

PERANCANGAN *SOFTWARE* UNTUK TAMPILAN

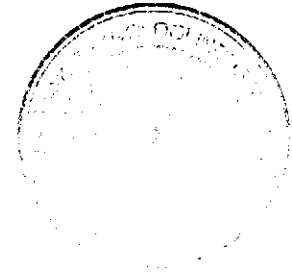
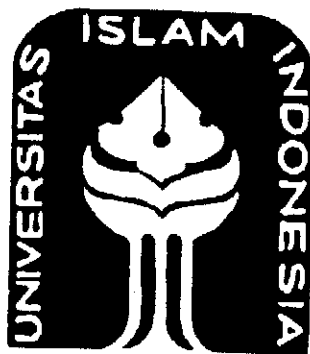
ANIMASI PADA SUSUNAN *DOT MATRIX* 8x8

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Teknik Elektro

No INV	645 / FTI / TE / 2009
TANGGAL	19 Mei 2009
ASAL	Beasiswa / Hadiah
HARGA	Rp -
PERPUSTAKAAN FAK. TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA YOGYAKARTA	



Disusun oleh :

Nama : Randi Pratama Putra

No.Mahasiswa : 04524020

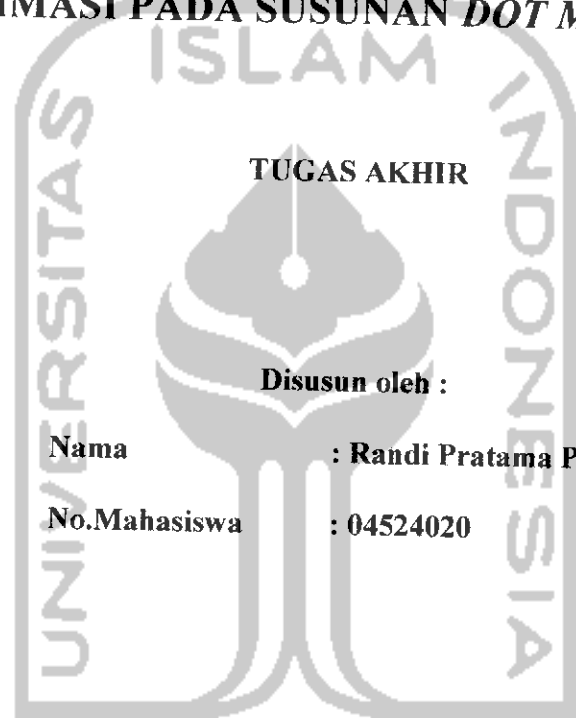
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA

2009

LEMBAR PENGESAHAN PEMBIMBING

PERANCANGAN *SOFTWARE* UNTUK TAMPILAN

ANIMASI PADA SUSUNAN *DOT MATRIX* 8x8



TUGAS AKHIR

Disusun oleh :

Nama : Randi Pratama Putra

No.Mahasiswa : 04524020

Yogyakarta, 2009

Pembimbing I

Pembimbing II

Wahyudi Budi Pramono, ST, M.Eng.

Medilla Kusriyanto, ST.

LEMBAR PENGESAHAN PENGUJI
PERANCANGAN *SOFTWARE* UNTUK TAMPILAN
ANIMASI PADA SUSUNAN *DOT MATRIX* 8X8
TUGAS AKHIR

oleh:

Nama : Randi Pratama Putra

No. Mahasiswa : 04524020

**Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia**

Yogyakarta, April 2009

Tim Penguji,

Wahyudi Budi Pramono, ST, M.Eng.
Ketua



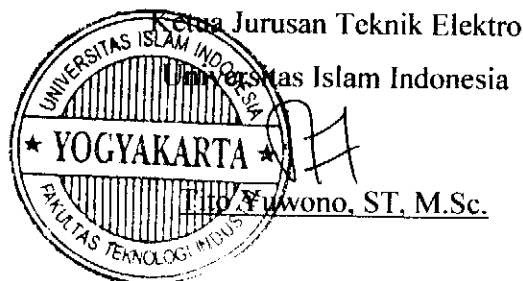
Tito Yuwono, ST, M.Sc.
Anggota I



Dwi Ana Ratna Wati, ST, M.Eng.
Anggota II



Mengetahui,



HALAMAN PERSEMBAHAN

Tugas akhir ini Ananda persembahkan Kepada :

Ayahanda dan Ibunda tercinta

Atas segalanya yang telah diberikan kepadaku

*Seluruh keluarga besarku yang telah memberikan
kepercayaan, semangat, tuntunan dan do'a
untukku*

Teman-temanku yang selalu baik hati kepadaku

MOTTO

"Sesungguhnya aku hanya diberi harta karena ilmu yang ada padaku"

(QS Al-Qashas:78)

"Barangsiapa yang menginginkan dunia, (mendapatkannya) harus memakai ilmu. Barang siapa yang menginginkan akhirat, (mendapatkannya) harus memakai ilmu. Barangsiapa yang menginginkan dunia dan akhirat (mendapatkannya pun) harus memakai ilmu."

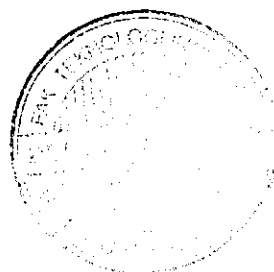
(Sabda Rasulullah SAW)

"Keridhaan Tuhan berada pada keridhaan kedua orang tua, dan kemurkaan Tuhan berada pada kemurkaan orang tua."

(HR Al-Turmudzi)

"Sebaik-baik manusia adalah orang yang banyak manfaatnya (kebaikannya) kepada manusia lainnya."

(H.R. Qadla'ie dari Jabir)



KATA PENGANTAR



Assalamu'alaikum Wr. Wb

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufiq dan hidayah-Nya, karena ridho-Nya penyusun dapat menyelesaikan penyusunan tugas akhir (TA) dengan judul **“Perancangan Software Untuk Tampilan Animasi Pada Susunan Dot Matrik 8x8 ”** dan tidak lupa juga dipanjatkan shalawat serta salam pada junjungan Nabi besar Muhammad SAW beserta keluarga dan pengikutnya sampai akhir zaman.

Adapun maksud dari penyusunan tugas akhir ini adalah untuk memenuhi kurikulum S-1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. disamping itu bertujuan untuk menambah pengetahuan terhadap ilmu yang dipelajari di bangku perkuliahan untuk diterapkan pada aplikasi sesungguhnya.

