperti pada Tabel 3.2 berikut :

Tabel 3.2 Baris dan kolom keypad ke port microcontroller

Colomn	PortC
C1	PC.0
C2	PC.1
C3	PC.2
C4	PC.3
Row	Port C
R1	PC.4
R2	PC.5
R3	PC.6
R4	PC.7

Dimana pada saat salah satu switch ditekan maka interrupt harus di bangkitkan selanjutnya microcontroller harus mengetahui switch mana yang ditekan tersebut. Sistem ini menggunakan INTO sebagai eksternal interruptnya. Pada saat switch ditekan maka akan ada hubungan antara baris dan kolom. Awalnya kolom bersifat sebagai output dengan logic low sedangkan baris bersifat sebagai input dengan logic high. Ketika switch ditekan maka pada salah satu kaki IC 74ALS20 (kaki 1, 2, 4, 5) akan bersifat logic low sedangkan ketiga kaki lainnya akan bersifat logic high. IC 74ALS20 merupakan IC yang bersifat NAND Gate sehingga apabila salah satu kakinya bersifat logic low sedangkan yang lainnya bersifat logic high, maka output dari IC 74ALS20 akan bersifat logic high, output dari IC 74ALS20 akan menjadi input IC 7407. Output dari gerbang NAND masuk ke rangkaian pereduksi bouncing yang terdiri dari IC 7407 sebuah resistor dan kapasitor, keluaran dari rangkaian tersebut masuk ke IC schimit trigger IC

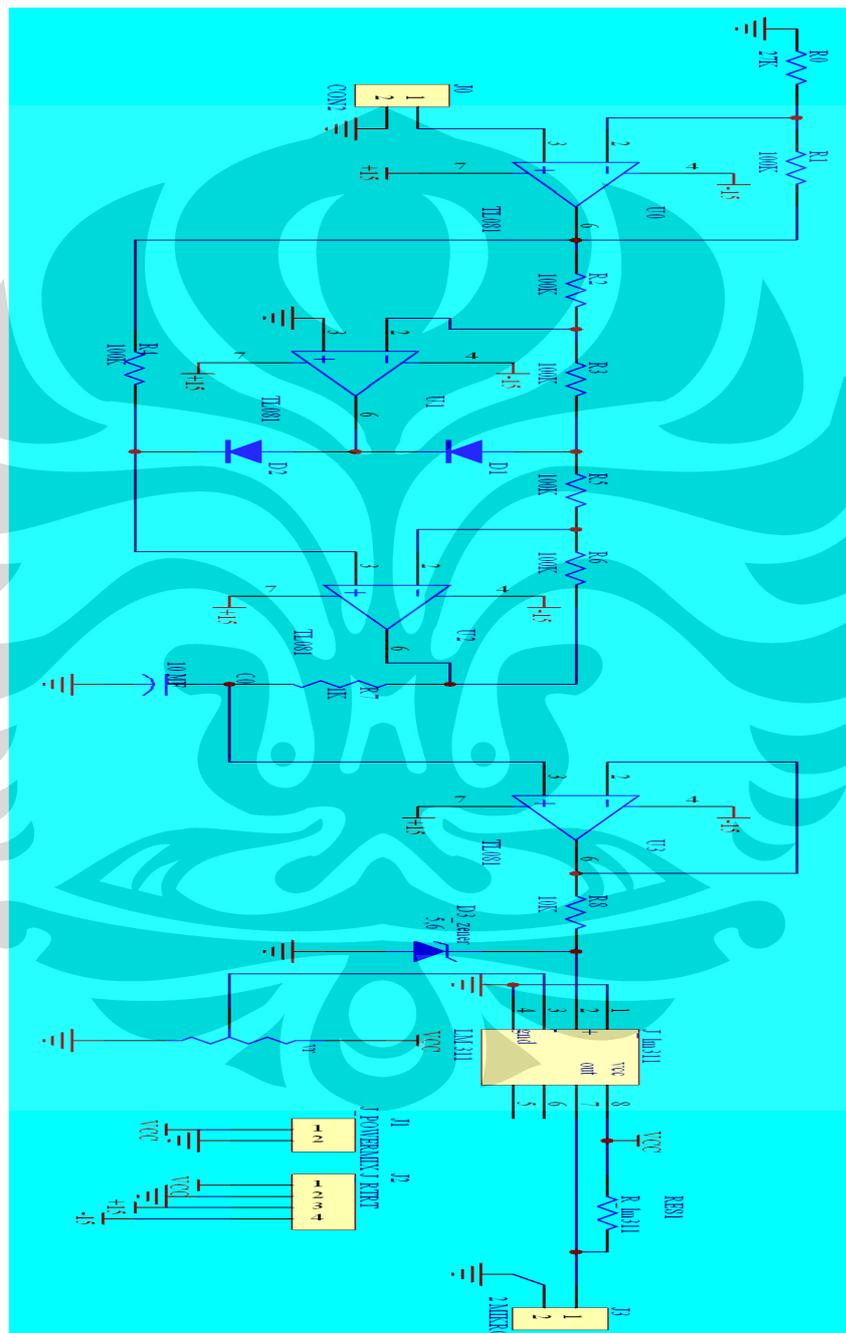
7414 untuk menghilangkan daerah terlarang. Pada saat output IC 7414 bersifat logic low maka akan menyebabkan terjadinya *interrupt*. Hal ini dikarenakan terjadi transisi pulsa dari high ke low sehingga terjadi interrupt. Adapun rangkaian interrupt untuk keypad seperti gambar 3.10 berikut :



Gambar 3.10 Skematik Rangkaian *interrupt keypad* 4 x 4

3.1.7 Rangkaian Pendeteksi Kondisi Peralatan Listrik

Berikut perancangan hardware pada system ditunjukkan seperti gambar di bawah ini.



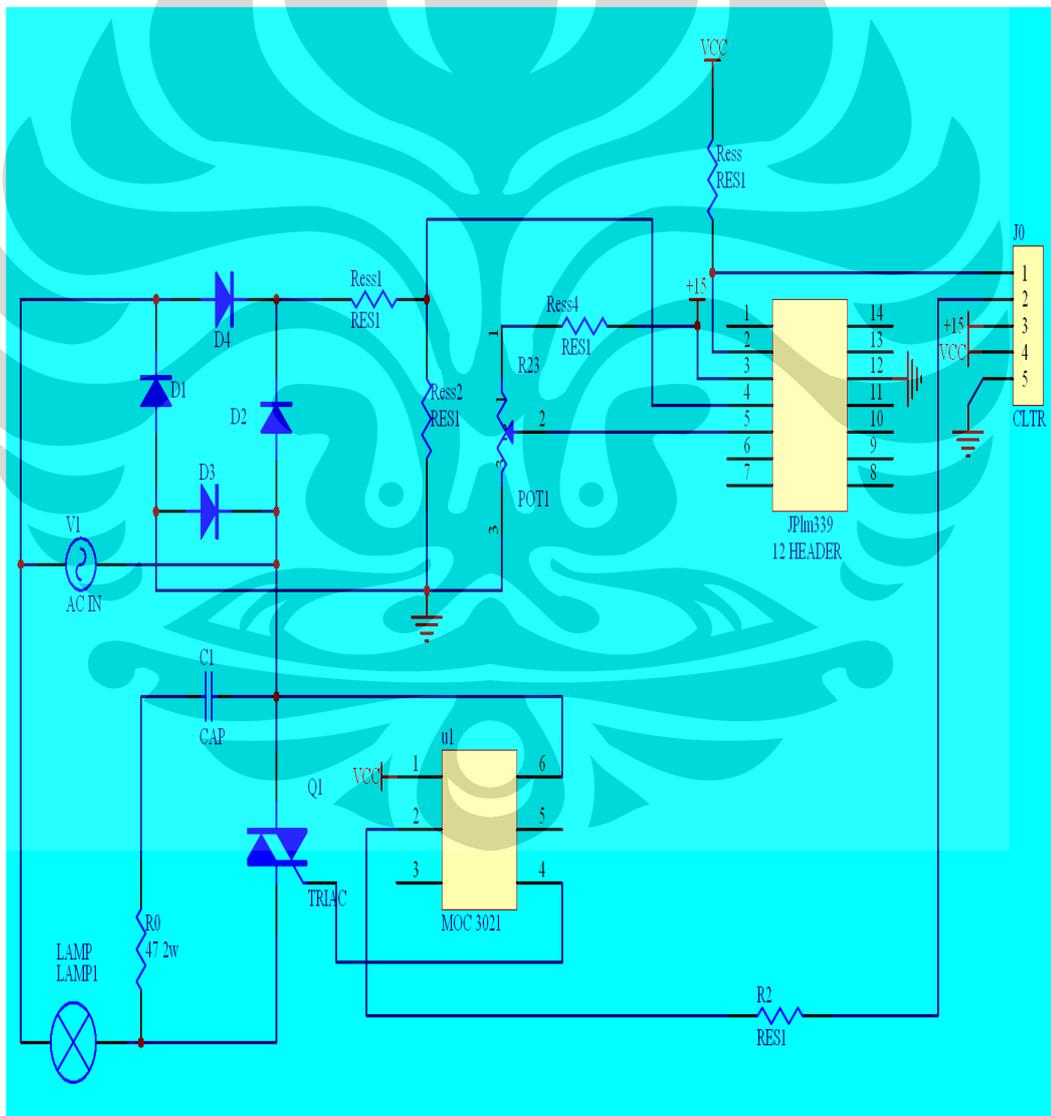
Gambar 3.11 Rangkaian pendeteksi peralatan listrik

Output dari sensor CT pertama kali akan masuk ke rangkaian *Non-inverting amplifier*. Output dari *Non Inverting amplifier* akan masuk ke rangkaian *precision rectifier*. Selanjutnya masuk ke rangkaian *differential amplifier*. Outputnya akan melewati sebuah rangkaian *low-pass filter*. Rangkaian ini akan menahan frekuensi tinggi dari sebuah sinyal dan hanya akan meloloskan frekuensi yang rendah dari sebuah sinyal. Sinyal yang memiliki frekuensi rendah tersebut masuk ke rangkaian *voltage follower*. Penambahan zener berfungsi untuk membuat output dari *voltage follower* dibawah + 5,6 v setelah itu akan masuk ke sebuah IC LM 311. Prinsip kerjanya adalah apabila tegangan input (+) kaki 2 lebih besar dari tegangan referensi (-) kaki 3 maka output dari IC LM 311 akan bersifat logic high (+5 v) begitupun sebaliknya jika tegangan input (+) kaki 2 lebih kecil dari tegangan referensi (-) kaki 3 maka output dari IC LM 311 akan bersifat logic low (0 v). Output dari IC LM 311 ini nantinya akan menjadi input bagi *microcontroller*.

Pada sistem ini menggunakan Porta.0 sebagai input. Apabila pada pina.0 mendapatkan input logic high maka pada PC akan memberitahu bahwa kondisi peralatan listrik dalam keadaan nyala sebaliknya bila inputnya bersifat logic low maka pada PC akan memberitahukan kondisi peralatan listrik dalam keadaan mati.

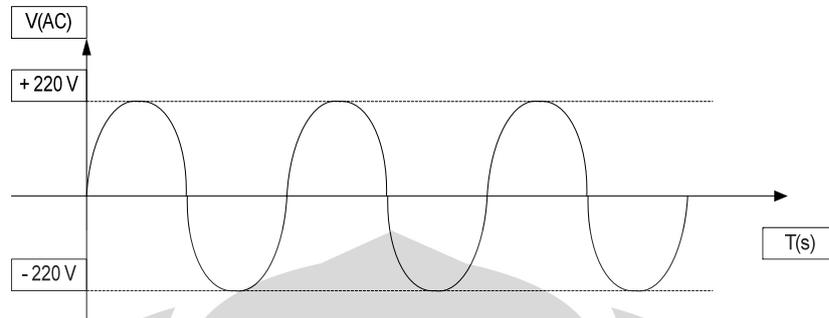
3.1.8 Rangkaian Pengendali Kecerahan Lampu AC 220 V

Sistem ini bertujuan untuk mendapatkan suatu lingkungan dengan tingkat intensitas cahaya yang baik sesuai dengan kenyamanan penghuninya. Sistem ini dapat mengatur tingkat kecerahan dari sebuah lampu 220 VAC. Biasa dikenal dengan sebutan rangkaian *dimmer*. Sistem ini adalah mengembangkan rangkaian dimmer agar pengaturannya dapat dijalankan melalui PC. Prinsip kerja dari rangkaian ini akan dijelaskan menggunakan gambar 3.12 berikut.



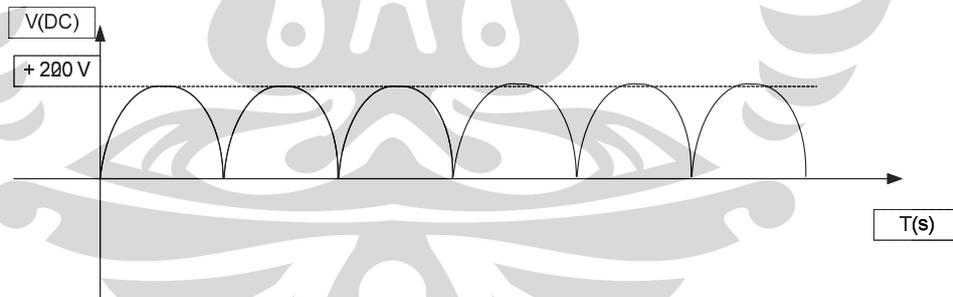
Gambar 3.12 Rangkaian Pengatur Intensitas Cahaya

Proses awal adalah tegangan PLN 220 VAC yang masuk akan memiliki bentuk gelombang sinus seperti pada gambar 3.13 dibawah ini.



Gambar 3.13 Gelombang Sinus Pada tegangan AC

Sinyal-sinyal AC tersebut nantinya setelah melewati bridge akan menjadi setengah dari sinyal aslinya karena bridge bersifat sebagai *fullwave rectifier* sehingga output keluaran dari bridge akan tampak seperti pada gambar di bawah ini.



Gambar 3.14 Gelombang setelah melewati bridge

Bridge tersebut berfungsi sebagai penyearah dan juga berfungsi merubah tegangan AC menjadi DC. Karena hasilnya masih terlalu besar, maka diperlukan suatu pembagi tegangan. Keluaran pembagi tegangan tersebut akan masuk ke kaki (4) IC LM339 kemudian hasilnya dibandingkan dengan V_{ref} pada kaki (5).

Tegangan referensi diatur sebesar 1,14 V. Rangkaian ini berfungsi sebagai *zero crossing detector*, seperti pada gambar 3.12. Rangkaian *zero detector* yang digunakan dalam sistem ini menggunakan IC komparator LM339. Outputnya dihubungkan dengan kaki INT0 pada IC AT90S2313 (Slave). Saat fase positif komparator akan menghasilkan output *high* (V_{cc}) dan saat fase negatif akan menghasilkan output *low* (0 volt). Jadi outputnya adalah gelombang kotak dengan frekuensi sesuai dengan frekuensi AC-nya yaitu 50 Hz. Keluaran dari IC ini akan masuk ke INT0, sinyal-sinyal tersebut akan dideteksi pada saat terjadi transisi dari *high* ke *low*, pada saat itulah terjadi interrupt. Bila kita perhatikan lagi pada gambar 3.11 disana terdapat suatu driver triac, *Driver triac* yang digunakan adalah tipe MOC3021. *Driver* ini termasuk jenis *optocoupler* sehingga relatif aman jika terjadi ketidaknormalan pada bagian beban. Pengaktifan Triac melalui *driver* ini adalah dengan pulsa *low* yang berasal dari portd.5. Pada saat ada pulsa *low* (0 volt) pada kaki 2 maka akan terjadi beda potensial antara kaki 1 dan 2 sehingga arus mengalir dan dioda dalam MOC3021 memancarkan cahaya sehingga *bilateral switch* ON, arus mengalir dari kaki 6 ke 4 akan mengaktifkan Triac menjadi ON sehingga Triac dapat mengalirkan arus.

Untuk bagian tegangan rendah (sebelah kiri) arus yang mengalir maksimum 30 mA. Apabila hambatan dioda dalam MOC3021 diabaikan maka arus yang mengalir dapat dihitung sebagai berikut :

$$I = 5/180 = 0,02778 \text{ A} = 27,78 \text{ mA (masih di bawah nilai yang diizinkan)}$$

Arus 27,78 mA bagi mikrokontroler sudah dianggap besar, sehingga pengaktifan MOC3021 dengan *active low* agar mikrokontroler tidak melakukan *sourcing*.

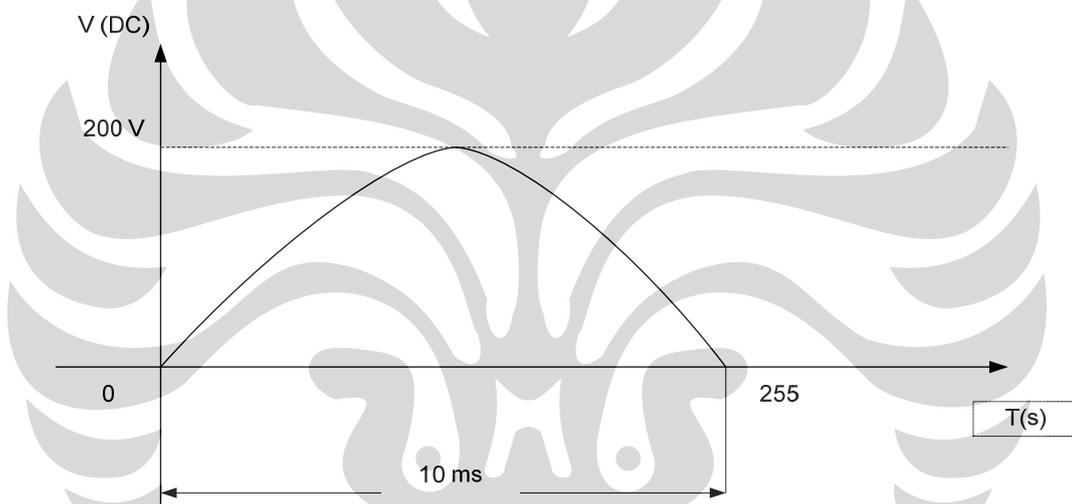
Tegangan PLN memiliki frekuensi 50 Hz sesuai dengan rumus periode dimana gelombang sinus PLN berjalan dalam satu detik.

$$T = 1 / f$$

$$T = 1 / 50 \text{ Hz}$$

$$T = 20 \text{ ms}$$

Setelah melewati bridge maka sinyal akan menjadi setengah periode pada fase positif sehingga $T = 20 \text{ ms} / 2 = 10 \text{ ms}$.



Gambar 3.15. Setengah periode gelombang

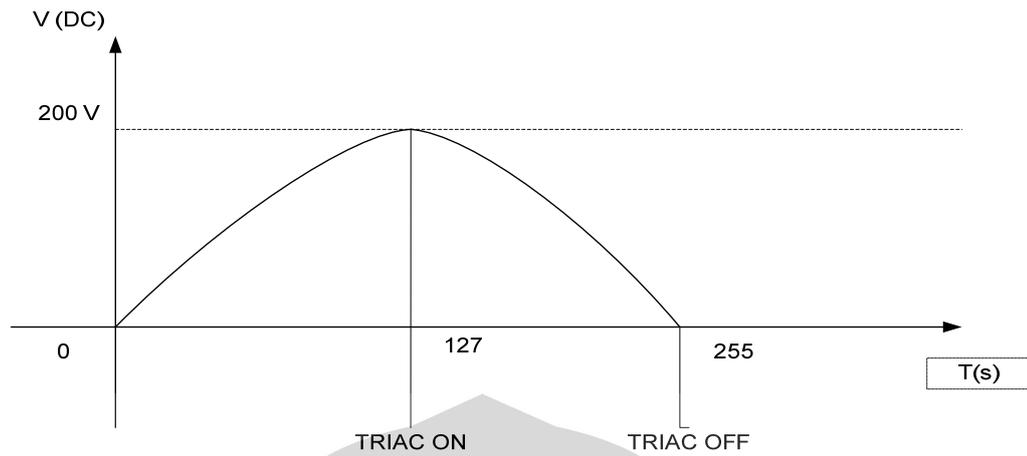
Gambar di atas menjelaskan bahwa 1 / 2 siklus dari gelombang PLN memiliki waktu (t) sebesar 10 ms dengan 256 skala (0 – 255). Bila kita hitung akan menjadi.

$$256 \text{ skala} = 10 \text{ ms}$$

$$1 \text{ skala} = 10 \text{ ms} / 256$$

$$1 \text{ skala} = 0,04 \mu\text{s} = 40 \mu\text{s}$$

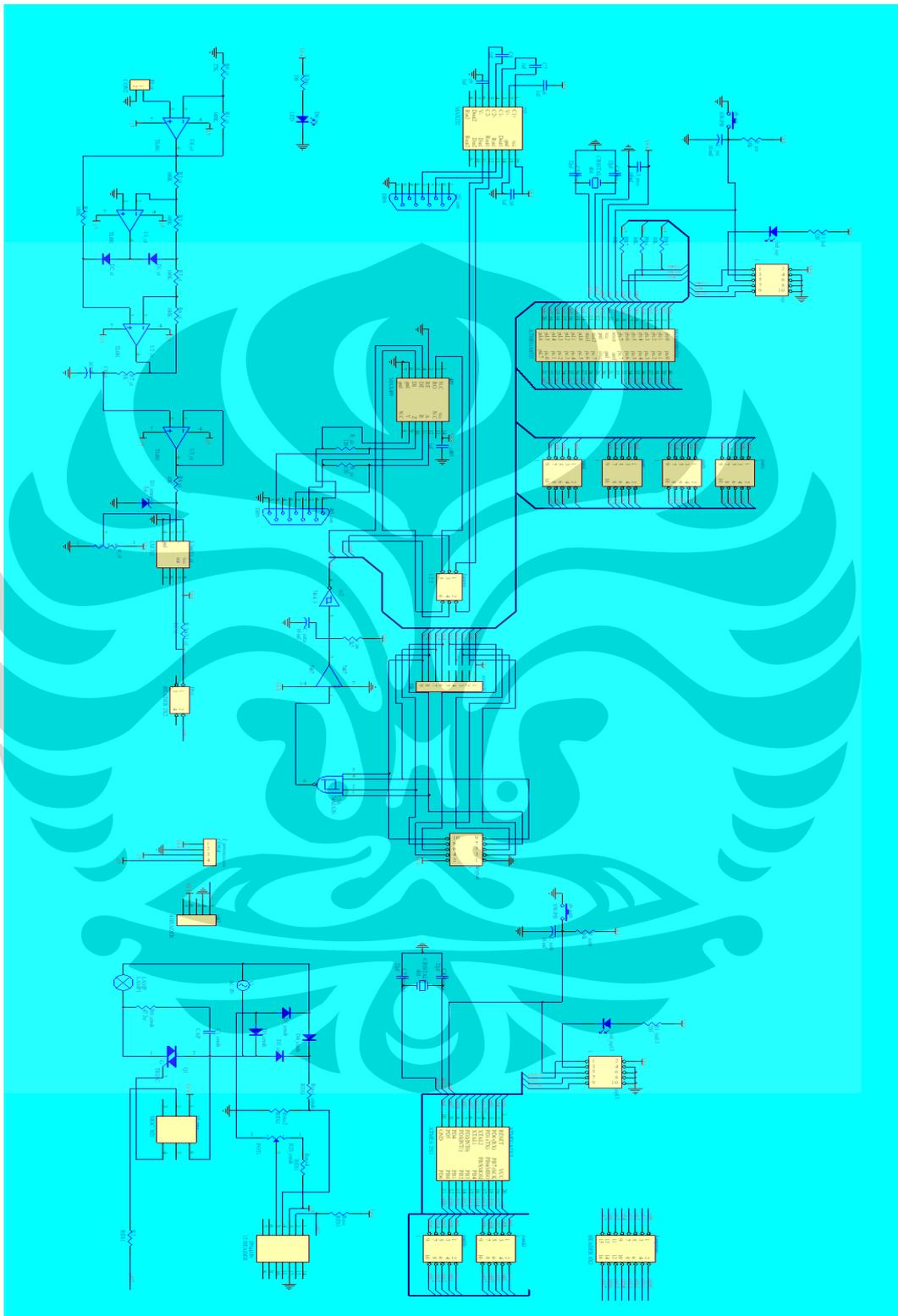
Sebagai contoh bila kita memakai lampu AC 220 V dengan daya sebesar 100 Watt, maka apabila kita ingin menurunkan dayanya menjadi 50 Watt maka artinya kita memiliki penerangan sebesar 50 %.



