PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

TUGAS AKHIR

SIMULASI SISTEM ANTRIAN LOKET BERBASIS MIKROKONTROLER ATMEGA8535 DAN ATMEGA8

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Program Studi Teknik Elektro

Oleh:

CHANDRA CHRISMARANTIKA
NIM: 075114010

PROGRAM STUDI JURUSAN TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

FINAL PROJECT

THE SIMULATION OF QUEUE COUNTER SYSTEM BASED ON MICROCONTROLLERS ATMEGA8535 AND ATMEGA8

Presented as Partial Fullfillment of Requirements

To Obtain the SarjanaTeknik Degree

In Electrical Engineering Study Program

CHANDRA CHRISMARANTIKA

NIM: 075114010

ELECTRICAL ENGINEERING STUDY PROGRAM

FACULTY OF SCIENCE AND TECHNOLOGY

SANATA DHARMA UNIVERSITY

YOGYAKARTA

2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PERSETUJUAN

TUGAS AKHIR

SIMULASI SISTEM ANTRIAN LOKET BERBASIS *MIKROKONTROLER ATMEGA8535 DAN MIKROKONTROLER ATMEGA8*

Oleh:

CHANDRA CHRISMARANTIKA
NIM: 075114010

telah disetujui oleh:

Pembimbing

Martanto, S.T., M.T.

Tanggal: 12 juli 2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PENGESAHAN

TUGAS AKHIR

SIMULASI SISTEM ANTRIAN LOKET BERBASIS MIKROKONTROLER ATMEGA8535 DAN ATMEGA8

Oleh:

Chandra Chrismarantika
NIM : 075114010

Telah dipertahankan di depan Panitia Penguji
Pada tanggal 29 Mei 2013
Dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap		Tanda Tangan
Ketua	: Wiwien Widyastuti, S.T., M.T.	
Sekretaris	: Martanto, S.T., M.T.	
Anggota	: Ir. Tjendro, M.Kom.	

Yogyakarta, 16 Juli 2013
Fakultas Sains dan Teknologi
Universitas Sanata Dharma
Dekan,

Paulina Heruningsih Prima Rosa, S.Si., M.Sc.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa tugas akhir ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, 12 Juli 2013

Chandra Chrismarantika

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PERSEMBAHAN DAN MOTTO HIDUP

MOTTO:

Talks to do more...

Dengan ini kupersembahkan karyaku ini untuk.....
Yesus Kristus Pembimbing Hidupku ,
Keluargaku tercinta,
Teman-teman seperjuanganku angkatan 2007,

Terima Kasih untuk semuanya.....

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma :

Nama : Chandra Chrismarantika

Nomor Mahasiswa : 075114010

Demi pengembangan ilmu pengetahuan, saya memberikan kepada Perpustakaan Universitas Sanata Dharma karya ilmiah saya yang berjudul :

SIMULASI SISTEM ANTRIAN LOKET BERBASIS ATMEGA8535 DAN ATMEGA8

beserta perangkat yang diperlukan (bila ada). Dengan demikian saya memberikan kepada kepada Perpustakaan Universitas Sanata Dharma hak untuk menyimpannya, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas, dan mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalty kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini saya buat dengan sebenarnya.

Yogyakarta, 12 Juli 2013

Chandra Chrismarantika

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

INTISARI

Di Indonesia perkembangan teknologi sudah sangat maju. Mayoritas sistem industri dan informasi banyak menggunakan perangkat control untuk mengendalikan berbagai peralatan agar mempermudah pekerjaan manusia. Peningkatan kebutuhan layanan antrian dinilai sangat kurang adil karena secara umum orang ingin segera dilayani dan tidak tertib dalam hal mengantri yang akhirnya saling menyerobot agar ingin cepat dilayani, sehingga petugas loket antrian kesulitan dalam melayani dalam antrian. Loket antrian biasanya dapat kita jumpai di bank,PLN,rumah sakit dan tempat layanan umum lainnya.

Simulasi sistem antrian loket berbasis mikrokontroler pada tugas akhir ini menghasilkan suatu alat yang dapat mengatur atau mengontrol pengambilan dan pemanggilan nomor antrian pada antrian loket. Sistem ini berbasis mikrokontroler Atmega8535 sebagai *master control*, mikrokontroler Atmega8 sebagai *slave control*/loket antrian dan *seven segmen* untuk mengetahui cacahan antrian dan pencatatan nomor antrian. Jika tombol antrian ditekan maka *master control* akan mulai mencacah dan akan menampilkan *seven segmen* nomor antrian pada loket antrian/*slave control*, jika pada *slave control* sudah tidak ada antrian/*user*, maka *slave control* akan mengirim data panggilan ke *master control*. Setelah *master control* menerima data panggil dari *slave control*, maka *master control* akan mengirim data nomor antrian berikutnya ke *slave control*.

Simulasi sistem antrian loket berbasis mikrokontroler Atmega8535 dan Atmega8 yang secara keseluruhan sudah dapat bekerja. Penampil menaikkan cacahan antrian dan melakukan pencatatan nomor antrian pada rangkaian Atmega8535, rangkaian Atmega8, rangkaian LCD, rangkaian *push button*, rangkaian *buzzer* ternyata mampu menampilkan data-data sesuai dengan perancangan. Simulasi sistem antrian loket pada data cacahan antrian dan pencacatan nomor antrian pada sistem keseluruhan memliki tingkat akurasi sempurna.

Kata kunci: Simulasi Sistem, Antrian Loket, Mikrokontroler Atmega8535 dan Atmega8.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

ABSTRACT

In Indonesia, the development of technology is very advanced. The majority of industrial systems and a lot of information using control devices to control a variety of equipment in order to facilitate the work of man. Increased need for services considered to be less fair queuing because generally people want to be served immediately and disorderly in the queue finally grab each other in order to want to be served quickly, so the attendant difficulties in serving counter queue in the queue. Counter queue can usually be encountered in the bank, PLN, hospitals and other public services.

Simulation of queuing systems on a microcontroller based counter this thesis produce a tool that can manage or control the retrieval and dialing the queue at the counter queue. The system is based on a master control microcontroller ATmega8535, ATmega8 microcontroller as slave control / counter queue and seven segments to know the registration number of counts queue and queue. If the button is pressed, the master control queue will start chopping and seven segment displays the queue number in counter queue / slave control, if the slave has no control queue / user, then the slave call control will send data to the master control. After the master control receives data from the slave call control, the master control will send data to the slave number next queue control.

Simulation queuing system counters based microcontroller ATmega8535 and ATmega8 overall should be working. Raise chopped queue viewer and make record number of queues on the circuit ATmega8535, ATmega8 series, LCD series, a series of push button, buzzer circuit was able to display the data in accordance with the design. Simulation of queuing systems on a data counter queues and mutilations number of counts in the queue exists at the level of accuracy of the overall system is perfect.

keyword: Simulation System, Queue Counters, Microcontroller ATmega8535 and ATmega8.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yesus Kristus karena telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan laporan akhir ini dengan baik. Laporan akhir ini disusun untuk memenuhi syarat memperoleh gelar sarjana.

Penulis mengucapkan terima kasih kepada :

1. Kedua orangtuaku yaitu : Eddy Christianto dan Emy Winarni, kakakku atas dukungan, doa, cinta, perhatian dan motivasi yang tiada henti.
2. Paulina Heruningsih Prima Rosa, M.Sc., Dekan Fakultas Sains dan Teknologi Universitas Sanata Dharma Yogyakarta.
3. Petrus Setyo Prabowo, S.T., M.T., Ketua Program Studi Teknik Elektro Universitas Sanata Dharma Yogyakarta.
4. Wiwien Widyastuti, S.T., M.T., dosen pembimbing akademik yang telah mendampingi dan membimbing penulis selama studi.
5. Martanto, S.T., M.T., dosen pembimbing yang dengan penuh kesabaran dan ketulusan hati memberi bimbingan, kritik, saran, serta motivasi dalam penulisan skripsi ini.
6. Wiwien Widyastuti, S.T., M.T.. dan Ir.Tjendro., dosen penguji yang telah memberikan masukan, bimbingan, saran dalam merevisi skripsi ini.
7. Bapak/ Ibu dosen yang telah mengajarkan banyak hal selama penulis menempuh pendidikan di Program Studi Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Sanata Dharma.
8. Rekan-rekan seperjuanganku angkatan 2007 Teknik Elektro, teman-teman Basecamp, dan teman-teman Universitas Sanata Dharma yang memberikan dukungan dalam penyelesaian skripsi ini.
9. Semua pihak yang tidak dapat disebutkan satu persatu atas semua dukungan yang telah diberikan dalam penyelesaian skripsi ini.
10. Seseorang yang telah memberikan semangat kuliah dan menyemangatkan kembali arti dari kehidupan.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini masih mengalami kesulitan dan tidak lepas dari kesalahan. Oleh karena itu, penulis mengharapkan masukan,

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

kritik dan saran yang membangun agar skripsi ini menjadi lebih baik. Dan semoga skripsi ini dapat bermanfaat sebagaimana mestinya.

Yogyakarta, 12 Juli 2013

Penulis

Chandra Chrismarantika

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PERSETUJUAN iii
DAFTAR ISI xii
DAFTAR GAMBAR xiv
DAFTAR TABEL xvi
BAB I PENDAHULUAN
1.1. Latar Belakang 1
1.2. Tujuan dan Manfaat 2
1.3. Batasan Masalah 2
1.4. Metodologi Penelitian 2
BAB II DASAR TEORI
2.1. Mikrokontroler AVR 4
2.1.1. Konfigurasi ATmega8535 4
2.1.2. *Reset* dan Osilator Eksternal 7
2.1.3. *Timer/Counter*0 8
2.1.3.1. Register Pengendali *Timer*0 8
2.1.4 Konfigurasi ATmega8 11
2.2. LED*(Light Emitting Dioda)* 14
2.3 LCD*(Liquid cristal Display)* 16
2.3.1. Konfigurasi *LCD* 16X2 16
2.4. Saklar *Push Button* 17
2.5. *Buzzer* 17
2.6. *Seven Segment* 18
2.6.1 Scanning *Seven Segment* 19
2.7. Transistor sebagai Saklar 21
2.8. Regulator tegangan 22
2.8.1 IC regulator tiga terminal 23

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB III PERANCANGAN

3.1. Diagram Blok Rangkaian 27
3.2. Perancangan Perangkat Keras 28
3.2.1. Perancangan Rangkaian *Mikrokontroler* Atmega8535 29
3.2.2. Perancangan Rangkaian *Mikrokontroler* Atmega8.. 29
3.2.3. Perancangan Rangkaian *Display LCD 16X2* 30
3.2.4. Perancangan Rangkaian *Display Seven Segmen Master Control* 31
3.2.5. Perancangan Rangkaian *Display Seven Segmen Slave Control* 32
3.2.6. Perancangan Rangkaian *Push Button* 33
3.2.7. Perancangan Rangkaian *Buzzer* 33
3.2.8. Perancangan Rangkaian Catu Daya 37
3.3. Perancangan Perangkat Lunak 38
3.3.1. *Flowchart Master Control* Atmega8535 38
3.3.2. *Flowchart Slave Control* Atmega8 40

BAB IV HASIL DAN PEMBAHASAN

4.1. Implementasi Alat 41
4.2. Proses Pengambilan data 42
4.3. Pengujian Sistem Keseluruhan 43
4.4. Pengujian Subsistem 59
4.4.1. Rangkaian Catu daya. 59
4.4.2. Pengujian *Buzzer*. 60
4.4.3. Pengujian Push Button dan LCD. 61
4.4.4. Pengujian Display 7segmen untuk *Slave Control* 62
4.4.5. Pengujian Display 7segmen untuk *Master Control* 64

BAB V KESIMPULAN

5.1. Kesimpulan 66
5.2. Saran 66
DAFTAR PUSTAKA 67

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR GAMBAR

Gambar 1.1. Blok model perancangan 3
Gambar 2.1. Konfigurasi *pin* ATmega8535 5
Gambar 2.2. Rangkaian *reset* 7
Gambar 2.3. Rangkaian Osilator Kristal 8
Gambar 2.4. Konfigurasi *pin* ATMega8 12
Gambar 2.5. Simbol dan bentuk LED 15
Gambar 2.6. Gambar rangkaian LED sederhana 15
Gambar 2.7. LCD 16X2 16
Gambar 2.8. Konfigurasi *pin* LCD 16
Gambar 2.9. *Push Button* 17
Gambar 2.10. Simbol *Buzzer* 18
Gambar 2.11. *Display seven segment* 18
Gambar 2.12. Rangkaian internal *common annoda* 19
Gambar 2.13. Rangkaian internal *common* katoda 19
Gambar 2.14. Transistor sebagai saklar 21
Gambar 2.15. Rangkaian Regulator Tegangan 21
Gambar 2.16. Bentuk IC Regulator dan Simbol rangkaian 24
Gambar 2.17. Regulator *Zener* 24
Gambar 2.18. Regulator *Zener Follower* 25
Gambar 3.1. Diagram blok perancangan 28
Gambar 3.2. Rangkaian mikrokontroler ATmega8535 29
Gambar 3.3. Rangkaian mikrokontroler ATmega8 30
Gambar 3.4. Rangkaian LCD 16x2 31
Gambar 3.5. Rangkaian *7segment Master Control* 32
Gambar 3.6. Rangkaian *7segment Slave Control* 33
Gambar 3.7. Rangkaian *push button* 33
Gambar 3.8. Rangkaian *buzzer* 35
Gambar 3.9. Rangkaian catu daya 37
Gambar 3.10. *Flow chart Master control* 38
Gambar 3.11. *Flow chart Slave control* 40

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.1. Bentuk alat antrian loket tampak dari depan 41
Gambar 4.2. Multimeter *digital* 43
Gambar 4.3. Catu daya 59
Gambar 4.4. Rangkaian *buzzer* 60
Gambar 4.5. Gambar *push button* dan *LCD* 61
Gambar 4.6. *Display* 7*segment* untuk *Slave Control* 63
Gambar 4.7. *Display* 7*segmen* untuk *master control* 64

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR TABEL

Tabel 2.1. Fungsi khusus *port* B 6
Tabel 2.2. Fungsi khusus *port* C 6
Tabel 2.3. Fungsi khusus *port* D 7
Tabel 2.4. Mode Operasi Kristal Osilator 8
Tabel 2.5. Register TCCR0 9
Tabel 2.6. *Prescaler timer/counter*0 9
Tabel 2.7. Mode operasi 9
Tabel 2.8. Mode Normal dan CTC 10
Tabel 2.9. Register TCNT0 10
Tabel 2.10. Register OCR0 10
Tabel 2.11. Register TIMSK 10
Tabel 2.12. Register TIFR 10
Tabel 2.13. Komunikasi asinkron data *master control* 20
Tabel 2.14. Karakteristik regulator IC 78XX 26
Tabel 3.1. Fungsi *pin* pada ATmega8535 28
Tabel 3.3. Fungsi *pin* pada ATmega8 30
Tabel 3.4. Fungsi konektor pada LCD ke ATmega8535 31
Tabel 4.1. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *slave control* 44
Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* 48
Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap* 53
Tabel 4.4. *Output* tegangan dari rangkaian regulator 5 volt 60
Tabel 4.5. *Output* tegangan dari rangkaian *buzzer* 60
Tabel 4.6. Cacahan antrian untuk *Push Button* dan LCD 62
Tabel 4.7. Cacahan antrian display 7segmen untuk *Slave Control* 63
Tabel 4.8. *Display* Cacahan antrian untuk *master control* 65

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada masa kini teknologi mengalami perkembangan sangat maju. Banyak sistem industri dan informasi menggunakan perangkat kontrol. Perangkat kontrol semakin banyak dibutuhkan untuk mengendalikan berbagai peralatan yang digunakan untuk mempermudah pekerjaan manusia. Pada layanan umum banyak orang yang harus dilayani dan kalau orang yang melayani kurang, biasanya diminta untuk mengantri. Dalam kenyataannya banyak setiap orang yang mengantri tidak tertib, tidak patuh dan saling menyerobot ingin cepat dilayani, maka petugas loket antrian kesulitan dalam menentukan siapa yang harus dilayani dalam antrian. Hal ini tentunya kurang menyenangkan dan cendrung tidak adil. Loket antrian biasa kita jumpai di bank, PLN, rumah sakit dan di tempat layanan umum lainnya.

Berdasarkan hal di atas, penulis ingin membuat suatu sistem otomatis yang lebih praktis dan efisien untuk mempermudah dan mempercepat sistem antrian loket. Beberapa peneliti telah membahas tentang sistem antrian loket dengan melibatkan komputer sebagai pengendali utamanya[2]. Secara umum sistem ini berbasis mikrokontroler, terdapat 1 mikrokontroler utama ATmega8535 dan 3 mikrokontroler ATmega8 sebagai pengatur tampilan 7*segment* dan masukan *pushbutton* sebagai penanda antrian berikutnya. Pada mikrokontroler utama akan terhubung *push button* sebagai tombol antrian dan 7*segment* sebagai penampilan nomor antrian dan loket tersebut. Ketika tombol antrian ditekan, maka nomor antrian akan tertampil di 7*segment*, menandakan nomor antrian tersebut. Orang yang telah menekan tombol antrian dan telah mengetahui nomor antriannya tersebut tinggal menunggu pemanggilan nomor yang telah dimiliki untuk melakukan transaksi pada loket antrian.

Dengan menggunakan sistem ini diharapkan dapat meningkatkan kenyamanan pelayanan, konsumen tidak perlu kelelahan berdiri menunggu dibarisan antrian yang panjang dan mereka dapat dengan bebas melakukan aktivitas sebelum nomor antrian mereka dipanggil atau tertampil.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

1.2 Tujuan dan Manfaat

Tujuan penelitian ini adalah menghasilkan suatu alat yang dapat mengatur atau mengontrol pengambilan dan pemanggilan nomor antrian pada loket antrian.

Manfaat dari penelitian ini adalah untuk memajukan fasilitas pelayanan sistem antrian yang ada.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

a. *Master control* berjumlah 1 menggunakan ATmega8535 mencacah nomor antrian 0-99, mengirim data nomor antrian ke *slave control* dan menerima data panggilan antrian berikutnya loket yang kosong dari masing-masing *slave control*.
b. *Slave control* berjumlah 3 menggunakan ATmega8 mengirim data panggilan antrian berikutnya ke *master control* dan menerima data nomor antrian dari *master control*.
c. Terdapat 3 loket antrian, masing masing terdapat *push button* dan penampil *led* sebagai pengirim data panggilan antrian dan antrian berikutnya.
d. Penampil nomor antrian berupa 7*segment* dan LCD 16x2.
e. Jarak kabel komunikasi *master* dan *slave* ±4 meter.