Dalam menyusun laporan tugas akhir ini tidak terlepas dari berbagai pihak yang memberikan bantuan dan dukungan. Untuk itu pada kesempatan ini penyusun mengucapkan banyak terima kasih yang sebesar-besarnya kepada :

1. Ibunda Hj. Asnelli dan Ayahanda Zailir yang senantiasa memberikan dukungan semangat, perhatian, moril, materil dan do'a setiap saat.
2. Bapak Fathul Wahid, ST, M.Sc, selaku Dekan Fakultas Teknik Industri

3. Bapak Tito Yuwono, ST, M.Sc, selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Wahyudi Budi Pramono, ST, selaku Dosen Pembimbing I yang telah memberikan arahan, masukan serta bimbingan sehingga penyusun dapat menyelesaikan penyusunan tugas akhir dengan baik.
5. Bapak Yusuf Aziz Amrullah, ST, selaku Dosen Pembimbing II yang telah memberikan saran-saran, kritik serta bimbingan sehingga penyusun dapat menyelesaikan penyusunan tugas akhir dengan baik.
6. Bapak Medilla Kusriyanto, ST, selaku Dosen Pengganti Pembimbing II yang bersedia menggantikan posisi Dosen Pembimbing II sebelumnya dalam memberikan arahan dan masukan untuk menyempurnakan penyusunan tugas akhir dengan baik.
7. Seluruh keluarga besarku yang selalu mendoakan dan memberikan dukungan dalam penyusunan tugas akhir.
8. Wahyu Edi Wibowo, Mas Fajar (Bejo) & Mas Galih Mustiko Aji yang telah berbaik hati membantu serta meluangkan waktu dan ilmunya.
9. Mas Agung selaku *Staff Lab.* IMEL dan Mas Heri selaku *Staff Lab.* Kendali dan Mikroprosesor yang telah berbaik hati meluangkan waktu dan ilmunya.
10. Musyafius Musyafaah yang selama ini menjadi penyemangat, pemberi warna dalam hidupku dan selalu menemaniku dalam menyelesaikan tugas akhir.
11. Dosen dan karyawan Fakultas Teknologi Industri, Ka.Lab Jurusan Teknik Elektro beserta para asisten lab. atas waktu, tempat dan ilmu yang diberikan.

12. Seluruh pihak yang tidak dapat disebutkan satu-persatu, yang telah memberikan *support* dan do'a.

Penyusun telah berupaya yang terbaik dalam menyusun tugas akhir ini, namun penyusun menyadari bahwa penulisan laporan ini tidak luput dari kekurangan, maka kritik dan saran yang konstruktif dari semua pihak sangat diperlukan untuk penulisan laporan yang selanjutnya dan penyusun terima dengan sepenuh hati sebagai bahan untuk peningkatan kemampuan dan keterampilan penyusun di lain kesempatan.

Akhirnya penyusun berharap semoga laporan ini dapat bermanfaat dan berguna bagi penyusun dan pembaca serta menjadikan amal ibadah yang diterima di sisi-Nya. Amin.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Maret 2009

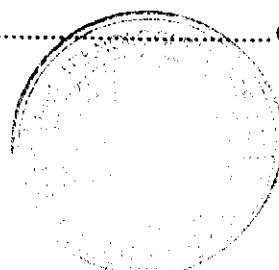
Penyusun

ABSTRAK

Suatu penampil informasi dalam bentuk sebuah *display* menggunakan *dot matrix* untuk menampilkan animasi bergambar akan menjadi salah satu alternatif pilihan dalam dunia periklanan saat ini untuk mempromosikan sesuatu, baik itu berupa iklan layanan masyarakat, pesan bergambar, iklan produk-produk industri ataupun informasi lainnya yang akan dipublikasikan di tengah-tengah masyarakat. Di samping itu, harga untuk sebuah papan *dot matrix display* berukuran besar jauh lebih murah jika dibandingkan dengan sebuah papan *display LCD*. Selain penggunaannya lebih praktis, *dot matrix display* ini lebih efisien bila hanya digunakan untuk sebuah penampil informasi layanan periklanan. Perancangan perangkat keras *dot matrix display* ini menggunakan beberapa buah *dot matrix 8x8* yang dirangkai sebanyak 7 baris dan 15 kolom yang dikendalikan oleh sebuah mikrokontroler AT89S52 untuk setiap barisnya. Sedangkan proses untuk menampilkan animasi gambarnya dilakukan secara pemrograman pada antarmuka perangkat lunak menggunakan *visual basic 6.0*. proses pembentukan sebuah animasi gambar dilakukan dengan cara mengirimkan *frame-frame* gambar yang membentuk sebuah pola gerak suatu gambar animasi yang telah dirancang sebelumnya secara satu per satu. Data yang diolah pada perangkat lunak hanya mengerti tipe data ASCII sedangkan data yang akan dibaca pada mikrokontrolernya adalah data bilangan biner. pengiriman data dilakukan secara *serial* sehingga untuk pengkonversian data ASCII ke bentuk bilangan biner dilakukan di *port serialnya* dengan menggunakan kontrol MSComm. Penyalaan *dot matrix* sesuai dengan input data (*frame* gambar) yang dikirim dari perangkat lunak yang dilakukan berdasarkan proses sistem *scanning* gambar (menguraikan gambar). Lama durasi animasi gambar tergantung seberapa banyak *frame* gambar yang dikirim dan dalam perancangan perangkat lunak menggunakan *visual basic* ini, jumlah *frame* gambar tidak dibatasi.

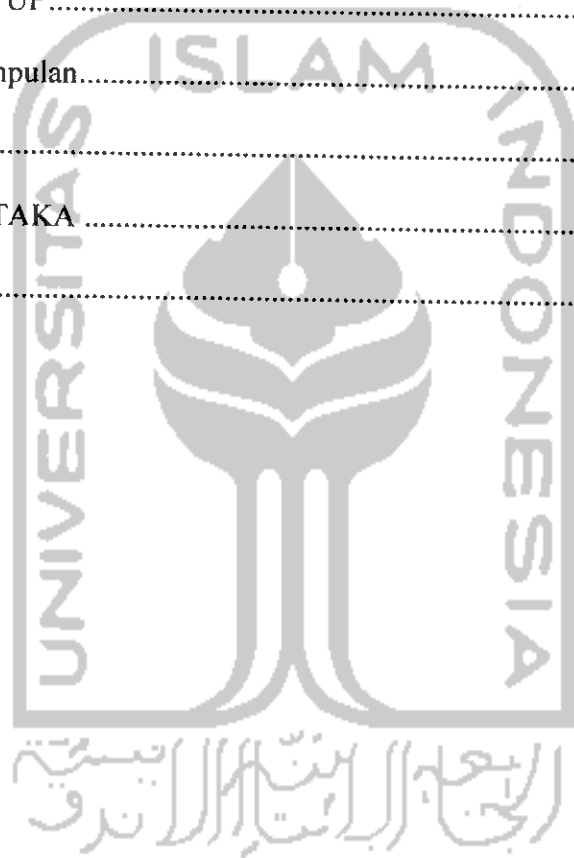
DAFTAR ISI

HALAMAN JUDUL.....	I
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	II
HALAMAN PENGESAHAN DOSEN PENGUJI.....	III
HALAMAN PERSEMBAHAN.....	IV
MOTTO	V
KATA PENGANTAR.....	VI
ABSTRAK	IX
DAFTAR ISI.....	X
DAFTAR GAMBAR.....	XIII
DAFTAR TABEL.....	XV
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Sistematika Penulisan.....	4
BAB II STUDI PUSTAKA.....	6
2.1 Tinjauan Pustaka.....	6