Gambar 3.16. Pengaturan Gelombang

Berdasarkan gambar di atas maka triac baru ON setelah mencapai nilai 127 setelah mencapai 255 maka triac akan off kembali dan begitu seterusnya dengan delay mulai pada saat triac on sampai off sebesar $5 \mu\text{s}$.

3.1.9 Rangkaian Minimum Sistem



Gambar 3.17 Skematik rangkaian minimum sistem

Gambar di atas adalah gambar dari penggabungan beberapa rangkaian, terdiri dari *minimum system* ATMEGA32 dan AT90S2313, rangkaian pendeteksi peralatan listrik serta rangkaian pengatur tegangan AC. Pada board tersebut terdapat dua buah *microcontroller* yaitu *microcontroller* AVR ATMEGA32 dan AT90S2313, masing-masing IC terdiri dari jalur ISP, rangkaian reset *microcontroller* dan *ISP programmer* serta kristal yang digunakan, 4 buah header 5x2 untuk ATMEGA32 dan 2 buah header 5x2 untuk AT90S2313 yang terhubung ke port I/O-nya *microcontroller*. Sebagai otak dari sistem ini. avr seri ATMEGA32 (*master*) dan AT90S2313 (*slave*). IC *microcontroller* ini memiliki 40 pin dengan 32 pin I/O, 8 Kbyte flash memori, dan 512 byte EEPROM. Untuk konfigurasi masing-masing port pada *microcontroller* ini dapat dilihat pada table 3.3 berikut :

Tabel 3.3 Konfigurasi I/O Port ATMEGA32 (Master)

Port A	Konfigurasi	PortB	Konfigurasi
PortA.0	Sensor Arus (CT)	PortB.0	Data ke Micro 2
PortA.1	Sensor OPT101	PortB.1	Data ke Micro 2
PortA.2	RS (LCD)	PortB.2	Data ke Micro 2
PortA.3	E (LCD)	PortB.3	Data ke Micro 2
PortA.4	DB.4 (LCD)	PortB.4	Data ke Micro 2
PortA.5	DB.5 (LCD)	PortB.5	Data ke Micro 2
PortA.6	DB.6 (LCD)	PortB.6	Data ke Micro 2
PortA.7	DB.7 (LCD)	PortB.7	Data ke Micro 2
PortC	Konfigurasi	PortD	Konfigurasi
PortC.0	Keypad Coloumn 1	PortD.0	Rx
PortC.1	Keypad Coloumn 2	PortD.1	Tx
PortC.2	Keypad Coloumn 3	PortD.2	-
PortC.3	Keypad Coloumn 4	PortD.3	INT1 Keypad
PortC.4	Keypad Row 1	PortD.4	-
PortC.5	Keypad Row 2	PortD.5	-
PortC.6	Keypad Row 3	PortD.6	Output ke L293D
PortC.7	Keypad Row 4	PortD.7	Output ke RS485

Adapun untuk IC AT90S2313 memiliki 2K byte Flash Memori dan 128 byte EEPROM. Untuk Konfigurasi masing-masing port pada IC ini dapat dilihat pada table 3.4 berikut ini :

Tabel 3.4 Konfigurasi I/O Port AT90S2313 (Slave)

PortB	Konfigurasi	PortD	Konfigurasi
PortB.0	Data Dari Micro 1	PortD.0	-
PortB.1	Data Dari Micro 1	PortD.1	-
PortB.2	Data Dari Micro 1	PortD.2	INT0
PortB.3	Data Dari Micro 1	PortD.3	-
PortB.4	Data Dari Micro 1	PortD.4	-
PortB.5	Data Dari Micro 1	PortD.5	Kontrol Triac
PortB.6	Data Dari Micro 1	PortD.6	-
PortB.7	Data Dari Micro 1	-	-

Kembali pada gambar 3.4 terdapat komunikasi serial RS-232 yang nantinya akan terhubung langsung ke port serial PC, sedangkan gambar yang kedua adalah rangkaian RS-485 yang akan terhubung ke rangkaian RS-485 – RS-232 *converter* dan kemudian terhubung ke port serial PC. Kedua rangkaian diatas di hubungkan ke header 3x6 dimana pada header tersebut telah terhubung pin TXD dan RXD dari *microcontroller*. Rangkaian diatas tidak bekerja secara bersamaan. Saat jumper dihubungkan pada pin 1-3 dan 2-4 dari header, maka kaki TXD dan RXD *microcontroller* akan terhubung ke rangkaian RS-232 *converter*, sedangkan saat jumper dihubungkan pada pin 2-5 dan 4-6 dari header, maka kaki TXD dan RXD *microcontroller* akan terhubung ke rangkaian RS-485. Pada rangkaian RS-485 menggunakan IC max 489 yang bersifat *full duplex* sehingga pada saat melakukan pengiriman data, *microcontroller* dapat melakukan penerimaan data

yang lain. Agar dapat melakukan pengiriman dan penerimaan data serial, kaki *enable transceiver* dan *receiver* pada IC 489 harus diaktifkan terlebih dahulu. Untuk mengaktifkan *enable transceiver* pada IC 489 dengan memberikan logika high atau “1” sedangkan untuk mengaktifkan *enable receiver* dengan memberikan logika low atau “0” pada kaki tersebut.

Level tegangan pada *microcontroller* adalah level tegangan TTL yang hanya mengenal logika 1 untuk 3-5 V dan logika 0 untuk 0-0.8 V, sedangkan pada PC memiliki level tegangan RS-232 yang menganggap tegangan -15 s/d -25 V sebagai logika 1 dan tegangan 15 s/d 25 V sebagai logika low (0). Saat menggunakan komunikasi RS-232 memiliki kendala yaitu masalah keterbatasan jarak dan rentan terhadap noise. Untuk mengatasi keterbatasan koneksi, digunakan komunikasi RS-485 yang dapat digunakan sampai sejauh 1,2 KM dan lebih kecil kemungkinan terkena gangguan interferensi sinyal dari luar, barulah nantinya jalur komunikasi ini terhubung ke rangkaian RS-485 to RS-232 *converter* pada saat akan dihubungkan ke PC untuk mengkonversi beda level tegangan. Fungsi dari rangkaian RS-485 diatas adalah agar system yang ada pada rumah dapat berkomunikasi dengan PC sentral, walaupun dengan jarak yang jauh. Fungsi dari rangkaian RS-232 *converter* di atas adalah mengkonversi beda tegangan antara mikrokontroler dengan PC.

Adapun jalur komunikasi yang nantinya akan digunakan oleh system untuk berkomunikasi dengan PC sentral adalah jalur komunikasi RS-485 dikarenakan jaraknya yang jauh, akan tetapi penulis disini tidak melakukan pengujian terhadap kabel yang digunakan pada saat melakukan percobaan komunikasi serial antara

mikrokontroler dengan PC, hanya untuk jarak yang dekat. Untuk melakukan pemilihan rangkaian komunikasi mana yang akan digunakan, penulis hanya tinggal merubah posisi jumper yang ada pada header 3x6 yang terhubung ke kaki RXD dan TXD dari mikrokontroler.

3.2 Perancangan Software

Pada bagian ini akan dijelaskan tentang perancangan software dari sistem yang telah dibuat termasuk protokol komunikasi serial antara PC – mikrokontroler yang digunakan.

3.2.1 Protokol Pengiriman Data serial

Perangkat elektronik yang ada pada tiap lantai adalah sama, yang membedakannya adalah alamat yang tersimpan di dalam EEPROM microcontroller. Secara default, tiap-tiap sistem pada tiap lantai sudah memiliki alamat tetapi alamat tersebut dapat dirubah-rubah dengan cara mengirimkan paket data inisialisasi alamat secara serial seperti yang ada pada tabel 3.5 dibawah ini. Berikut ini adalah format protokol yang digunakan pada saat melakukan komunikasi antara komputer dengan mikrokontroler:

Tabel 3.5 Format protokol komunikasi serial PC-mikrokontroler dan mikrokontroler-PC.

Format Protokol					Keterangan
Start Frame '*'	Alamat 4 byte	Command 'd'	Alamat Baru 4 byte	Stop Frame '#'	Format melakukan inialisasi alamat dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'e'	Stop Frame '#'		Format meminta alamat dari PC- μ Controller. Alamat = 0001 (lantai 1) / 0002 (lantai 2)
Start Frame '*'	Alamat 4 byte	Command 'f'	Stop Frame '#'		Format meminta data orang, kondisi peralatan listrik dan sensor dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'g'	Stop Frame '#'		Format Meminta data nama & password yang tersimpan di EEPROM dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'h'	Stop Frame '#'		Format mengganti password lama dengan password baru dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'i'	Stop Frame '#'		Format melakukan penghapusan data nama & password di EEPROM dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'j'	Stop Frame '#'		Format melakukan registrasi nama & password ke EEPROM dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'k'	Stop Frame '#'		Format mengganti nama lama dengan nama baru ke EEPROM dari PC- μ Controller
Start Frame '*'	Alamat 4 byte	Command 'l'	Stop Frame '#'		Format melakukan pengontrolan intensitas lampu AC 220 V dari PC- μ Controller

Hal yang pertama kali dilakukan microcontroller saat menerima kiriman data serial dari PC adalah mengecek apakah data tersebut data berupa angka ascii dari karakter '*' atau bukan, jika benar maka cek data selanjutnya yaitu alamat sebesar 4 byte. Jika alamat benar maka data selanjutnya yang harus di cek adalah command bit. Jika command bit berupa angka ascii dari karakter 'd' maka proses selanjutnya adalah menerima 4 byte alamat baru lalu bila di akhiri angka ascii dari karakter '#' maka data tersebut akan disimpan di EEPROM microcontroller.

apabila komputer sentral ingin mengetahui alamat sistem pada setiap lantai, maka paket data yang harus dikirim adalah seperti pada baris ke-2 dari tabel 3.1. Alamat yang digunakan adalah '0000', dan command bitnya adalah 'e' dan di akhiri angka ascii dari karakter '#'. Ketika microcontroller menerima paket data tersebut, maka akan segera mengirimkan alamatnya, sebagai contoh saat alamat pada system tersebut adalah '1234' maka paket data *reply* dari microcontroller adalah *1234#.

Jika command bit yang diterima adalah angka ascii dari karakter 'f' dan di akhiri angka ascii dari karakter '#' maka proses selanjutnya adalah mengirimkan data-data orang yang berada di lantai tersebut, melakukan pengecekan kondisi peralatan listrik dengan melakukan pengecekan kondisi porta.0 berada dalam kondisi logic high / logic low. Logic high berarti peralatan listrik masih dalam keadaan menyala sebaliknya jika logic low maka kondisi peralatan listrik dalam keadaan mati. Selain itu microcontroller akan mengirimkan data intensitas.

Jika command bit yang diterima adalah angka ascii dari karakter 'g' dan di akhiri angka ascii dari karakter '#' maka proses selanjutnya adalah mengirimkan

data-data nama & password yang telah teregistrasi dan tersimpan di EEPROM microcontroller.

Jika command bit yang diterima adalah angka ascii dari karakter 'h' dan diakhiri angka ascii dari karakter '#' maka proses selanjutnya adalah mengganti dan menyimpan password lama dengan password baru di EEPROM microcontroller.

Jika command bit yang diterima adalah angka ascii dari karakter 'i' dan diakhiri angka ascii dari karakter '#' maka proses selanjutnya adalah menghapus data nama & password yang tersimpan di EEPROM microcontroller.

Jika command bit yang diterima adalah angka ascii dari karakter 'j' dan diakhiri angka ascii dari karakter '#' maka proses selanjutnya adalah melakukan registrasi nama & password dan menyimpannya di EEPROM microcontroller.

Jika command bit yang diterima adalah angka ascii dari karakter 'k' dan diakhiri angka ascii dari karakter '#' maka proses selanjutnya adalah mengganti dan menyimpan nama lama dengan nama baru di EEPROM microcontroller.

Apabila paket data yang di terima oleh mikrokontroler tidak sesuai seperti protokol pada tabel 3.5, maka data tersebut akan diabaikan dan mikrokontroler tidak akan merespon permintaan PC.

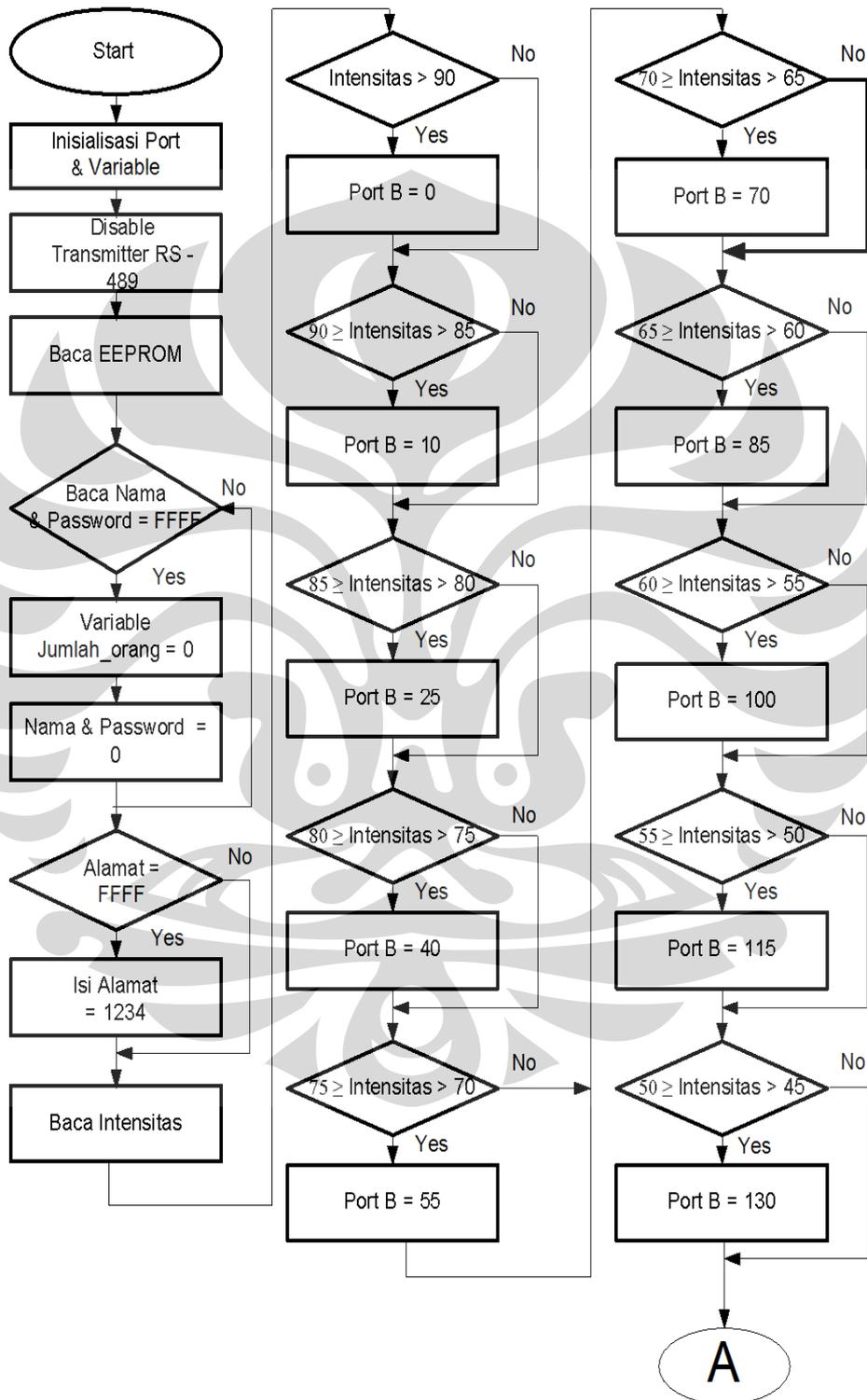
3.2.2 Flowchart Program

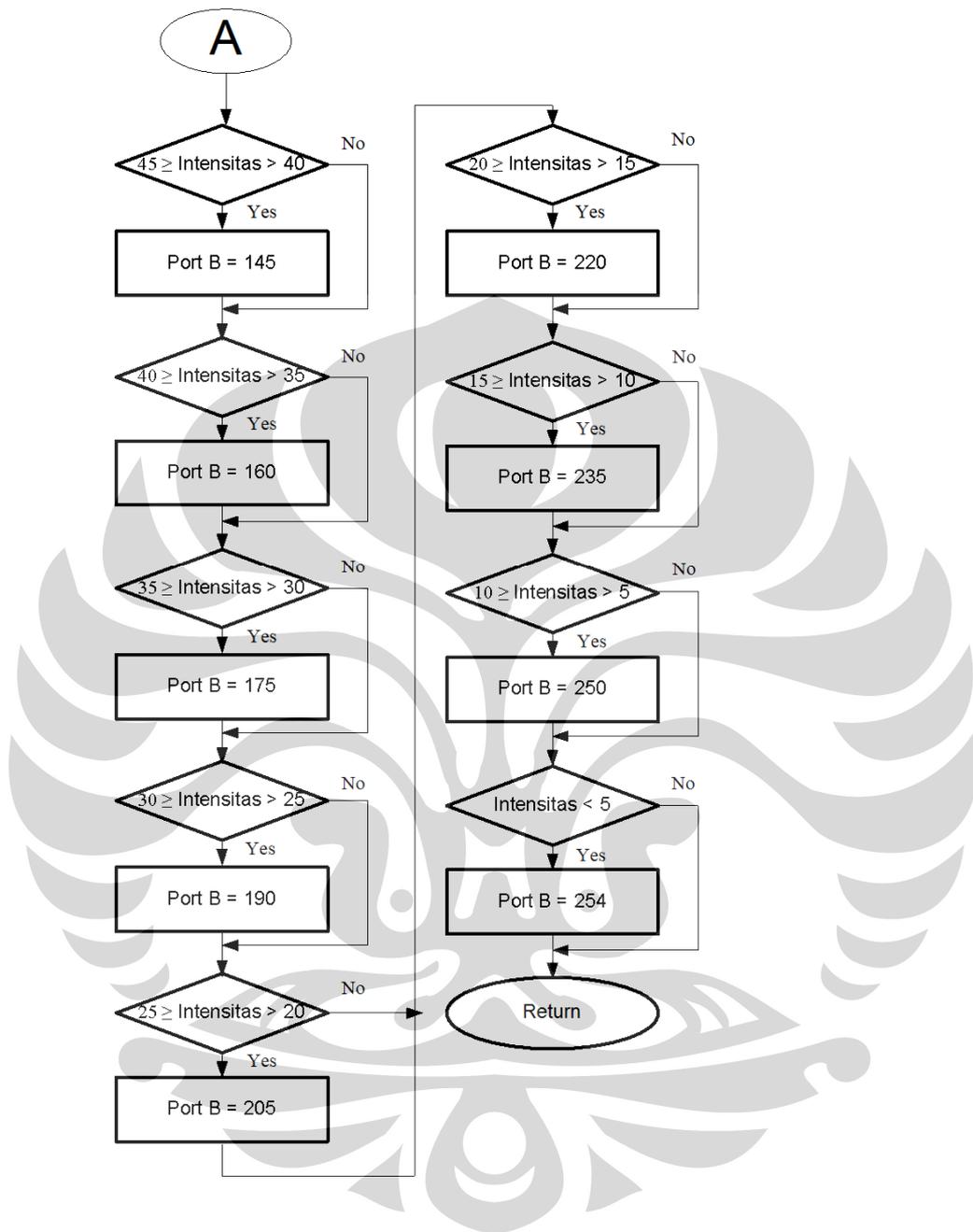
Perancangan sistem ini tidak hanya memiliki *hardware* saja tetapi sistem pengendalian ini juga memanfaatkan *software*. *Software* yang digunakan yaitu BASCOM-AVR dengan menggunakan *microcontroller* ATMEGA32 dan AT90S2313.