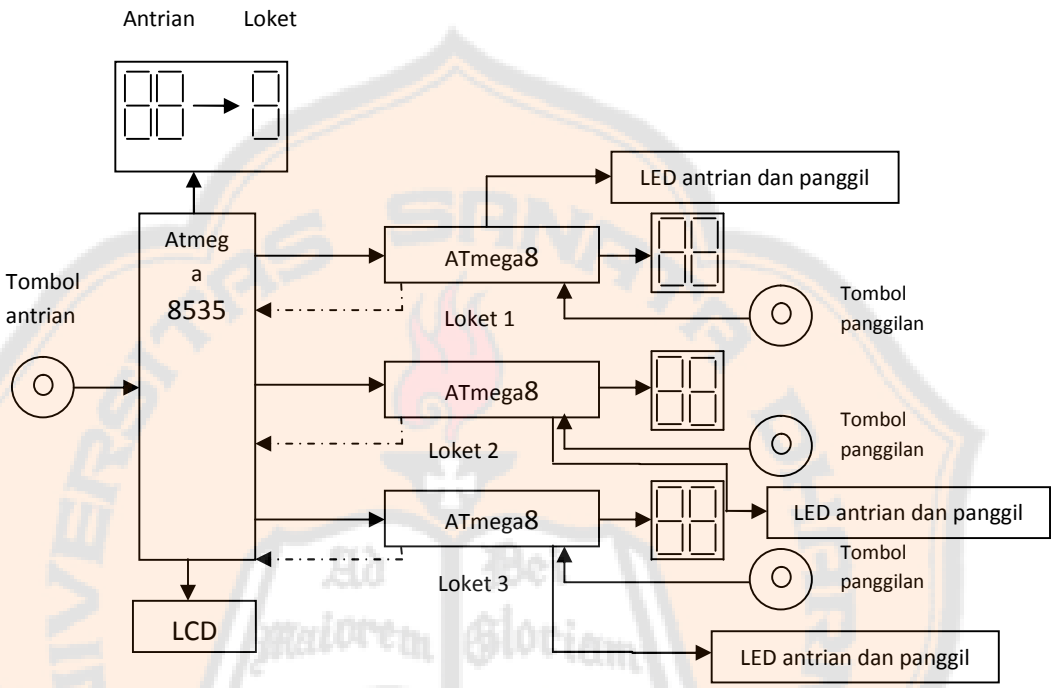
1.4 Metodologi Penelitian

Penulisan ini menggunakan metode:

a. Pengumpulan bahan-bahan refrensi berupa *website* dan buku-buku.
b. Perancangan subsistem berupa *hardware* dan *software*. Pada tahap ini bertujuan mencari bentuk model yang optimal dari sistem yang akan dibuat dengan mempertimbangkan dari berbagai faktor-faktor permasalahan dan kebutuhan yang telah ditentukan.
c. Pembuatan subsistem *hardware* dan *software*. Berdasarkan Gambar 1.1, rangkaian akan bekerja apabila pada tombol antrian ditekan. Setelah tombol antrian ditekan maka ATmega8535 akan mulai mencacah dan akan menampilkan *seven segmen* nomor antrian pada loket tertentu, dan ATmega8535 sebagai *master control* akan menerima data dari tiga buah *push button* yang

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

masing-masing terdapat Atmega8 pada *slave control*. Jika pada *slave control* sudah tidak ada antrian/*user*, maka *slave control* akan mengirim data panggilan ke *master control*. Setelah *master control* telah menerima data panggil dari *slave control*, maka *master control* akan mengirim data nomor antrian berikutnya ke *slave control*.



Gambar 1.1. Blok model perancangan

d. Proses pengambilan data. Teknik pengambilan data dengan cara menaikan cacahan antrian dan melakukan pencatatan nomor antrian. Dengan cara seperti ini data dapat dibandingan dan dapat diketahui jika terdapat kesalahan.
e. Analisa dan penyimpulan hasil percobaan. Analisa data dilakukan dengan membandingkan data hasil percobaan dengan perancangan sesuai spesifikasi yang telah ditentukan. Penyimpulan hasil percobaan dapat dilakukan dengan melihat hasil analisa.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB II
DASAR TEORI

2.1 Mikrokontroler AVR

Mikrokontroler adalah sebuah sistem *microprocessor* yang di dalamnya sudah terdapat CPU, ROM, RAM, I/O, *clock* dan peralatan internal lainya yang sudah saling terhubung dan terorganisasi (teralamati) dengan baik oleh pabrik pembuatnya dan dikemas dalam satu *chip* yang siap pakai[3]. Sehingga pengguna tinggal memprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya. AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator*eksternal karena di dalamnya sudah terdapat internal oscillator. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 byte.

2.1.1 Konfigurasi ATmega8535

Mikrokontroler ATmega8535 memiliki 3 jenis memori, yaitu memori *flash*, memori data dan memori EEPROM [2]. Ketiganya memiliki ruang sendiri dan terpisah.

a. Memori *flash*

ATmega8535 memiliki kapasitas memori *flash* sebesar 8 Kbyte yang terpetakan dari alamat 0000h-0FFFh, masing-masing alamat memiliki lebar data 16 bit. Memori program ini terbagi menjadi 2 bagian yaitu bagian program *boot* dan bagian program aplikasi.

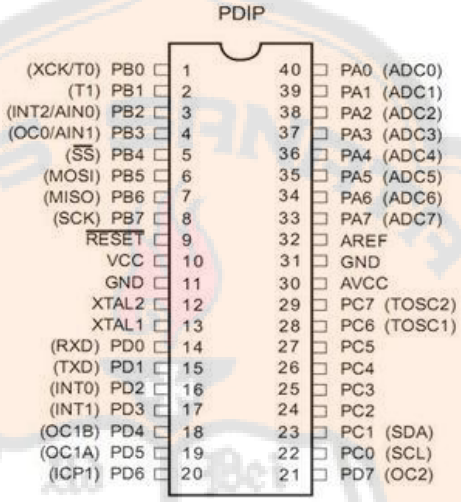
b. Memori data

ATmega8535 memiliki kapasitas memori data sebesar 608 byte yang terbagi menjadi 3 bagian yaitu register serba guna, register I/O dan SRAM. ATmega8535 memiliki 32 byte register serba guna, 64 byte register I/O yang dapat diakses sebagai bagian dari memori RAM (menggunakan instuksi LD atau ST) atau dapat juga diakses sebagai I/O (menggunakan instruksi IN atau OUT), dan 512 byte digunakan untuk memori data SRAM.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

c. Memori EEPROM

ATmega8535 memiliki memori EEPROM sebesar 512 byte yang terpisah dari memori program maupun memori data. Memori EEPROM ini hanya dapat diakses dengan menggunakan register-register I/O yaitu register EEPROM *Address*, register EEPROM *Data*, dan register EEPROM *Control*. Untuk mengakses memori EEPROM ini diperlakukan seperti mengakses data eksternal, sehingga waktu eksekusinya relatif lebih lama bila dibandingkan dengan mengakses data dari SRAM.



Gambar 2.1. Konfigurasi *pin* ATMega8535 [2]

Konfigurasi *pin* ATmega8535 dapat dilihat pada Gambar 2.1. Dari gambar dapat dijelaskan fungsi dari masing-masing *pin* Atmega8535 sebagai berikut [2]:

a. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.

b. GND merukan *pin Ground*.

c. *Port* A (PortA0…PortA7) merupakan *pin input/output* dua arah dan *pin* masukan ADC.

d. *Port* B (PortB0…PortB7) merupakan *pin input/output* dua arah dan dan *pin* fungsi khusus, seperti dapat dilihat pada Tabel 2.1.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2.1 Fungsi Khusus Port B [3]

Pin	**Fungsi Khusus**
PB7	SCK (SPI *Bus Serial Clock*)
PB6	MISO (SPI *Bus Master Input/ Slave Output*)
PB5	MOSI (SPI *Bus Master Output/ Slave Input*)
PB4	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 T1 (*Timer/Counter External Counter Input*) XCK (USART *External Clock Input/Output*)

e. *Port* C (PortC0…PortC7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus, seperti dapat dilihat pada Tabel 2.2.

Tabel 2.2. Fungsi Khusus Port C [2]

Pin	**Fungsi khusus**
PC7	TOSC2 (*Timer Oscillator Pin2*)
PC6	TOSC1 (*Timer Oscillator Pin1*)
PC5	*Input/Output*
PC4	*Input/Output*
PC3	*Input/Output*
PC2	*Input/Output*
PC1	SDA (*Two-wire Serial Buas Data Input/Output Line*)
PC0	SCL (*Two-wire Serial Buas Clock Line*)

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

f. *Port* D (PortD0…PortD7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus, seperti yang terlihat pada Tabel 2.3.

Tabel 2.3. Fungsi Khusus Port D [2]

Pin	**Fungsi khusus**
PD7	OC2 (*Timer/Counter Output Compare Match Output*)
PD6	ICP (*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 Interrupt1Input*)
PD2	INT0 (*External Interrupt* 0 *Input*)
PD1	TXD (USART *Output Pin*)
PD0	RXD (USART *Input Pin*)

g. RESET merupakan *pin* yang digunakan untuk me-*reset* mikrokontroler.

h. XTAL1 dan XTAL2 merupakan *pin* masukan *clock* eksternal.

i. AVCC merupakan *pin* masukan tegangan untuk ADC.

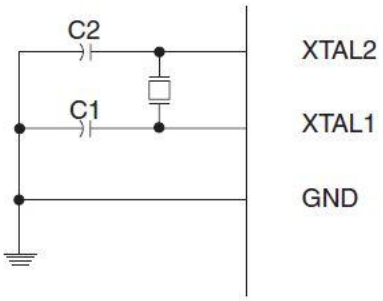
j. AREFF merupakan *pin* masukan tegangan referensi ADC.

2.1.2 *Reset* dan Osilator Eksternal

Chip akan *reset* jika tegangan catu nol atau pin RST dipaksa 0[6]. Jika membutuhkan tombol reset, dapat ditambah dengan rangkaian reset seperti pada Gambar 2.2. External Oscillator menggunakan rangkaian yang ditunjukkan pada Gambar 2.3.

Gambar 2.2. Rangkaian *Reset* [4]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 2.3. Rangkaian Osilator Kristal [7]

Osilator dapat beroperasi dalam tiga mode yang berbeda, masing-masing dioptimalkan untuk frekuensi tertentu jangkauan. Modus operasi dipilih oleh sekering CKSEL3 .. 1 seperti yang ditunjukkan pada Tabel 2.4.

Tabel 2.4. Mode Operasi Kristal Osilator [7]

CKOPT	CKSEL3..1	Frequency Range(1) (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101(2)	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 - 16.0	12 - 22

2.1.3 *Timer/Counter* 0

Timer/counter 0 adalah sebuah *timer/counter* yang dapat mencacah sumber pulsa/*clock* baik dari dalam *chip* (*timer*) ataupun dari luar *chip* (*counter*) dengan kapasitas 8-*bit* atau 256 cacahan [6].

Dapat digunakan untuk :

a. *Timer/counter* biasa
b. *Clear Timer on Compare Match* (selain Atmega8)
c. Generator frekuensi (selain Atmega8)
d. *Counter* pulsa eksternal.

Mempunyai hingga 10-*bit* (1024) *Clock Prescaler* (pemilihan *clock* yang masuk ke *timer/counter*).

2.1.3.1 Register Pengendali *Timer*0

1. *Timer/Counter Control Register* – TCCR0

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2.5. Register TCCR0 [4]

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit CS00 s.d. 02 bertugas untuk memilih (*prescaler*) atau mendefinisikan pulsa/*clock* yang akan masuk ke dalam *timer*/*counter*0 [6]. Tabel 2.5 menunjukkan register pada TCCR0 dan Tabel 2.6 menunjukkan *prescaler timer/counter0.*

Tabel 2.6. *Prescaler timer*/*counter*0 [4]

CS02	CS01	CS00	Keterangan
0	0	0	*Stop*/berhenti
0	0	1	Clk (sama dengan clok cpu)
0	1	0	Clk/8 (1 clk *timer*/counter0 = 8 clk cpu)
0	1	1	Clk/64 (1 clk *timer/counter0* = 64 clk cpu)
1	0	0	Clk/256 (1 clk *timer/counter0* = 256 clk cpu)
1	0	1	Clk/1024 (1 clk *timer/counter0* = 1024 clk cpu)
1	1	0	Clk eksternal dari *pin* T0 pada kondisi *falling edge*
1	1	1	Clk eksternal dari *pin* T0 pada kondisi *rising edge*

(1 clok *timer*/*counter*0= 8 clk cpu) artinya tiap 8 *clock* CPU yang masuk ke dalam timer/counter0 dihitung satu oleh register pencacah TCNT0.*Falling edge* adalah perubahan pulsa/*clock* dari 1 ke 0. *Rising edge* adalah perubahan pulsa/*clock* dari 0 ke 1.

Bit 7-F0C0 : *Force Output Compare*

Bit ini hanya dapat digunakan untuk metode pembanding . Jika *bit*-F0C0 di-*set* maka akan memaksa terjadinya *compare-match* (TCNT0==OCR0).*Bit* 3, 6-WGM01:0:*Waveform Generation Mode* Kedua *bit* ini digunakan memilih mode yang digunakan, seperti yang terlihat pada Tabel 2.7.

Tabel 2.7. Mode operasi [6]

WGM01	WGM00	Mode operasi	TOP	Update OCR0	Flag TOV0 on
0	0	Normal	0xFF	Imidiet	MAX
0	1	Phase correct PWM	0xFF	TOP	Bottom
1	0	CTC	OCR0	Imidiet	MAX
1	1	Fast PWM	0xFF	Bottom	MAX

Bit 5:4-COM01:0:*Compare Match Output Mode*

Kedua *bit* ini berfungsi mendefinisikan *pin* OC0 sebagai *output timer*0 (atau sebagai saluran *output* PWM). Tabel 2.8 menunjukkan *output pin* OC0 pada mode Normal dan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

CTC, Tabel 2.9 menunjukkan *output pin* OC0 pada mode *Fast* PWM dan Tabel 2.10 menunjukan *output pin* OC0 pada mode *Phase Correct* PWM.

Tabel 2.8. Mode Normal dan CTC [6]

COM01	COM00	Output pin OC0
0	0	Tidak dihubungkan
0	1	*Toggle* saat *compare match*
1	0	*Clear* saat *compare match*
1	1	Set saat *compare match*

1. *Timer*/*Counter Register*-TCNT0 [6]

Tabel 2.9. Register TCNT0 [6]

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Register ini bertugas menghitung pulsa yang masuk ke dalam *timer*/*counter*, seperti terlihat pada Tabel 2.9. [4]. Kapasitas register ini 8-*bit* atau 255 hitungan, setelah mencapai hitungan maksimal maka akan kembali ke nol (*overflow*/limpahan).

2. *Output Compare Register* – OCR0

Tabel 2.10. Register OCR0 [6]

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Register ini bertugas sebagai register pembanding yang bisa kita tentukan besarnya sesuai dengan kebutuhan, seperti terlihat pada Tabel 2.10. [6]. Dalam praktiknya pada saat TCNT0 mencacah maka otomatis oleh CPU aka membandingkan dengan isi OCR0 secara kontinyu dan jika isi TCNT0 sama dengan isi OCR0 maka akan terjadi *compare match* yang dapat dimanfaatkan untuk *mode* CTC dan PWM.

3. *Timer*/*Counter Interrupt Mask Register* – TIMSK

Tabel 2.11. Register TIMSK [6]

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Tabel 2.11 menunjukan register TIMSK [6].

Bit 0-TOIE0: T/Co *Overflow Interrupt Enable*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dalam register TIMSK *timer*/*conter*0 memiliki *bit* TOIE0 sebagai *bit* peng-aktif interupsi *timer*/*counter*0 (TOIE0=1 *enable*, TOIE0=0 *disable*).

Bit 1-OCIE0: T/Co *Output Compare Match Interrupt Enable*

Selain ATmega8, TIMSK *timer*/*counter*0 memiliki *bit* OCIE0 sebagai *bit* peng-aktif interupsi *compare match timer*/*counter*0 (OCIE0=1 *enable*, OCIE0=0 *disable*).

4. *Timer*/*Counter Interrupt Flag Register* – TIFR

Tabel 2.12. Register TIFR [4]

Bit	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Tabel 2.12 menunjukan register TIFR [6].

Bit 1-OCF0: *Output Compare Flag* 0

Flag OCF0 akan *set* sebagai indikator terjadinya *compare match*, dan akan *clear* sendiri bersamaan eksekusi vektor interupsi t*imer*0 *compare match*.

Bit 0-TOC0: *Timer* /*Counter* 0 *Overflow flag*

Bit status *timer*/*counter*0 dalam register TIFR, di mana *bit*-TOV0 (*Timer*/*Counter*0 *overflow*) akan *set* secara otomatis ketika terjadi limpahan/*overflow* pada register TCNT0 dan akan *clear* bersamaan dengan eksekusi vektor interupsi.

2.1.4 Konfigurasi ATmega8

Mikrokontroler ATmega8 memiliki 3 jenis memori, yaitu memori *flash*, memori data dan memori EEPROM yaitu:

a. Memori *Flash*

Memori flash adalah memori ROM tempat kode-kode program berada. Kata flash menunjukan jenis ROM yng dapat ditulis dan dihapus secara elektrik. Memori flash terbagi menjadi dua bagian yaitu bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program apikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting* awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer/downloader*, misalnya melalui USART.

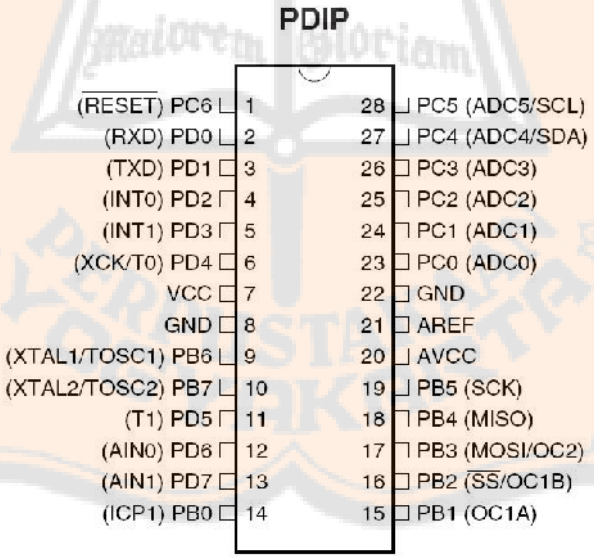
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

b. Memori Data

Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu : 32 GPR (*General Purphose Register*) adalah register khusus yang bertugas untuk membantu eksekusi program oleh ALU (*Arithmatich Logic Unit*), dalam instruksi assembler setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk variabel global atau nilai balik fungsi dan nilai-nilai yang dapat memperingan kerja ALU. Dalam istilah processor komputer sahari-hari GPR dikenal sebagai *"chace memory"*.I/O register dan Aditional I/O register adalah *register* yang difungsikan khusus untuk mengendalikan berbagai pheripheral dalam mikrokontroler seperti *pin port, timer/counter, usart* dan lain-lain. Register ini dalam keluarga mikrokontrol MCS51 dikenal sebagai SFR(*Special Function Register*).

c. EEPROM

EEPROM adalah memori data yang dapat mengendap ketika chip mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.



Gambar 2.4. Konfigurasi *pin* ATMega8 [6]

Konfigurasi *pin* ATmega8 dapat dilihat pada Gambar 2.4. Dari gambar dapat dijelaskan fungsi dari masing-masing *pin* Atmega8 sebagai berikut [3]:

- *VCC*

Merupakan *supply* tegangan digital.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

- *GND*

Merupakan ground untuk semua komponen yang membutuhkan grounding.

- Port B (PB7...PB0)

Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. Port B merupakan sebuah 8-bit *bi-directional* I/O dengan internal pull-up resistor. Sebagai *input*, pin-pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Khusus PB6 dapat digunakan sebagai *input* Kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock* internal, bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai *output* Kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber clock yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *Asyncronous Timer/Counter2*, maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.

- Port C (PC5…PC0)

Port C merupakan sebuah 7-*bit bi-directional* I/O port yang di dalam masing-masing pin terdapat *pull-up* resistor. Jumlah pin nya hanya 7 buah mulai dari *pin* C.0 sampai dengan *pin* C.6. Sebagai keluaran/*output port* C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus *(source*).

- RESET/PC6

Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin* I/O.*Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port* C lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika *level* tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

- Port D (PD7…PD0)

Port D merupakan 8-*bit bi-directional* I/O dengan internal *pull-up* resistor.Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

- AVcc

Pin ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low passfilter*.