2.2	Rencana Penelitian	10
BAB III PERANCANGAN SISTEM		12
3.1	Perancangan Sistem Perangkat Lunak (<i>Software</i>)	12
3.1.1	Membuat Tampilan/Antarmuka	13
3.1.2	Pengaturan Properti Kontrol.....	14
3.1.3	Menulis Kode	17
3.1.3.1	Prosedur Private Sub Form_Load ().....	17
3.1.3.2	Prosedur Private Sub Command1_Click ().....	19
3.1.3.3	Prosedur Private Sub Command2_Click ().....	21
3.1.3.4	Prosedur Private Sub Command3_Click.....	23
3.1.3.5	Prosedur Private Sub Command4_Click ().....	24
3.1.3.6	Prosedur Private Sub Timer1_Timer ().....	26
3.2	Pengaturan Properti <i>Serial Port</i> Pada PC (<i>Personal Computer</i>).....	31
3.3	Rancangan Perangkat Keras (<i>Hardware</i>).....	32
BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN SISTEM.....		35
4.1	Pengujian <i>User Interface</i> /Antarmuka Pada Perangkat Lunak (<i>Software</i>)...35	
4.1.1	Pengujian Tombol <i>Command1</i> (“Kirim Gambar”).....	35
4.1.2	Pengujian tombol <i>command2</i> (“Mainkan” dan “Berhenti”).....	37
4.1.3	Pengujian tombol <i>command3</i> (“Hapus Gambar”).....	39
4.1.4	Pengujian tombol <i>command4</i> (“Tutup”)	40

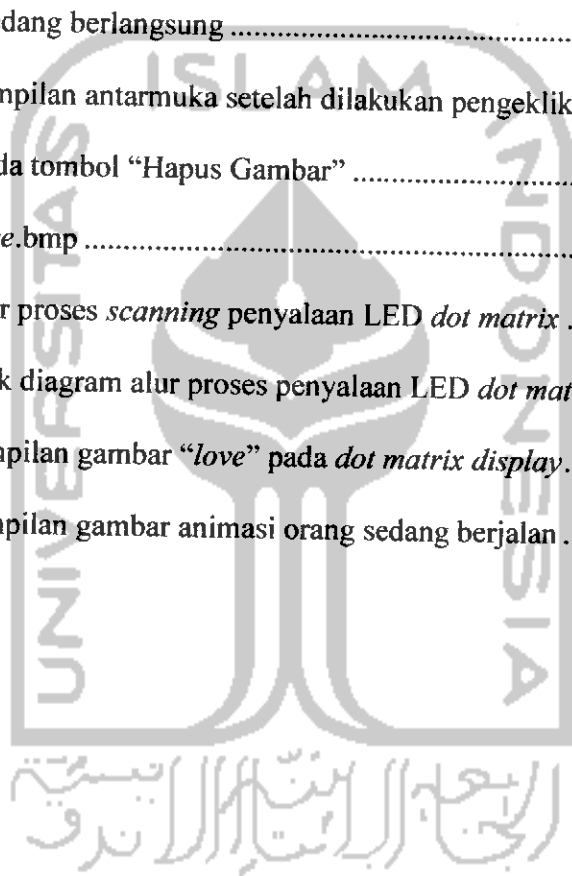
4.2	Pengujian Perangkat Lunak Terhadap Perangkat Keras <i>Dot Matrix Display</i>	40
4.2.1	Pengujian Nilai <i>Baudrate</i>	40
4.2.2	Pengujian Pengiriman <i>Frame</i> gambar	43
BAB V PENUTUP		49
5.1	Kesimpulan	49
5.2	Saran	50
DAFTAR PUSTAKA		51
LAMPIRAN		53



DAFTAR GAMBAR

Gambar 3.1 Blok diagram perancangan perangkat lunak	12
Gambar 3.2 Rancangan tampilan program kendali LED <i>display dot matrix</i>	13
Gambar 3.3 Rancangan tampilan program kendali LED <i>display dot matrix</i> setelah pengaturan properti	17
Gambar 3.4 Diagram alir proses inialisasi <i>port MSCComm1</i>	18
Gambar 3.5 Indikator kesalahan pemakaian COM	19
Gambar 3.6 Diagram alir aksi yang dilakukan tombol “Kirim Gambar” saat diklik..	20
Gambar 3.7 Diagram alir aksi yang terjadi saat tombol “Mainkan” diklik.....	22
Gambar 3.8 Diagram alir aksi yang terjadi saat tombol “Hapus Gambar” diklik.....	23
Gambar 3.9 Diagram alir aksi yang terjadi saat tombol “Tutup” diklik	25
Gambar 3.10 Diagram alir aksi yang terjadi saat <i>timer1</i> aktif pada mikrokontroler “A”	27
Gambar 3.11 Pengaturan properti <i>communication port</i> pada PC (Personal Computer)	31
Gambar 3.12 Tampilan pengaturan COM <i>port number</i> pada PC (Personal Computer)	32
Gambar 3.13 Skema rangkaian <i>dot matrix display</i> untuk satu rangkaian sistem minimum dan beberapa buah <i>dot matrix</i>	33

Gambar 3.14 Blok diagram rancangan <i>dot matrix display</i>	34
Gambar 4.1 Tampilan antarmuka saat tombol “Kirim Gambar” diklik	36
Gambar 4.2 Tampilan antarmuka ketika <i>frame</i> gambar telah dikirim ke <i>listbox</i>	37
Gambar 4.3 Tampilan antarmuka saat proses penganimasian gambar sedang berlangsung	38
Gambar 4.4 Tampilan antarmuka setelah dilakukan pengeklikkan pada tombol “Hapus Gambar”	39
Gambar 4.5 <i>Love.bmp</i>	44
Gambar 4.6 Alur proses <i>scanning</i> penyalaaan LED <i>dot matrix</i>	44
Gambar 4.7 Blok diagram alur proses penyalaaan LED <i>dot matrix</i>	46
Gambar 4.8 Tampilan gambar “ <i>love</i> ” pada <i>dot matrix display</i>	41
Gambar 4.9 Tampilan gambar animasi orang sedang berjalan	42



DAFTAR TABEL

Tabel 3.1 Pengaturan properti kontrol.....	14
Tabel 4.1 Konfigurasi <i>baudrate</i> dalam bahasa <i>assembly</i>	41
Tabel 4.2 Uji coba pengaruh peningkatan nilai <i>baudrate</i> terhadap LED <i>dot matrix</i> ..	42
Tabel 4.3 Uji coba pengaruh pengubahan nilai waktu tunda.....	43
Tabel 4.4 Proses pengolahan data dari bilangan biner ke bentuk bilangan ASCII.....	45



BAB I

PENDAHULUAN

1.1 Latar Belakang

Manfaat dari perkembangan teknologi saat ini sangat dapat dirasakan aplikasinya dalam semua bidang kehidupan, khususnya penerapan teknologi di dunia periklanan dan hiburan, banyak sekali media-media canggih yang digunakan hanya untuk menyampaikan atau mempromosikan sesuatu ke masyarakat diantaranya seperti produk bernilai komersil, info layanan publik, tampilan animasi hiburan dan lain sebagainya. Media-media canggih tersebut tidak terlepas dari penerapan piranti elektronika terprogram dalam bentuk sebuah *display* atau penampil.