Ada beberapa hal yang dilakukan dalam penelitian kali ini dimana setiap sistem memiliki main program masing-masing. Berikut ini akan dijelaskan hal-hal yang menyangkut pengerjaan softwarena seperti telah dikatakan sebelumnya bahwa sistem ini terdiri atas sistem keamanan (*security*), pendeteksian kondisi peralatan listrik dengan sensor arus (CT) serta pengontrolan lampu AC. Berikut adalah beberapa flowchart yang menggambarkan software pendukung sistem yang dibuat.

3.2.2.1 Flowchart Program Micro Master (ATMEGA32)

Berikut ini adalah flowchart dari program utama sistem yang dibuat :





Gambar 3.18 Flowchart program utama sistem microcontroller (ATMEGA32)

Apa yang dikerjakan oleh program utama mikrokontroler hanya melakukan inisialisasi, melakukan pengecekan EEPROM dan mengirim data ke microcontroller (slave) sebagai respons dari intensitas yang terbaca. Hal tersebut

akan dilakukan secara terus menerus sampai terjadi interrupt. Dikarenakan semua inputannya menggunakan interrupt, ketika salah satu interrupt yang digunakan aktif, maka program akan loncat ke sub routine interrupt tersebut lalu menjalankan programnya, dan setelah selesai mengerjakan akan kembali ke program utama, yaitu menunggu terjadinya pengaktifan interrupt lagi. Apabila interrupt terjadi secara bersamaan, maka yang akan dikerjakan terlebih dahulu adalah interrupt yang memiliki prioritas lebih tinggi yaitu interrupt eksternal 0 yang digunakan pada keypad, dan baru interrupt serial yang memiliki prioritas lebih rendah.

Saat pertama kali program di download, EEPROM masih dalam keadaan kosong, dan ketika dilakukan pengujian berapa nilai yang ada di dalam EEPROM tersebut ditampilkan angka 65535 atau FFFF heksadecimal. Untuk memberikan alamat default (alamat awal) pada sistem tersebut, maka dilakukan pengecekan isi dari EEPROM. Ketika isi dari EEPROM sama dengan 65535 berarti sistem masih belum memiliki alamat dan proses selanjutnya adalah menyimpan alamat awal (1234) kedalam EEPROM.

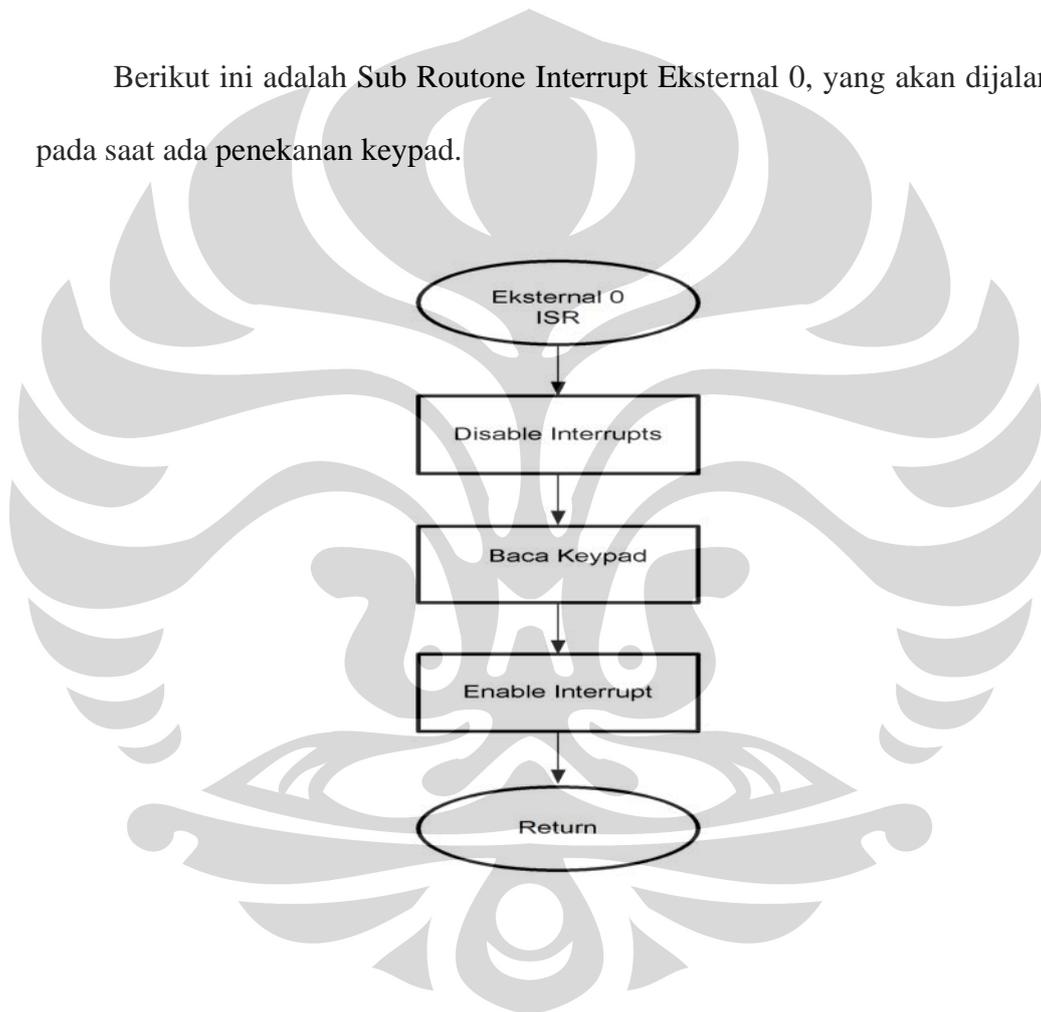
Ketika isi EEPROM tidak sama dengan 65535, ini berarti sistem tidak perlu di beri alamat awal (sudah memiliki alamat). Berikut ini adalah potongan listing program ketika melakukan pengecekan EEPROM :

```
Readeeprom Bc_alamat , Label1  
If Bc_alamat = 65535 Then  
Writeeprom Alamat_awl , Label1  
End If
```

Sama halnya untuk memberikan default (data nama & password) pada sistem tersebut dilakukan pengecekan isi dari EEPROM. Apabila isi dari EEPROM sama dengan 65535 berarti sistem masih belum memiliki data nama & password.

3.2.2.2 Flowchart Program Interrupt Eksternal INT0 (Keypad)

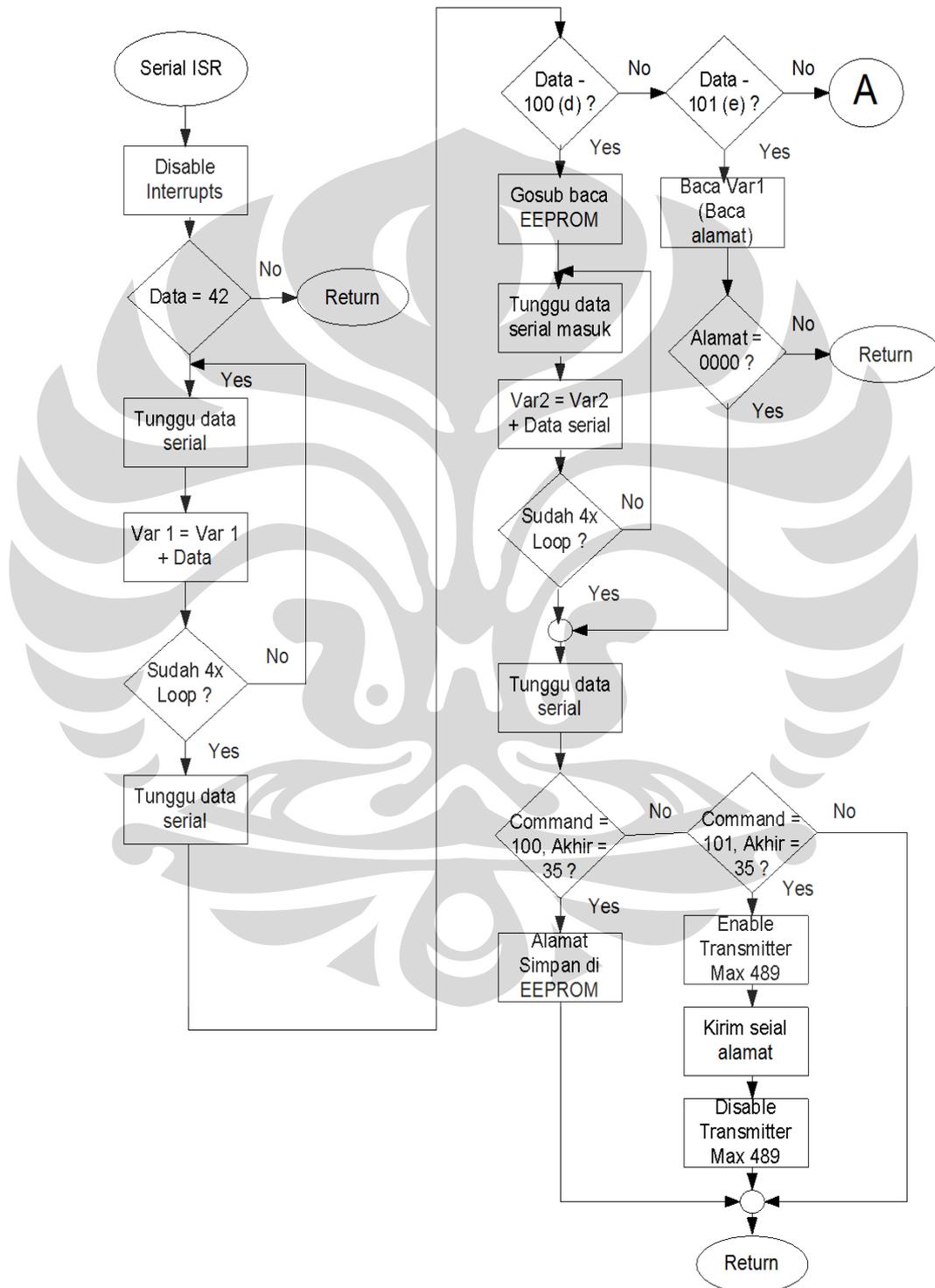
Berikut ini adalah Sub Routine Interrupt Eksternal 0, yang akan dijalankan pada saat ada penekanan keypad.

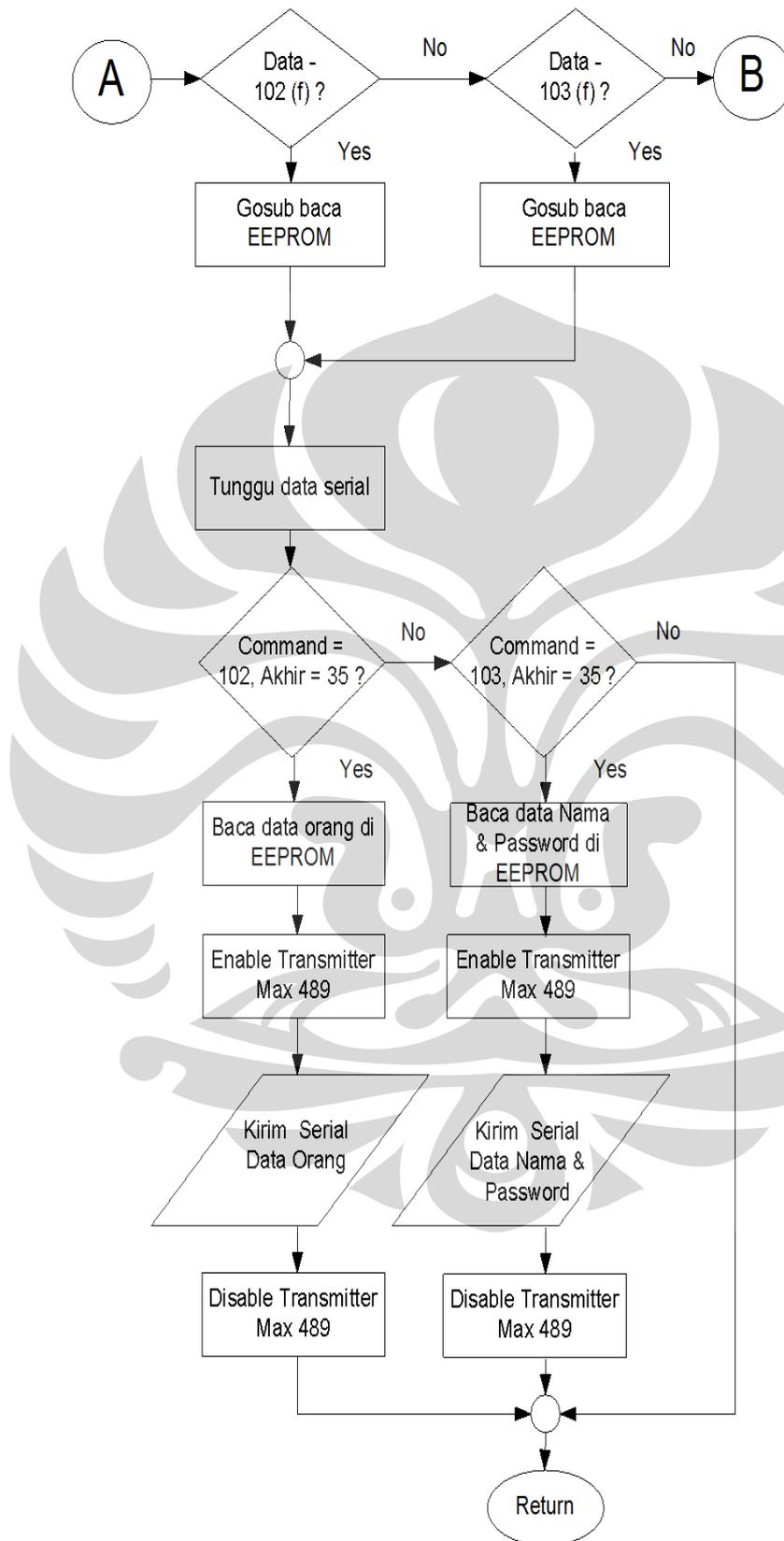


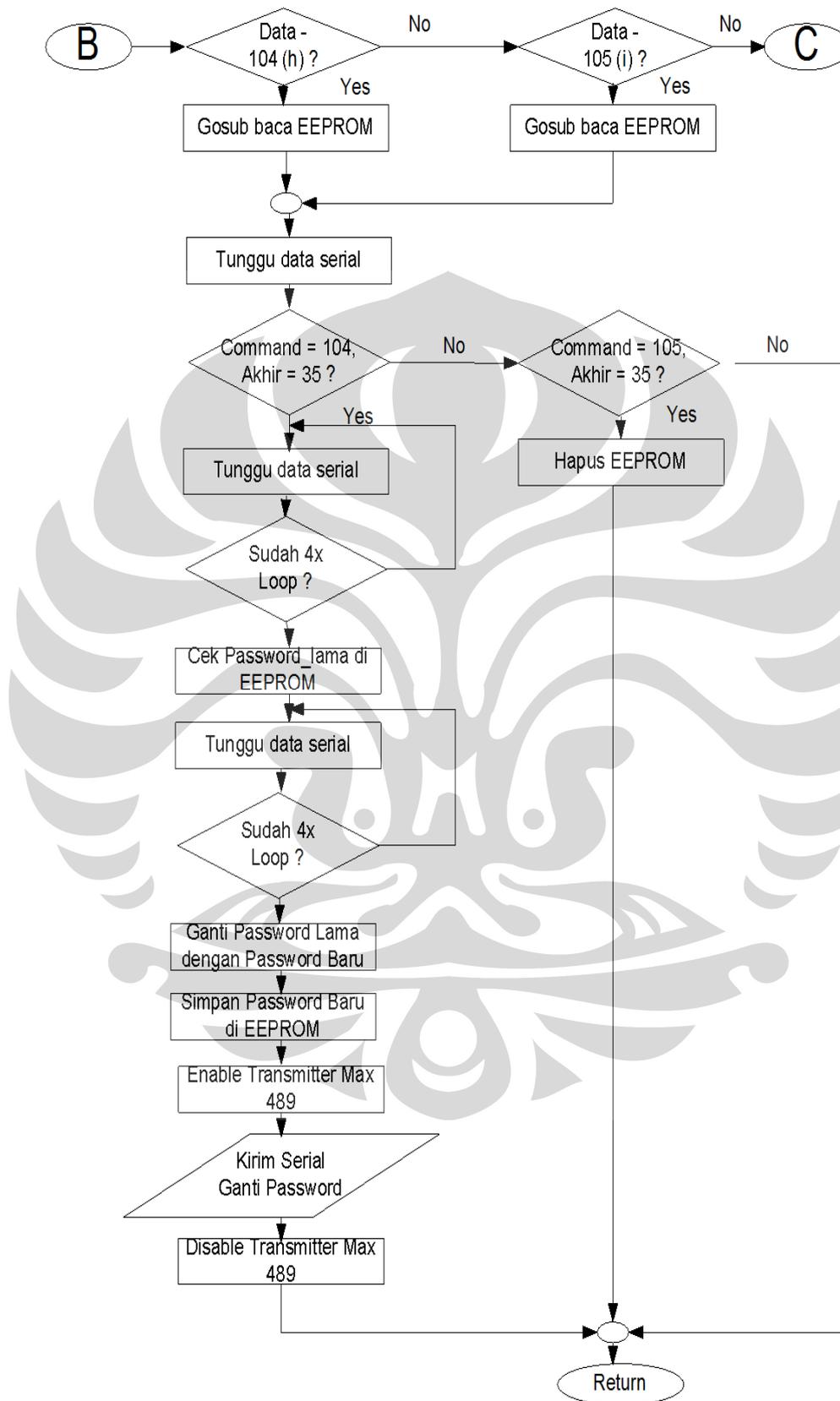
Gambar 3.19 Flowchart program interrupt eksternal INT0 (keypad)

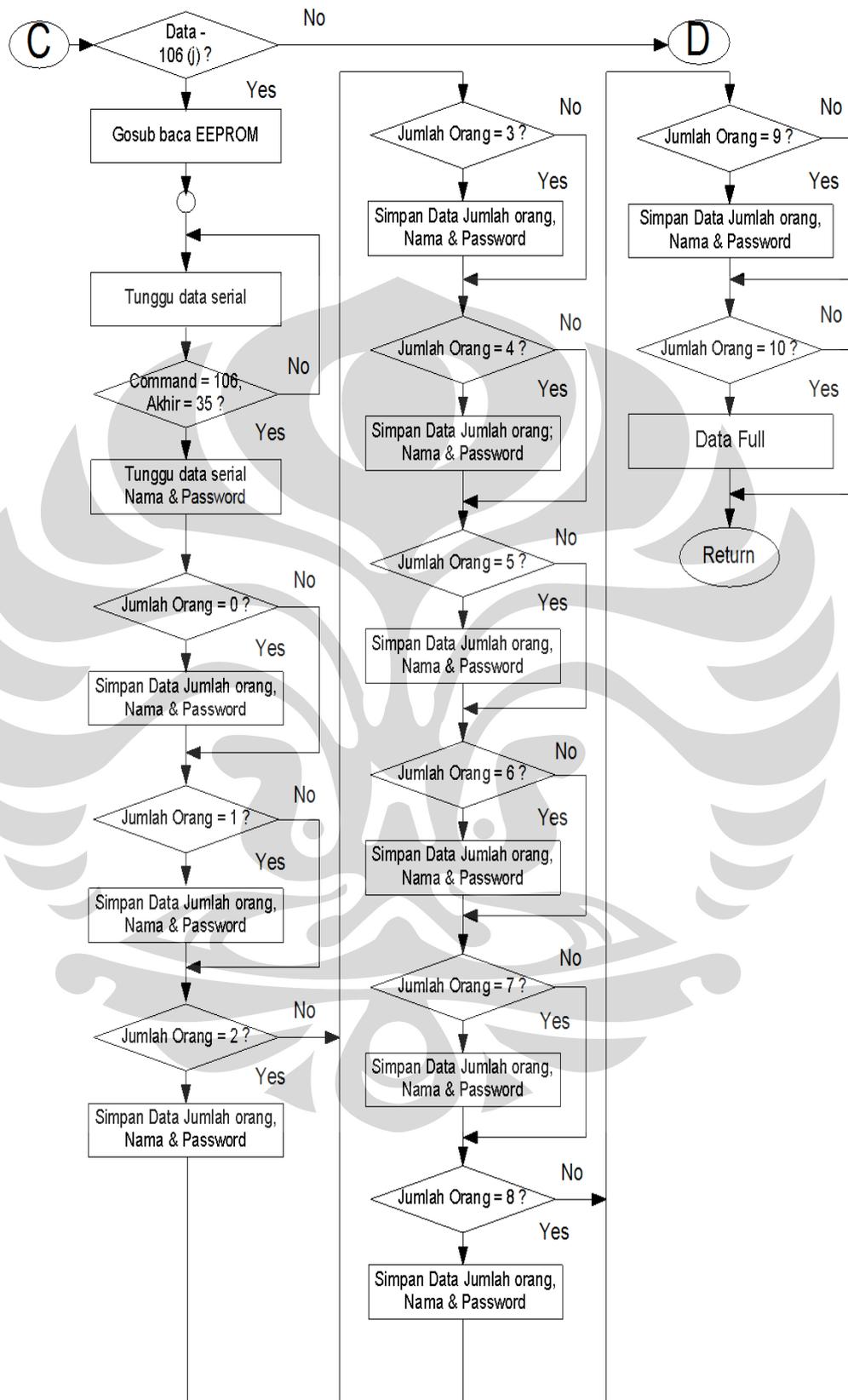
3.2.2.3 Flowchart Program Interrupt Serial