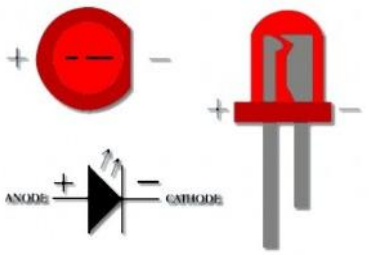
- AREF

Merupakan pin referensi jika menggunakan ADC

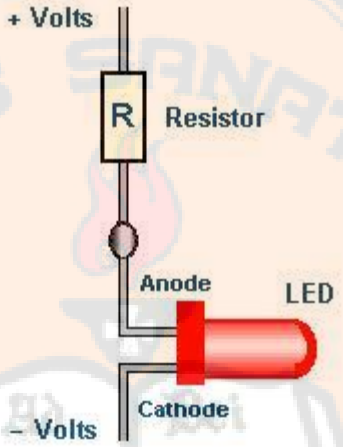
2.2 LED *(Light Emitting Dioda)*

LED merupakan salah satu jenis dioda yang mengubah energi perpindahan elektron – elektron yang jatuh dari pita konduksi ke pita valensi menjadi cahaya. Berwana –warninya cahaya yang dipancarkan ini, dikarenakan jenis bahan yang digunakan berbeda – beda. Bahan – bahannya antara lain gallium, arsen dan fosfor. LED akan memancarkan cahaya apabil diberikan tegangan listrik dengan konfigurasi forward bias.Pada umumnya, kemampuan mengalirkan arus pada LED cukup rendah yaitu maksimal 20 mA. Apabila LED dialiri arus lebih besar dari 20 mA maka LED akan rusak, sehingga pada rangkaian LED dipasang sebuah resistor sebagai pembatas arus.Penggunaan led biasanya berhubungan dengan segala hal yang dilihat oleh manusia, seperti untuk mesin hitung, jam digital, dan lain – lain. LED dapat menyala jika diberikan tegangan yang cukup untuk memicu dioda persambungannya, biasanya nilainya adalah 1.5V, LED tersebut juga harus dialiri arus listrik yang cukup untuk menyalakan LED tersebut, biasanya sebesar 10mA[11] seperti pada gambar 2.6. Simbol dan bentuk fisik dari LED dapat dilihat pada gambar 2.5[8]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 2.5. Simbol dan bentuk LED[8]



Gambar 2.6. Gambar rangkaian LED sederhana[8]

Dari gambar 2.5. dapat ketahui bahwa LED memiliki kaki 2 buah seperti dengan di-oda yaitu kaki anoda dan kaki katoda. Pada gambar diatas kaki anoda memiliki ciri fisik lebih panjang dari kaki katoda pada saat masih baru, kemudian kaki katoda pada LED ditandai dengan bagian body LED yang di papas rata. Kaki anoda dan kaki katoda pada LED disimbolkan seperti pada gambar 2.5. Pemasangan LED agar dapat menyala adalah dengan memberikan tegangan bias maju yaitu dengan memberikan tegangan positif ke kaki anoda dan tegangan negatif ke kaki katoda[8]. Dari rangkaian gambar 2.6. jika pin di beri tegangan vcc 5V, maka akan diketahui tegangan pada resistor dengan rumus :

$$VR = Vin - Vled \qquad [2.1]$$

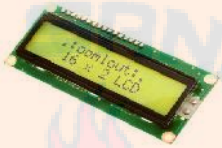
Arus yang mengalir pada rangkaian gambar 2.6. adalah :

$$I = VR/R \qquad [2.2]$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

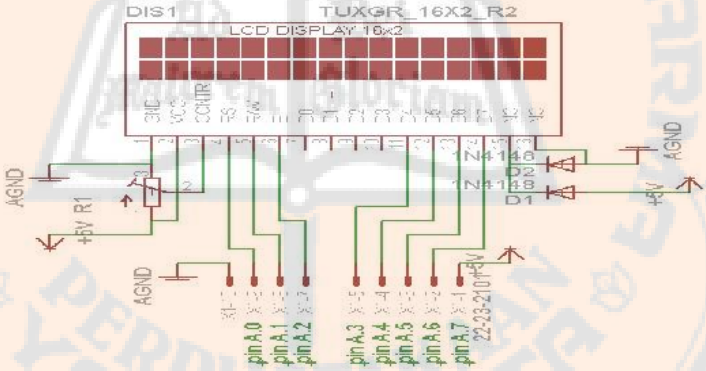
2.3 LCD *(Liquid Cristal Display)*

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD adalah salah satu jenis display elektronik yang dibuat dengan teknologi *CMOS* logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.Berikut ini adalah bentuk fisik LCD 16X2 4 *bit* ditunjukkan pada gambar 2.7[15].



Gambar 2.7. LCD 16X2 4 *bit*[15]

2.3.1 Konfigurasi Pin LCD 16X2



Gambar 2.8. Konfigurasi *pin* LCD 4 *bit*[9]

Konfigurasi *pin* dapat dilihat pada Gambar 2.8. Dari gambar dapat dijelaskan fungsi dari masing-masing *pin* LCD 4 *bit* sebagai berikut[9]:

- Kaki-kaki LCD nomor 1, 2, dan 3 adalah kaki VSS/GND**,** VCC, dan VEE/VO**.** VEE berfungsi untuk mengatur kecerahan tampilan karakter LCD. Untuk mengaturnya, digunakan potensio 10K yang dapat diputar-putar untuk mendapatkan kecerahan tampilan yang diinginkan.
- Kaki LCD nomor 4 (RS) adalah kaki Register Selector yang berfungsi untuk memilih Register Kontrol atau Register Data. Register kontrol digunakan untuk

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

mengkonfigurasi LCD. Register Data digunakan untuk menulis data karakter ke memori display LCD.

- Kaki LCD nomor 5 **(**R/W**)** digunakan untuk memilih aliran data apakah READ ataukah WRITE. Karena kita tidak memerlukan fungsi untuk membaca data dari LCD dan hanya perlu menulis data saja ke LCD, maka kaki ini dihubungkan ke GND (WRITE).
- Kaki LCD nomor 6 (ENABLE) digunakan untuk mengaktifkan LCD pada proses penulisan data ke Register Kontrol dan Register Data LCD.

2.4 Saklar *Push Button*

Saklar merupakan komponen elektronika yang berfungsi untuk menghubungkan dan memutuskan dua titik atau lebih dalam suatu rangkaian elektronika. Salah satu jenis saklar adalah saklar *Push ON* yaitu saklar yang hanya akan menghubungkan dua titik atau lebih pada saat tombolnya ditekan dan pada saat tombolnya tidak ditekan maka akan memutus-kan dua titik atau lebih dalam suatu rangkaian elektronika. Saklar push *ON* dapat ber-bentuk berbagai macam, ada yang menggunakan tuas dan ada yang tanpa tuas. Saklar *push ON* sering diaplikasikan pada tombol-tombol perangkat elektronik *digital*. Saklar *push ON* juga dikenal sebagai saklar *push button*. Simbol saklar *Push ON* ditunjukan pada gambar 2.9 berikut [10]:

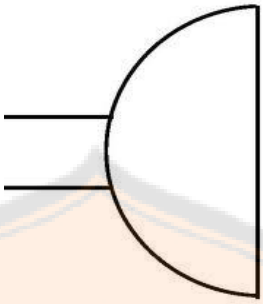
Gambar 2.9. *Push Button*[10]

2.5 *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

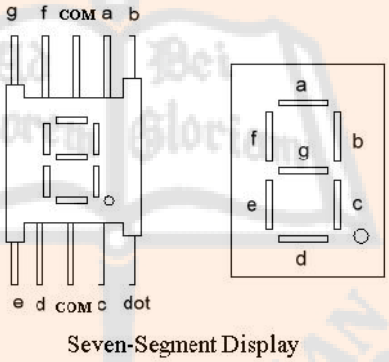
menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*). Berikut ini adalah simbol *buzzer* yang ditunjukkan gambar 2.10[14].



Gambar 2.10. Simbol *Buzzer*[14]

2.6 *Seven Segment*

Seven segment pada prinsipnya merupakan gambungan dari 8 buah led yang tersusun seperti gambar 2.11. Sehingga sebuah display 7-*segment* memerlukan masukan sebanyak 8 buah data.



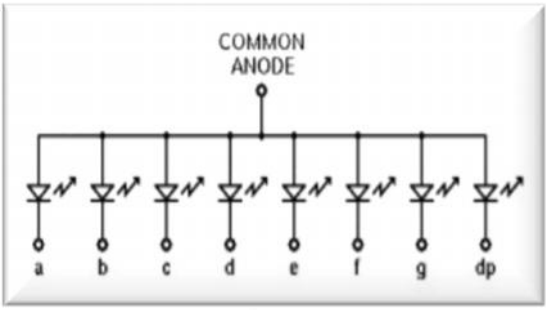
Gambar 2.11. *Display seven segment*[16]

Jenis-jenis Seven Segment[16] :

1. *Common Anoda*

Semua anoda dari led dalam seven segmen disatukan secara parallel dan semua itu dihubungkan ke vcc, dan kemudian led dihubungkan melalui tahanan pembatas arus keluar dari penggerak led. Karena dihubungkan ke vcc, maka common annoda ini berada pada kondisi AKTIF *LOW* (led akan menyala/aktif bila diberi logika 0). Rangkaian internal display *seven segment common annoda* dapat dilihat pada gambar 2.12.

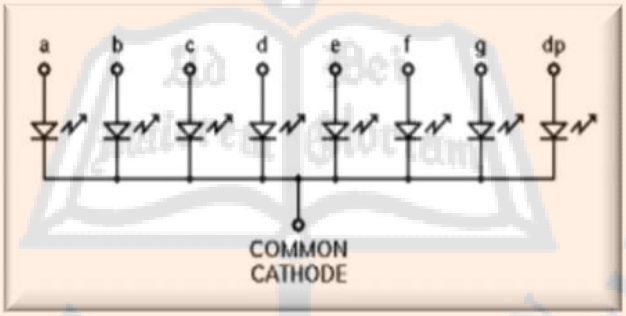
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 2.12. Rangkaian internal *common annoda*

2. *Common Catoda*

Merupakan kebalikan dari *Common Anoda*. Di sini semua katoda disatukan secara parallel dan dihubungkan ke *ground*. Karena seluruh katoda dihubungkan ke *ground*, maka *common* katoda ini berada pada kondisi AKTIF *HIGH* (led akan menyala/aktif bila diberi logika 1). Rangkaian internal *display seven segment common* katoda dapat dilihat pada gambar 2.13.



Gambar 2.13. Rangkaian internal *common* katoda

2.6.1 *Scanning Sevent Segment*

Scanning merupakan sebuah metode *interfacing*/antarmuka yang akan menghubungkan mikrokontroler dengan piranti elektronik dan penundaan waktu yang tepat maka dapat menimbulkan efek *seven segment* aktif dalam waktu yang bersamaan. Dalam cara kerja *scanning display seven segment* memakai output data dari kaki *common* katoda atau *common annoda* yang diatur melaui transistor dan *driver* untuk setiap satu *seven segment*, sedangkan pengontrolan datanya dengan memparallel tujuh jalur setiap kaki-kaki dari *seven segment* yang dihubungkan dengan mikrokontroler dan mengatur *supply* tegangan menuju ke kaki *common* secara bergantian disertai pemberian data dan kontrol data ke tujuh jalur *seven segment*.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Komunikasi data yang diterapkan pada master control(Atmega8535) adalah komunikasi asinkron antar perangkat. Komunikasi asinkron antar perangkat harus memiliki protokol yang sama antar perangkat 1 dengan perangkat lain, apabila protokolnya berbeda maka akan terjadi kesalahan komunikasi data. Protokol data merupakan susunan data yang diatur sedemikian rupa sebagai sarana komunikasi antara dua atau lebih yang menghubungkan suatu perangkat *master* dengan *slave control*.

Komunikasi data antara lain :

1. *Start bit* : ketika komunikasi UART (serial asinkron) akan diberikan, terlebih dahulu dimulai dengan pemberian *start bit*. Fungsinya sebagai pemicu (tanda) kepada penerima (RxD) bahwa akan ada data yang diberikan oleh pemancar (TxD) dan juga akan memicu clock penerima dan pemancar haruslah akurasi dengan toleransi 10% sehingga tidak terjadi kesalahan data.
2. *Data bit* : data yang dikirim secara UART dimulai dari LSB (bit ke 0) hingga MSB (bit terakhir). Menentukan banyaknya bit tersebut haruslah sama antara pemancar dengan penerima. Banyaknya data bits pada AVR bisa bernilai 7,8 atau 9 *data bit.*
3. *Parity* : sebagai pengecekan error data yang ditransfer. Parity bisa bernilai ODD (ganjil), EVEN (Genap), dan NONE selain itu pemancar dan penerima harus menggunakan parity yang sama. Jika ODD parity maka jumlah total nilai 1 pada data bits + parity berjumlah ganjil, contoh ODD, jika data bits 00110101 maka parity bernilai 1. Sedangkan jika EVEN parity maka jumlah total nilai 1 pada data bits + parity berjumlah genap. Contoh even, jika data bits 00110101 maka parity bernilai 0.
4. *Stop bits* : berfungsi sebagai akhir dari komunikasi data dan kemudian masuk pada IDLE *state*. Pengiriman data selanjutnya dapat dilakukan setelah *stop bit* diberikan.
5. *idle state* : kondisi tidak terjadinya komunikasi data dan jalur data berlogika 1 secara terus menerus.

Tabel 2.13. Komunikasi asinkron data *master control.*

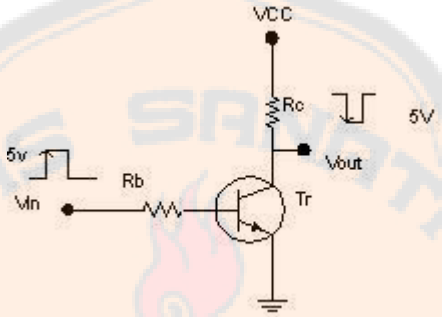
Start	D1	La1	Lp1	D2	La2	Lp2	D3	La3	Lp3	Parity(*crosscheck*)	Stop

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada tabel 2.13. Memperlihatkan serangkaian protokol komunikasi asinkron data pada *master control*.

2.7 Transistor sebagai Saklar

Transistor sebagai saklar dengan mengatur bias sebuah transistor sampai transistor jenuh, maka seolah akan didapat hubung singkat antara kaki kolektor dan emitor. Dengan memanfaatkan fenomena ini, maka transistor dapat difungsikan sebagai saklar elektronik.



Gambar 2.14. Transistor sebagai saklar[12]

Pada gambar 2.14 transistor tidak dibias (pada keadaan normal OFF). Hal ini berarti bahwa titik kerja (question point) transistor adalah pada daerah cut off. Jika Vin sudah cukup besar (0.6V untuk silikon, 0.3 untuk germanium), maka transistor akan saturasi, tegangan pada kaki kolektor emitor menjadi sekitar 0.2 V. Pada keadaan ini transistor seperti sebuah saklar yang sedang tertutup.

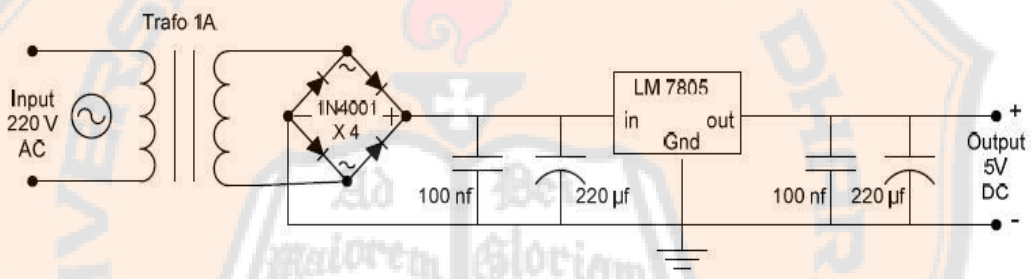
Penggunaaan suatu transisitor dalam suatau rangkaian dapat berfungsi sebagai penguat, pembagi tegangan maupun pembalik fasa dan sebagai driver. Transistor sebagai driver berfungsi menggerakkan komponen lain yang akan mengaktifkan pensaklaran, seperti mengaktifkan relay. Untuk mengaktifkan komponen tersebut ada beberapa hal yang harus dipertimbangkan yaitu :

- Jenis transistor (Silikon atau germanium) dan nilai hfe dari transistor yang digunakan.
- Karateristik relay yang digunakan (catu relay dan nilai tahanannya).
- Besar tegangan input driver.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.8 Regulator Tegangan

Regulator tegangan adalah bagian power supply yang berfungsi untuk memberikan stabilitas output pada suatu power supply. Output tegangan DC dari penyearah tanpa regulator mempunyai kecenderungan berubah harganya saat dioperasikan. Adanya perubahan pada masukan AC dan variasi beban merupakan penyebab utama terjadinya ketidakstabilan pada power supply. Pada sebagian peralatan elektronika, terjadinya perubahan catu daya akan berakibat cukup serius. Untuk mendapatkan pencatu daya yang stabil diperlukan regulator tegangan. Regulator tegangan untuk suatu power supply paling sederhana adalah menggunakan dioda zener. Rangkaian dasar penggunaan dioda zener sebagai regulator tegangan dapat dilihat pada gambar rangkaian 2.15[18].



Gambar 2.15. Rangkaian Regulator Tegangan[19]

Rangkaian pencatu daya (power supply) dengan regulator diode zener pada gambar rangkaian diatas, merupakan contoh sederhana cara pemasangan regulator tegangan dengan dioda zener. Diode zener dipasang paralel atau shunt dengan L dan R . Regulator ini hanya memerlukan sebuah diode zener terhubung seri dengan resistor RS . Perhatikan bahwa diode zener dipasang dalam posisi reverse bias. Dengan cara pemasangan ini, diode zener hanya akan berkonduksi saat tegangan reverse bias mencapai tegangan breakdown dioda zener. Penyearah berupa rangkaian diode tipe jembatan (bridge) dengan proses penyaringan atau filter berupa filter-RC. Resistor seri pada rangkaian ini berfungsi ganda. Pertama, resistor ini menghubungkan C1 dan C2 sebagai rangkaian filter. Untuk kapasitor yang sebesar ini banyak tersedia tipe elco yang memiliki polaritas dan tegangan kerja maksimum tertentu. Tegangan kerja kapasitor yang digunakan harus lebih besar dari tegangan keluaran catu daya. Anda barangkali sekarang paham mengapa rangkaian audio yang anda buat mendengung, coba periksa kembali rangkaian penyearah catu daya yang

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

anda buat, apakah tegangan *ripple* ini cukup mengganggu. Jika dipasaran tidak tersedia kapasitor yang demikian besar, tentu bisa dengan memparalel dua atau tiga buah kapasitor[17].

2.8.1 IC Regulator Tiga Terminal

Regulator tiga terminal adalah " *Integrated Voltage Regulator Circuit* " yang dirancang untuk mempertahankan tegangan outputnya tetap dan mudah untuk dirangkai. Keuntungannya adalah [17]:

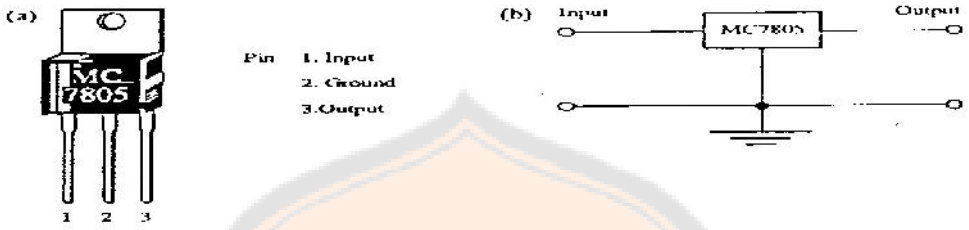
1. Membutuhkan penambahan komponen luar yang sangat sedikit, ukuran kecil
2. Mempunyai proteksi terhadap arus hubung singkat.
3. Mempunyai automatic thermal shutdown.
4. Mempunyai tegangan output yang sangat konstan.
5. Mempunyai arus rendah.
6. Mempunyai ripple output yang sangat kecil.
7. Pembiyaan rendah.

Sumber catu daya menggunakan sumber DC (Direct Current) atau arus searah agar mampu bekerja dengan baik. Banyak terdapat sumber catu daya dengan arus searah seperti baterai, accu. Tetapi penggunaan catu daya ini tentunya disesuaikan dengan perangkat elektronika yang digunakan. Dipasaran jarang ditemukan catu daya DC yang sesuai dengan kebutuhan, untuk itu kita perlu melakukan modifikasi atau merubah sumber AC (Alternating Current) arus bolak –balik menjadi arus searah DC. Selain menggunakan arus searah tentunya kinerja dari catu daya ini diharapkan stabil, karena apabila tidak stabil ini akan menyebabkan pengaruh yang sifatnya merusak fungsi kerja yang dicatunya. Untuk menjaga agar catu daya ini dipasaran banyak dijual komponen-komponen penyetabil tegangan yang berbentuk IC (Interegated Circuit) atau yang biasa disebut IC regulator. IC regulator ini sendiri secara umum dapat dibagi mejadi dua yaitu regulator tegangan tetap dan regulator yang dapat diatur. Untuk regulator tetap sekarang ini yang paling terkenal adalah keluarga 78xx untuk output tegangan positif dan keluarga 79xx untuk keluaran tegangan negatif. Untuk nilai output yang dihasilkan ini sendiri dapat diketahui dengan membaca dua digit terakhirnya. Sebagai contoh 7805 adalah regulator tegangan tetap untuk keluaran positif dengan nilai output sebesar 5 volt. Sedangkan 7912 adalah regulator

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

tegangan negatif dengan tegangan keluaran -12 volt[21].