Salah satu jenis *display* yang sering digunakan untuk keperluan periklanan dan tata rias kota adalah *LED dot matrix display*, *LED dot matrix display* ini merupakan susunan dari sejumlah LED (*Light Emitting Dioda*) yang tersusun membentuk jejeran beberapa baris dan kolom. Jika dibandingkan dengan LCD (*Liquid Crystal Display*), biaya yang dibutuhkan untuk membuat sebuah *display dot matrik* berukuran besar jauh lebih murah dan lebih efisien apabila hanya diperuntukkan sebagai papan iklan atau informasi. *LED dot matrix display* dapat diprogram dengan sentuhan animasi (gambar gerak) layaknya sebuah LCD, salah satu bahasa pemrograman yang mampu memprogram pengendalian *LED dot matrix*

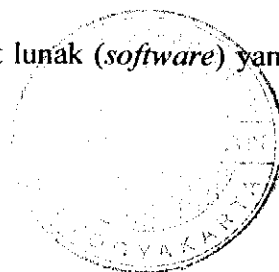
display ini adalah bahasa pemrograman VB (*visual basic*), dengan VB dapat dirancang sebuah tampilan pengendalian *LED dot matrix display* yang mampu mengubah input data yang bersifat statis yaitu objek (gambar atau karakter) diam menjadi input data yang bersifat dinamis yaitu objek (gambar atau karakter) dengan animasi (gerak).

Bahasa pemrograman *visual basic* memiliki beberapa kelebihan jika dibandingkan dengan menggunakan bahasa pemrograman lain, salah satunya yaitu untuk membuat sebuah aplikasi *visual basic* dengan langkah-langkah seperti membuat tampilan atau antarmuka dengan menggunakan *activeX control* seperti *textbox*, *command button*, dan sebagainya dalam sebuah form, sedikit pengaturan pada properti masing-masing kontrol dan selanjutnya penulisan kode-kode, hanya membutuhkan waktu yang relatif singkat tanpa banyak mengalami kompleksitas.

Atas berbagai pertimbangan di atas, maka dirancanglah perangkat lunak untuk memprogram dan mengendalikan tampilan sebuah *LED dot matrix display* dengan sentuhan animasi menggunakan bahasa pemrograman *visual basic 6.0* (VB 6.0). *LED dot matrix display* yang akan diprogram adalah *LED dot matrix display* yang berdimensi 56 baris dan 120 kolom dengan 1 warna (*monochrome*) berbasis mikrokontroler ATMEL AT89S52 dengan pengiriman data menggunakan *serial port* (DB9).

1.2 Rumusan Masalah

Dengan melihat latar belakang di atas, maka dapat dirumuskan permasalahan yaitu “Bagaimana merancang sebuah perangkat lunak (*software*) yang



mampu memprogram dan mengendalikan tampilan sebuah LED *dot matrix display* 1 warna (*monochrome*) yang berdimensi 56 baris dan 120 kolom dengan sentuhan animasi”.

1.3 Batasan Masalah

Pembahasan tentang perancangan perangkat lunak ini mempunyai cakupan yang luas, maka dari itu untuk membatasi masalah agar tidak meluas kepermasalahan lain dan lebih terarah sebagaimana tujuan, penulis membatasi penelitian agar diperoleh suatu solusi yang diharapkan. Batasan-batasan yang dimaksud antara lain:

1. Perancangan tampilan/antarmuka yang terdiri atas kontrol pengiriman input *data/frame* gambar, kontrol jalankan animasi, kontrol hentikan animasi, kontrol tutup aplikasi, *listbox* input *data/frame* gambar dan *picturebox* simulasi animasi.
2. Pengiriman data dari PC ke *hardware* menggunakan *serial port* dengan spesifikasi *boudrate* 9600 bps, *no parity*, *stop bit* 1 dan data dikirim per 8 bit. Jenis format *data/frame* gambar yang di kirim adalah *bitmap*.
3. Mikrokontroler yang akan diprogram adalah jenis ATMEL AT89S52
4. Dimensi LED *dot matrix display* yang digunakan adalah 56 baris dan 120 kolom dengan jenis *dot matrix* ukuran 8x8 1 warna (*monochrome*)

5. Bahasa pemrograman yang digunakan adalah VB 6.0 *enterprise edition*.

1.4 Tujuan Penelitian

Tujuan dari pembuatan tugas akhir ini adalah untuk membuat program aplikasi menggunakan bahasa pemrograman *visual basic 6.0* yang dapat menampilkan gambar animasi pada *LED dot matrix display*, dengan demikian diharapkan tampilan animasi pada *LED dot matrix display* ini dapat membuat menjadi lebih menarik sebuah papan *display*.

1.5 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir dengan judul “Perancangan *Software* Untuk Tampilan Animasi Pada Susunan *Dot Matrix 8x8*” terdiri dari 5 bab, dengan masing-masing bab adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang masalah yang akan diteliti, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

BAB II STUDI PUSTAKA

Bab ini memuat pembahasan tentang penelitian-penelitian sejenis yang sudah pernah dilakukan dan menjadi acuan sebagai pembanding terhadap rencana penelitian yang akan dirancang.

BAB III PERANCANGAN SISTEM

Bab ini menjelaskan perancangan sistem yang digunakan, cara mengimplementasikan rancangan.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN SISTEM

Bab ini membahas perihal hasil-hasil pengujian yang diperoleh dari uji coba perakitan.

BAB V PENUTUP

Bab ini memuat kesimpulan dan saran-saran dari proses perancangan yang telah dilakukan.



BAB II

STUDI PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan dari penelitian yang sudah ada diperlukan untuk dilakukannya suatu pengembangan dan inovasi terhadap unjuk kerja sistem yang pernah dirancang sebelumnya dengan perancangan sistem yang akan dilakukan pada penelitian tugas akhir kali ini. Penelitian-penelitian yang sudah pernah dilakukan tersebut diantaranya yaitu:

Hari Satriyo Basuki, 2006, Telah melakukan penelitian tentang perancangan dan implementasi penampil nama stasiun untuk kereta api lokal berbasis mikrokontroler dengan *dot matrix display*. Dalam aplikasinya, penampil nama stasiun ini dibuat sesederhana mungkin. Akan tetapi, alat ini cukup kuat untuk bertahan dari keadaan yang mengancam peralatan bisa jadi rusak seperti guncangan, suhu yang cukup tinggi, debu yang banyak serta tegangan yang kurang stabil. Dalam menampilkannya ke papan *dot matrix display*, operator hanya tinggal menekan tombol *keypad* yang telah ditentukan untuk menampilkan nama stasiun yang dikehendaki untuk ditampilkan.