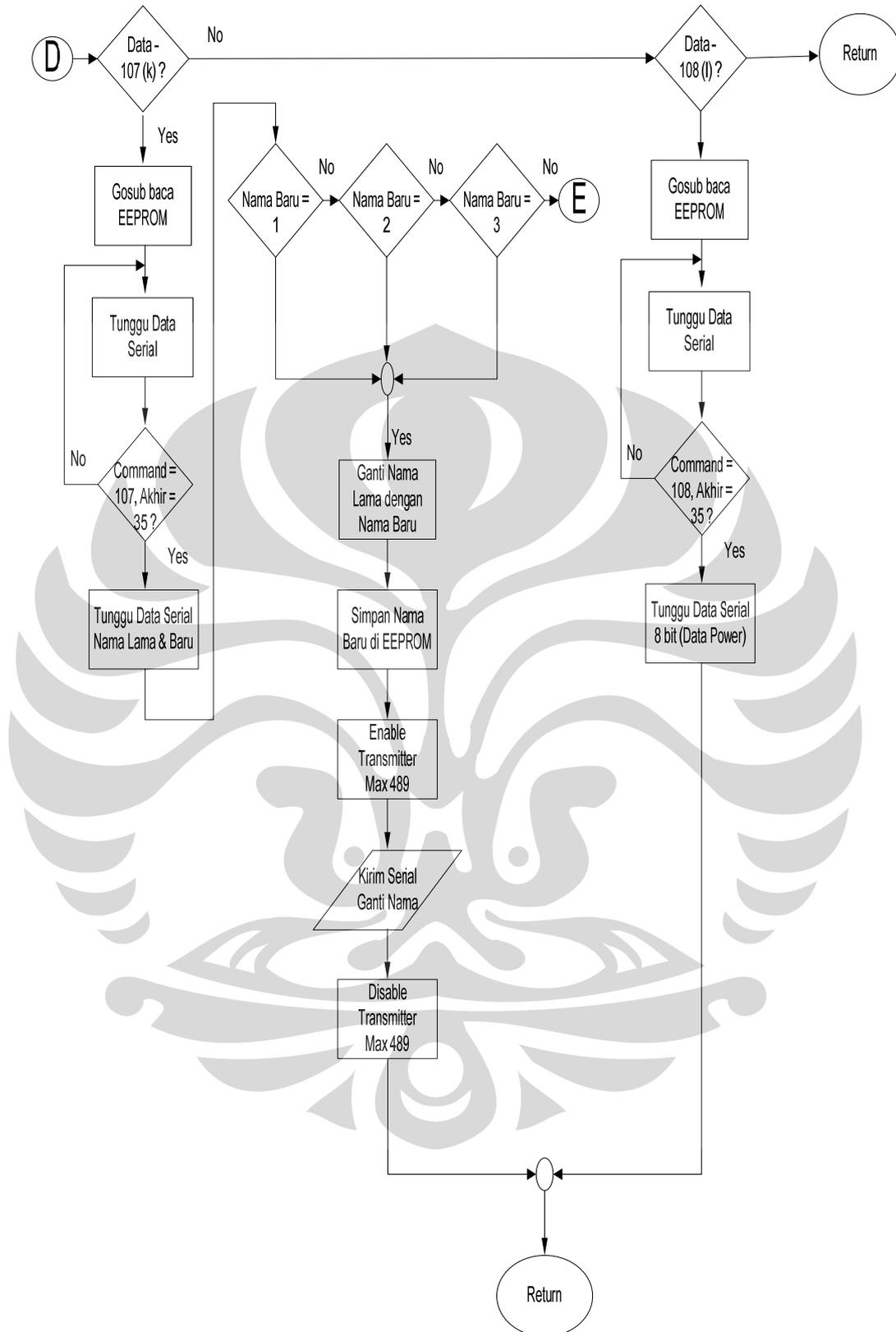
Berikut ini adalah flowchart dari subroutine interrupt serial :











Gambar 3.20 Flowchart program interrupt serial

Sama seperti input keypad, input melalui komunikasi serial (pengiriman data dari komputer sentral ke microcontroller) tidak dapat dipantau secara terus menerus oleh microcontroller. Apabila microcontroller memantau data masukan komunikasi serial secara terus-menerus, maka inputan yang lain tidak dapat ditangani. Solusi untuk masalah ini adalah dengan mengaktifkan interrupt serial dan meletakkan program input data serial pada sub routine interrupt serial. Hal ini akan mengakibatkan pembacaan data serial baru akan dilakukan pada saat data serial diterima mikrokontroler.

Prosedur dari pengiriman data serial pada sistem ini adalah sangat erat kaitannya dengan protokol dari data serial yang terdiri dari 1 startbit, 4 byte data, 1 command bit dan 1 stopbit. Satu sistem microcontroller memiliki satu alamat masing-masing, alamat default pada sistem ini = 1234. Dimana pada saat terjadi permintaan data dari komputer sentral ke microcontroller yang membedakannya adalah command bit.

Hal yang pertama kali dilakukan microcontroller saat menerima kiriman data serial dari PC adalah mengecek apakah data tersebut data berupa angka ascii dari karakter '*' atau bukan, jika benar maka cek data selanjutnya yaitu alamat sebesar 4 byte. Jika alamat benar maka data selanjutnya yang harus di cek adalah command bit. Jika command bit berupa angka ascii dari karakter 'd' maka proses selanjutnya adalah menerima 4 byte alamat baru lalu bila di akhiri angka ascii dari karakter '#' maka data tersebut akan disimpan di EEPROM microcontroller. Jika command bit yang diterima adalah angka ascii dari karakter 'e'. Hal itu berarti ada permintaan alamat oleh komputer sentral.

Alamat yang digunakan adalah '0000', dan command bitnya adalah 'd'. Ketika microcontroller menerima paket data tersebut, maka akan segera mengirimkan alamatnya, sebagai contoh saat alamat pada system tersebut adalah '1234' maka paket data *reply* dari mikrokontroler adalah *1234#. Apabila paket data yang di terima oleh mikrokontroler tidak sesuai, maka data tersebut akan diabaikan dan mikrokontroler tidak akan merespon permintaan PC.

Jika command bit yang diterima adalah angka ascii dari karakter 'f' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah mengirimkan data nama yang teridentifikasi, data kondisi peralatan listrik serta data intensitas. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

```
Kirim_orang:
  Set Portd.7
  Waitms 5
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
  Readeeprom Enam
  Readeeprom Tujuh
  Readeeprom Delapan
  Readeeprom Sembilan
  Readeeprom Sepuluh
  Print "*";
  Print "i" ; ";";
  Readeeprom Bc_eeeprom , Nama_1
```

```
If Bc_eeprom <> 65535 Then
Print Satu ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeprom , Nama_2
If Bc_eeprom <> 65535 Then
Print Dua ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeprom , Nama_3
If Bc_eeprom <> 65535 Then
Print Tiga ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeprom , Nama_4
If Bc_eeprom <> 65535 Then
Print Empat ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeprom , Nama_5
If Bc_eeprom <> 65535 Then
Print Lima ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeprom , Nama_6
If Bc_eeprom <> 65535 Then
Print Enam ; ";";
```

```

Else
Print "-" ; ";";
End If
Readeeprom Bc_eeeprom , Nama_7
If Bc_eeeprom <> 65535 Then
Print Tujuh ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeeprom , Nama_8
If Bc_eeeprom <> 65535 Then
Print Delapan ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeeprom , Nama_9
If Bc_eeeprom <> 65535 Then
Print Sembilan ; ";";
Else
Print "-" ; ";";
End If
Readeeprom Bc_eeeprom , Nama_10
If Bc_eeeprom <> 65535 Then
Print Sepuluh ; ";";
Else
Print "-" ; ";";
End If

```

Krm_lamp:

Cuk:

```

Debounce Pina.0 , 1 , Lamp_nyala
Debounce Pina.0 , 0 , Lamp_mati
Goto Cuk

```

Lamp_nyala:

```
Print "lampu nyala" ;";";
Gosub Krm_intensitas
Goto Coyo
```

Lamp_mati:

```
Print "lampu mati" ;";";
```

Intnsits:

```
Gosub Krm_intensitas
```

Coyo:

```
Adc_iw = Fusing(adc_iwan , "#.##")
Print Adc_iw ;";";
Print "#"
Goto Balix
```

```
Waitms 5
```

```
Reset Portd.7
```

Jika command bit yang diterima adalah angka ascii dari karakter 'g' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah mengirimkan data nama & password yang tersimpan di EEPROM. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

Kirim_data:

```
Set Portd.7
```

```
Waitms 5
```

```
Readeeprom Satu , Dat_1
```

```
Readeeprom Dua
```

```
Readeeprom Tiga
```

```
Readeeprom Empat
```

```
Readeeprom Lima
```

Readeeprom Enam
Readeeprom Tujuh
Readeeprom Delapan
Readeeprom Sembilan
Readeeprom Sepuluh
Readeeprom Pras , Psw_1
Readeeprom Prass , Psw_2
Readeeprom Prasss , Psw_3
Readeeprom Prassss , Psw_4
Readeeprom Prasssss , Psw_5
Readeeprom Prassssss , Psw_6
Readeeprom Prasssssss , Psw_7
Readeeprom Prassssssss , Psw_8
Readeeprom Prasssssssss , Psw_9
Readeeprom Prassssssssss , Psw_10
Print Satu;
Print ";";
Print Pras;
Print ";";
Print Dua ;
Print ";";
Print Prass;
Print ";";
Print Tiga;
Print ";";
Print Prasss;
Print ";";
Print Empat;
Print ";";
Print Prassss;
Print ";";
Print Lima;

```

Print ";";
Print Prasssss;
Print ";";
Print Enam;
Print ";";
Print Prassssss;
Print ";";
Print Tujuh;
Print ";";
Print Prasssssss;
Print ";";
Print Delapan;
Print ";";
Print Prassssssss;
Print ";";
Print Sembilan;
Print ";";
Print Prasssssssss;
Print ";";
Print Sepuluh;
Print ";";
Print Prassssssssss
Goto Balix
Waitms 5
Reset Portd.7

```

Jika command bit yang diterima adalah angka ascii dari karakter 'h' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah melakukan penggantian password dan menyimpannya di EEPROM. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

Ganti_password:

```
Set Portd.7
Waitms 5
Z = 0
Inputstring = ""
Do
Dato = Waitkey()
nputstring = Inputstring + Chr(dato)
Incr Z
Loop Until Z = 4
Ps_lama = Val(inputstring)
  Readeeprom Bc1 , Psw_1
  Readeeprom Bc2 , Psw_2
  Readeeprom Bc3 , Psw_3
  Readeeprom Bc4 , Psw_4
  Readeeprom Bc5 , Psw_5
  Readeeprom Bc6 , Psw_6
  Readeeprom Bc7 , Psw_7
  Readeeprom Bc8 , Psw_8
  Readeeprom Bc9 , Psw_9
  Readeeprom Bc10 , Psw_10
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
  Readeeprom Enam
  Readeeprom Tujuh
  Readeeprom Delapan
  Readeeprom Sembilan
  Readeeprom Sepuluh
```

```

Z = 0
Inputstring = ""
Do
Dato = Waitkey()
Inputstring = Inputstring + Chr(dato)
Incr Z
Loop Until Z = 4
Ps_br = Val(inputstring)
If Ps_lama = Bc1 Then
    Writeeprom Ps_br , Psw_1
    Print "password " ; Satu ; " diganti dari " ; Bc1 ; " jd " ; Ps_br
Elseif Ps_lama = Bc2 Then
    Writeeprom Ps_br , Psw_2
    Print "password " ; Dua ; " diganti dari " ; Bc2 ; " jd " ; Ps_br
Elseif Ps_lama = Bc3 Then
    Writeeprom Ps_br , Psw_3
    Print "password " ; Tiga ; " diganti dari " ; Bc3 ; " jd " ; Ps_br
Elseif Ps_lama = Bc4 Then
    Writeeprom Ps_br , Psw_4
    Print "password " ; Empat ; " diganti dari " ; Bc4 ; " jd " ; Ps_br
Elseif Ps_lama = Bc5 Then
    Writeeprom Ps_br , Psw_5
    Print "password " ; Lima ; " diganti dari" ; Bc5 ; " jd " ; Ps_br
Elseif Ps_lama = Bc6 Then
    Writeeprom Ps_br , Psw_6
    Print "password " ; Enam ; " diganti dari" ; Bc6 ; " jd " ; Ps_br
Elseif Ps_lama = Bc7 Then
    Writeeprom Ps_br , Psw_7
    Print "password " ; Tujuh ; " diganti dari" ; Bc7 ; " jd " ; Ps_br
Elseif Ps_lama = Bc8 Then
    Writeeprom Ps_br , Psw_8
    Print "password " ; Delapan ; " diganti dari" ; Bc8 ; " jd " ; Ps_br

```

```

Elseif Ps_lama = Bc9 Then
    Writeeprom Ps_br , Psw_9
    Print "password " ; Sembilan ; " diganti dari" ; Bc9 ; " jd " ; Ps_br
Elseif Ps_lama = Bc10 Then
    Writeeprom Ps_br , Psw_10
    Print "password " ; Sepuluh ; " diganti dari" ; Bc10 ; " jd " ; Ps_br
Else
    Print "password anda tdk d kenal! coba lagi!!"
    Goto Balix
End If
Goto Balix
Waitms 5
Reset Portd.7

```

Jika command bit yang diterima adalah angka ascii dari karakter 'i' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah melakukan penghapusan data nama & password yang tersimpan di EEPROM. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

```

Hapus_data:
Wan = 0
Iew = &H0000
    Writeeprom Wan , Awali
Entire_password = ""
Writeeprom Entire_password , Dat_1
Entire_password = ""
Writeeprom Entire_password
Entire_password = ""
Writeeprom Entire_password

```

```

Entire_password = ""
Writeeprom Entire_password
Entire_password = ""
Writeeprom Iew , Psw_1
Writeeprom Iew , Psw_2
Writeeprom Iew , Psw_3
Writeeprom Iew , Psw_4
Writeeprom Iew , Psw_5
Writeeprom Iew , Psw_6
Writeeprom Iew , Psw_7
Writeeprom Iew , Psw_8
Writeeprom Iew , Psw_9
Writeeprom Iew , Psw_10
Goto Balix

```

Jika command bit yang diterima adalah angka ascii dari karakter 'j' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah melakukan registrasi nama & password yang kemudian disimpan di EEPROM. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

Registrasi_orang:

```

Set Portd.7
Waitms 5
Readeeprom Wan , Awali
If Wan = 10 Then
Print "Data Full...Sudah Ga Bisa SimDelapan Data Lagi!! max cm 10!!"
Goto Balix
End If
Input "masukan nama = " , Entire_name
Input "masukan password = " , Entire_password
Pras = Val(entire_password)
If Wan = 0 Then
    Writeeprom Entire_name , Dat_1
    Incr Wan
    Writeeprom Wan , Awali
    Writeeprom Pras , Psw_1
    Goto Balix
Elseif Wan = 1 Then
    Readeeprom Satu , Dat_1
    Writeeprom Satu , Dat_1
    Writeeprom Entire_name
    Incr Wan
    Writeeprom Wan , Awali
    Writeeprom Pras , Psw_2
    Goto Balix
Elseif Wan = 2 Then
    Readeeprom Satu , Dat_1
    Readeeprom Dua
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Entire_name
    Incr Wan

```

```
Writeeprom Wan , Awali
Writeeprom Pras , Psw_3
Elseif Wan = 3 Then
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Writeeprom Satu , Dat_1
  Writeeprom Dua
  Writeeprom Tiga
  Writeeprom Entire_name
  Incr Wan
  Writeeprom Wan , Awali
  Writeeprom Pras , Psw_4
Elseif Wan = 4 Then
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Writeeprom Satu , Dat_1
  Writeeprom Dua
  Writeeprom Tiga
  Writeeprom Empat
  Writeeprom Entire_name
  Incr Wan
  Writeeprom Wan , Awali
  Writeeprom Pras , Psw_5
Elseif Wan = 5 Then
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
```

```
Writeeprom Satu , Dat_1
Writeeprom Dua
Writeeprom Tiga
Writeeprom Empat
Writeeprom Lima
Writeeprom Entire_name
Incr Wan
Writeeprom Wan , Awali
Writeeprom Pras , Psw_6
Elseif Wan = 6 Then
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
  Readeeprom Enam
  Writeeprom Satu , Dat_1
  Writeeprom Dua
  Writeeprom Tiga
  Writeeprom Empat
  Writeeprom Lima
  Writeeprom Enam
  Writeeprom Entire_name
  Incr Wan
  Writeeprom Wan , Awali
  Writeeprom Pras , Psw_7
Elseif Wan = 7 Then
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
```

```
Readeeprom Enam
Readeeprom Tujuh
Writeeprom Satu , Dat_1
Writeeprom Dua
Writeeprom Tiga
Writeeprom Empat
Writeeprom Lima
Writeeprom Enam
Writeeprom Tujuh
Writeeprom Entire_name
Incr Wan
Writeeprom Wan , Awali
Writeeprom Pras , Psw_8
Elseif Wan = 8 Then
    Readeeprom Satu , Dat_1
    Readeeprom Dua
    Readeeprom Tiga
    Readeeprom Empat
    Readeeprom Lima
    Readeeprom Enam
    Readeeprom Tujuh
    Readeeprom Delapan
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Entire_name
Incr Wan
```

```
Writeeprom Wan , Awali
Writeeprom Pras , Psw_9
Else
  Readeeprom Satu , Dat_1
  Readeeprom Dua
  Readeeprom Tiga
  Readeeprom Empat
  Readeeprom Lima
  Readeeprom Enam
  Readeeprom Tujuh
  Readeeprom Delapan
  Readeeprom Sembilan
  Writeeprom Satu , Dat_1
  Writeeprom Dua
  Writeeprom Tiga
  Writeeprom Empat
  Writeeprom Lima
  Writeeprom Enam
  Writeeprom Tujuh
  Writeeprom Delapan
  Writeeprom Sembilan
  Writeeprom Entire_name
  Incr Wan
  Writeeprom Wan , Awali
  Writeeprom Pras , Psw_10
End If
Waitms 5
Reset Portd.7
Goto Balix
```

Jika command bit yang diterima adalah angka ascii dari karakter 'k' maka dan diakhiri dengan karakter '#' proses selanjutnya yang dilakukan microcontroller adalah mengganti nama lama dengan nama baru yang kemudian disimpan di EEPROM. Berikut adalah potongan program yang dijalankan ketika diterima paket data tersebut :

Ganti_nama:

Set Portd.7

Waitms 5

Input "Nama yg mau d ganti : " , Nama_baru

Readeeprom Satu , Dat_1

Readeeprom Dua

Readeeprom Tiga

Readeeprom Empat

Readeeprom Lima

Readeeprom Enam

Readeeprom Tujuh

Readeeprom Delapan

Readeeprom Sembilan

Readeeprom Sepuluh

Input "Nama baru : " , Inputstring

If Nama_baru = Satu Then

Writeeprom Inputstring , Dat_1

Writeeprom Dua

Writeeprom Tiga

Writeeprom Empat

Writeeprom Lima

Writeeprom Enam

Writeeprom Tujuh

Writeeprom Delapan

```

Writeeprom Sembilan
Writeeprom Sepuluh
Print "Nama diganti dari " ; Satu ; " jd " ; Inputstring
Elseif Nama_baru = Dua Then
    Writeeprom Satu , Dat_1
    Writeeprom Inputstring
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Sembilan
    Writeeprom Sepuluh
Print "Nama diganti dari " ; Dua ; " jd " ; Inputstring
Elseif Nama_baru = Tiga Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Inputstring
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Sembilan
    Writeeprom Sepuluh
Print "Nama diganti dari " ; Tiga ; " jd " ; Inputstring
Elseif Nama_baru = Empat Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Inputstring