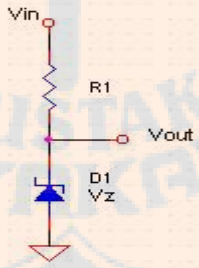
Gambar 2.16. memperlihatkan contoh IC regulator Tegangan Positif tiga terminal MC 7805[17].



Gambar 2.16. Bentuk IC Regulator dan Simbol rangkaian[17]

7805 adalah regulator untuk mendapat tegangan +5 volt, 7812 regulator tegangan +12 volt dan seterusnya. Sedangkan seri 79XX misalnya adalah 7905 dan 7912 yang berturut-turut adalah regulator tegangan -5 dan -12 volt.

Selain dari regulator tegangan tetap ada juga IC regulator yang tegangannya dapat diatur. Prinsipnya sama dengan regulator OP-amp yang dikemas dalam satu IC misalnya LM317 untuk regulator variable positif dan LM337 untuk regulator variable negatif. Bedanya resistor R1 dan R2 ada di luar IC, sehingga tegangan keluaran dapat diatur melalui resistor eksternal tersebut[17]. Rangkaian regulator yang paling sederhana ditunjukkan pada gambar 2.17.

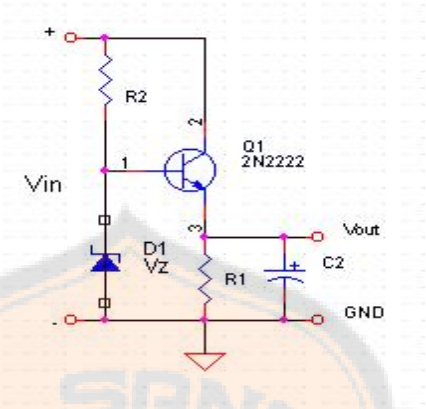


Gambar 2.17. Regulator *Zener*[17]

Pada rangkaian ini, zener bekerja pada daerah *breakdown*, sehingga menghasilkan tegangan output yang sama dengan tegangan zener atau Vout = Vz. Namun rangkaian ini hanya bermanfaat jika arus beban tidak lebih dari 50mA[17]. Prinsip rangkaian catu daya yang seperti ini disebut *shunt regulator*, salah satu ciri khasnya adalah komponen regulator yang paralel dengan beban. Ciri lain dari shunt regulator adalah, rentan terhadap *short-circuit*. Perhatikan jika Vout terhubung singkat (*short-circuit*) maka arusnya tetap I =

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Vin/R1. Disamping *regulator shunt*, ada juga yang disebut dengan *regulator seri*. Prinsip utama regulator seri seperti rangkaian pada gambar 2.18 berikut ini[17] :



Gambar 2.18. Regulator *Zener Follower*[17]

Jika diperlukan catu arus yang lebih besar, tentu perhitungan arus base I_B pada rangkaian di atas tidak bisa diabaikan lagi. Dimana seperti yang diketahui, besar arus IC akan berbanding lurus terhadap arus IB atau dirumuskan dengan $I_C = I_B$. Untuk keperluan itu, transistor Q1 yang dipakai bisa diganti dengan transistor *Darlington* yang biasanya memiliki nilai _ yang cukup besar. Dengan transistor *Darlington*, arus base yang kecil bisa menghasilkan arus IC yang lebih besar[17].

Seri LM 78XX adalah regulator dengan tiga terminal, dapat diperoleh dengan berbagai tegangan tetap.

Beberapa IC regulator mempunyai kode yang dibuat oleh pabrik pembuat komponen , sebagai contoh : IC LM7805 AC Z yang artinya sebagai berikut [17]:

LM Linear Monolithic

78L Bagian nomor dasar yang menyatakan tegangan positip

Tegangan output

AC Standart ketepatan

Z Tipe pembungkus , ZTO-92 Plastic

Seri LM 78XXC dapat diperoleh dalam kemasan TO-3 alamunium , arus keluaran (output) 1A ,boleh lebih asalkan IC regulator dilengkapi dengan pendingin (heat- sink). Regulator LM 78XXC mudah dipakai dan tambahan komponen-komponen ektern tidak banyak[17].

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Sifat-sifat IC regulator LM 78XX adalah sebagai berikut[17]:

1. pembebanan lebih termik
2. Tidak diperlukan komponen tambahan
3. Ada pengamanan untuk transistor keluaran (output).
4. Dapat diperoleh dalam kemasan TO-3 alu-minium.

Arus maksimum regulator IC ya ng dikirim ke beban tergantung pada tiga faktor, yaitu[17]:

1. Temperatur.
2. Perbedaan antara tegangan input dan output atau disebut diferensial input output.
3. Arus Beban.

Jika arus pada beban yang dibutuhkan pada rangkaian catu daya adalah I_{dc}, tegangan puncak V_m dan tegangan masuk minimal regulator adalah V_i(min) maka tegangan puncak ripple adalah :

$$V_r(\text{peak}) = V_m - V_i(\text{min}) \qquad [2.3]$$

Tegangan ripple rms adalah :

$$V_r(\text{rms}) = \frac{V_{r(peak)}}{\sqrt{3}} \qquad [2.4]$$

Nilai kapasitansi minimal yang diperlukan adalah :

$$C_{23} = \frac{2{,}4 I_{dc}}{\text{V}_\text{m} - V_i(\text{min})} \qquad [2.5]$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2.14. Karakteristik regulator IC 78XX[17]

Tipe	U out (V)	I out (A)			U in (V)	
		78XX C	78 LXX	78 MXX		
7805	5	1	0,1	0,5	7,5	20
7806	6	1	0,1	0,5	8,6	21
7808	8	1	0,1	0,5	10,6	23
7810	10	1	0,1	0,5	12,7	25
7812	12	1	0,1	0,5	14,8	27
7815	15	1	0,1	0,5	18	30
7818	18	1	0,1	0,5	21	33
7824	24	1	0,1	0,5	27,3	38

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB III
PERANCANGAN ALAT

3.1 Diagram Blok Perancangan

Perancangan alat ini terdiri dari 1 *master*, yaitu mikrokontroler ATmega8535 dan 3 *slave*, yaitu terdiri dari 3 buah mikrokontroler ATmega8. Pada *master* terdapat *display* berupa *7segment* dan LCD. *Display 7segment* berupa dua *digit* (penampil nomor antrian giliran) dan satu *digit* (penampil loket tujuan). *Display* LCD 16x2 digunakan sebagai penampil cetak nomor antrian. Pada tiga *slave* juga terdapat *display*, yang masing-masing *slave* terdapat *7segment* dua *digit*, yang menampilkan nomor antrian pada saat berlangsung. Gambar 3.1 menujukkan diagram blok perancangan alat sistem antrian loket berbasis mikrokontroler ATmega8535.

Sistem kontrol pada perancangaan alat ini terdapat pada tombol *push button* pada *master* dan pada masing-masing *slave*. Jika tombol *pushbutton* pada *master* ditekan, maka akan dijadikan *input master* sebagai cacahan antrian. *Master* akan mengolah data *input*, mencacah, menampilkan *printout* nomor antrian pada LCD, menerima data panggil pada masing-masing loket/*slave*, setelah itu *master* memproses menentukan loket yang kosong untuk antrian berikutnya lalu menampilkan nomor antrian giliran saat itu dan menampilkan loket yang akan dituju, pada *7segment* dua *digit* dan satu *digit* dan pada saat yang sama *master* mengirimkan data antrian 7*segment* pada loket/*slave* tujuan. Pada masing-masing *slave* terdapat tombol *pushbutton* dan *led* antrian yang berfungsi sebagai pengirim data panggilan antrian berikutnya ke *master*. Pemanggilan nomor antrian giliran berikutnya pada loket ke berapa, ditandai dengan bunyi *buzzer* dan menyalanya *led* panggil.

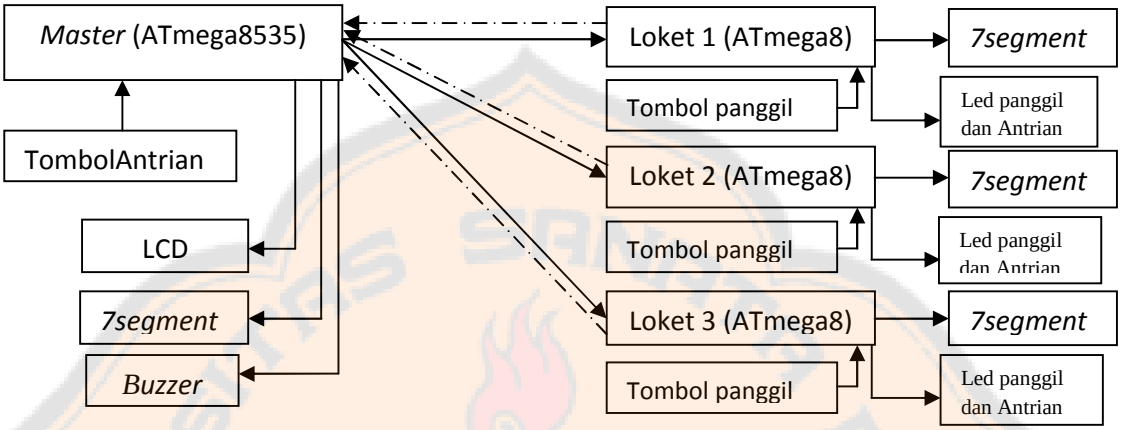
Proses pengiriman data terjadi ketika *master* menerima data panggil dari *slave*. Setelah *master* menerima data panggil dari masing-masing *slave*, maka *master* akan mengirimkan data antrian ke masing-masing *slave*. Selanjutnya ketika *slave* menerima data antrian dari *master*, maka *slave* akan mengolah data tersebut. Proses ini berlanjut secara terus-menerus.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Keterangan :

Kirim data master ke slave : ——→

Kirim data panggil slave ke master : –·–·–·→



Gambar 3.1. Diagram blok perancangan

3.2 Perancangan Perangkat Keras

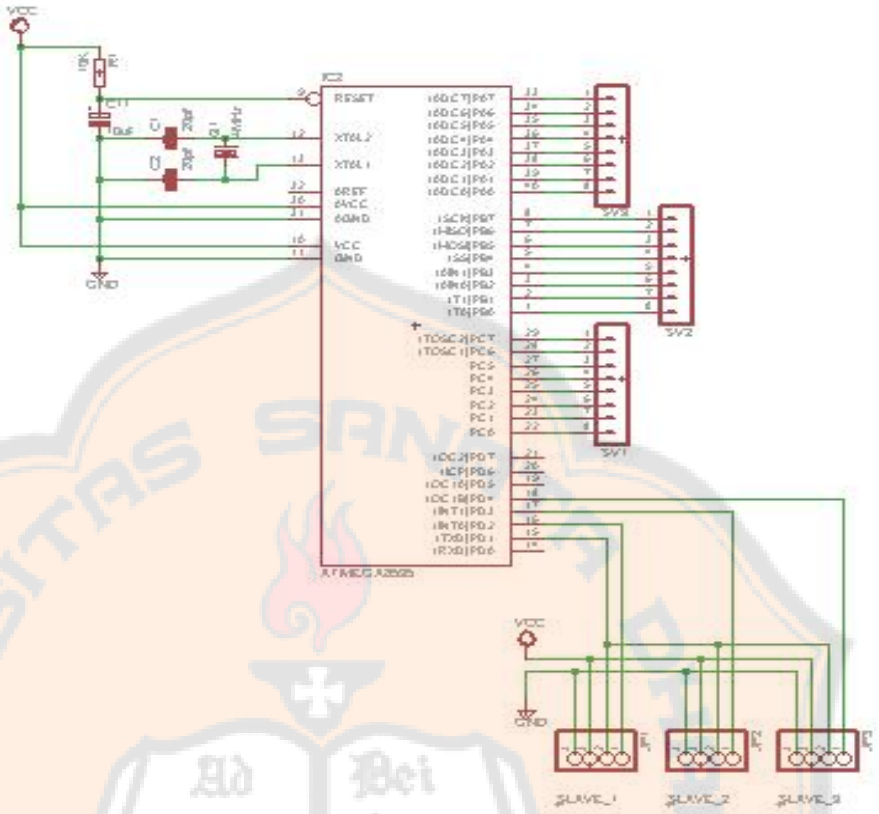
3.2.1 Perancangan Rangkaian Mikrokontroler ATmega8535

Pada rangkaian mikrokontroler AT mega8535 seperti pada gambar 3.2, *terdiri dari port* A, *port* B, *port* C dan *port* D. Seperti pada tabel 3.1, *port* A digunakan sebagai keluaran dari *7segment*. Ada 3 buah *7segment* yang digunakan dengan metode *scaning*. *Port* B digunakan sebagai kontrol *scaning* pada *7segment* (PB5, PB6 dan PB7). PC0, PC1, PC2 dan PC3 digunakan sebagai *port* data, sedangkan PC5, PC6 dan PC7 digunakan sebagai *port* pengaturan *interface* LCD. Pada *port* D digunakan sebagai pengirim dan penerima pada tiap-tiap *slave*/loket (PD1, PD2, PD3 dan PD4).

Tabel 3.1. Fungsi *pin* pada ATmega8535

Pin	Fungsi
PA7-PA0	*Output* data *7segment* (7 konektor pada *7segment*)
PB7-PB5	Kontrol *scaning* ke transistor (3 kaki basis transistor)
PC7-PC0	*Output* data LCD (7 konektor pada LCD)
PD4-PD1	PD1 pengirim data dari *master*, PD2-PD4 penerima data panggil dari *slave*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



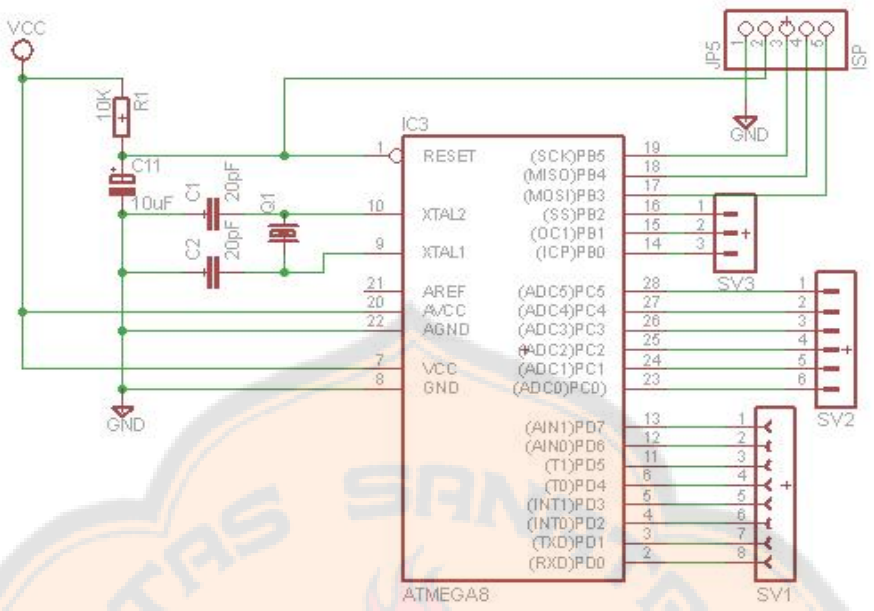
Gambar 3.2. Rangkaian mikrokontroler ATmega8535

3.2.2 Perancangan Rangkaian Mikrokontroler ATmega8

Pada rangkaian mikrokontroler ATmega8 seperti pada gambar 3.3, *terdiri dari port* B, *port* C dan *port* D. Seperti pada tabel 3.3, *port* B digunakan sebagai kontrol *scaning 7segment* (PB0 dan PB1). Ada 2 buah *7segment* yang digunakan dengan metode *scaning*. *Port* C digunakan sebagai penanda lampu indikator panggil dan isi antrian (PC5 dan PC4), serta masukan untuk tombol panggil (PC3). Pada *port* D digunakan sebagai keluaran dari *7segment*.

Komunikasi pada ATmega8 yaitu adalah menerima data antrian dari *master control*. Prinsipnya yaitu bahwa penerima hanya perlu mendeteksi *start bit* sebagai awal pengiriman data antrian, selanjutnya komunikasi data terjadi antar dua buah *shift register* yang ada pada pengirim maupun penerima. Setelah 8 *bit* data antrian diterima, penerima akan menunggu adanya *stop bit* sebagai tanda bahwa 1 *byte* data antrian telah dikirim dan penerima dapat siap untuk menunggu pengiriman data antrian berikutnya.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.3. Rangkaian mikrokontroler ATmega8

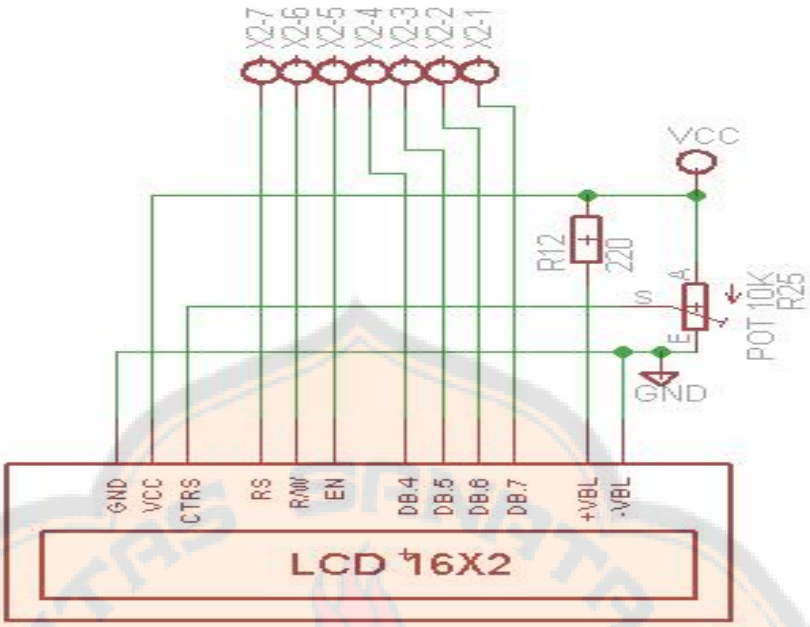
Tabel 3.3. Fungsi *pin* pada ATmega8

Pin	Fungsi
PD0-PD7	*Output* data *7segment* (7 konektor pada *7segment*)
PC5	Lampu indikator panggil (LED)
PC4	Lampu indikator isi antrian (LED)
PC3	*Input* data panggilan *push button*
PB1	Kontrol *scaning* basis transistor
PB0	Kontrol *scaning* basis transistor

3.2.3 Perancangan Rangkaian *Display* LCD 16x2

LCD yang digunakan yaitu LCD LMB162 dengan lebar *display* 2 baris 16 kolom yang konfigurasinya dapat dilihat pada gambar 3.4. Pada perancangan LCD digunakan satu buah potensiometer sebesar 10KΩ dengan fungsi untuk mengatur *contrast* dan *backlight* dari LCD. Tabel 3.4 memperlihatkan fungsi konektor LCD ke *pin* ATmega8535.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.4. Rangkaian LCD 16x2

Tabel 3.4. Fungsi konektor pada LCD ke ATmega8535

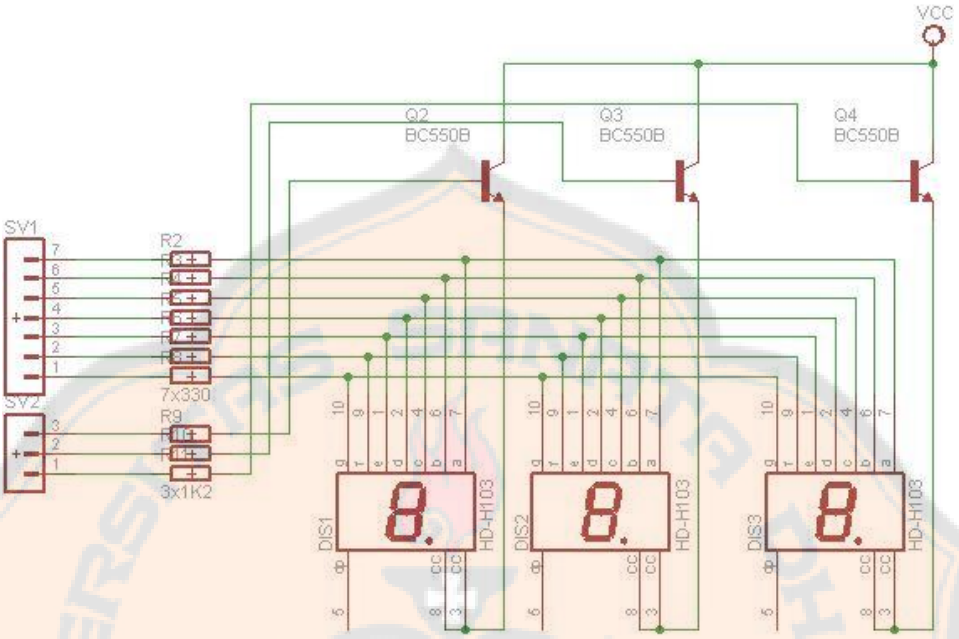
Pin	Fungsi
PC7	RS
PC6	R/W
PC5	EN
PC3	DB4
PC2	DB5
PC1	DB6
PC0	DB7

3.2.4 Perancangan Rangkaian *Display 7segment* untuk *Master Control*

Pada rangkaian 7*segment* untuk *Master Control* seperti pada gambar 3.5, metode yang digunakan dalam menyalakan atau menampilkan karakter angka adalah metode *scaning*. Metode *scanning* yang digunakan yaitu dengan memparalel kaki-kaki pada 7*segment* ke pemberi data antrian pada mikrokontroler. *Control scanning* dengan komponen elektronika transistor sebagai saklar elektronis. Pada *pin* mikrokontroler AT-

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

mega8535 PA7-PA0 digunakan sebagai *output* data dari ketiga 7*segment* master dan *control scanning* ke transistor di *pin* PB7-PB5.*Pin-pin* yang terhubung pada AT-mega8535 tersebut mengacu pada tabel 3.1.