Proses dalam menampilkan data ke *dot matrix display* adalah pemanfaatan fungsi *keypad* sebagai tombol perintah untuk menampilkan karakter ke papan *display*

tersebut. Sebagaimana yang diketahui bahwa *keypad* adalah suatu alat yang berupa kumpulan tombol-tombol yang dapat memberikan logika-logika digital yang akan diproses oleh mikrokontroler. Mikrokontroler yang digunakan adalah seri AT89C51 dan dalam perancangannya setiap tombol mewakili satu nama stasiun sehingga tombol tersebut dipergunakan untuk memilih nama stasiun yang dikehendaki untuk ditampilkan.

Adisatya Pramardianto, 2007, Telah melakukan penelitian tentang penampil informasi *dot matrix* dengan animasi terprogram berbasis mikrokontroler AT89C52. Perancangan sistem penampil informasi *dot matrix* ini diperuntukkan sebagai media informasi di tempat-tempat keramaian seperti swalayan, stasiun, bandara dan lain sebagainya. Dalam perancangannya menggunakan beberapa komponen utama yaitu mikrokontroler AT89C52, EEPROM AT24C64, demultiplexer 74LS138, transistor, *dot matrix* 10 buah dan rangkaian catu daya. Mikrokontroler AT89C52 berfungsi untuk mengendalikan kerja alat, EEPROM AT24C64 digunakan untuk menyimpan karakter, transistor digunakan untuk *scanning* baris, untuk *scanning* kolom digunakan demultiplexer 74LS138. Untuk menjalankan alat ini, pertama-tama karakter yang akan ditampilkan pada *dot matrix* diketik terlebih dahulu lewat komputer, lalu dikirim ke mikrokontroler. Karakter-karakter tersebut disimpan di memori EEPROM. Setelah seluruh karakter yang akan ditampilkan disimpan di memori EEPROM, lalu karakter tersebut ditampilkan satu persatu pada *dot matrix*. Ada dua macam animasi yang dapat ditampilkan pada *dot*

matrix yaitu karakter berjalan dari kanan ke kiri serta normal dan berkedip. Alat ini mampu menampilkan karakter berjalan dengan batas maksimal sampai 1000 karakter.

Sistem penampil informasi dalam menampilkan karakter dilandaskan atas dasar pemanfaatan fungsi IC memori SEEPROM (*Serial Erasable Electrically Programmable Read Only Memory*) AT24C64 sebagai tempat menyimpan data (karakter) yang siap dipanggil ketika akan ditampilkan pada alat penampil informasi yang menggunakan *dot matrix* ini. Dimana IC SEEPROM dalam melakukan penghapusan data menggunakan sinyal-sinyal listrik dan untuk menyimpan atau mengambil data dari IC memori ini hanya membutuhkan jalur dua bit dari mikrokontroler. Karena prosesnya menggunakan sistem serial, dua bit ini dipergunakan untuk jalur sinyal data dan sinyal *clock*. Untuk menstabilkan perubahan kondisi logika dari sinyal data dan sinyal *clock* dari IC SEEPROM ini dibutuhkan resistor eksternal *pull up*.

Jatmiko Heru Supriyono, 2008, telah melakukan penelitian tentang Pengembangan tulisan berjalan (*running text*) pada *dot matrix* dengan pengisian karakter berbasis layanan *Short Message Service* (SMS) jaringan GSM. Penelitian ini bertujuan untuk merancang dan mengimplementasikan fasilitas SMS sebagai media transmisi data karakter yang akan ditampilkan dalam penampil *dot matrix*. Metode penelitian yang dipakai adalah dengan cara pengembangan sistem secara keseluruhan dengan menggunakan mikrokontroler keluarga MCS 51 dan telpon seluler *ericsson* T230. Komunikasi antara telpon seluler dengan mikrokontroler dilakukan secara serial. *Dot matrix* yang digunakan berjumlah 10 buah *dot matrix* 8x8 dimana karakter

atau tulisan yang dikirim oleh *handphone* dapat tertampil pada *dot matrix display*. Komunikasi serial antara mikrokontroler dan *handphone* berhasil dilakukan dan diuji menggunakan fasilitas *hyperterminal* pada PC. Penampil *dot matrix* dapat menampilkan karakter yang dimasukkan melalui *handphone* pengirim hanya dengan satu jenis animasi dengan panjang karakter maksimal 13 karakter yang dapat berupa huruf, angka, tanda baca maupun karakter khusus yang ada di *handphone*. Waktu yang dibutuhkan oleh sistem secara keseluruhan untuk menampilkan karakter mulai dari mendapatkan SMS adalah sekitar 9 detik. Oleh karena itu SMS yang masuk dalam waktu kurang dari 9 detik setelah *handphone* penerima menerima SMS tidak dapat diproses (diabaikan) sehingga tidak dapat ditampilkan pada *dot matrix*.

Sistem pengiriman karakter ke *dot matrix* lewat SMS ini adalah karakter yang akan ditampilkan (baik huruf maupun angka) dituliskan terlebih dahulu pada telpon genggam pengirim kemudian dikirimkan ke telpon genggam penerima. Selanjutnya isi dari SMS yang diterima oleh telpon genggam penerima akan dikirimkan ke rangkaian pengendali berbasis mikrokontroler AT89S52 yang kemudian akan mengecek isi SMS tersebut dan ditampilkan ke dalam penampil *dot matrix*. Disini komunikasi antara telpon genggam dan rangkaian pengendali akan dilakukan secara serial standar RS 232 dengan piranti IC MAX 232. telpon genggam yang digunakan pada penerima adalah *Ericsson T230*.

Wahyu Widayanto, 2008, telah melakukan penelitian tentang pengendalian *running text* dengan SMS menggunakan mikrokontroler AT89S52. Tujuan penelitian ini adalah bagaimana mengaplikasikan teknologi SMS sebagai

pengendali jarak jauh dalam menampilkan karakter berjalan pada *dot matrix display* yang lebih praktis dan canggih. Perancangan sistem ini dikendalikan oleh sebuah *chip* mikrokontroler AT89S52 yang akan mengolah masukan data dari *handphone* yang berupa data PDU (*Protocol Data Unit*). Data PDU tersebut diolah dan dikonversi lagi ke tabel penyalan dot matrik.