```

```

Writeeprom Lima
Writeeprom Enam
Writeeprom Tujuh
Writeeprom Delapan
Writeeprom Sembilan
Writeeprom Sepuluh
Print "Nama diganti dari " ; Empat ; " jd " ; Inputstring
Elseif Nama_baru = Lima Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Inputstring
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Sembilan
    Writeeprom Sepuluh
Print "Nama diganti dari " ; Lima ; " jd " ; Inputstring
Elseif Nama_baru = Enam Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Inputstring
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Sembilan
    Writeeprom Sepuluh
Print "Nama diganti dari " ; Enam ; " jd " ; Inputstring
Elseif Nama_baru = Tujuh Then

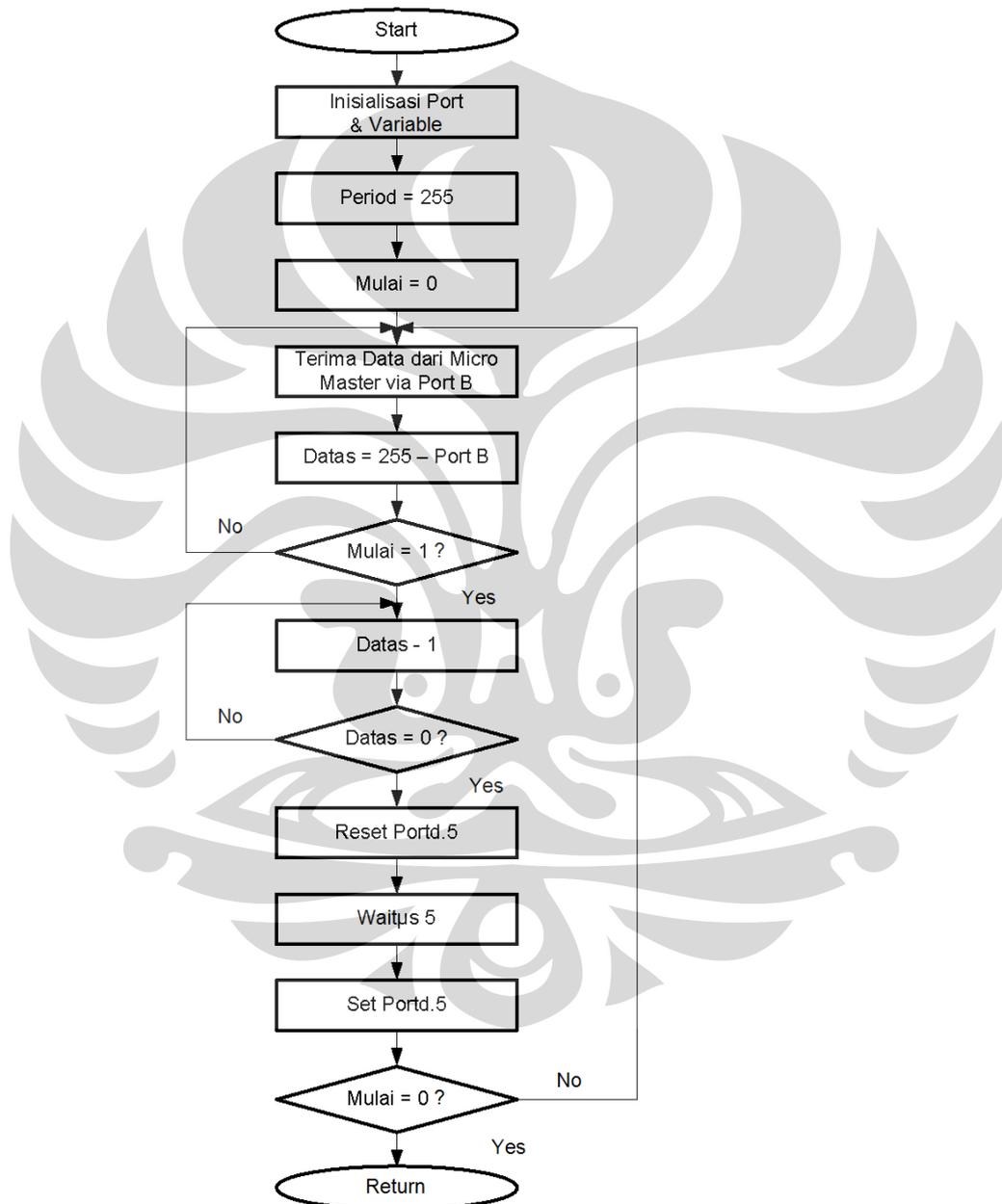
```

```
Writeeprom Satu , Dat_1
Writeeprom Dua
Writeeprom Tiga
Writeeprom Empat
Writeeprom Lima
Writeeprom Enam
Writeeprom Inputstring
Writeeprom Delapan
Writeeprom Sembilan
Writeeprom Sepuluh
Print "Nama diganti dari " ; Tujuh ; " jd " ; Inputstring
Elseif Nama_baru = Delapan Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Inputstring
    Writeeprom Sembilan
    Writeeprom Sepuluh
Print "Nama diganti dari " ; Delapan ; " jd " ; Inputstring
Elseif Nama_baru = Sembilan Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
```

```
Writeeprom Inputstring
Writeeprom Sepuluh
Print "Nama diganti dari " ; Sembilan ; " jd " ; Inputstring
Elseif Nama_baru = Sepuluh Then
    Writeeprom Satu , Dat_1
    Writeeprom Dua
    Writeeprom Tiga
    Writeeprom Empat
    Writeeprom Lima
    Writeeprom Enam
    Writeeprom Tujuh
    Writeeprom Delapan
    Writeeprom Sembilan
    Writeeprom Inputstring
Print "Nama diganti dari " ; Sepuluh ; " jd " ; Inputstring
Else
    Print "Nama anda tdk d kenal! coba lagi!!"
    Waitms 5
    Reset Portd.7
    Goto Balix
End If
Goto Balix
```

3.2.2.4 Flowchart Program Micro Slave (AT90S2313)

Berikut ini adalah flowchart dari program utama sistem microcontroller (slave) yang dibuat :



Gambar 3.21 Flowchart program utama sistem microcontroller (AT90S2313)

Microcontroller ini berfungsi untuk mengatur intensitas cahaya lampu AC. Microcontroller ini akan menerima data dari micro master melalui port B. Perancangan softwrenya terdiri atas main program serta interrupt eksternal INT0. Flowchart program di atas menggambarkan bahwa microcontroller (slave) menerima data dari microcontroller master dengan memakai data delapan bit yang telah dikonfigurasi sebagai input. Data tersebut digunakan untuk mengurangi periode gelombang sehingga didapatkan data yang baru. Data baru merupakan hasil dari $255 - \text{Data (port B)}$. Pada saat terjadi interrupt maka data ini akan terus dikurangkan dengan satu hingga nilai datanya sama dengan nol.

Jika nol maka intruksi selanjutnya adalah mereset triac dalam kondisi ini lampu akan menyala kemudian menset kembali triac tersebut sehingga lampu mati. Delay pada saat kondisi nyala dan mati sebesar $5 \mu\text{s}$. Karena proses interruptnya terjadi sangat cepat sehingga lampu terlihat lebih stabil.

Pada saat terjadi terjadi interrupt maka program akan menjalankan subroutine berikut.

Sindoro:

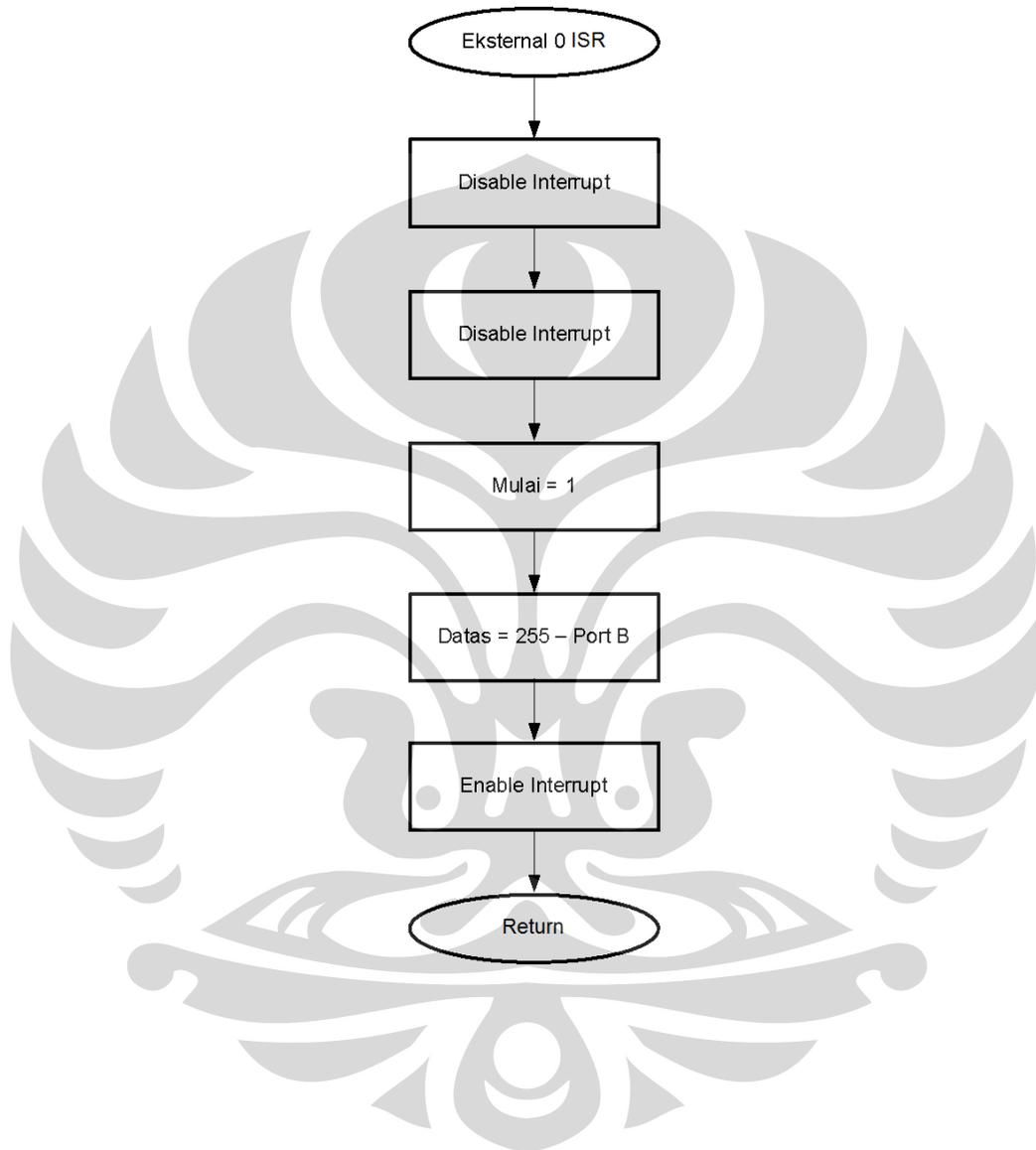
Mulai = 1

Datas = Iwiw

Return

3.2.2.5 Flowchart Program Micro Interrupt Eksternal INT0 (AT90S2313)

Berikut ini adalah flowchart untuk interrupt eksternal INT0



Gambar 3.22 Flowchart program interrupt eksternal INT0 (AT90S2313)

BAB 4

HASIL EKSPERIMEN DAN PEMBAHASAN

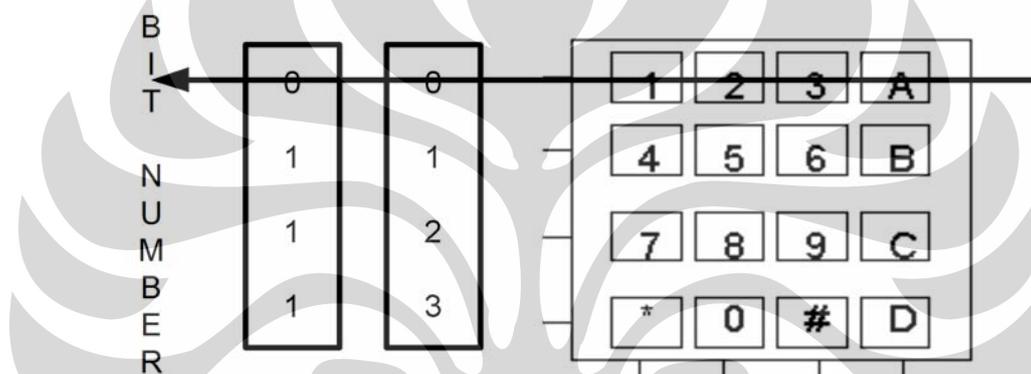
Setelah dilakukan pengerjaan keseluruhan sistem, maka perlu dilakukan pengujian alat serta penganalisaan terhadap alat, apakah sistem sudah bekerja dengan baik atau tidak. Pengujian-pengujian tersebut meliputi pengujian terhadap sistem keamanan (security), pengujian pendeteksian kondisi peralatan listrik, pengujian pengaturan lampu AC baik pada lantai 1 maupun pada lantai 2.

4.1 Pengujian Sistem Keamanan (Security)

Pada sistem ini kita dapat memasukkan data nama dan password ke dalam EEPROM yang masih kosong, sehingga terjadi suatu proses registrasi / penyimpanan nama dan password ke dalam EEPROM. Setelah melakukan registrasi nama & password tersebut, maka jika sewaktu-waktu kita ingin menghapus atau menggantinya dengan data yang baru kita dapat melakukannya. Berikut ini data yang didapatkan selama pengujian terhadap lantai 1 & 2 seperti pada tabel 4.1 & 4.2 yang terdapat pada bagian lampiran.

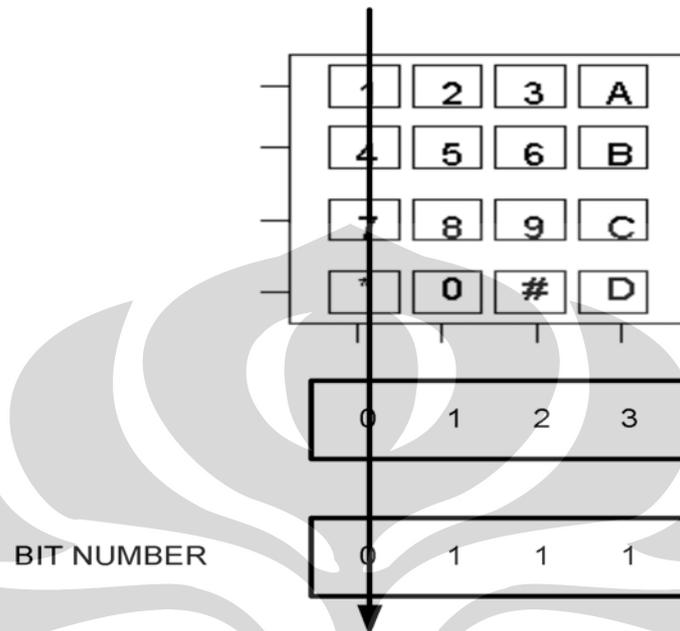
Seperti pada table tersebut setelah nama & password terigristrasi, maka hanya nama-nama orang tersebut yang berhak masuk ke dalam ruangan tersebut dengan memasukkan password tertentu sesuai yang tersimpan di EEPROM. Pada saat memasukkan password berarti akan terjadi interrupt yang disebabkan karena adanya penekanan tombol pada keypad.

Sebagai contoh apabila kita memiliki password 1234 maka proses pembacaan untuk angka 1 pada keypad maka seperti pada gambar 3.9 pada skematik keypad 4 x 4, maka ketika angka 1 ditekan maka akan terjadi hubungan antara baris (input, logic high) dengan kolom (output, logic low). Sehingga output (pin 6) IC 74ALS20 akan bersifat logic high karena salah satu input yaitu pin 1 bersifat logic high. Setelah melewati IC 7414 maka akan bersifat logic low.



Gambar 4.1 Bit number baris pembacaan angka 1

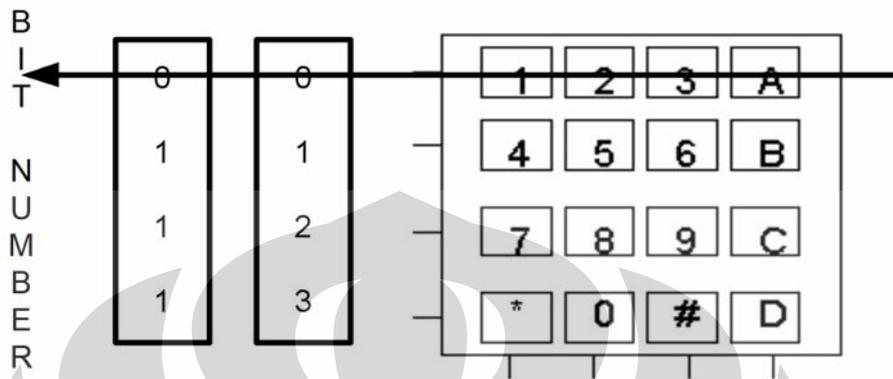
Seperti pada gambar di atas pada saat angka 1 ditekan maka pada baris akan memiliki bit number 1110. Nilai bit tersebut apabila kita konversi ke nilai decimal akan memiliki nilai 14 dan akan menjalankan case 14 : keycode = 1 pada subroutine interrupt keypad. Kemudian bit number untuk bagian kolom penjelasannya dapat dilihat pada gambar berikut.



Gambar 4.2 Bit number kolom pembacaan angka 1

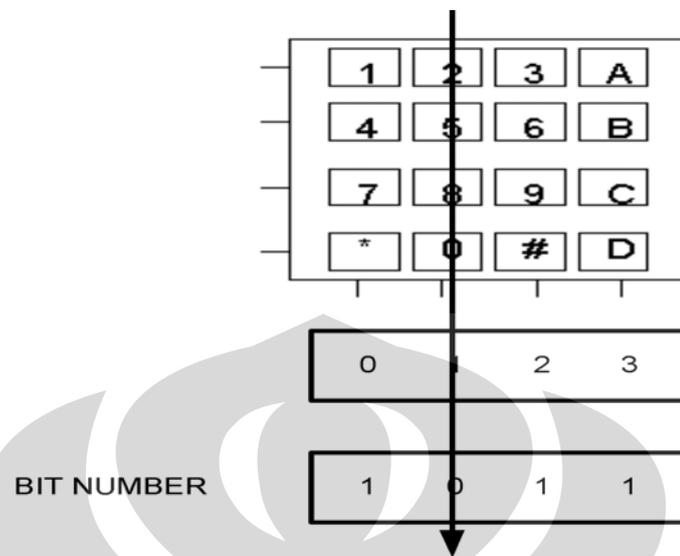
Seperti pada gambar di atas nilai bit number pada bagian kolom memiliki nilai 1110 dan bila dikonversi ke nilai decimal akan memiliki nilai decimal 14 dan akan menjalankan case 14 : $\text{keycode} = \text{keycode} + 0$. Karena pada baris didapat $\text{keycode} = 1$. Berarti $\text{keycode} = \text{keycode} + 0$, $\text{keycode} = 1 + 0 = 1$.

Proses pembacaan pada angka 2 dapat dijelaskan pada gambar berikut.



Gambar 4.3 Bit number baris pembacaan angka 2

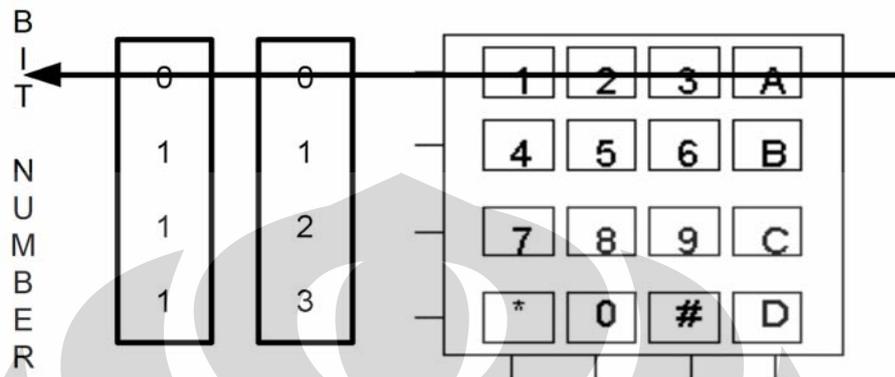
Pada pembacaan angka 2 bit number barisnya akan memiliki nilai 1110 dan bila dikonversi ke nilai decimal akan memiliki nilai 14. Maka secara software akan menjalankan case 14 : keycode = 1. Kemudian pembacaan untuk kolomnya dapat dipahami pada gambar berikut di bawah ini.



Gambar 4.4 Bit number kolom pembacaan angka 2

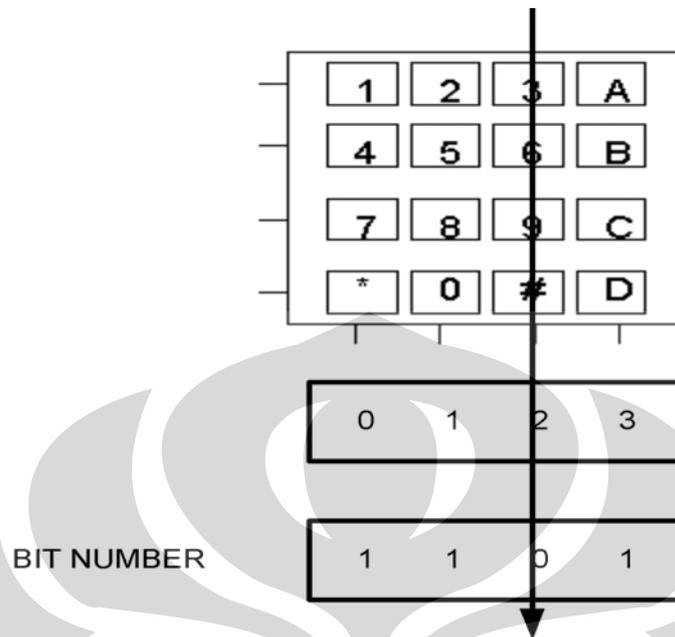
Seperti pada gambar di atas nilai bit number pada bagian kolom memiliki nilai 1101 dan bila dikonversi ke nilai decimal akan memiliki nilai decimal 13 dan akan menjalankan case 13 : $\text{keycode} = \text{keycode} + 1$. Karena pada baris didapat $\text{keycode} = 1$. Berarti $\text{keycode} = \text{keycode} + 1$, $\text{keycode} = 1 + 1 = 2$.

Proses pembacaan pada angka 3 dapat dijelaskan pada gambar berikut.