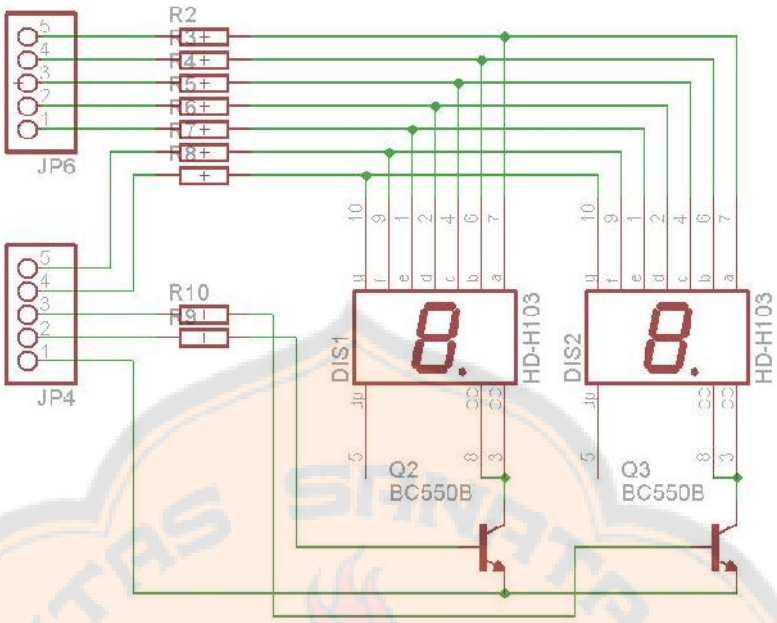


Gambar 3.5. Rangkaian *7segment Master Control*

3.2.5 Perancangan Rangkaian *Display 7segment* untuk *Slave Control*

Pada rangkaian 7*segment* untuk *Slave Control* seperti pada gambar 3.6, metode yang digunakan dalam menyalakan atau menampilkan karakter angka adalah metode *scaning*. Metode *scaning* yang digunakan yaitu dengan memparalel kaki-kaki pada 7*segment* ke pemberi data antrian pada mikrokontroler. Kontrol *scanning* dengan komponen elektronika transistor sebagai saklar elektronis. Sedangkan *pin* mikrokontroler ATmega8 PD0-PD7 sebagai *output* data dari ketiga 7*segment slave control* dan pin PC5-PC4 indikator led, *pin* PB0-PB1 kontrol *scanning* basis transistor. *Pin-pin* yang terhubung pada ATmega8 tersebut mengacu pada tabel 3.2.

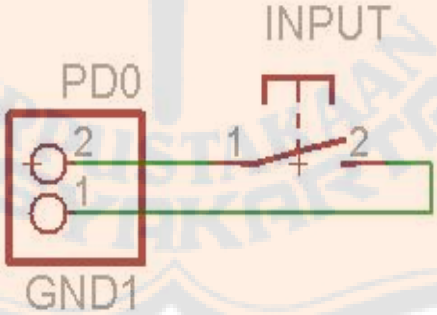
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.6. Rangkaian *7segment Slave Control*

3.2.6 Perancangan Rangkaian *Push Button*

Pada rangkaian *push button* seperti pada gambar 3.7 digunakan sebagai *input* nomor antrian *user*. Pada rangkaian ini *push button* disambungkan dengan PD0 dan *ground*. Jika pada PD0 berlogika 1(*high*), maka mikrokontroler Atmega8535 akan menganggap ada antrian *user*.



Gambar 3.7. Rangkaian *push button*

3.2.7 Perancangan *Buzzer*

Rangkaian *buzzer* digunakan sebagai penanda adanya antrian berikutnya. Gambar 3.8 memperlihatkan rangkaian *buzzer*. PD0 digunakan sebagai masukan pada transistor sebagai switch atau saklar otomatis untuk menghidupkan *buzzer*. Transistor yang digunakan BC550B jenis NPN (I_C-maks=100mA, β= 290) maka I_B dapat diketahui.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Ketika tidak ada data 1 dari mikrokontroler, maka arus ke basis akan 0 sehingga hubungan antara kolektor dan emitter terputus. Ketika mikrokontroler mengeluarkan data 1 ke *input driver buzzer*, maka kolektor akan terhubung dengan emitter sehingga kolektor transistor akan terhubung dengan emitter-nya. Karena emitter terhubung dengan *ground*, maka akan terjadi aliran arus dari kolektor ke emitter sehingga buzzer akan aktif.

Resistansi pada R_{buzzer}= 8Ω dengan V_{CC} = 5 V, pada keadaan saturasi maka V_{CE}=0;

$$I_C(sat) = \frac{V_{CC} - V_{CE}}{R_{relay}}$$

$$I_C(sat) = \frac{5\ V}{8\Omega}$$

$$I_C(sat) = 0{,}625\ A$$

Dengan β=292, maka I_B pada keadaan saturasi:

$$I_B(sat) = \frac{I_{C\ (sat)}}{\beta}$$

$$I_B(sat) = \frac{0{,}625\ A}{290}$$

$$I_B(sat) = 2{,}15\ .10^{-3} A$$

Untuk tegangan keluaran mikrokontroler 4,8 V dan V_{BE} = 0.7V :

$$R_B = \frac{V_{BB} - V_{BE}}{I_B}$$

$$R_B = \frac{4{,}8V - 0{,}7}{2{,}15\ .10^{-3} A}$$

$$R_B = 1902{,}4\ \Omega$$

R_B maksimal yang dapat digunakan adalah $1902{,}4\ \Omega$. Pemilihan R_B supaya $I_B > I_B(sat)$, misalnya I_B = 3,41mA dapat diperoleh dengan:

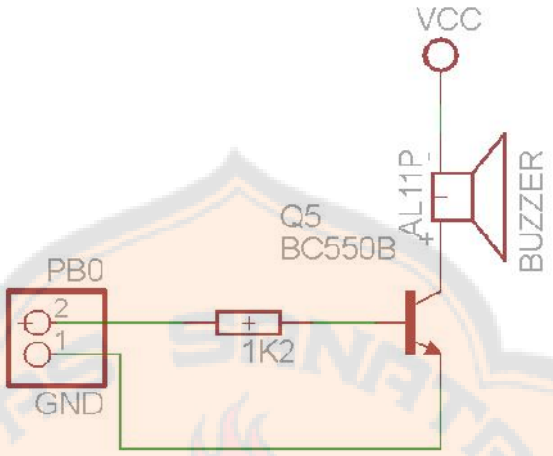
$$R_B = \frac{V_{BB} - V_{BE}}{I_B}$$

$$R_B = \frac{4{,}8V - 0{,}7V}{3{,}41mA}$$

$$R_B = 1200\Omega$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Resistor yang digunakan adalah 1200Ω.



Gambar 3.8. Rangkaian *buzzer*

3.2.8 Perancangan Rangkaian Catu Daya

Rangkaian catu daya yang digunakan menghasilkan tegangan catu sebesar +5 volt. Rangkaian catu daya memperoleh catu atau sumber tegangan dari jala-jala listrik PLN. Tegangan AC 220 volt harus diturunkan terlebih dahulu melalui trafo 1 A, penurunan tegangan menjadi sekitar 6 V. Tegangan AC tersebut kemudian diserahkan oleh dioda *bridge*, sehingga menghasilkan gelombang penuh.

Pengatur tegangan yaitu IC LM7805 digunakan untuk menghasilkan tegangan arus searah keluaran sebesar +5 volt, kemudian dihubungkan ke TIP41C agar arus yang dihasilkan lebih besar dari kemampuan IC LM7805 yaitu sebesar I_{dc} = 1000 mA, $V_m = 12\sqrt{2}V$, dan V_i(min) = 7,5. Pada persamaan 2.3, 2.4 dan 2.5 maka nilai kapasitansi pada C_{23} pada gambar 3.9 adalah :

$$C_{23} = \frac{2{,}4 I_{dc}}{V_{r(rms)}}$$

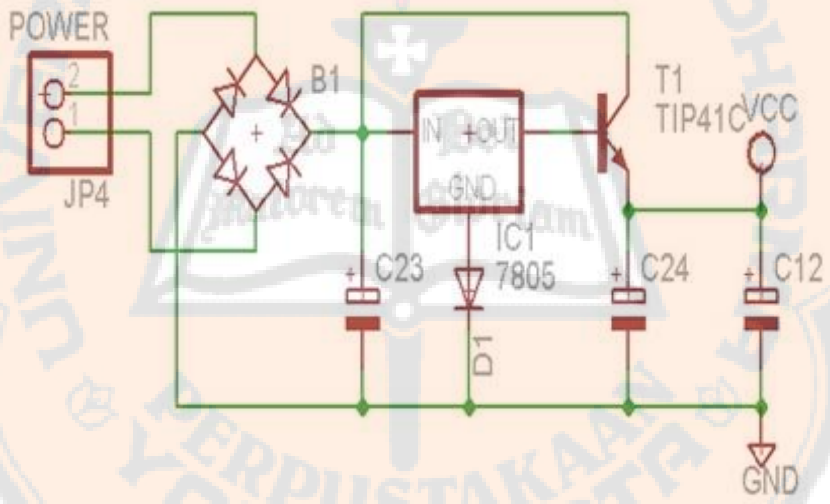
$$= \frac{2{,}4 I_{dc}}{\mathrm{V_m} - V_i(\min)/\sqrt{3}}$$

$$= \frac{(2{,}4)(1000\ mA)}{12\sqrt{2} - 7.5/\sqrt{3}}$$

$$= 439 \text{ « F}$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rangkaian catu daya dapat dilihat pada gambar 3.9, pemasangan kapasitor C23 sebesar 439µF, akan tetapi untuk memantapkan tegangan +5 volt maka menggunakan nilai kapasitor C_{23} 1000µF supaya pada rangkaian ini untuk mengurangi tegangan *ripple* yang mempunyai frekuensi rendah. Sedangkan pemasangan kapasitor C24 sekitar 100uF digunakan untuk mengurangi tegangan yang mempunyai frekuensi yang lebih tinggi. Diperlukan ketiga capasitor bertujuan untuk menyanggah tegangan agar mensupport kebutuhan tegangan saat drop tegangan. Pada IC 7805 dipasang dioda untuk pengaman polaritas dan menstabilkan tegangan agar +5 volt pada keluaran TIP41C.

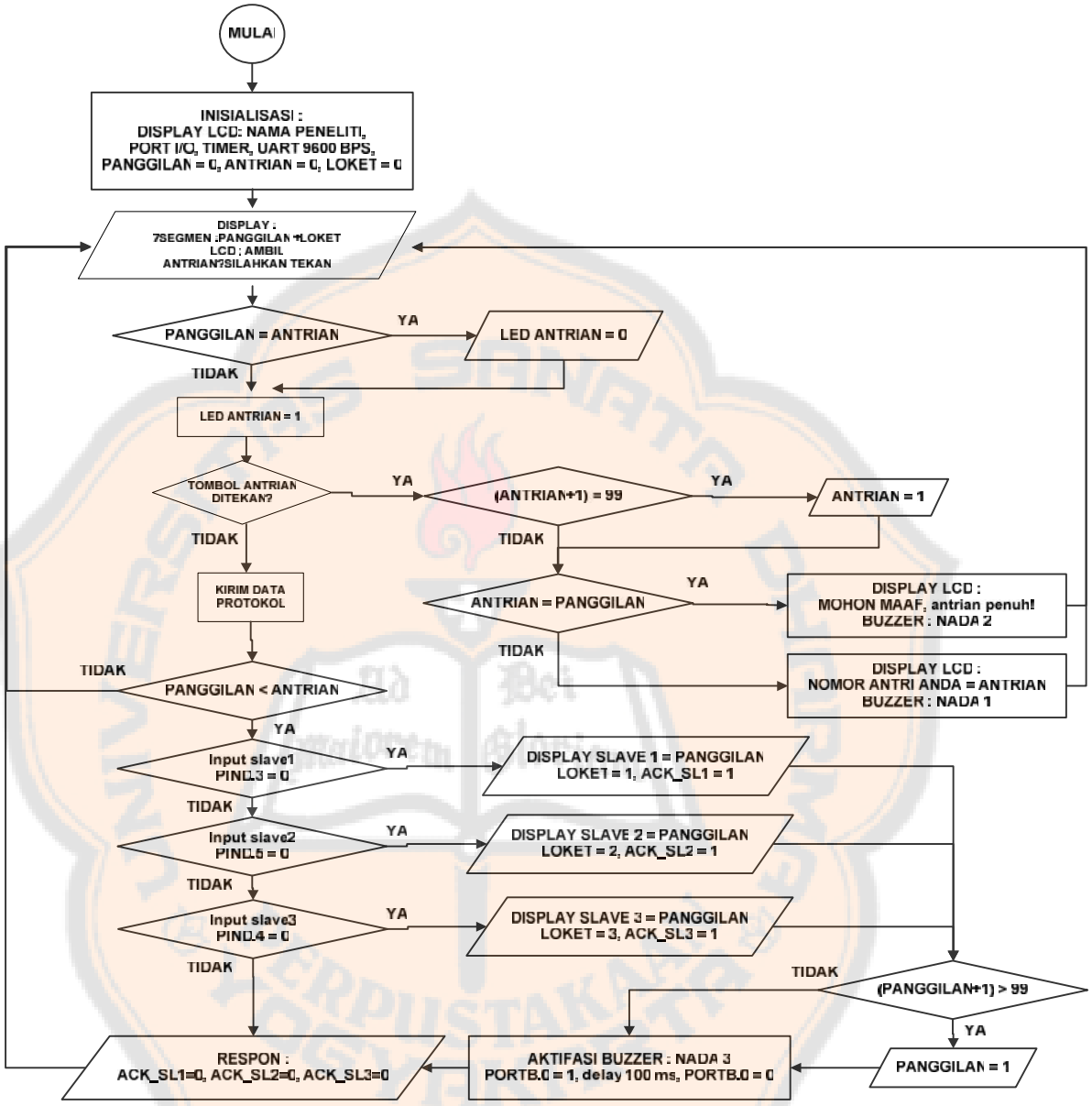


Gambar 3.9. Rangkaian catu daya

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.3 Perancangan Perangkat Lunak

3.3.1 *Flow chart Master Control* ATmega8535



Gambar 3.10. *Flow chart Master control*

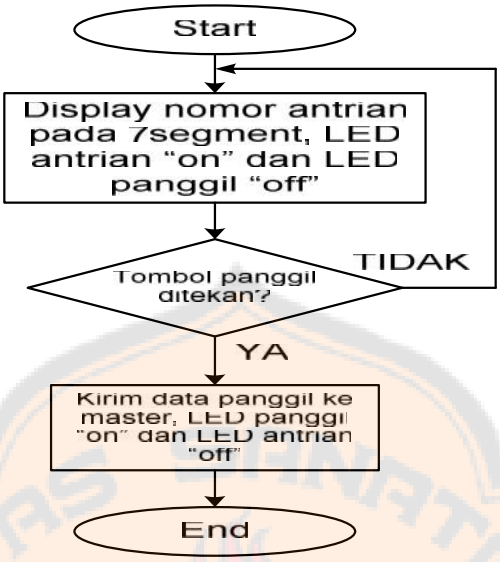
Gambar 3.10 menunjukkan *flow chart master control*. Pada *flow chart master control* terdiri dari beberapa proses. Pada proses pertama sistem akan dan melakukan inisialisasi. Tombol antrian ditekan makan inisialisai *port* I/O mikrokontroler, mengaktifkan *display* LCD nama peneliti, mengaktifkan *setting timer*, mengaktifkan UART 9600 BPS, dan setting panggilan = 0, antrian = 0, loket = 0 atau tidak ada antrian. Proses kedua *display* 7segmen pada *master control* dan *slave control* akan memproses panggilan dan loket yang akan mengaktifkan LCD, lalu akan menampilkan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

ambil antrian lalu silahkan ditekan . Jika antrian sama dengan panggilan atau nomor antrian panggilan sudah *overlap* akan memproses jika YA maka led antrian akan bernilai 0 atau tidak ada antrian pada *slave control* dan jika TIDAK maka led antrian akan bernilai 1 yang berarti ada antrian pada *slave control*. Tombol antrian ditekan kembali maka PIND.2 = 0 jika YA nomor antrian akan ditambah 1 apakah sama dengan 99, jika YA maka akan menampilkan *display* antrian pada *master control* bernilai 1. Jika TIDAK akan memproses antrian sama dengan panggilan yang berarti *overlap* pada *slave control* maka jika YA akan mengaktifkan *printout* nomor antrian pada LCD bahwa antrian penuh dan nada 2 buzzer. Jika TIDAK maka akan mengaktifkan LCD bahwa nomor antrian anda = nomor antrian selanjutnya. Proses ketiga Tombol antrian ditekan kembali maka PIND.2 = 0 jika TIDAK maka akan mengirimkan data antrian ke masing-masing input *slave control* kalau panggilan < antrian. Masing –masing input *slave control* dan *master control* akan menampilkan *display* 7segmen loket1, loket2, loket3 dan mengaktifkan ACK_SL1, ACK_SL2, ACK_SL3 = 1 lalu proses selanjutnya panggilan + 1 > 99 jika YA maka akan memanggil urutan nomor antrian 1 jika YA dan TIDAK akan mengaktifkan *buzzer* nada3, mengaktifkan PORTB=0, mengaktifkan PORTB=1 dengan delay 100ms . Proses selanjutkan akan mengaktifkan respon ACK_SL1, ACK_SL2, ACK_SL3 dan mengaktifkan *display* antrian dan loket antrian pada *master control*.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.3.2 *Flow chart Slave Control* ATmega8



Gambar 3.11. *Flow chart slave control*

Gambar 3.11 menunjukkan *flow chart slave control*. Pada *flow chart slave control*, proses pertama sistem akan menerima data antrian dari *master* dan menampilkan nomor antrian pada 7*segment*, LED antrian ON dan LED panggil OFF. Selanjutnya akan melanjutkan proses selanjutnya, yaitu pengambilan keputusan tombol panggil. Jika tombol panggil di tekan (YA) maka data panggil akan dikirim ke master, LED panggil ON dan LED antrian OFF. Jika TIDAK, maka akan melakukan proses sebelumnya.