Perancangan perangkat keras ini dibangun oleh beberapa rangkaian seperti rangkaian pengendali utama yaitu rangkaian mikrokontroler AT89S52 dengan *oscillator* yang digunakan adalah 11,0592 MHz, ELCO dan *resistor* sebagai rangkaian *reset*, rangkaian MAX 232 yang terdiri dari IC MAX 232 dan beberapa ELCO sebagai penghubung antara mikrokontroler dengan *handphone* sehingga dapat terkoneksi. Selanjutnya, rangkaian *dot matrix* 5x7 sebanyak 8 buah sebagai penampil tulisannya dilengkapi dengan 5 buah IC TTL 74LS164 sebagai *driver* penggeser data yang masuk ke kaki kolom data *dot matrix* sehingga tulisan akan bergeser.

2.2 Rencana Penelitian

Setelah melakukan peninjauan terhadap penelitian yang sudah ada diatas, yaitu bagaimana cara memprogram sebuah penampil *dot matrix* yang merupakan susunan dari beberapa *dot matrix* yang dirangkai 1 baris dan beberapa kolom untuk menampilkan karakter dengan dua pilihan animasi yaitu karakter berjalan dan karakter berkedip, maka dapat diketahui bahwa perancangan perangkat lunak dengan menggunakan bahasa *visual basic* 6.0 untuk menampilkan animasi pada *dot matrix display* berdimensi lebih besar yaitu 56x120 *pixel* (susunan *dot matrix* berukuran 8x8

yang dirangkai 7 baris dan 15 kolom) lewat sebuah antarmuka jauh lebih inovatif, karena pada penelitian tugas akhir yang akan dirancang ini lebih menonjolkan sisi penganimasian gambar dari pada karakter yang memiliki batasan jumlah maksimal karakter yang akan ditampilkan dan ruang gerak penganimasian sangat sempit. Tujuan dari penelitian ini adalah bagaimana memprogram sebuah papan *dot matrix display* berukuran besar untuk ditampilkan sebuah data bergambar dengan gerak animasi yang tidak terbatas.

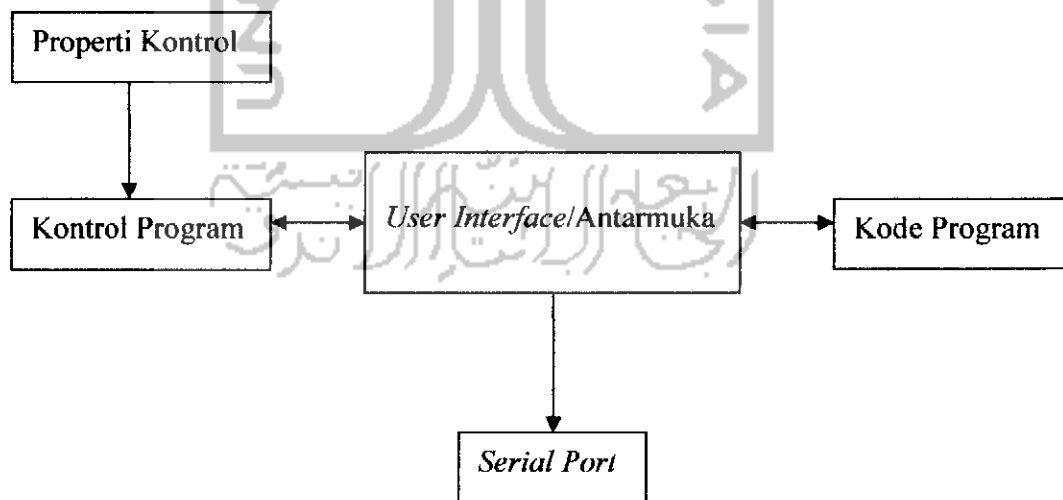
Perancangan perangkat lunak pada *dot matrix display* berdimensi 56x120 *pixel* ini menggunakan *serial port* sebagai jalur komunikasi dari PC (*Personal Computer*) ke perangkat keras *dot matrix display* dan sebagai kontrol pengiriman datanya menggunakan *MSComm*. Komunikasi antara perangkat lunak *visual basic* dengan perangkat keras *dot matrix display* dibangun berdasarkan sinkronisasi antara data yang diolah di *visual basic* dengan program *assembly* yang diisikan di mikrokontrolernya.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang dirancang adalah untuk memprogram pengendalian LED *dot matrix display* dengan menggunakan bahasa pemrograman *visual basic 6.0*, mempunyai tiga tahap pengerjaan utama, diantaranya adalah membuat tampilan/antarmuka, pengaturan properti kontrol program dan menuliskan kode program. Blok diagram sistem rancangan perangkat lunak ini disajikan dalam gambar berikut:

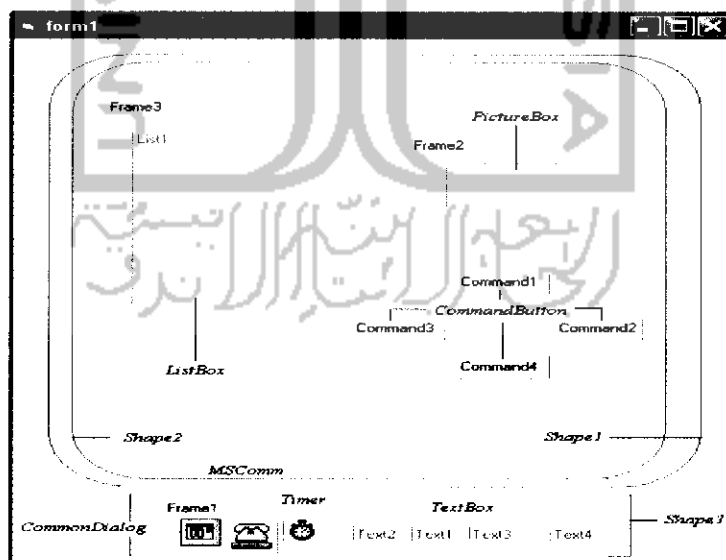


Gambar 3.1 Blok diagram perancangan perangkat lunak

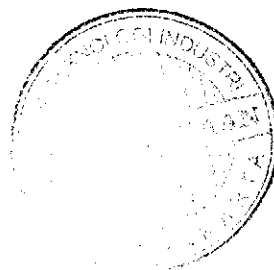
Blok diagram pada gambar 3.1 menggambarkan tentang bagian-bagian dari tahapan pengerjaan yang akan dilakukan untuk membuat sebuah *user interface*/antarmuka yang berfungsi sebagai pengendali utama tampilan animasi pada perangkat lunak. Antarmuka yang sudah dirancang akan dihubungkan dengan *serial port* sebagai media penghubung pengiriman data dari perangkat lunak ke perangkat keras *dot matrix display*.

Komunikasi antara *user interface*/antarmuka dengan *serial port* dibangun dengan menggunakan kontrol *MSComm* yang disediakan oleh *visual basic*. Setiap kontrol *MSComm* yang dipasang pada antarmuka hanya dapat menangani satu jalur komunikasi dengan *serial port*.