Gambar 4.5 Bit number baris pembacaan angka 3

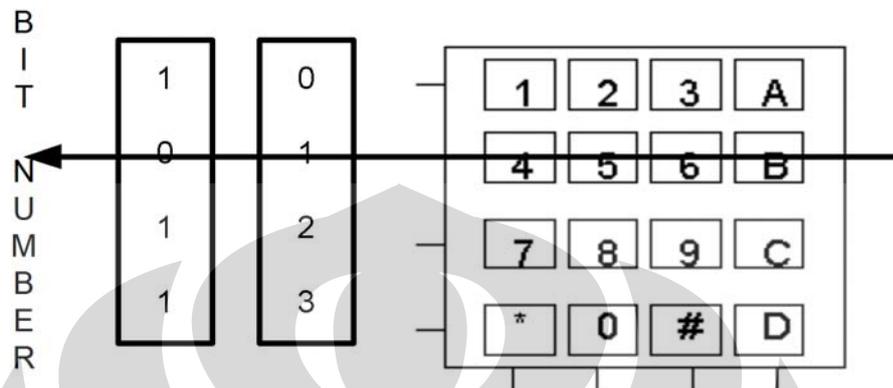
Pada pembacaan angka 3 bit number barisnya akan memiliki nilai 1110 dan bila dikonversi ke nilai decimal akan memiliki nilai 14. Maka secara software akan menjalankan case 14 : keycode = 1. Kemudian pembacaan untuk kolomnya dapat dipahami pada gambar berikut di bawah ini. Kemudian untuk pembacaan kolomnya dapat dipahami pada gambar berikut.



Gambar 4.6 Bit number kolom pembacaan angka 3

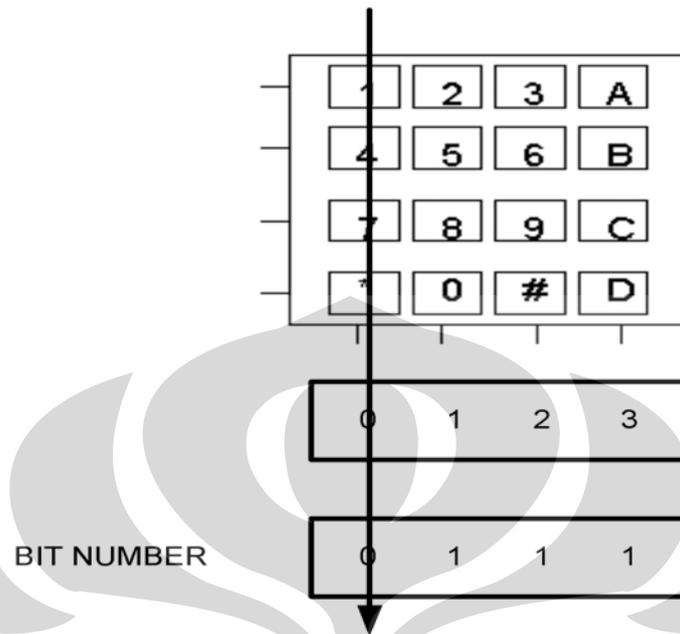
Seperti pada gambar di atas nilai bit number pada bagian kolom memiliki nilai 1011 dan bila dikonversi ke nilai decimal akan memiliki nilai decimal 11 dan akan menjalankan case 11 : $\text{keycode} = \text{keycode} + 2$. Karena pada baris didapat $\text{keycode} = 1$. Berarti $\text{keycode} = \text{keycode} + 2$, $\text{keycode} = 1 + 2 = 3$.

Proses pembacaan pada angka 4 dapat dijelaskan pada gambar berikut.



Gambar 4.7 Bit number baris pembacaan angka 4

Pada pembacaan angka 3 bit number barisnya akan memiliki nilai 1101 dan bila dikonversi ke nilai decimal akan memiliki nilai 13. Maka secara software akan menjalankan $\text{case } 13 : \text{keycode} = 4$. Kemudian pembacaan untuk kolomnya dapat dipahami pada gambar berikut di bawah ini. Kemudian untuk pembacaan kolomnya dapat dipahami pada gambar berikut.



Gambar 4.8 Bit number kolom pembacaan angka 4

Seperti pada gambar di atas nilai bit number pada bagian kolom memiliki nilai 1110 dan bila dikonversi ke nilai decimal akan memiliki nilai decimal 14 dan akan menjalankan case 14 : $\text{keycode} = \text{keycode} + 0$. Karena pada baris didapat $\text{keycode} = 4$. Berarti $\text{keycode} = \text{keycode} + 0$, $\text{keycode} = 4 + 0 = 4$.

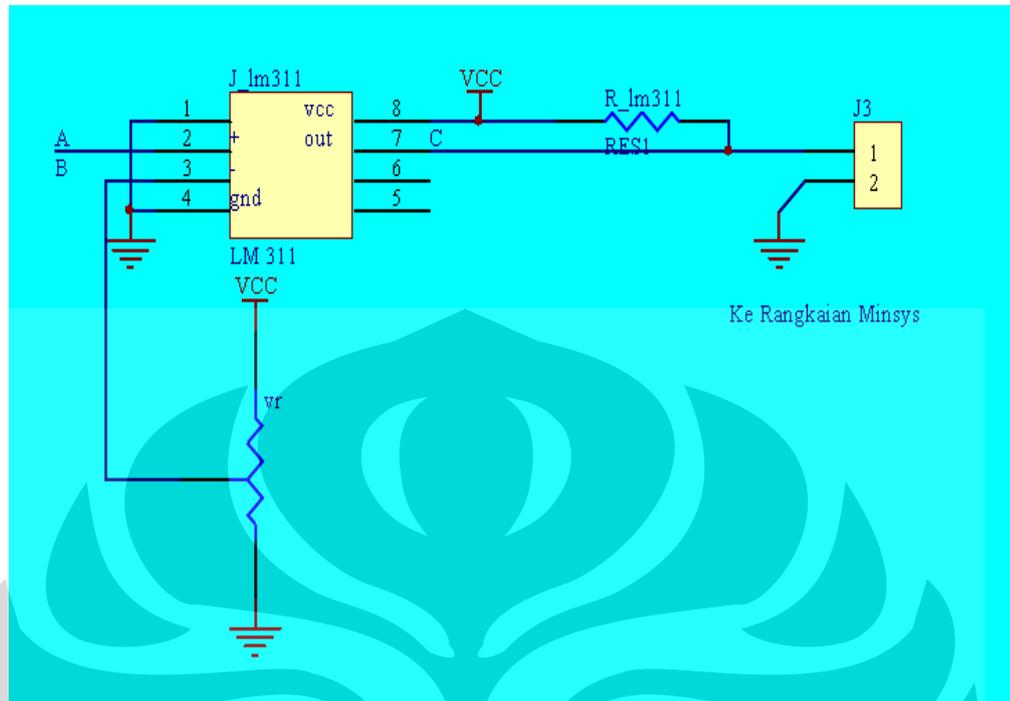
Selain itu data nama dan password pada sistem ini bisa dirubah baik satu persatu maupun secara keseluruhan seperti yang tercantum pada table 4.3 dan 4.4 yang terdapat pada bagian lampiran.

Pada tabel 4.1 & 4.2 merupakan pengambilan data dimana pada awalnya EEPROM dalam keadaan kosong, kemudian dimasukkan data nama dan password secara satu persatu hingga mencapai 10 orang. Sedangkan pada tabel 4.3 & 4.4 merupakan data dimana sistem diuji kehandalannya yaitu dengan mengganti data nama dan password baik satu persatu maupun secara keseluruhan. Pada pengujian tersebut data berhasil disimpan, dihapus maupun diganti-ganti dengan baik. Pengujian juga dilakukan terhadap membuka atau tidaknya pintu pada sistem tersebut dengan melakukan pengecekan data yang telah dimasukkan. Berdasarkan data yang diperoleh pada sistem ini tidak ditemukan masalah yang berarti.

4.6 Pengujian Pendeteksi Kondisi Peralatan Listrik

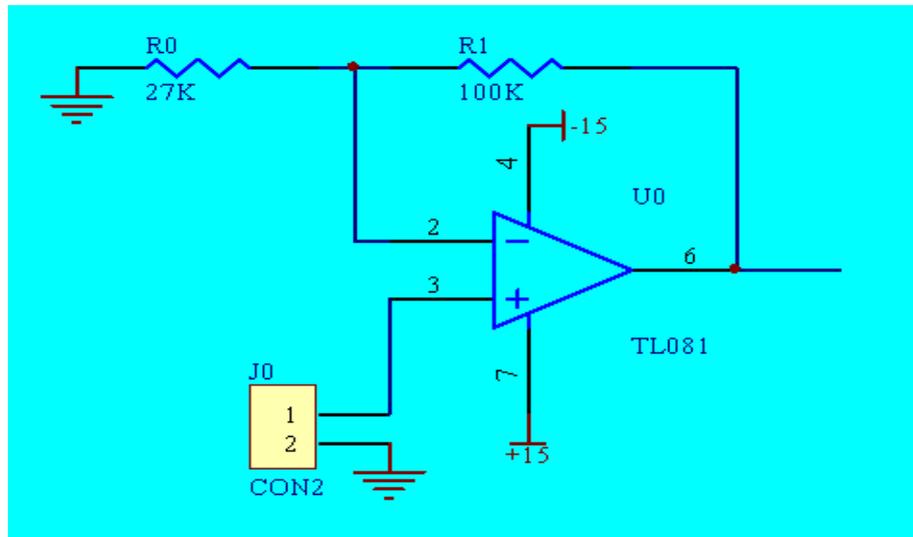
Pengambilan data dilakukan dengan memainkan saklar baik dalam kondisi ON / OFF yang dilakukan pada tiga titik, yaitu :

- Pada titik A yang merupakan tegangan input : $V_A = \frac{R_{CT}}{R_{CT} + R_1} V_{CC}$
- Pada titik B yang merupakan tegangan referensi.
- Pada titik C yang merupakan tegangan output sensor.



Gambar 4.9 Rangkaian comparator pada rangkaian pendeteksi

Seperti pada tabel 4.5 & 4.6 yang terdapat pada bagian lampiran tercatat bahwa pada saat sensor arus dialiri arus listrik maka akan ada input yang masuk ke rangkaian non inverting amplifier (pin3) seperti pada gambar di bawah ini.



Gambar 4.10 Rangkaian non-inverting amplifler pada rangkaian pendeteksi

Karena sebagai rangkaian non-inverting amplifler maka bentuk gelombang keluarannya sama dengan gelombang masukannya atau sefasa. Pada rangkaian ini penguatannya adalah :

$$A_1 = A_{fb} = \frac{R_f}{R_1} + 1$$

Dimana :

$$R_0 = R_1 = 27 K\Omega$$

$$R_1 = R_f = 100 K\Omega$$

Maka sesuai persamaan diatas pada rangkaian non-inverting pada rangkaian pendeteksi penguatannya adalah sebagai berikut :

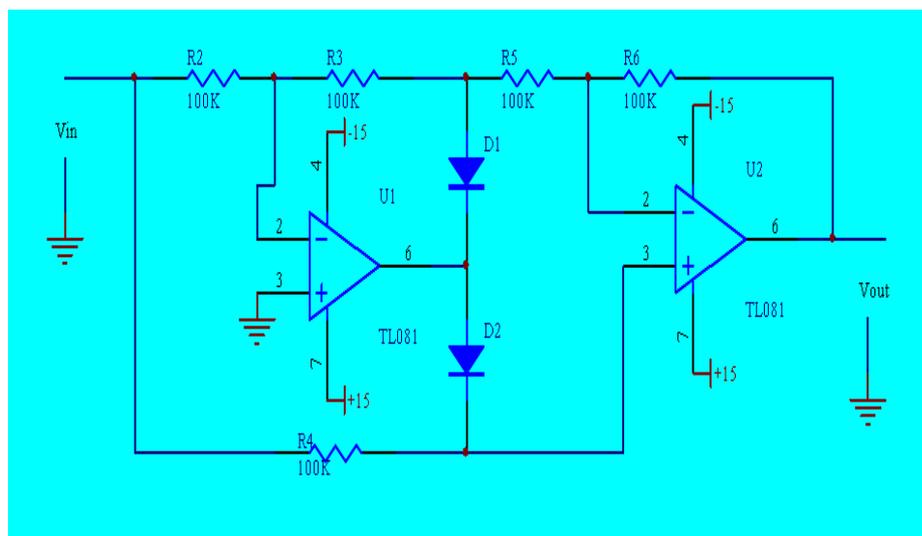
$$A_1 = A_{fb} = \frac{100K\Omega}{27K\Omega} + 1$$

$$A_1 = A_{fb} = \frac{100K\Omega + 27K\Omega}{27K\Omega}$$

$$A_1 = A_{fb} = \frac{127K\Omega}{27K\Omega}$$

$$A_1 = A_{fb} = 4.7$$

Pada keluaran sensor *current transformer* terukur outputnya adalah sekitar 46,7 mV yang masuk pada pin 2 (-) op-amp pertama pada saat dialiri arus listrik. Hal tersebut berarti $V_{in} = 46,7$ mV dengan penguatan pada op-amp pertama $A_1 = 4,7$ kali dari tegangan input maka tegangan outputnya menjadi 219,49 mV. Tegangan ini kemudian masuk ke rangkaian *precision rectifier* seperti pada gambar berikut.



Gambar 4.11 Rangkaian *precision rectifier* pada rangkaian pendeteksi

Pada rangkaian ini terminal inverting (pin2) dihubungkan ke virtual ground melalui resistor R3 dan R5. Sehingga penguatan pada op-amp ke tiga memenuhi persamaan berikut :

$$A_3 = A_{fb} = 1 + \frac{R_6}{R_3 + R_5}$$

Dimana :

$$R_3 = R_5 = R_6 = 100K\Omega$$

Maka sesuai persamaan di atas penguatan pada op-amp ke tiga adalah sebagai berikut.

$$A_3 = A_{fb} = 1 + \frac{100K\Omega}{100K\Omega + 100K\Omega}$$

$$A_3 = A_{fb} = \frac{200K\Omega + 100K\Omega}{200K\Omega}$$

$$A_3 = A_{fb} = \frac{300K\Omega}{200K\Omega}$$

$$A_3 = A_{fb} = 1,5$$

Adapun penguatan pada op-amp kedua (*inverting amplifier*) A_2 tidak melebihi satu. Hal itu disebabkan karena tidak semua arus yang mengalir melalui resistor R2 mengalir melalui R4. Tegangan pada titik C akan sama dengan tegangan pada titik B pada saat V_{in} pada op-amp ketiga hampir nol. Sehingga ada input arus yang mengalir melalui R3 dan R5 serta R4. Sehingga tegangan pada titik B sama

dengan $\frac{2}{3}$ Vin. Dimana pada saat tegangan di titik C sama dengan di titik B maka penguatan pada op-amp ke dua A_2 memenuhi persamaan berikut.

$$A_2 = A_{fb} = \frac{(R_3 + R_5) \parallel R_4}{R_2}$$

Dimana :

$$R_2 = R_3 = R_4 = R_5 = 100K\Omega$$

Sehingga penguatan pada op-amp ke dua adalah sebagai berikut.

$$A_2 = A_{fb} = \frac{(100K\Omega + 100K\Omega) \parallel 100K\Omega}{100K\Omega}$$

$$A_2 = A_{fb} = \frac{200K\Omega}{300K\Omega}$$

$$A_2 = A_{fb} = 0,66$$

Karena pada rangkaian ini semua resistor memiliki nilai yang sama maka penguatan totalnya adalah sebagai berikut.

$$A_{fbTotal} = A_1 \times A_2 = \left(\frac{2}{3}\right) \left(\frac{3}{2}\right)$$

$$A_{fbTotal} = 1$$

Dengan V_{in} sebesar 46,7 mV yang masuk ke rangkaian precision rectifier maka setelah melewatinya dari hasil pengukuran didapat hasil dari V_{out} adalah sebesar 8,73 V. Setelah itu tegangan ini akan melewati rangkaian *low pass filter* kemudian masuk ke rangkaian *voltage follower*. Adapun besarnya tegangan terukur pada pin 3 (+) op-amp ke empat sebesar 7,4 V, karena bersifat sebagai pengikut tegangan maka tegangan output yang dihasilkan pada *voltage follower* akan sama dengan tegangan inputnya yaitu sebesar 7,4 V. Setelah itu akan melewati dioda zener 5,6 V dan setelah melewati dioda ini didapat tegangan terukur sebesar 5,1 V yang nantinya akan menjadi input (pin2) bagi rangkaian komparator.

Pada saat tegangan (pin2) lebih besar dari tegangan referensi (pin3) maka akan dihasilkan output (pin7) yang bersifat logic high. Begitu pun sebaliknya pada saat peralatan listrik dimatikan tegangan input (pin2) lebih kecil dari tegangan referensi (pin3) maka akan dihasilkan output tegangan (pin7) yang bersifat logic low. Tegangan output yang bersifat logic high / low ini akan menjadi input bagi microcontroller.

Dari data-data yang didapatkan pada table 4.7 & 4.8 yang terdapat pada bagian lampiran diketahui bahwa pendeteksian dengan sensor arus berjalan dengan baik artinya selama kabel penghantar dialiri arus yang melewatinya, maka sensor arus ini masih akan dapat membaca kondisi suatu peralatan listrik. Sebaliknya jika kabel tersebut tidak mendapatkan supply tegangan maka sama artinya tidak ada arus yang melewati sensor tersebut, sehingga keluaran dari rangkaian pendeteksi akan bersifat logic low. Jika output dari kondisi ini bersifat logic low maka artinya peralatan listrik tersebut dalam keadaan OFF sebaliknya jika output dari rangkain pendeteksi bersifat

logic high maka kondisi peralatan listrik dalam keadaan ON. Secara hardware dapat dilihat dari keluarain IC LM311.

4.3 Pengujian Pengaturan Tegangan AC

Pengujian pada sistem ini dibandingkan dengan intensitas dan pembacaan pada Luxmeter. Berikut adalah pengambilan data pengaturan lampu AC. Pada pengujian ini data mulai dari 0 – 255 dimasukkan secara berurut dengan kelipatan 5. Hal ini dilakukan untuk mendapatkan data tegangan yang terukur. Adapun pengontrolan tegangan ini dilakukan melalui PC dengan memberikan nilai data tersebut (0 – 255) ke Port B microcontroller seperti yang terdapat pada tabel 4.9 & 4.10 yang tercantum pada bagian lampiran.

Dari hasil pengambilan data tersebut didapatkan bahwa semakin besar nilai data (0 – 255) yang kita masukkan maka tegangan yang masuk ke lampu akan semakin besar sehingga intensitas yang terbaca oleh sensor OPT101 dan Luxmeter akan semakin besar. Begitupun sebaliknya apabila nilai data yang kita masukkan semakin kecil maka tegangan yang diberikan ke lampu akan semakin kecil pula sehingga akan didapatkan intensitas yang semakin kecil.

Pengambilan datanya dilakukan dengan memberikan power melalui paket data serial 8 bit ke microcontroller master pada port B yang dikonfigurasi sebagai output. Kemudian data 8 bit tersebut nantinya akan diterima oleh microcontroller slave yang dikonfigurasi sebagai input. Data 8 bit inilah yang nantinya mengatur besar kecilnya tegangan yang akan diberikan ke lampu.

4.6 Pengujian Komunikasi Serial

Pengujian ini dimaksudkan sebagai pengujian awal apakah *microcontroller* dapat menerima data dari PC dan mengirimkannya kembali ke PC dengan benar. Pengujian komunikasi serial ini menggunakan program *HyperTerminal* yang sudah disediakan oleh windows. Pengujian dilakukan dengan menggunakan komunikasi RS-232 dan RS-485. Adapun parameter yang digunakan untuk mengkonfigurasi *Hyper Terminal* adalah sebagai berikut :

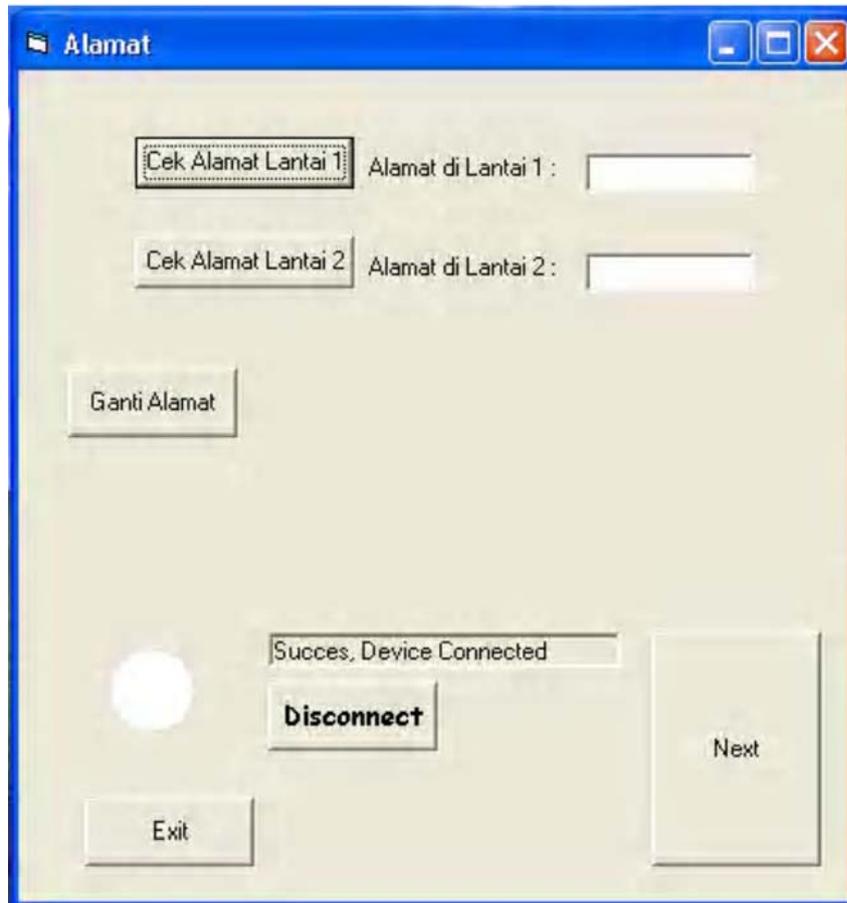
Baudrate = 19200 bps, Data Bit = 8, Parity = none, Stop Bit = 1

Dari data-data yang terdapat pada dua tabel 11 & 12 pada bagian lampiran, dapat dilihat bahwa tidak ada masalah dengan rangkaian komunikasi serial RS-232 dan RS-485 yang ada pada minimum system yang dibuat.