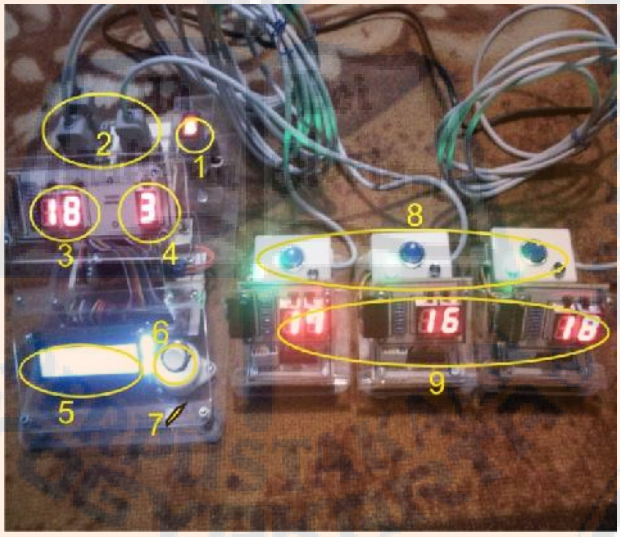
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB IV
HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil pengamatan dari Simulasi Sistem Antrian Loket Berbasis *Mikrokontroler* Atmega8535 dan Atmega8. Hasil pengamatan dan pengambilan data dilakukan yang dilakukan berupa menaikkan cacahan antrian dan melakukan pencatatan nomor antrian.

4.1.Implementasi Alat

Hasil implementasi dari Simulasi Sistem Antrian Loket Berbasis *Mikrokontroler* Atmega8535 dan Atmega8 tersusun atas rangkaian Atmega8535, rangkaian Atmega8, rangkaian LCD, rangkaian *push button*, rangkaian *buzzer*. Gambar 4.1 menunjukkan bentuk fisik dari antrian loket.



Gambar 4.1. Bentuk alat antrian loket dari depan

Pada gambar 4.1 memperlihatkan dari Simulasi antrian loket berbasis Atmega8535 dan Atmega8, yang terdiri dari:

1. Tombol *On/Off* catu daya
2. Kabel serial *Slave Control*
3. *Display 7segmen Master Control(*penampil nomor antrian *slave control)*
4. *Display 7segmen Master Control(* penampil loket antrian *slave control)*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

5. LCD 16x2
6. Tombol *pushbutton* antrian
7. *Buzzer*
8. *Display 7segmen Master Control(*penampil nomor antrian *master control)*
9. Tombol *pushbutton slave control*

Simulasi antrian loket berbasis mikrokontroler atmega8535 dan atmega8 rangkaian akan bekerja jika tombol *pushbutton* antrian ditekan. Pada alat ini terdapat tombol *pushbutton* pada *master* dan pada masing-masing *slave.* Jika tombol *pushbutton* pada *master* ditekan, maka akan dijadikan *input master* sebagai cacahan antrian. *Master* akan mengolah data *input*, mencacah, menampilkan *printout* nomor antrian pada LCD, menerima data panggil pada masing-masing loket/*slave*, setelah itu master memproses menentukan loket yang kosong untuk antrian berikutnya lalu menampilkan nomor antrian giliran saat itu dan menampilkan loket yang akan dituju, pada *7segment* dua *digit* dan satu *digit* dan pada saat yang sama *master* mengirimkan data antrian 7*segment* pada loket/*slave* tujuan. Pada masing-masing slave terdapat tombol push button dan *led* antrian yang berfungsi sebagai pengirim data panggilan antrian berikutnya ke *master*. Pemanggilan nomor antrian giliran berikutnya pada loket ke berapa, ditandai dengan bunyi *buzzer* dan menyalanya *led* panggil. Proses ini berlanjut terus menerus hingga tombol *power* ditekan dari *ON* menjadi *OFF*.

4.2.Proses Pengambilan Data

Penelitian dilakukan di Laboratorium Tugas Akhir Teknik Elektro, Universitas Sanata Dharma Yogyakarta. Pengambilan data pada penelitian ini bagi menjadi 2 kategori, yaitu pengambilan data pada sub sistem dan pengambilan data pada sistem keseluruhan. Langkah – langkah yang dilakukan dalam pengambilan data adalah:

1. Menyiapkan peralatan yang dibutuhkan dalam penelitian. Alat ukur yang dibutuhkan yaitu multimeter. Gambar 4.2 memperlihatkan alat ukur dan peralatan yang dibutuhkan dalam penelitian.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.2. Multimeter *digital*

2. Melakukan uji coba alat, pengukuran dan pengambilan data. Pada tahap ini dilakukan uji coba alat dengan cara melakukan cacahan antrian dan pencatatan nomor antrian sesuai dengan teori dasar. Proses selanjutnya adalah pengambilan data. Pengambilan data dilakukan cara menaikkan cacahan dan nomor antrian.

4.3. Pengujian Sistem Keseluruhan

Pengujian sistem keseluruhan dilakukan dengan cara pengambilan data cacahan antrian dan pencatatan nomor antrian. Teknik pengambilan data dilakukan dengan cara yaitu menaikan cacahan antrian 1 hingga 99 dan melakukan pencatatan nomor antrian. Setelah nomor antrian hingga 99, maka akan berlanjut kembali dari awal nomor antrian hingga *overlap* yaitu *user* sudah melewati batas dari antrian pada loket. Jika terjadi *overlap* maka *user* akan diberitahukan pada *printout* nomor antrian LCD bahwa antrian penuh.

Tabel 4.1 memperlihatkan data cacahan antrian dan pencatatan nomor antrian sistem keseluruhan pada *slave control* dengan cara menaikan cacahan antrian 1 hingga 99. Pada tabel 4.2 memperlihatkan data pencatatan nomor antrian pada *master control*, yang ditampilkan *display* antrian 7*segmen* dan LCD, sedangkan tabel 4.3 memperlihatkan data cacahan antrian dan pencatatan nomor antrian sistem keseluruhan pada *subsistem* dengan cara menaikan cacahan antrian hingga *overlap*. Perbedaan antara ketiga data cacahan antrian tersebut agar dapat dibandingkan dan dapat diketahui jika terdapat kesalahan.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.1. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *slave control*

Nomor Antrian	*Display 7segmen* Loket 1	*Display 7segmen* Loket 2	*Display 7segmen* Loket 3	Hasil
1	1	Kosong	Kosong	Bekerja dengan baik
2	Ada antrian	2	Kosong	Bekerja dengan baik
3	Ada antrian	Ada antrian	3	Bekerja dengan baik
4	4	Ada antrian	Ada antrian	Bekerja dengan baik
5	Ada antrian	5	Ada antrian	Bekerja dengan baik
6	Ada antrian	Ada antrian	6	Bekerja dengan baik
7	7	Ada antrian	Ada antrian	Bekerja dengan baik
8	Ada antrian	8	Ada antrian	Bekerja dengan baik
9	Ada antrian	Ada antrian	9	Bekerja dengan baik
10	10	Ada antrian	Ada antrian	Bekerja dengan baik
11	Ada antrian	11	Ada antrian	Bekerja dengan baik
12	Ada antrian	Ada antrian	12	Bekerja dengan baik
13	13	Ada antrian	Ada antrian	Bekerja dengan baik
14	Ada antrian	14	Ada antrian	Bekerja dengan baik
15	Ada antrian	Ada antrian	15	Bekerja dengan baik
16	16	Ada antrian	Ada antrian	Bekerja dengan baik
17	Ada antrian	17	Ada antrian	Bekerja dengan baik
18	Ada antrian	Ada antrian	18	Bekerja dengan baik
19	19	Ada antrian	Ada antrian	Bekerja dengan baik
20	Ada antrian	20	Ada antrian	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.1. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *slave control*

21	Ada antrian	Ada antrian	21	Bekerja dengan baik
22	22	Ada antrian	Ada antrian	Bekerja dengan baik
23	Ada antrian	23	Ada antrian	Bekerja dengan baik
24	Ada antrian	Ada antrian	24	Bekerja dengan baik
25	25	Ada antrian	Ada antrian	Bekerja dengan baik
26	Ada antrian	26	Ada antrian	Bekerja dengan baik
27	Ada antrian	Ada antrian	27	Bekerja dengan baik
28	28	Ada antrian	Ada antrian	Bekerja dengan baik
29	Ada antrian	29	Ada antrian	Bekerja dengan baik
30	Ada antrian	Ada antrian	30	Bekerja dengan baik
31	31	Ada antrian	Ada antrian	Bekerja dengan baik
32	Ada antrian	32	Ada antrian	Bekerja dengan baik
33	Ada antrian	Ada antrian	33	Bekerja dengan baik
34	34	Ada antrian	Ada antrian	Bekerja dengan baik
35	Ada antrian	35	Ada antrian	Bekerja dengan baik
36	Ada antrian	Ada antrian	36	Bekerja dengan baik
37	37	Ada antrian	Ada antrian	Bekerja dengan baik
38	Ada antrian	38	Ada antrian	Bekerja dengan baik
39	Ada antrian	Ada antrian	39	Bekerja dengan baik
40	40	Ada antrian	Ada antrian	Bekerja dengan baik
41	Ada antrian	41	Ada antrian	Bekerja dengan baik
42	Ada antrian	Ada antrian	42	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.1. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *slave control*

43	43	Ada antrian	Ada antrian	Bekerja dengan baik
44	Ada antrian	44	Ada antrian	Bekerja dengan baik
45	Ada antrian	Ada antrian	45	Bekerja dengan baik
46	46	Ada antrian	Ada antrian	Bekerja dengan baik
47	Ada antrian	47	Ada antrian	Bekerja dengan baik
48	Ada antrian	Ada antrian	48	Bekerja dengan baik
49	49	Ada antrian	Ada antrian	Bekerja dengan baik
50	Ada antrian	50	Ada antrian	Bekerja dengan baik
51	Ada antrian	Ada antrian	51	Bekerja dengan baik
52	52	Ada antrian	Ada antrian	Bekerja dengan baik
53	Ada antrian	53	Ada antrian	Bekerja dengan baik
54	Ada antrian	Ada antrian	54	Bekerja dengan baik
55	55	Ada antrian	Ada antrian	Bekerja dengan baik
56	Ada antrian	56	Ada antrian	Bekerja dengan baik
57	Ada antrian	Ada antrian	57	Bekerja dengan baik
58	58	Ada antrian	Ada antrian	Bekerja dengan baik
59	Ada antrian	59	Ada antrian	Bekerja dengan baik
60	Ada antrian	Ada antrian	60	Bekerja dengan baik
61	61	Ada antrian	Ada antrian	Bekerja dengan baik
62	Ada antrian	62	Ada antrian	Bekerja dengan baik
63	Ada antrian	Ada antrian	63	Bekerja dengan baik
64	64	Ada antrian	Ada antrian	Bekerja dengan baik
65	Ada antrian	65	Ada antrian	Bekerja dengan baik
66	Ada antrian	Ada antrian	66	Bekerja dengan baik
67	67	Ada antrian	Ada antrian	Bekerja dengan baik
68	Ada antrian	68	Ada antrian	Bekerja dengan baik
69	Ada antrian	Ada antrian	69	Bekerja dengan baik
70	70	Ada antrian	Ada antrian	Bekerja dengan baik
71	Ada antrian	71	Ada antrian	Bekerja dengan baik
72	Ada antrian	Ada antrian	72	Bekerja dengan baik
73	73	Ada antrian	Ada antrian	Bekerja dengan baik
74	Ada antrian	74	Ada antrian	Bekerja dengan baik
75	Ada antrian	Ada antrian	75	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.1. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *slave control*

76	76	Ada antrian	Ada antrian	Bekerja dengan baik
77	Ada antrian	77	Ada antrian	Bekerja dengan baik
78	Ada antrian	Ada antrian	78	Bekerja dengan baik
79	79	Ada antrian	Ada antrian	Bekerja dengan baik
80	Ada antrian	80	Ada antrian	Bekerja dengan baik
81	Ada antrian	Ada antrian	81	Bekerja dengan baik
82	82	Ada antrian	Ada antrian	Bekerja dengan baik
83	Ada antrian	83	Ada antrian	Bekerja dengan baik
84	Ada antrian	Ada antrian	84	Bekerja dengan baik
85	85	Ada antrian	Ada antrian	Bekerja dengan baik
86	Ada antrian	86	Ada antrian	Bekerja dengan baik
87	Ada antrian	Ada antrian	87	Bekerja dengan baik
88	88	Ada antrian	Ada antrian	Bekerja dengan baik
89	Ada antrian	89	Ada antrian	Bekerja dengan baik
90	Ada antrian	Ada antrian	90	Bekerja dengan baik
91	91	Ada antrian	Ada antrian	Bekerja dengan baik
92	Ada antrian	92	Ada antrian	Bekerja dengan baik
93	Ada antrian	Ada antrian	93	Bekerja dengan baik
94	94	Ada antrian	Ada antrian	Bekerja dengan baik
95	Ada antrian	95	Ada antrian	Bekerja dengan baik
96	Ada antrian	Ada antrian	96	Bekerja dengan baik
97	97	Ada antrian	Ada antrian	Bekerja dengan baik
98	Ada antrian	98	Ada antrian	Bekerja dengan baik
99	Ada antrian	Ada antrian	99	Bekerja dengan baik

Dari Tabel 4.1 menunjukkan bahwa *slave control* melakukkan proses pengiriman data ketika *master* menerima data panggil dari *slave*. Setelah *master* menerima data panggil dari masing-masing *slave*, maka *master* akan mengirimkan data antrian ke masing-masing *slave*. Selanjutnya ketika *slave* menerima data antrian dari *master*, maka *slave* akan mengolah data tersebut. Proses ini berlanjut secara terus-menerus.

Tingkat keberhasilan alat dapat dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\textbf{Banyaknya percobaan yang berhasil}}{\textbf{Jumlah prcobaan yang dilakukan}} \times 100\%$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$= \frac{99}{99} x100\% = 100\%$$

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam cacahan antrian dan pencatatan nomor antrian untuk *slave control*.

Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control*

Nomor Antrian	*Display 7segmen Master Control* Nomor Antrian	*Display* LCD	*Display 7segmen Master Control* Loket Antrian	Hasil
1	1	Nomor antrian =1=	1	Bekerja dengan baik
2	2	Nomor antrian =2=	2	Bekerja dengan baik
3	3	Nomor antrian =3=	3	Bekerja dengan baik
4	4	Nomor antrian =4=	1	Bekerja dengan baik
5	5	Nomor antrian =5=	2	Bekerja dengan baik
6	6	Nomor antrian =6=	3	Bekerja dengan baik
7	7	Nomor antrian =7=	1	Bekerja dengan baik
8	8	Nomor antrian =8=	2	Bekerja dengan baik
9	9	Nomor antrian =9=	3	Bekerja dengan baik
10	10	Nomor antrian =10=	1	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control*

11	11	Nomor antrian =11=	2	Bekerja dengan baik
12	12	Nomor antrian =12=	3	Bekerja dengan baik
13	13	Nomor antrian =13=	1	Bekerja dengan baik
14	14	Nomor antrian =14=	2	Bekerja dengan baik
15	15	Nomor antrian =15=	3	Bekerja dengan baik
16	16	Nomor antrian =16=	1	Bekerja dengan baik
17	17	Nomor antrian =17=	2	Bekerja dengan baik
18	18	Nomor antrian =18=	3	Bekerja dengan baik
19	19	Nomor antrian =19=	1	Bekerja dengan baik
20	20	Nomor anrian =20=	2	Bekerja dengan baik
21	21	Nomor antrian =21=	3	Bekerja dengan baik
22	22	Nomor antrian =22=	1	Bekerja dengan baik
23	23	Nomor antrian =23=	2	Bekerja dengan baik
24	24	Nomor antrian =24=	3	Bekerja dengan baik
25	25	Nomor antrian =25=	1	Bekerja dengan baik
26	26	Nomor antrian =26=	2	Bekerja dengan baik
27	27	Nomor antrian =27=	3	Bekerja dengan baik
28	28	Nomor antrian =28=	1	Bekerja dengan baik
29	29	Nomor anrian =29=	2	Bekerja dengan baik
30	30	Nomor antrian =30=	3	Bekerja dengan baik
31	31	Nomor antrian =31=	1	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control*

32	32	Nomor anrian =32=	2	Bekerja dengan baik
33	33	Nomor anrian =33=	3	Bekerja dengan baik
34	34	Nomor antrian =34=	1	Bekerja dengan baik
35	35	Nomor antrian =35=	2	Bekerja dengan baik
36	36	Nomor antrian =36=	3	Bekerja dengan baik
37	37	Nomor antrian =37=	1	Bekerja dengan baik
38	38	Nomor antrian =38=	2	Bekerja dengan baik
39	39	Nomor antrian =39=	3	Bekerja dengan baik
40	40	Nomor anrian =40=	1	Bekerja dengan baik
41	41	Nomor antrian =41=	2	Bekerja dengan baik
42	42	Nomor antrian =42=	3	Bekerja dengan baik
43	43	Nomor antrian =43=	1	Bekerja dengan baik
44	44	Nomor antrian =44=	2	Bekerja dengan baik
45	45	Nomor antrian =45=	3	Bekerja dengan baik
46	46	Nomor antrian =46=	1	Bekerja dengan baik
47	47	Nomor antrian =47=	2	Bekerja dengan baik
48	48	Nomor antrian =48=	3	Bekerja dengan baik
49	49	Nomor antrian =49=	1	Bekerja dengan baik
50	50	Nomor antrian =50=	2	Bekerja dengan baik
51	51	Nomor antrian =51=	3	Bekerja dengan baik
52	52	Nomor antrian =52=	1	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control*

53	53	Nomor antrian =53=	2	Bekerja dengan baik
54	54	Nomor antrian =54=	3	Bekerja dengan baik
55	55	Nomor antrian =55=	1	Bekerja dengan baik
56	56	Nomor antrian =56=	2	Bekerja dengan baik
57	57	Nomor antrian =57=	3	Bekerja dengan baik
58	58	Nomor antrian =58=	1	Bekerja dengan baik
59	59	Nomor antrian =59=	2	Bekerja dengan baik
60	60	Nomor antrian =60=	3	Bekerja dengan baik
61	61	Nomor antrian =61=	1	Bekerja dengan baik
62	62	Nomor antrian =62=	2	Bekerja dengan baik
63	63	Nomor antrian =63=	3	Bekerja dengan baik
64	64	Nomor antrian =64=	1	Bekerja dengan baik
65	65	Nomor antrian =65=	2	Bekerja dengan baik
66	66	Nomor antrian =66=	3	Bekerja dengan baik
67	67	Nomor antrian =67=	1	Bekerja dengan baik
68	68	Nomor antrian =68=	2	Bekerja dengan baik
69	69	Nomor antrian =69=	3	Bekerja dengan baik
70	70	Nomor antrian =70=	1	Bekerja dengan baik
71	71	Nomor antrian =71=	2	Bekerja dengan baik
72	72	Nomor antrian =72=	3	Bekerja dengan baik
73	73	Nomor antrian =73=	1	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.2. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control*

74	74	Nomor antrian =74=	2	Bekerja dengan baik
75	75	Nomor antrian =75=	3	Bekerja dengan baik
76	76	Nomor antrian =76=	1	Bekerja dengan baik
77	77	Nomor antrian =77=	2	Bekerja dengan baik
78	78	Nomor antrian =78=	3	Bekerja dengan baik
79	79	Nomor antrian =79=	1	Bekerja dengan baik
80	80	Nomor antrian =80=	2	Bekerja dengan baik
81	81	Nomor antrian =81=	3	Bekerja dengan baik
82	82	Nomor antrian =82=	1	Bekerja dengan baik
83	83	Nomor antrian =83=	2	Bekerja dengan baik
84	84	Nomor antrian =84=	3	Bekerja dengan baik
85	85	Nomor antrian =85=	1	Bekerja dengan baik
86	86	Nomor antrian =86=	2	Bekerja dengan baik
87	87	Nomor antrian =87=	3	Bekerja dengan baik
88	88	Nomor antrian =88=	1	Bekerja dengan baik
89	89	Nomor antrian =89=	2	Bekerja dengan baik
90	90	Nomor antrian =90=	3	Bekerja dengan baik
91	91	Nomor antrian =91=	1	Bekerja dengan baik
92	92	Nomor antrian =92=	2	Bekerja dengan baik
93	93	Nomor antrian =93=	3	Bekerja dengan baik
94	94	Nomor antrian =94=	1	Bekerja dengan baik
95	95	Nomor antrian =95=	2	Bekerja dengan baik
96	96	Nomor antrian =96=	3	Bekerja dengan baik
97	97	Nomor antrian =97=	1	Bekerja dengan baik
98	98	Nomor antrian =98=	2	Bekerja dengan baik
99	99	Nomor antrian =99=	3	Bekerja dengan baik

Pengujian Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* bertujuan untuk mengetahui rangkaian ini dapat berfungsi sesuai perancangan. Dari Tabel 4.2 menunjukkan bahwa *master* akan mengolah data *input*, mencacah, menampilkan *printout* nomor antrian pada LCD, menerima data panggil pada masing-masing loket/*slave*, setelah itu *master* memproses menentukan loket yang kosong untuk antrian berikutnya lalu menampilkan nomor antrian giliran saat itu dan menampilkan loket yang akan dituju. Tingkat keberhasilan alat dapat dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{Banyaknya percobaan yang berhasil}}{\text{Jumlah percobaan yang dilakukan}} \times 100\%$$

$$= \frac{99}{99} \times 100\% = 100\%$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam cacahan antrian dan pencatatan nomor antrian untuk *master control*.

Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

Nomor Antrian	*Display 7segmen Master Control* Nomor Antrian	*Display* LCD	*Display 7segmen Master Control* Loket Antrian	Hasil
1	1	Nomor antrian =1=	1	Bekerja dengan baik
2	2	Nomor antrian =2=	2	Bekerja dengan baik
3	3	Nomor antrian =3=	3	Bekerja dengan baik
4	4	Nomor antrian =4=	1	Bekerja dengan baik
5	5	Nomor antrian =5=	2	Bekerja dengan baik
6	6	Nomor antrian =6=	3	Bekerja dengan baik
7	7	Nomor antrian =7=	1	Bekerja dengan baik
8	8	Nomor antrian =8=	2	Bekerja dengan baik
9	9	Nomor antrian =9=	3	Bekerja dengan baik
10	10	Nomor antrian =10=	1	Bekerja dengan baik
11	11	Nomor antrian =11=	2	Bekerja dengan baik
12	12	Nomor antrian =12=	3	Bekerja dengan baik
13	13	Nomor antrian =13=	1	Bekerja dengan baik
14	14	Nomor antrian =14=	2	Bekerja dengan baik
15	15	Nomor antrian =15=	3	Bekerja dengan baik
16	16	Nomor antrian =16=	1	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

17	17	Nomor antrian =17=	2	Bekerja dengan baik
18	18	Nomor anrian =18=	3	Bekerja dengan baik
19	19	Nomor antrian =19=	1	Bekerja dengan baik
20	20	Nomor antrian =20=	2	Bekerja dengan baik
21	21	Nomor antrian =21=	3	Bekerja dengan baik
22	22	Nomor antrian =22=	1	Bekerja dengan baik
23	22	Nomor antrian =23=	2	Bekerja dengan baik
24	24	Nomor antrian =24=	3	Bekerja dengan baik
25	23	Nomor antrian =25=	3	Bekerja dengan baik
26	23	Nomor antrian =26=	3	Bekerja dengan baik
27	23	Nomor anrian =27=	3	Bekerja dengan baik
28	23	Nomor antrian =28=	3	Bekerja dengan baik
29	23	Nomor antrian =29=	3	Bekerja dengan baik
30	23	Nomor anrian =30=	3	Bekerja dengan baik
31	23	Nomor anrian =31=	3	Bekerja dengan baik
32	23	Nomor antrian =32=	3	Bekerja dengan baik
33	23	Nomor antrian =33=	3	Bekerja dengan baik
34	23	Nomor antrian =34=	3	Bekerja dengan baik
35	23	Nomor antrian =35=	3	Bekerja dengan baik
36	23	Nomor antrian =36=	3	Bekerja dengan baik
37	23	Nomor antrian =37=	3	Bekerja dengan baik
38	23	Nomor anrian =38=	3	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

39	23	Nomor antrian =39=	3	Bekerja dengan baik
40	23	Nomor antrian =40=	3	Bekerja dengan baik
41	23	Nomor antrian =41=	3	Bekerja dengan baik
42	23	Nomor antrian =42=	3	Bekerja dengan baik
43	23	Nomor antrian =43=	3	Bekerja dengan baik
44	23	Nomor antrian =44=	3	Bekerja dengan baik
45	23	Nomor antrian =45=	3	Bekerja dengan baik
46	23	Nomor antrian =46=	3	Bekerja dengan baik
47	23	Nomor antrian =47=	3	Bekerja dengan baik
48	23	Nomor antrian =48=	3	Bekerja dengan baik
49	23	Nomor antrian =49=	3	Bekerja dengan baik
50	23	Nomor antrian =50=	3	Bekerja dengan baik
51	23	Nomor antrian =51=	3	Bekerja dengan baik
52	23	Nomor antrian =52=	3	Bekerja dengan baik
53	23	Nomor antrian =53=	3	Bekerja dengan baik
54	23	Nomor antrian =54=	3	Bekerja dengan baik
55	23	Nomor antrian =55=	3	Bekerja dengan baik
56	23	Nomor antrian =56=	3	Bekerja dengan baik
57	23	Nomor antrian =57=	3	Bekerja dengan baik
58	23	Nomor antrian =58=	3	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

59	23	Nomor antrian =59=	3	Bekerja dengan baik
60	23	Nomor antrian =60=	3	Bekerja dengan baik
61	23	Nomor antrian =61=	3	Bekerja dengan baik
62	23	Nomor antrian =62=	3	Bekerja dengan baik
63	23	Nomor antrian =63=	3	Bekerja dengan baik
64	23	Nomor antrian =64=	3	Bekerja dengan baik
65	23	Nomor antrian =65=	3	Bekerja dengan baik
66	23	Nomor antrian =66=	3	Bekerja dengan baik
67	23	Nomor antrian =67=	3	Bekerja dengan baik
68	23	Nomor antrian =68=	3	Bekerja dengan baik
69	23	Nomor antrian =69=	3	Bekerja dengan baik
70	23	Nomor antrian =70=	3	Bekerja dengan baik
71	23	Nomor antrian =71=	3	Bekerja dengan baik
72	23	Nomor antrian =72=	3	Bekerja dengan baik
73	23	Nomor antrian =73=	3	Bekerja dengan baik
74	23	Nomor antrian =74=	3	Bekerja dengan baik
75	23	Nomor antrian =75=	3	Bekerja dengan baik
76	23	Nomor antrian =76=	3	Bekerja dengan baik
77	23	Nomor antrian =77=	3	Bekerja dengan baik
78	23	Nomor antrian =78=	3	Bekerja dengan baik
79	23	Nomor antrian =79=	3	Bekerja dengan baik
80	23	Nomor antrian =80=	3	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

81	23	Nomor antrian =81=	3	Bekerja dengan baik
82	23	Nomor antrian =82=	3	Bekerja dengan baik
83	23	Nomor antrian =83=	3	Bekerja dengan baik
84	23	Nomor antrian =84=	3	Bekerja dengan baik
85	23	Nomor antrian =85=	3	Bekerja dengan baik
86	23	Nomor antrian =86=	3	Bekerja dengan baik
87	23	Nomor antrian =87=	3	Bekerja dengan baik
88	23	Nomor antrian =88=	3	Bekerja dengan baik
89	23	Nomor antrian =89=	3	Bekerja dengan baik
90	23	Nomor antrian =90=	3	Bekerja dengan baik
91	23	Nomor antrian =91=	3	Bekerja dengan baik
92	23	Nomor antrian =92=	3	Bekerja dengan baik
93	23	Nomor antrian =93=	3	Bekerja dengan baik
94	23	Nomor antrian =94=	3	Bekerja dengan baik
95	23	Nomor antrian =95=	3	Bekerja dengan baik
96	23	Nomor antrian =96=	3	Bekerja dengan baik
97	23	Nomor antrian =97=	3	Bekerja dengan baik
98	23	Nomor antrian =98=	3	Bekerja dengan baik
99	23	Nomor antrian =99=	3	Bekerja dengan baik
1	23	Nomor antrian =1=	3	Bekerja dengan baik
2	23	Nomor antrian =2=	3	Bekerja dengan baik
3	23	Nomor antrian =3=	3	Bekerja dengan baik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lanjutan Tabel 4.3. Data cacahan antrian dan pencatatan nomor antrian keseluruhan untuk *master control* hingga *overlap*

4	23	Nomor antrian =4=	3	Bekerja dengan baik
5	23	Nomor antrian =5=	3	Bekerja dengan baik
6	23	Nomor antrian =6=	3	Bekerja dengan baik
7	23	Nomor antrian =7=	3	Bekerja dengan baik
8	23	Nomor antrian =8=	3	Bekerja dengan baik
9	23	Nomor antrian =9=	3	Bekerja dengan baik
10	23	Nomor antrian =10=	3	Bekerja dengan baik
11	23	Nomor antrian =11=	3	Bekerja dengan baik
12	23	Nomor antrian =12=	3	Bekerja dengan baik
13	23	Nomor antrian =13=	3	Bekerja dengan baik
14	23	Nomor antrian =14=	3	Bekerja dengan baik
15	23	Nomor antrian =15=	3	Bekerja dengan baik
16	23	Nomor antrian =16=	3	Bekerja dengan baik
17	23	Nomor antrian =17=	3	Bekerja dengan baik
18	23	Nomor antrian =18=	3	Bekerja dengan baik
19	23	Nomor antrian =19=	3	Bekerja dengan baik
20	23	Nomor anrian =20=	3	Bekerja dengan baik
21	23	Maaf antrian Penuh	3	Bekerja dengan baik

Dari Tabel 4.3 menunjukkan bahwa antrian terakhir pada loket antrian yaitu nomor antrian 24 di loket 3. Terjadi *overlap* karena *user* sudah melewati batas dari antrian pada loket sehingga display 7segmen pada master control menampilkan nomor antrian yang terakhir yaitu nomor antrian 24. Prosesnya yaitu *master* akan mengolah data *input*, mencacah, menampilkan *printout* nomor antrian pada LCD, menerima data panggil pada masing-masing loket/*slave*, setelah itu *master* memproses menentukan loket yang kosong untuk antrian berikutnya lalu menampilkan nomor antrian giliran saat itu dan menampilkan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

loket yang akan dituju. Jika loket penuh atau ada antrian semua maka slave tidak mengirimkan data panggil ke *master* sehinggga disaat nomor antrian melebihi data antrian di *slave* maka akan ditampilkan pada LCD bahwa antrian penuh. Tingkat keberhasilan alat dapat dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{Banyaknya percobaan yang berhasil}}{\text{Jumlah percobaan yang dilakukan}} \times 100\%$$

$$= \frac{99}{99} \times 100\% = 100\%$$

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam cacahan antrian dan pencatatan nomor antrian untuk *master control*.

4.4. Pengujian Sub Sistem

Pengujian sub sistem terdiri dari beberapa pengujian, diantaranya pengujian rangkaian catu daya, pengujian *buzzer*, pengujian *push button* dan *LCD*, pengujian *display 7segment* untuk *Slave Control*, pengujian *Display 7segment* untuk *Master Control*.

4.4.1. Rangkaian Catu daya

Simulasi antrian loket berbasis atmega8535 dan atmega8 menggunakan rangkaian catu daya yang menghasilkan 5 volt berfungsi sebagai *supply* tegangan seluruh sistem antrian loket. Rangkaian catu daya terdiri dari rangkaian catu daya 5 volt kemudian dihubungkan dihubungkan ke TIP41C agar arus yang dihasilkan lebih besar dari kemampuan IC LM7805 yaitu sebesar 1 A. Sumber tegangan yang digunakan berasal dari trafo CT *step down* 15 volt, kemudian disearahkan oleh dioda *bridge*, sehingga menghasilkan gelombang penuh. Selanjutnya gelombang penuh dari dioda *bridge* akan distabilkan oleh sebuah *filter* melalui kapasitor yang nantinya akan diteruskan ke IC regulator tegangan. Gambar catu daya ditunjukkan pada gambar 4.3.

Gambar 4.3. Catu daya

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rangkaian regulator 5 volt digunakan sebagai *supply* tegangan *master control* dan *slave control*. Tabel 4.4 adalah pengujian *output* tegangan pada catu daya.

Tabel 4.4. *Output* tegangan dari rangkaian regulator 5 volt

	Input	*Output* (teori)	*Output*	*Error*
Regulator 5 volt	11,7 V	5 V	5,005 V	0,11 %

Tabel 4.4 memperlihatkan output tegangan dari catu daya. Pada pengujian ini dilakukan 2 kali. Catu daya 5 volt digunakan sebagai *supply* seluruh sistem pada *master control* dan *slave control*, terdapat error output sebesar 0,11 %. Sesuai dengan *datasheet*, tegangan kerja *output* pada LM7805 adalah 4,8 volt sampai 5,2 volt, sehingga *error* output tidak berpengaruh terhadap kerja IC komparator LM7805, supaya arus yang dihasilkan 1 A maka diberi pendukung TIP41C yang kemampuannya lebih besar dari LM7805 yaitu 1 A. Pada catu daya *error* yang terjadi juga tidak berpengaruh karena masih bisa ditoleransi, karena melalui pengujian, rangkaian ini masih biasa bekerja.

4.4.2. Pengujian *Buzzer*

Rangkaian *buzzer* digunakan sebagai pendeteksi adanya antrian berikutnya pada *master control*. *Buzzer* berfungsi jika tombol *pushbutton* loket ditekan. Gambar 4.4 menunjukkan rangkaian fisik *buzzer*. Tabel 4.5 adalah pengujian *output* tegangan pada catu daya.

Gambar 4.4. Rangkaian *buzzer*

Tabel 4.5. *Output* tegangan dari rangkaian *buzzer*

	Input	*Output* (teori)	*Output*	*Error*
Buzzer	3 V	2,5 V	2,34 V	0,064 %

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada pengujian rangkaian *buzzer*, tombol *pushbutton* loket jika ditekan maka pada *slave control* akan mengirimkan data panggil yang bernilai 1(*high*) ke *master control* yang akan membunyikan *buzzer* bertanda adanya panggilan antrian berikutnya dan hasilnya. Terdapat error output sebesar 0,064 % karena sesuai dengan *datasheet* yaitu tegangan kerja pada *buzzer* sebesar 2,5 V-3,5 V.

4.4.3. Pengujian *Push Button* dan LCD

Pada rangkaian LCD dan *push button* akan bekerja jika *pushbutton* antrian ditekan. Jika *pushbutton* ditekan maka *push button* mengirimkan data 1(*high*) ke *master control* yang akan ditampilkan ke LCD menunjukkan adanya *user*. LCD dan *Push button* ditunjukkan pada gambar 4.5.

Gambar 4.5. Gambar *push button* dan *LCD*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.6. Cacahan antrian untuk *Push Button* dan LCD

Nomor antri-an/*push button*	*Display* LCD	Loket	Hasil
1	Nomor an-trian =1=	1	Bekerja dengan Baik
2	Nomor an-trian =2=	2	Bekerja dengan Baik
3	Nomor an-trian =3=	3	Bekerja dengan Baik

Pengujian *push button* dan LCD bertujuan untuk mengetahui apakah rangkaian ini dapat berfungsi sesuai perancangan. Pengujian pada tabel 4.6 dilakukan dengan menekan tombol *push button* antrian pada *master control* akan berlogika 1(*high*), maka mikrokontroler Atmega8535 akan menganggap ada antrian *user* yang menampilkan printout nomor antrian pada LCD. Tingkat keberhasilan alat dapat dihitung dengan per-samaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{Banyaknya percobaan yang berhasil}}{\text{Jumlah percobaan yang dilakukan}} \times 100\%$$

$$= \frac{3}{3} \times 100\%$$

$$= 100\%$$

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam menampilkan *printout* nomor antrian ke LCD.

4.4.4. Pengujian *Display 7segmen* untuk *Slave Control*

Pada rangkaian display 7*segment* untuk *Slave Control*, metode yang digunakan dalam menyalakan atau menampilkan karakter angka adalah metode *scaning*. *Display* 7*segmen* untuk *slave control* ditunjukkan pada gambar 4.5.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.6. *Display* 7*segment* untuk *Slave Control*

Melalui pengujian dengan melakukan cacahan antrian agar mengetahui adanya kesalahan pada *slave control*. Berikut ini adalah tabel cacahan antrian *display* 7*segmen* untuk *slave control* yang ditunjukkan pada tabel 4.7.

Tabel 4.7. Cacahan antrian display 7segmen untuk *Slave Control*

NO	DISPLAY 7SEGMEN			HASIL
ANTRIAN	LOKET 1	LOKET 2	LOKET 3	
1	1	KOSONG	KOSONG	BEKERJA DENGAN BAIK
2	TERISI	2	KOSONG	BEKERJA DENGAN BAIK
3	ADA	ADA	3	BEKERJA DENGAN BAIK
4	ADA	ADA	4	BEKERJA DENGAN BAIK
5	5	ADA	ADA	BEKERJA DENGAN BAIK
6	ADA	6	ADA	BEKERJA DENGAN BAIK
7	ADA	ADA	7	BEKERJA DENGAN BAIK
8	ADA	8	ADA	BEKERJA DENGAN BAIK
9	ADA	ADA	9	BEKERJA DENGAN BAIK
10	10	ADA	ADA	BEKERJA DENGAN BAIK

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pengujian dilakukan agar sesuai dengan perancangan. Pada tabel 4.7 dilakukan dengan cara menekan tombol pushbutton pada slave control yang terdapat pada masing-masing loket/slave control yang berfungsi mengirim data panggilan antrian berikutnya ke master. Tingkat keberhasilan alat dapat dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{Banyaknya percobaan yang berhasil}}{\text{Jumlah prcobaan yang dilakukan}} \times 100\%$$

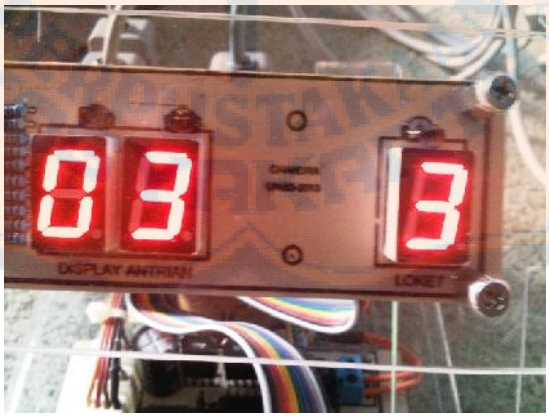
$$= \frac{10}{10} \times 100\%$$

$$= 100\%$$

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam menampilkan *display* 7*segmen* untuk *Slave Control*.

4.4.5. Pengujian Display 7*segmen* untuk *Master Control*

Pada rangkaian 7segmen untuk *master control* menggunakan metode *scanning*. Metode *scanning* yang digunakan yaitu dengan memparalel kaki-kaki pada 7*segment* ke pemberi data antrian pada mikrokontroler. Master control akan menampilkan *Display* 7*segmen* yang ditunjukkan pada gambar 4.7.