3.1.1 Membuat Tampilan/Antarmuka



Gambar 3.2 Rancangan tampilan program kendali LED *display dot matrix*



3.1.2 Pengaturan Properti Kontrol

Bertujuan untuk menentukan sifat-sifat atau karakteristik kontrol yang dipasang, misalnya warna, jenis huruf, ukuran dan sebagainya. jendela properties akan secara otomatis menampilkan properti sesuai dengan objek atau kontrol yang dipilih. Berikut nilai-nilai properti yang perlu dirubah sesuai dengan keperluan untuk membuat antarmuka perangkat lunak pengendalian LED *display dot matrix*:

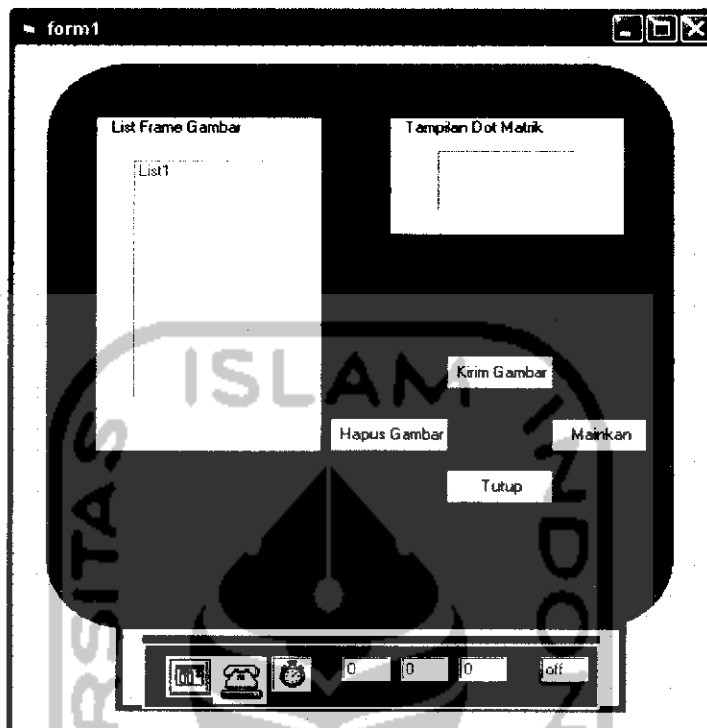
Tabel 3.1 Pengaturan properti kontrol

Objek	Properti	Pengaturan	Keterangan
<i>Form1</i>	- <i>ControlBox</i> - <i>Picture</i> - <i>WindowState</i>	- <i>False</i> - (<i>Bitmap</i>) - <i>2-maximized</i>	- Menyembunyikan tampilan kontrol <i>maximize</i> , <i>minimize</i> dan <i>close</i> . - Menambahkan <i>background</i> gambar pada <i>form</i> . - Memperbesar ukuran <i>form</i> menjadi satu layar penuh.
<i>Command1</i>	- <i>Caption</i> - <i>TabIndex</i>	- Kirim Gambar - 1	- Menampilkan teks "Kirim Gambar" pada kontrol. - Urutan fokus ke-1 pada <i>form</i> saat menekan tombol TAB.
<i>Command2</i>	- <i>Caption</i> - <i>Enabled</i> - <i>TabIndex</i>	- Mainkan - <i>False</i> - 2	- Menampilkan teks "Mainkan" pada kontrol. - Menonaktifkan <i>command2</i> secara default. - Urutan fokus ke-2 saat menekan tombol TAB.
<i>Command3</i>	- <i>Caption</i>	- Hapus Gambar	- Menampilkan teks "Hapus Gambar"

Objek	Properti	Pengaturan	Keterangan
<i>Command3</i>	- <i>Enabled</i> - <i>TabIndex</i>	- <i>False</i> - 3	- Menyembunyikan tampilan kontrol <i>maximize, minimize</i> . - Urutan fokus ke-3 saat menekan tombol TAB.
<i>Command4</i>	- <i>Caption</i> - <i>TabIndeks</i>	- Tutup - 4	- Menampilkan teks "Tutup" pada kontrol. - Urutan fokus ke-4 saat menekan tombol TAB
<i>Text1, Text2 dan Text3</i>	- <i>Text</i> - <i>TabIndex</i>	- 0 - 5, 6 dan 7	- Memberikan nilai awal 0 pada kontrol. - Urutan fokus ke-5, 6 dan 7 saat menekan tombol TAB.
<i>Text4</i>	- <i>Text</i> - <i>TabIndex</i>	- <i>Off</i> - 8	- Memberikan nilai awal <i>off</i> pada kontrol.] - Urutan fokus ke-8 saat menekan tombol TAB.
<i>Timer1</i>	- <i>Enabled</i> - <i>Interval</i>	- <i>False</i> - 1	- Menonaktifkan <i>timer1</i> secara <i>default</i> . - Jarak waktu minimal
<i>CommandDialog1</i>	- <i>Default</i>	- <i>Default</i>	- <i>Default</i>
<i>List</i>	- <i>TabIndex</i>	- 9	- Urutan fokus ke-9 saat menekan tombol TAB
<i>Picture1</i>	- <i>Font</i> - <i>ScaleMode</i>	- <i>Arial</i> - 3- <i>Pixel</i>	- Menentukan ukuran atau jenis <i>font</i> teks. - Menentukan jenis satuan ukuran data pada objek.
<i>MSComm</i>	- <i>RThreshold</i> - <i>RTSenable</i> - <i>Setting</i>	- 1 - <i>True</i> - 9600,n,8,1	- Menentukan jumlah batas permintaan data sebanyak 1. - Mengaktifkan <i>Request To Send</i> . - Menentukan <i>baudrate</i> 9600 <i>bits per second</i> , tanpa paritas, 8 bit pengiriman data dan <i>stop bits = 1</i> .

Objek	Properti	Pengaturan	Keterangan
<i>Frame1</i>	- <i>Caption</i> - <i>Back Color</i> - <i>Fore Color</i>	- - &H80000008& - &H80000009&	- Tidak diberi teks - Menampilkan warna belakang <i>window text</i> . - Menampilkan warna depan <i>highlight text</i> .
<i>Frame2</i>	- <i>Caption</i>	- Tampilan Dot Matrik	- Menampilkan <i>frame</i> dengan nama "Tampilan Dot Matrik".
<i>Frame3</i>	- <i>Caption</i>	- <i>List Frame Gambar</i>	- Menampilkan <i>frame</i> dengan nama "List <i>Frame Gambar</i> ".
<i>Shape1</i>	- <i>Border Color</i> - <i>Border Width</i> - <i>FillColor</i> - <i>FillStyle</i> - <i>Shape</i>	- &H8000000D& - 5 - &H00000080& - 0-Solid - <i>Rounded Rectangle</i>	- Menampilkan warna <i>highlight</i> pada garis <i>shape</i> . Ukuran lebar garis <i>shape</i> = 5. - Menampilkan warna isi <i>shape</i> merah tua. - Jenis tampilan isi <i>shape</i> . - Jenis <i>shape</i> yang digunakan.
<i>Shape2</i>	- <i>Border Color</i> - <i>BorderWidth</i> - <i>FillColor</i> - <i>FillStyle</i> - <i>Shape</i>	- &H80000008& - 5 - &H00C00000& - 0-Solid - <i>Rounded Rectangle</i>	- Menampilkan warna <i>window text</i> pada garis <i>shape</i> . - Ukuran lebar garis <i>shape</i> = 5 - Menampilkan warna isi <i>shape</i> biru tua. - Jenis tampilan isi <i>shape</i> . - Jenis <i>shape</i> yang digunakan
<i>Shape3</i>	- <i>Border Width</i>	- 5	- Ukuran lebar garis <i>shape</i> = 5