4.6 Pengujian Protokol Komunikasi Serial

Pengambilan data pada protokol komunikasi serial ini dilakukan dengan cara memanggil protokol-protokol sesuai format protokol yang telah di jelaskan pada bab 3.2.1 tabel 3.5. Pengujian tersebut dilakukan dengan menggunakan komunikasi RS-485 dengan visualisasi visual basic. Parameter yang digunakan adalah sebagai berikut:

Baudrate = 19200 bps, Data Bit = 8, Parity = none, Stop Bit = 1

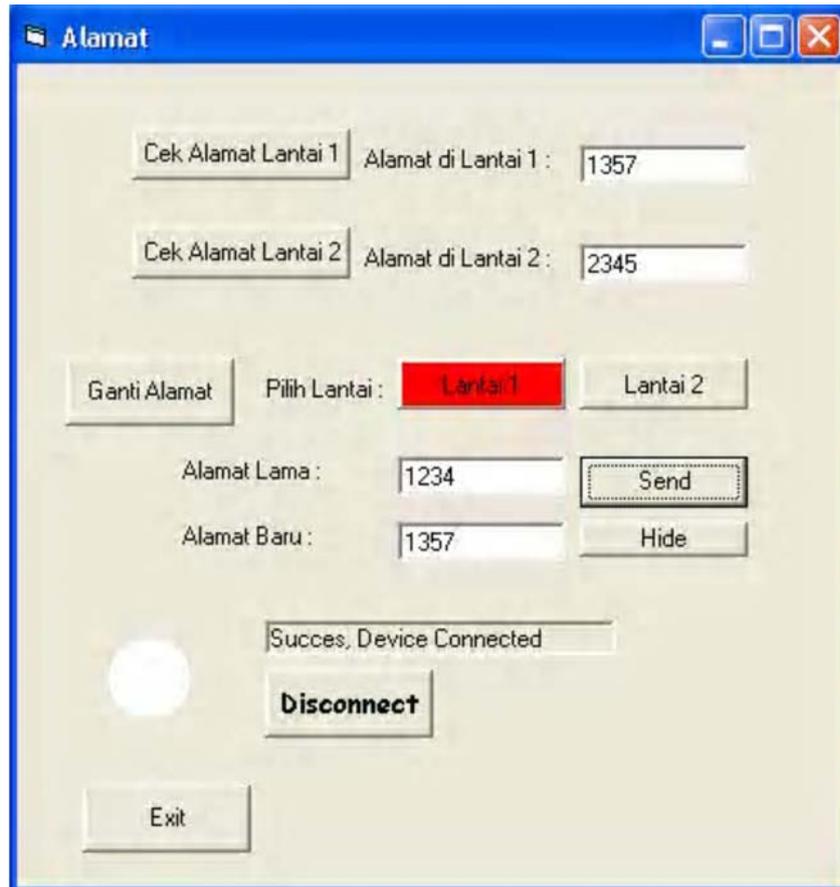


Gambar 4.13 Tampilan awal ketika program pertama kali dijalankan

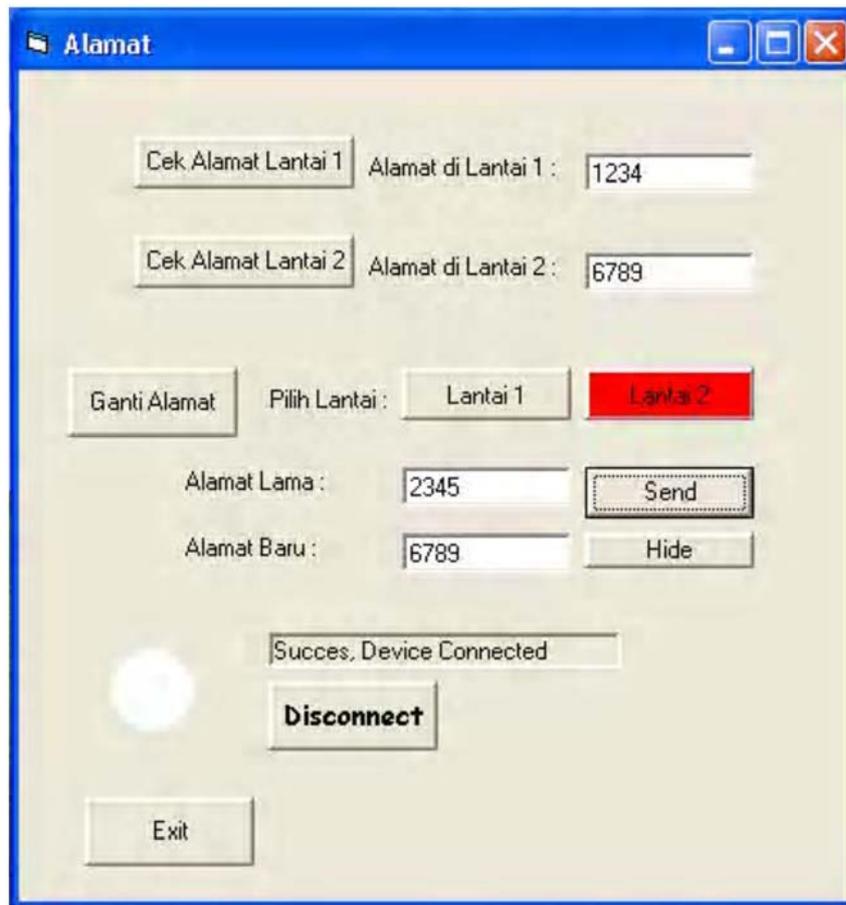
Pada saat pertama kali dijalankan (pertama kali program di *download*) , sistem tersebut memiliki alamat default yaitu 1234 (lantai 1) / 2345 (lantai 2) yang nantinya dapat diubah-ubah sesuai keinginan. Selanjutnya ketika dijalankan kembali, sistem akan membaca alamat yang tersimpan pada EEPROM internal microcontroller.

Perintah yang diketikkan untuk mengganti alamat adalah : *1234d1357# (lantai 1) atau *2345d6789# (lantai 2), yang digunakan saat PC ingin mengubah alamat sebelumnya yang ada pada sistem (1234 / 2345) dengan alamat yang baru seperti

pada gambar di bawah ini. Proses ini akan mengakibatkan data alamat yang sebelumnya tersimpan pada EEPROM akan dihapus dan digantikan oleh data alamat yang baru.



Gambar 4.14 Tampilan pada saat ada permintaan penggantian alamat pada lantai 1

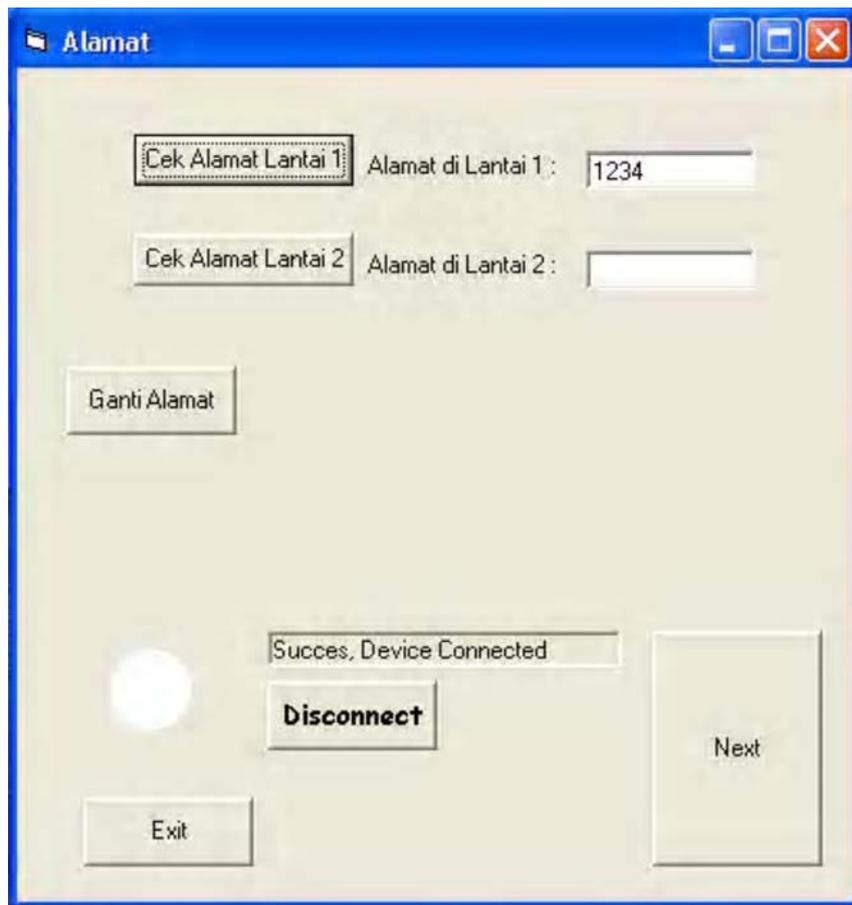


Gambar 4.15 Tampilan pada saat ada permintaan penggantian alamat pada lantai 2

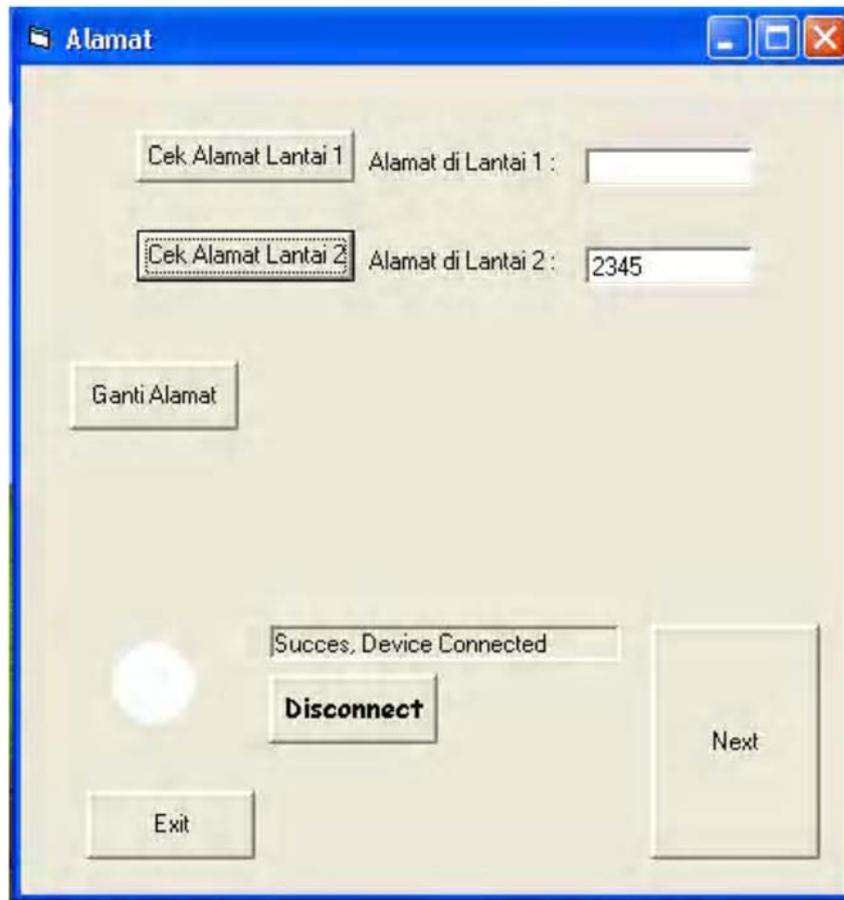
Setelah alamat diganti, maka seluruh protokol pengiriman data dari PC ke mikrokontroler menggunakan alamat yang baru. Saat menerima paket data dari PC, sistem akan membandingkan alamat yang diterima dengan alamat yang tersimpan pada EEPROM. Bila alamat yang digunakan tidak cocok, maka mikrokontroler akan mengabaikan paket data tersebut.

Perintah yang diketikkan selanjutnya: *0001e# (lantai 1) atau *0002e# (lantai 2) , yang digunakan saat PC ingin meminta alamat dari sistem tersebut. Alamat yang

digunakan adalah 0001 untuk lantai 1 dan 0002 untuk lantai 2. Pada saat mikrokontroler menerima paket data tersebut, maka akan segera membaca data alamat yang tersimpan di EEPROM dan mengirimkannya kembali ke PC.

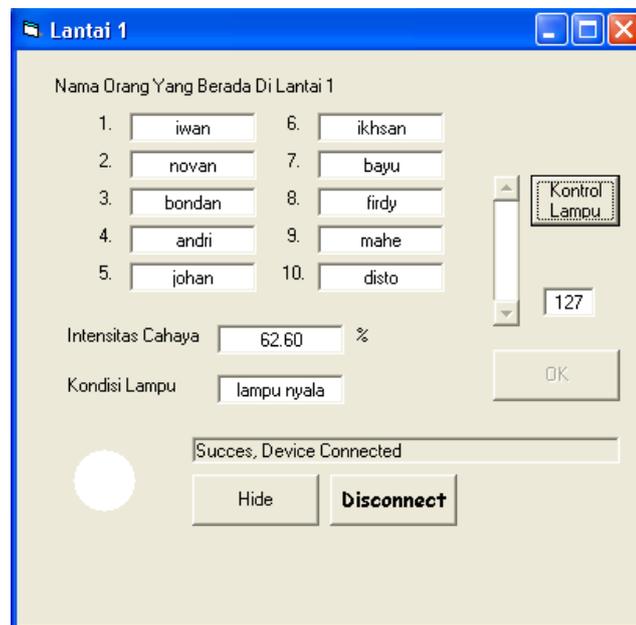


Gambar 4.16 Tampilan pada saat ada permintaan alamat pada lantai 1

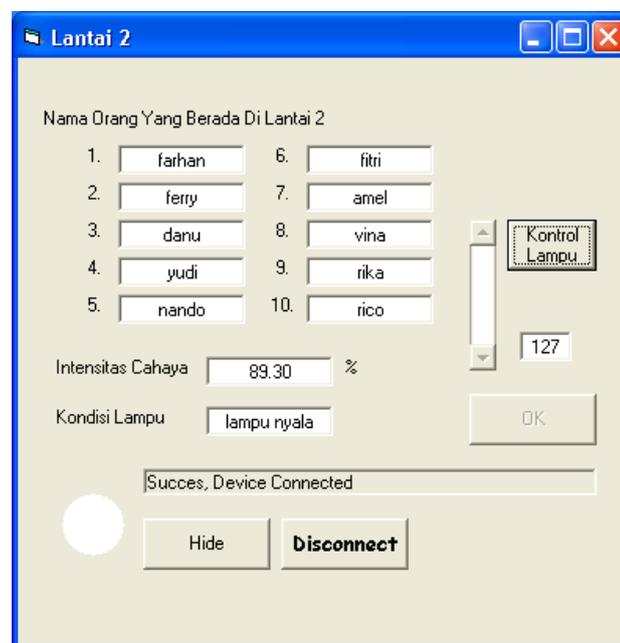


Gambar 4.17 Tampilan pada saat ada permintaan alamat pada lantai 2

Perintah yang diketikkan selanjutnya: *1234f# (lantai 1) atau *2345f# (lantai 2), yang digunakan saat PC ingin meminta data-data kondisi pada lantai tersebut. Data-data tersebut adalah data orang yang berada di lantai tersebut kemudian data kondisi peralatan listrik apakah masih dalam keadaan hidup atau mati serta data intensitas kecerahan pada lantai tersebut.

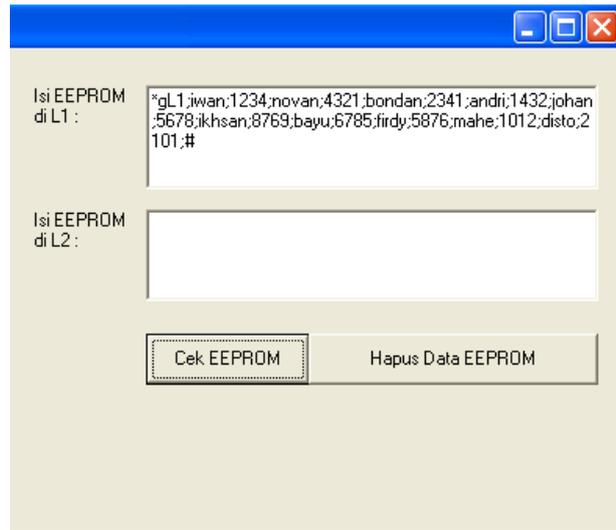


Gambar 4.18 Tampilan pada saat ada permintaan data pada lantai 1

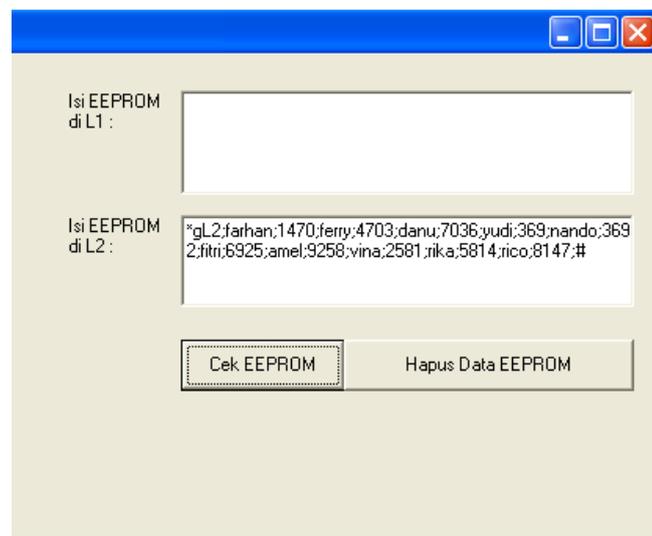


Gambar 4.19 Tampilan pada saat ada permintaan data pada lantai 2

Perintah yang diketikkan selanjutnya: *1234g# (lantai 1) atau *2345g# (lantai 2), yang digunakan saat PC ingin meminta data nama & password yang telah terigristrasi dan tersimpan di EEPROM.

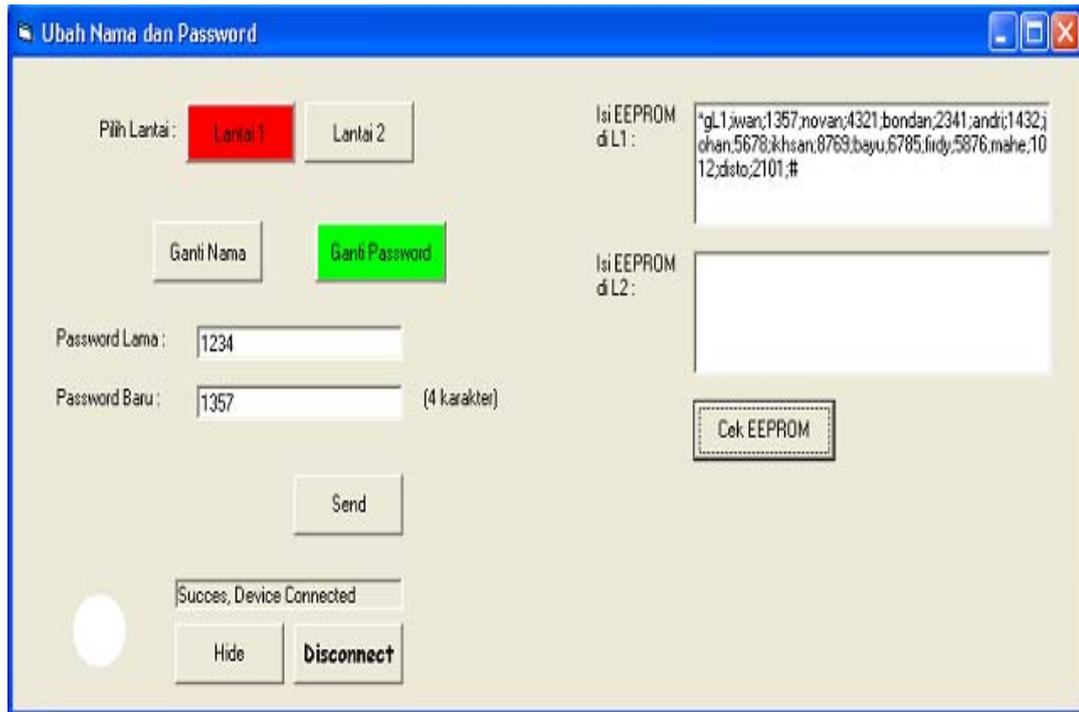


Gambar 4.20 Tampilan pada saat ada permintaan data nama & password pada lantai 1

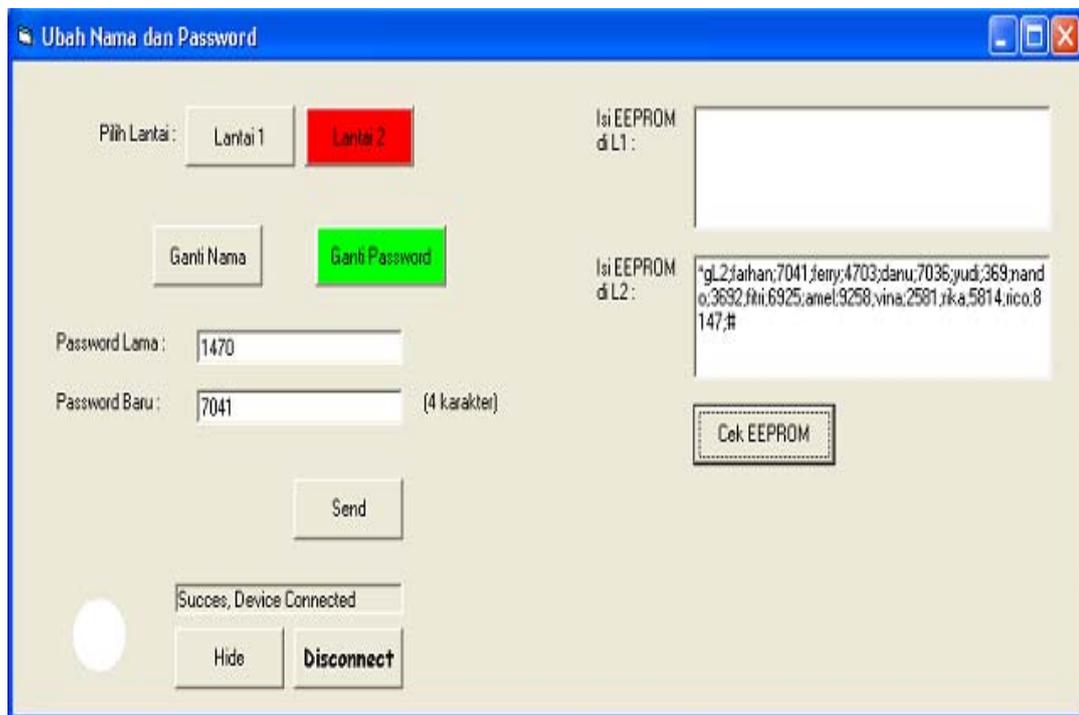


Gambar 4.21 Tampilan pada saat ada permintaan data nama & password pada lantai 2

Perintah yang diketikkan selanjutnya: *1234h# (lantai 1) atau *2345h# (lantai 2), yang digunakan saat PC ingin mengganti password lama yang tersimpan di EEPROM dengan password baru.