Gambar 4.7. *Display* 7*segmen* untuk *master control*

Melalui pengujian dengan melakukan cacahan antrian agar mengetahui adanya kesalahan pada *master control*. Berikut ini adalah tabel cacahan antrian pada *master control* yang ditunjukkan pada tabel 4.8.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.8. *Display* Cacahan antrian untuk *master control*

Nomor antrian	*Display* Antrian	Loket	Hasil
1	1	1	Bekerja dengan Baik
2	2	2	Bekerja dengan Baik
3	3	3	Bekerja dengan Baik

Tingkat keberhasilan alat dapat dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{Banyaknya percobaan yang berhasil}}{\text{Jumlah prcobaan yang dilakukan}} \times 100\%$$

$$= \frac{3}{3} \times 100\%$$

$$= 100\%$$

Simulasi antrian loket berbasis Atmega8535 dan Atmega8 memiliki tingkat keberhasilan 100% dalam menampilkan cacahan antrian *display* 7*segmen* untuk *Master Control*.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB V
KESIMPULAN DAN SARAN

5.1.Kesimpulan

Berdasarkan pengujian dan pengambilan data Simulasi Antrian Loket Berbasis Atmega8535 dan Atmega8, dapat disimpulkan bahwa:

1. Tingkat akurasi sempurna yaitu 100% pada data cacahan antrian dan pencacatan nomor antrian pada sistem keseluruhan.
2. Pada data cacahan antrian dan pencacatan nomor antrian pada *subsistem* memiliki tingkat akurasi sempurna yaitu 100%.
3. Pada *subsistem hardware* catu daya memiliki *error output* sebesar 0,11% sehingga mendekati akurasi sempurna dari *output* teori.
4. *Subsistem hardware buzzer* memiliki *error output* sebesar 0,64% mendekati akurasi sempurna *output* teori.

5.2.Saran

1. Perlunya penambahan *subsistem* printout nomor antrian, agar lebih akurat dalam proses pengambilan nomor antrian.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR PUSTAKA

[1] [http://eprints.undip.ac.id/33779/6/1602_chapter_II.pdf diakses tanggal 29 september 2012

[2] http://repository.usu.ac.id/bitstream/123456789/14072/1/09E02904.pdf diakses tanggal 29 september 2012

[3] http://library.binus.ac.id/eColls/eThesis/Bab2/2010-2-00480-TIAS%20Bab%202.pdf.diakses tanggal 30 september 2012

[4] http://devilzc0de.org/forum/printthread.php?tid=1414 diakses tanggal 30 september 2012

[5] http://www.mytutorialcafe.com/Microcontroller%20Application%20C%207segmen.htm diakses tanggal 30 september 2012

[6] Winoto, Ardi,*Mikrokontroler AVR ATmega8/32/16/8535*, Bandung, 2008

[7] http://agfi.staff.ugm.ac.id/blog/index.php/2009/12/mengenal-system-clock-pada-mikrokontroler-avr/ diakses tanggal 1 oktober 2012

[8] http://elektronika-dasar.com/komponen/led-light-emitting-dioda/ diakses tanggal 1 oktober 2012

[9] http://menanamilmu.blogspot.com/2010/09/interfacing-lcd-dengan-atmega-8535.html diakses tanggal 1 oktober 2012

[10] http://elektronika-dasar.com/komponen/limit-switch-dan-saklar-push-on/ diakses tanggal 1 oktober 2012

[11] http://atmelmikrokontroler.wordpress.com/2009/06/12/light-emiting-dioda-led/ diakses tanggal 1 oktober 2012

[12] http://abisabrina.wordpress.com/2010/08/14/fungsi-dasar-transistor/ diakses tanggal 1 oktober 2012

[13] http://fisikaher.blogspot.com/2010/12/rangkaian-power-supply.html diakses tanggal 4 oktober 2012

[14] https://tipsntrik88.wordpress.com/2009/07/07/simbol-dan-gambar-komponen/ diakses tanggal 22 oktober 2012

[15] http://elektronika-dasar.com/teori-elektronika/lcd-liquid-cristal-display/ diakses tanggal 22 oktober 2012

[16] http://jendelascience.blogspot.com/2010/11/seven-segment.html diakses tanggal 22 oktober 2012

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

[17] http://smkn2-singosari.sch.id/downlot.php?file=rangkaian_penyearah.pdf diakses tanggal 10 november 2012

[18] http://elektronika-dasar.com/teori-elektronika/regulator-tegangan/ diakses tanggal 10 november 2012

[19] http://sharekan.wordpress.com/category/electronics/ diakses tanggal 27 november 2012

[20] http://www.fairchildsemi.com/ds/LM/LM7805.pdf diakses tanggal 14 maret 2013

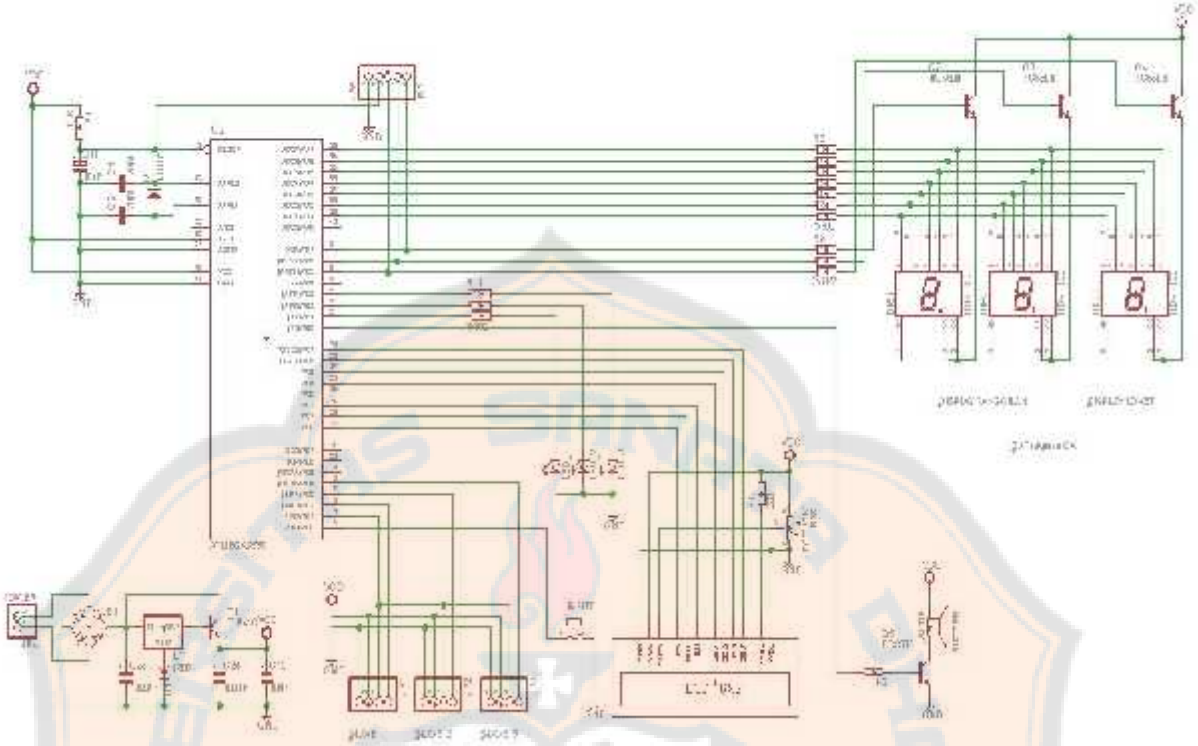
[21] http://elektronika-dasar.web.id/teori-elektronika/konsep-dasar-penyearah-gelombang-rectifier/ diakses tanggal 07 juli 2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

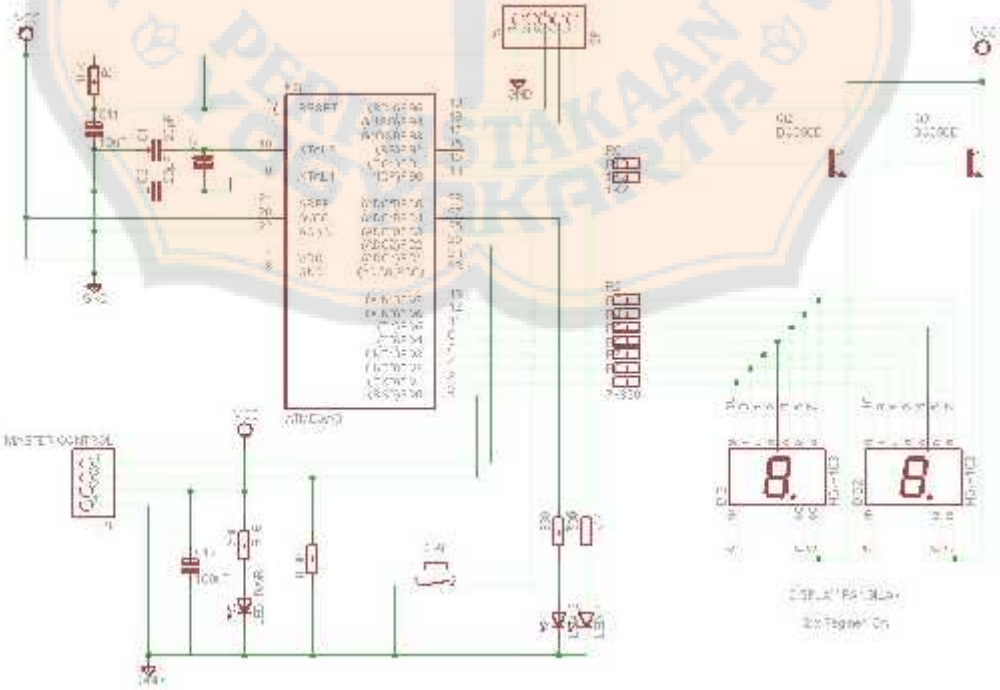
LAMPIRAN

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rangkaian Master Control



Rangkaian Slave Control



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LISTING PROGRAM

Master Control

```
#include <mega8535.h>          // atmega library file
#include <stdio.h>             // standard i/o file
#include <delay.h>             // delay function modul
#include <alcd.h>              // lcd function modul

#define tbl_x   PIND.2         // alokasi port i/o
#define tbl_1   PIND.3
#define tbl_2   PIND.5
#define tbl_3   PIND.4
#define buzzer  PORTB.0

#define total   99             // total jumlah antrian 99

//=================================================================
==============

bit ovl_antri=0,ovl_panggil=0;                  // register bit
unsigned char no_antri=1,no_panggil=1;          // register byte
unsigned char dsp,clk,detik,index_antri=1;      // register byte
unsigned char dsp_1=1,dsp_2=2,dsp_3=3,call=0,loket=0;   // register byte
unsigned char lcd,tick,antri,leds_1,leds_2,leds_3;      // register byte
unsigned char buffer[33];                       // reg_array 33 byte

flash unsigned char segmen[] =                  // data segmen karakter
{
   0b00000011,    // 0
   0b11001111,    // 1
   0b00100101,    // 2
   0b10000101,    // 3
   0b11001001,    // 4
   0b10010001,    // 5
   0b00010001,    // 6
   0b11000011,    // 7
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
      0b00000001,     // 8
      0b10000001,     // 9
      0b11111111      // off
   };

//==================================================================
==============

interrupt [TIM2_COMP] void timer2_comp_isr(void)        // 150 frame/detik
{
   PORTB&=0x1F;
   if ( ++dsp>2 ) dsp=0;
       if ( dsp==0 ) PORTA=segmen[((call%100 )/10)],PORTB.7=1;
   else if ( dsp==1 ) PORTA=segmen[((call%100 )%10)],PORTB.6=1;
   else if ( dsp==2 ) PORTA=segmen[((loket%100)%10)],PORTB.5=1;
}

//==================================================================
==============

interrupt [TIM1_COMPA] void timer1_compa_isr(void)      // setiap 100 mSec
{
   if  ( lcd==0 )                          // batal jika lcd aktif
   {   #asm("sei");                          // kirim data protokol
      printf("*%u%02u%u%02u%u%02u%u\r\n",
      antri,dsp_1,leds_1,dsp_2,leds_2,dsp_3,leds_3 );
   };
}

//==================================================================
==============

void tone1()                                // nada 1
{
   buzzer=1,delay_ms(10),buzzer=0,delay_ms(100);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
}
void tone2()                                // nada 2
{
   tone1(),tone1();
}
void tone3()                                // nada panggil
{
   buzzer=1,delay_ms(100),buzzer=0; delay_ms(50);
   buzzer=1,delay_ms(100),buzzer=0;
}

//=================================================================
==============

void panggil( unsigned char set )                // panggil antrian
{
   if  ( no_panggil==no_antri ) return;          // batal jika sama

       if ( set==1 ) dsp_1 = no_panggil;         // display loket 1
   else if ( set==2 ) dsp_2 = no_panggil;          // display loket 2
   else if ( set==3 ) dsp_3 = no_panggil;          // display loket 3
   call  = no_panggil;                          // display panggilan
   loket = set;                              // display loket n
   tone3();                                 // nada panggil

   if  ( ++no_panggil > total ) no_panggil=1,ovl_panggil=1;    // reset counter
   if  ( no_panggil==no_antri ) antri=0;                    // reset led
   delay_ms(1000);                                  // delay 1 sec
}

//=================================================================
==============
void main(void)
{
   PORTA=0xFF,DDRA=0xFF;                        // Initialisasi Port A
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
   PORTB=0x10,DDRB=0xFF;                    // Initialisasi Port B
   PORTC=0xFF,DDRC=0xFF;                    // Initialisasi Port C
   PORTD=0xFF,DDRD=0xC3;                    // Initialisasi Port D

   TCCR1A=0,TCCR1B=0x0C,TCNT1=0,OCR1A=0x5A0;          // Timer1 14.400kHz,
CTC
   ASSR=0,TCCR2=0x0C,TCNT2=0,OCR2=128,TIMSK=0x90;     // Timer2 57.600kHz,
CTC
   UCSRA=0,UCSRB=0x08,UCSRC=0x06,UBRRH=0,UBRRL=0x17;    // UART Baud
Rate 9600

   lcd_init(16);
   lcd_putsf(" SYSTEM ANTRIAN \nCHANDRA USD-2013");   // display judul
   delay_ms(1000);                          // delay awal
   #asm("sei")

   delay_ms(2000),tone1();
   dsp_1=0,dsp_2=0,dsp_3=0;
   lcd_clear(),lcd_putsf(" AMBIL ANTRIAN? \nsilahkan ditekan");   // start_up


//-----------------------------------------------------------------------------

   while (1)                                // loop utama
   {
      if  ( !tbl_x )                        // tombol antrian = 0
      {
         if  ( ovl_antri==1 && no_antri==(no_panggil-1) )
         {
            lcd_clear();
            lcd_putsf("   MOHON MAAF   \n antrian Penuh! "),tone2();
            while ( !tbl_x ); delay_ms(500),no_antri-=1;
         }
         else
         {   lcd=1;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
            sprintf(buffer,"Nomer antri anda\n    =  %02u  =    ",no_antri);
            lcd=0,lcd_clear(),lcd_puts(buffer),tone1();
            while ( !tbl_x ); delay_ms(1000);
          };

          if  ( ++no_antri > total ) ovl_antri=1,no_antri=1;
          lcd_clear(),lcd_putsf(" AMBIL ANTRIAN? \nsilahkan ditekan");
        };

        if  ( !tbl_1 ) leds_1=1,panggil(1),leds_1=0;   // tombol slave 1 = 0

        if  ( !tbl_2 ) leds_2=1,panggil(2),leds_2=0;   // tombol slave 2 = 0

        if  ( !tbl_3 ) leds_3=1,panggil(3),leds_3=0;   // tombol slave 3 = 0

        if  ( no_panggil!=no_antri ) antri=1;          // led ada antrian

        delay_ms(50);
    };
}

//=================================================================
==============
//
//      Interupt Timer 1    : Set CTC = 0x05A0 = 1440
//                          : (1/14400 kHz) x 1440 = 0.01 detik
//      Frekuensi Data      : 1/0.1 = 10 Hz
//
//      Interupt Timer 2    : (1/57.600 kHz) x 127 = 453.54 Hz
//      Frekuensi Scanning  : 453.54 Hz / 3 segmen = 151.181 Frame / detik
//
// end of file
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Slave Control

```
#include <mega8.h>                    // atmega library file
#include <stdio.h>                    // standard i/o file
#include <delay.h>                    // delay function modul

#define slave   1                     // nomor slave : 1/2/3
#define sw_inp  PINC.3                // alokasi port i/o
#define sw_out  PORTC.2
#define led1    PORTC.4
#define led2    PORTC.5

#define FRAMING_ERROR (1<<FE)         // serial data register
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

//=================================================================
==============

bit tick=0;                           // register data bit
unsigned char respon=0;               // register data byte
unsigned char satuan=10;              // register data byte
unsigned char puluhan=10;             // register data byte
unsigned char f_antrian=0;            // register data byte
unsigned char rx_counter=0;           // register data byte

flash unsigned char segmen[] =        // data segmen karakter
{
   0b00000011,    // 0
   0b11001111,    // 1
   0b00100101,    // 2
   0b10000101,    // 3
   0b11001001,    // 4
   0b10010001,    // 5
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
    0b00010001,     // 6
    0b11000011,     // 7
    0b00000001,     // 8
    0b10000001,     // 9
    0b11111111      // off
  };

//==================================================================
==============

interrupt [USART_RXC] void usart_rx_isr(void)              // seria data-in
{
  char status,data;

  status=UCSRA,data=UDR;
  if  ( (status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0 )
  {
    rx_counter++;                              // counter + 1
    if ( data=='*' ) rx_counter=0;                  // header: reset
    else if ( rx_counter==1 ) f_antrian = (data&0x0F);     // ada antrian ?
    else if ( slave==1 )
    {                                          // slave = 1
        if ( rx_counter==2 ) puluhan = (data&0x0F);    // data puluhan
      else if ( rx_counter==3 ) satuan  = (data&0x0F);    // data satuan
      else if ( rx_counter==4 ) respon  = (data&0x0F);    // respon master
    }
    else if ( slave==2 )
    {                                          // slave = 2
        if ( rx_counter==5 ) puluhan = (data&0x0F);    // data puluhan
      else if ( rx_counter==6 ) satuan  = (data&0x0F);    // data satuan
      else if ( rx_counter==7 ) respon  = (data&0x0F);    // respon master
    }
    else if ( slave==3 )
    {                                          // slave = 3
        if ( rx_counter==8 ) puluhan = (data&0x0F);    // data puluhan
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
        else if ( rx_counter==9 ) satuan  = (data&0x0F);    // data satuan
        else if ( rx_counter==10) respon  = (data&0x0F);    // respon master
      };
   };
}

//=================================================================
==============

interrupt [TIM2_COMP] void timer2_comp_isr(void)    // timer 2 interupt
{
   PORTD=0xFF,PORTB&=0xFC;
   if  ( tick )                          // proses 1
   {
      PORTD = segmen[(puluhan&0x0F)];          // data display puluhan
      PORTB.1 = 1;                          // nyalakan segmen puluhan
      tick = 0;
   }
   else                                  // proses 2
   {
      PORTD = segmen[(satuan&0x0F)];           // data display satuan
      PORTB.0 = 1;                          // nyalakan segmen satuan
      tick = 1;
   };
}

//=================================================================
==============

void main(void)
{
   PORTB=0x38,DDRB=0x3F;                    // Initialisasi Port B
   PORTC=0x0F,DDRC=0x37;                    // Initialisasi Port C
   PORTD=0xFF,DDRD=0xFE;                     // Initialisasi Port D
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
   ASSR=0,TCCR2=0x0C,TCNT2=0;                    // Timer 2 57.600 kHz
   OCR2=127,TIMSK=0x80;                          // Interupt 2 Mode CTC
   UCSRA=0,UCSRB=0x90,UCSRC=0x86;                // UART Mode receive only
   UBRRH=0,UBRRL=0x17,delay_ms(100);             // UART Baud Rate: 9600
   #asm("sei")                                   // Aktifasi sistem interupt

//-----------------------------------------------------------------------------

   while (1)                                     // loop utama
   {
      if  ( sw_inp==0 ) led2=1,sw_out=0;         // switch = 0, kirim 0
      if  ( respon==1 ) led2=0,sw_out=1;         // respon dari master = 1
      if  ( f_antrian==1 ) led1=1; else led1=0;  // ada antrian = 1 > led1 on
      delay_ms(50);                              // delay 50 mili detik
   };
}

//=================================================================
==============

//     Interupt timer2     : (1/57.600 kHz) x 127 = 453.54 Hz
//     Frekuensi Scanning  : 453.54 Hz / 2 segmen = 226.77 Frame / detik
//
// end of file
```