Gambar 4.22 Tampilan pada saat ada penggantian password pada lantai 1



Gambar 4.23 Tampilan pada saat ada penggantian password pada lantai 2

Perintah yang diketikkan selanjutnya: *1234j# (lantai 1) atau *2345j# (lantai 2), yang digunakan saat PC ingin melakukan registrasi nama & password sekaligus menyimpannya di EEPROM.

The screenshot shows the 'Registrasi' application window. The 'Pilih Lantai' section has 'Lantai 1' selected. The 'Masukkan Nama' field is empty with '(Max 10 Karakter)' next to it. The 'Masukkan Password' field is empty with '(4 Karakter)' next to it. The 'Isi EEPROM di L1' field contains the text: 'gL1;iwan;1234;novan;4321;bondan;2341;andii;1432;johan;5678;khsan;8769;bayu;6785;firdy;5676;mahe;1012;disto;2101;#'. The 'Isi EEPROM di L2' field is empty. There are buttons for 'Send', 'Cek EEPROM', and 'Hapus Data EEPROM'. A status bar at the bottom shows 'Sukses, Device Connected' with 'Hide' and 'Disconnect' buttons.

Gambar 4.24 Tampilan pada saat melakukan registrasi nama & password pada lantai 1

The screenshot shows the 'Registrasi' application window. The 'Pilih Lantai' section has 'Lantai 2' selected. The 'Masukkan Nama' field is empty with '(Max 10 Karakter)' next to it. The 'Masukkan Password' field is empty with '(4 Karakter)' next to it. The 'Isi EEPROM di L1' field is empty. The 'Isi EEPROM di L2' field contains the text: 'gL2;farhan;1470;reza;3074;Danu;7036;Yudi;369;nando;3692;fitri;6925;amel;9258;vina;2581;nika;5814;rico;8147;#'. There are buttons for 'Send', 'Cek EEPROM', and 'Hapus Data EEPROM'. A status bar at the bottom shows 'Sukses, Device Connected' with 'Hide' and 'Disconnect' buttons.

Gambar 4.25 Tampilan pada saat melakukan registrasi nama & password pada lantai 2

Perintah yang diketikkan selanjutnya: *1234k# (lantai 1) atau *2345k# (lantai 2), yang digunakan saat PC ingin mengganti nama lama yang tersimpan di EEPROM dengan nama baru.

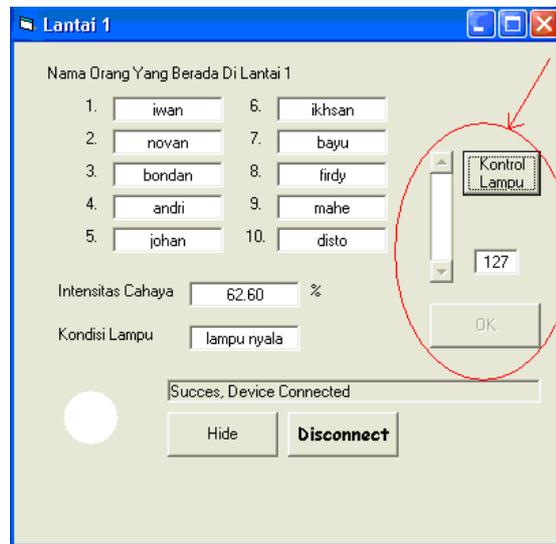


Gambar 4.26 Tampilan pada saat ada penggantian nama pada lantai 1

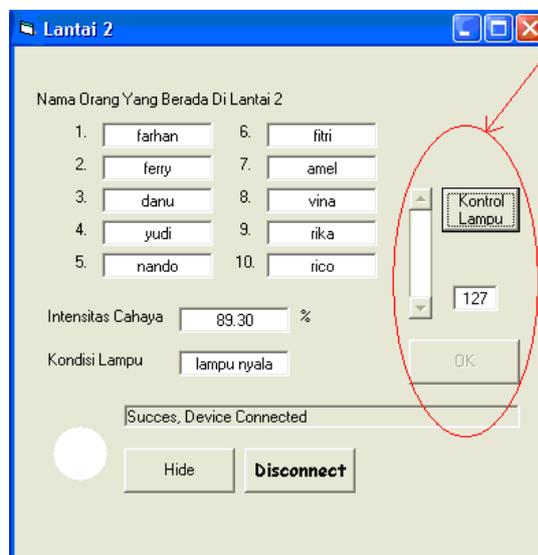


Gambar 4.27 Tampilan pada saat ada penggantian nama pada lantai 2

Perintah yang diketikkan selanjutnya: *1234I# (lantai 1) atau *2345I# (lantai 2), yang digunakan saat PC ingin melakukan pengontrolan lampu AC 200 volt.



Gambar 4.28 Tampilan pada saat melakukan pengontrolan lampu AC 220 V pada lantai 1



Gambar 4.29 Tampilan pada saat melakukan pengontrolan lampu AC 220 V pada lantai 2

4.6 Pengujian Panjang Kabel Komunikasi Serial

Pada pengujian kali ini dilakukan pengujian terhadap panjang kabel yang memiliki jarak 200 meter. Adapun pengujian ini dilakukan secara berulang-ulang. Setelah dilakukan pengujian baik itu pada waktu transmit maupun receive pada jarak 200 meter masih dapat berjalan dengan baik. RS-485 dapat mengakomodir pentransmisiian dengan jarak yang jauh karena fungsinya sebagai repeater.

Adapun pengetesan dilakukan dengan menggunakan kabel UTP dengan konfigurasi straight. Dimana pada konfigurasi ini memiliki konfigurasi ujung yang satu dengan ujung yang lainnya sama.

Adapun jarak dari komunikasi ini bisa jauh dikarenakan sistem ini tidak rentan terhadap noise. Hal itu dikarena prinsip pembacaan logic high dan logic lownya berdasarkan beda tegangan antara kabel 1 dan kabel 2 pada sistem komunikasi ini.

BAB 5

KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang diperoleh penulis setelah melakukan penelitian tugas akhir serta saran-saran untuk perbaikan sistem dan hasil yang lebih baik lagi dimasa yang akan datang.

5.1 Kesimpulan

Setelah menyelesaikan perancangan sistem serta melakukan pengujian terhadap sistem tersebut, maka penulis dapat menarik suatu kesimpulan bahwa:

1. Hasil pengujian terhadap sistem keamanan, pendeteksian peralatan listrik dengan sensor arus (CT) serta pengendalian lampu AC dengan OPT101 berjalan dengan baik.
2. Sistem melakukan pemantauan keadaan pada sistem tersebut secara terus menerus, dan baru akan mengirimkan hasil pemantauan tersebut jika ada permintaan dari PC (dummy).
3. Banyaknya data nama dan password yang dapat dimasukkan ke dalam sistem keamanan bergantung dari kapasitas eeprom microcontroller yang digunakan.
4. Pengecekan saklar manual (push button) memiliki prioritas yang lebih tinggi bila dibandingkan dengan pembacaan data serial.
5. Sistem pendeteksian peralatan listrik dapat mengetahui kondisi setiap peralatan listrik yang digunakan pada bangunan tersebut.

6. Sistem menggunakan interrupt serial, sehingga pembacaan data serial yang dikirim dari PC baru dilakukan pada saat data tersebut diterima oleh *microcontroller*.
7. Pada sistem pendeteksian peralatan listrik dengan sensor arus (CT) output keluarannya ditentukan kondisi high dan lownya keluaran IC LM311.
8. Setiap lantai menggunakan dua buah *microcontroller*, yang pertama berlaku sebagai *master* dan yang kedua berlaku sebagai *slave*.
9. Pada pengujian panjang kabel 200 meter dengan kabel UTP pentransmision data masih dapat berjalan dengan baik.

5.2 Saran

Berikut ini adalah saran dari penulis agar dimasa yang akan datang sistem ini dapat dibuat jauh lebih baik lagi :

- ✚ Sistem ini dibangun dengan dua buah *microcontroller* pada setiap lantainya, diharapkan adanya sistem yang lebih baik lagi dengan meminimalisasi jumlah *microcontroller* yang digunakan.
- ✚ Pada sistem pendeteksian peralatan listrik masih menggunakan sensor analog yang keluarannya masih harus ditambahkan dengan rangkaian pengkondisi sinyal, diharapkan nantinya untuk kedepannya menggunakan smart sensor sehingga keluarannya bisa langsung dihubungkan ke *microcontroller*.

DAFTAR PUSTAKA

- [1] Atmel, 2007, *8-Bit AVR[®] Microcontroller with 8K byte in-system programmable flash AT Mega32*, Atmel.inc.(<http://www.atmel.com>), 12 Januari 2008, pk.15.30.
- [2] Jan Axelson, 21 agustus 1995, *Networks for Monitoring and Control Using an RS-485 Interface*, Microcomputer journal.
- [3] Frank D.Petruzella, 1996, *Elektronik industri*, Terj.dari *Industrial electronic*, oleh Sumanto, Penerbit Andi, Yogyakarta.
- [4] Fredrick W. Hughes, 1981, *Panduan Op-amp*, Terj.dari *Op-Amp Handbook*, oleh Ignatius Hartono, PT. Elek Media Komputindo.
- [5] Mazidi, Muhammad Ali, *The 8086 IBM PC & Compatible Computers*, Prentice Hall, New York, 1995.
- [6] *Data Sheet Maxim RS-485/RS-422 Transceiver*
<http://www.alldatasheet.com>, 3 Maret 2008, pk.13.00.

LAMPIRAN

LAMPIRAN

Tabel 4.1 Pengujian sistem keamanan lantai 1

Nama	Password	Buka Pintu	Simpan / Baca EEPROM	Hapus EEPROM
Iwan	1234	Berhasil	Berhasil	Berhasil
Novan	4321	Berhasil	Berhasil	Berhasil
Bondan	2341	Berhasil	Berhasil	Berhasil
Andri	1432	Berhasil	Berhasil	Berhasil
Johan	5678	Berhasil	Berhasil	Berhasil
Ikhsan	8769	Berhasil	Berhasil	Berhasil
Bayu	6785	Berhasil	Berhasil	Berhasil
Firdy	5876	Berhasil	Berhasil	Berhasil
Mahe	1012	Berhasil	Berhasil	Berhasil
Disto	2101	Berhasil	Berhasil	Berhasil

Tabel 4.2 Pengujian sistem keamanan lantai 2

Nama	Password	Buka Pintu	Simpan / Baca EEPROM	Hapus EEPROM
Farhan	1470	Berhasil	Berhasil	Berhasil
Ferry	4703	Berhasil	Berhasil	Berhasil
Danu	7036	Berhasil	Berhasil	Berhasil
Yudi	0369	Berhasil	Berhasil	Berhasil
Nando	3692	Berhasil	Berhasil	Berhasil
Fitri	6925	Berhasil	Berhasil	Berhasil
Amel	9258	Berhasil	Berhasil	Berhasil
Vina	2581	Berhasil	Berhasil	Berhasil
Rika	5814	Berhasil	Berhasil	Berhasil
Rico	8147	Berhasil	Berhasil	Berhasil

Tabel 4.3 Pengujian perubahan data lantai 1

Nama	Password	Nama	Password	Simpan / Hapus
-------------	-----------------	-------------	-----------------	-----------------------

Sebelum	Sebelum	Sesudah	Sesudah	EEPROM
Iwan	1234	Eksa	1357	Berhasil
Novan	4321	Deni	7531	Berhasil
Bondan	2341	Fitra	3571	Berhasil
Andri	1432	Waras	1753	Berhasil
Johan	5678	Rusma	9135	Berhasil
Ikhsan	8769	Dadi	5319	Berhasil
Bayu	6785	Doni	7913	Berhasil
Firdy	5876	Desi	3197	Berhasil
Mahe	1012	Dodo	1035	Berhasil
Disto	2101	Ocha	5301	Berhasil

Tabel 4.4 Pengujian perubahan data lantai 2

Nama	Password	Nama	Password	Simpan / Hapus
------	----------	------	----------	----------------

Sebelum	Sebelum	Sesudah	Sesudah	EEPROM
Farhan	1470	Frengky	0741	Berhasil
Ferry	4703	Reza	3074	Berhasil
Danu	7036	Fandi	6307	Berhasil
Yudi	0369	Hary	9630	Berhasil
Nando	3692	Adi	2963	Berhasil
Fitri	6925	Visi	5296	Berhasil
Amel	9258	Tessa	8529	Berhasil
Vina	2581	Wenny	1852	Berhasil
Rika	5814	Lia	4185	Berhasil
Rico	8147	Ariah	7418	Berhasil

Tabel 4.5 Pengukuran tegangan output rangkaian sensor arus (CT) pada lantai 1

Kondisi	Tegangan Titik A (V)	Tegangan Titik B	Tegangan Titik C (V)
---------	----------------------	------------------	----------------------

Saklar PLN		(V)	
ON	5.350	3.124	4.790
OFF	0.298	3.133	0.155
ON	5.254	3.122	4.680
OFF	0.304	3.107	0.154
ON	5.226	3.133	4.530
OFF	0.312	3.123	0.162
ON	5.351	3.105	4.480
OFF	0.310	3.120	0.145
ON	5.252	3.109	4.420
OFF	0.324	3.133	0.148
ON	5.325	3.124	4.780
OFF	0.289	3.133	0.135
ON	5.135	3.114	4.778
OFF	0.292	3.133	0.164
ON	5.349	3.128	4.870
OFF	0.278	3.132	0.152
ON	5.335	3.187	4.789
OFF	0.304	3.133	0.165

Tabel 4.6 Pengukuran tegangan output rangkaian sensor arus (CT) pada lantai 2

Kondisi Saklar PLN	Tegangan Titik A (V)	Tegangan Titik B (V)	Tegangan Titik C (V)
ON	5.351	3.224	4.690
OFF	0.398	3.135	0.178
ON	5.354	3.122	4.586
OFF	0.324	3.127	0.153
ON	5.426	3.143	4.630
OFF	0.312	3.123	0.172
ON	5.251	3.135	4.580
OFF	0.316	3.220	0.165
ON	5.352	3.119	4.620
OFF	0.344	3.133	0.147
ON	5.315	3.134	4.488
OFF	0.326	3.132	0.130
ON	5.235	3.114	4.778
OFF	0.382	3.143	0.194
ON	5.249	3.228	4.570
OFF	0.265	3.130	0.142
ON	5.335	3.213	4.689
OFF	0.298	3.153	0.185

Tabel 4.7 Pengambilan data pendeteksian peralatan listrik lantai 1 oleh PC

Data (0 – 255)	Tegangan (AC)	Intensitas Cahaya	Luxmeter (Sc.2000)
0	0,60	5,3	36,8
5	0,90	8,6	38,2
10	1,24	10,3	40
15	5	11,7	41,6
20	6,1	14,5	42
25	7,3	16,4	43,1
30	10,5	18,5	44,9
35	11,8	19,1	46,7
40	13,6	19,7	47,5
45	15,5	20,3	48,7
50	16,8	21,9	49,8
55	19	23,7	50,2
60	20,6	24,1	51,4
65	22	25,6	52,1
70	23,1	26,2	53,3
75	25,4	26,7	54
80	26,8	27,1	55,9
85	27,3	27,4	56,8
90	29,6	28,1	57,5
95	31,8	28,5	58,1
100	32,8	28,7	59
105	33,8	29,2	59,4
110	36,4	29,7	60,2
115	38,3	31,1	61,4
120	39,7	31,7	62,3
125	89,5	32,2	64,3

130	90,5	32,6	65,4
135	100	33,8	66,8
140	102	35,7	67,1
145	109,5	36,4	69
150	110	38,9	69,4
155	120	40,7	70,4
160	123,6	41,7	71,3
165	125	42,6	72
170	130	43,4	72,5
175	136	47,4	73,1
180	141	49,2	74
185	141,4	50,6	74,7
190	147	51,8	75,4
195	155,2	52,6	76
200	157	54,3	76,3
205	158	56,7	77
210	161,3	58,1	77,2
215	168,2	59,70	78
220	169	60,8	78,8
225	171	62,4	79
230	172,7	63,2	79,1
235	182,5	64,3	80,9
240	185	65,5	82,1
245	188,3	66,1	82,9
250	192	68,2	83,2
254	192,2	69,7	84,3
255	0,6	5,4	36,2

Tabel 4.10 Data dan tegangan pada pengaturan lampu AC pada lantai 2

Data (0 – 255)	Tegangan (AC)	Intensitas Cahaya	Luxmeter (Sc.2000)
0	0,56	5,4	36,7
5	0,91	8,7	38,3
10	1,23	10,4	40
15	5	11,8	41,5
20	6,2	14,6	42,1
25	7,2	16,5	43,2
30	10,6	18,6	44,6
35	11,2	19,2	46,8
40	13,8	19,8	47,6
45	15,4	20,4	48,8
50	16,9	21,8	49,9
55	19,1	23,8	50,3
60	20,5	24,2	51,5
65	22	25,5	52,5
70	23,2	26,3	53,6
75	25,1	26,8	54
80	25,7	27,2	55,8
85	27,4	27,6	56,9
90	29,2	28,2	57,6
95	30,6	28,6	58,2
100	32,7	28,8	59
105	33,1	29,4	59,5
110	36,3	29,9	60,3
115	38,2	31,5	61,5
120	38,5	31,9	62,6
125	88,4	32,1	64,4

130	90,5	32,7	65,4
135	100	33,9	66,8
140	102	35,8	67,2
145	108,4	36,6	69
150	111	38,7	69,5
155	121	40,4	70,5
160	123,7	41,8	71,4
165	126	42,9	72
170	131	43,5	72,6
175	135	47,5	73,2
180	140	49,3	74
185	141,3	50,8	74,8
190	148	51,9	75,5
195	155	52,7	76
200	157,2	54,2	76,4
205	158,1	56,8	77
210	161,2	58,3	77,3
215	168,3	59,7	78
220	168	60,9	78,7
225	173	62,3	79
230	173,7	63,6	79,3
235	181,5	64,5	80,8
240	186	65,6	82,3
245	189,4	66,2	82,7
250	192,2	68,7	83,3
254	192,7	69,2	84,5
255	0,67	5,5	36,4

Tabel 4.11 Pengujian komunikasi serial menggunakan RS-232.

Karakter yang dikirim PC (diketikkan melalui keyboard)	Data yang dikirim <i>microcontroller</i> (Tampilan pada hyperterminal)	Keterangan
		(Keberhasilan pengiriman)
'a'	A	berhasil
'b'	B	berhasil
'c'	C	berhasil
'd'	D	berhasil
'e'	E	berhasil
'f'	F	berhasil
'g'	G	berhasil
'h'	H	berhasil
'i'	I	berhasil
'j'	J	berhasil
'k'	K	berhasil
'l'	L	berhasil
'm'	M	berhasil
'n'	N	berhasil
'o'	O	berhasil
'p'	P	berhasil
'q'	Q	berhasil
'r'	R	Berhasil
's'	S	Berhasil
't'	T	Berhasil
"u"	U	Berhasil
"v"	V	Berhasil
"w"	W	Berhasil
"x"	X	Berhasil
"y"	Y	Berhasil
"z"	Z	Berhasil
"0"	0	Berhasil
"1"	1	Berhasil
"2"	2	Berhasil
"3"	3	Berhasil

Tabel 4.12 Pengujian komunikasi serial menggunakan RS-485 to RS-232

Karakter yang dikirim PC (diketikkan melalui keyboard)	Data yang dikirim <i>microcontroller</i> (Tampilan pada hyperterminal)	Keterangan
		(Keberhasilan pengiriman)
'a'	A	berhasil
'b'	B	berhasil
'c'	C	berhasil
'd'	D	berhasil
'e'	E	berhasil
'f'	F	berhasil
'g'	G	Berhasil
'h'	H	Berhasil
'i'	I	Berhasil
'j'	J	Berhasil
'k'	K	Berhasil
'l'	L	Berhasil
'm'	M	Berhasil
'n'	N	Berhasil
'o'	O	Berhasil
'p'	P	Berhasil
'q'	Q	Berhasil
'r'	R	Berhasil
's'	S	Berhasil
't'	T	Berhasil
"u"	U	Berhasil
"v"	V	Berhasil
"w"	W	Berhasil
"x"	X	Berhasil
"y"	Y	Berhasil
"z"	Z	Berhasil
"0"	0	Berhasil
"1"	1	Berhasil
"2"	2	Berhasil
"3"	3	Berhasil