Setelah melakukan pengaturan properti di atas, maka kontrol-kontrol program yang telah dirangkai sebelumnya akan menjadi seperti gambar tampilan antarmuka berikut:



Gambar 3.3 Rancangan tampilan program kendali LED display dot matrix setelah pengaturan properti

3.1.3 Menulis Kode

Kode-kode program yang dituliskan adalah kode-kode program yang mengaitkan fungsi kontrol satu dengan fungsi kontrol lainnya dalam membangun sistem program aplikasi yang akan dijalankan.

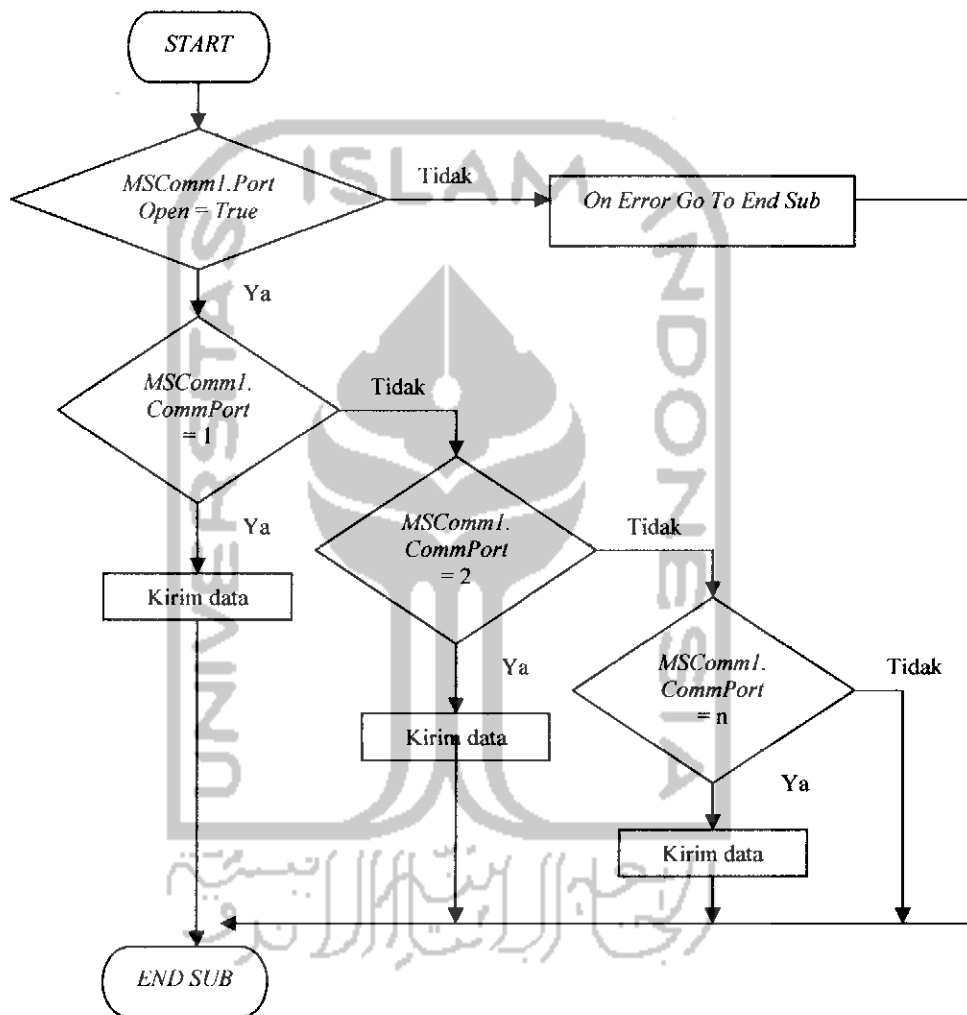
3.1.3.1 Prosedur *Private Sub Form_Load ()*

```
Private Sub Form_Load()
    MSComm1.CommPort = 1
    On Error GoTo end_sub
```


MSComm1.PortOpen = True

end_sub:

End Sub

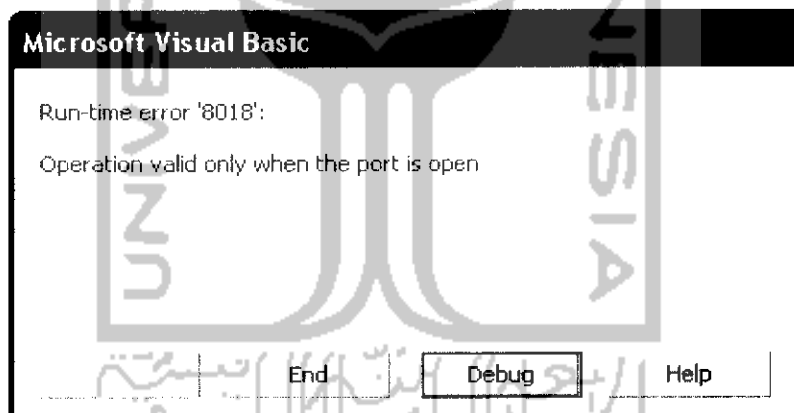


Gambar 3.4 Diagram alir proses inisialisasi port *MSComm1*

Prosedur akan menjalankan *event load* pada *form* atau *event* yang terjadi saat *form* dijalankan. Prosedur ini akan mengaktifkan *port* pada *MSComm* yaitu

COM1 sebagai jalur pengiriman data dari PC (*Personal Computer*) ke mikrokontroler melalui *serial port*. Apabila COM yang digunakan tidak sesuai dengan COM yang dituliskan pada kode program di atas ataupun sebaliknya, maka pernyataan *On Error Go To end sub* akan memberikan tampilan indikator kesalahan pemakaian COM dan membatalkan proses jalannya program saat tombol “mainkan” ditekan.

Tampilan indikator tersebut memberikan dua jenis tombol pilihan dimana yang satu untuk mengakhiri jalannya program dan yang satunya lagi melakukan pengecekan (*debug*) letak kesalahan yang terjadi.



Gambar 3.5 Indikator kesalahan pemakaian COM

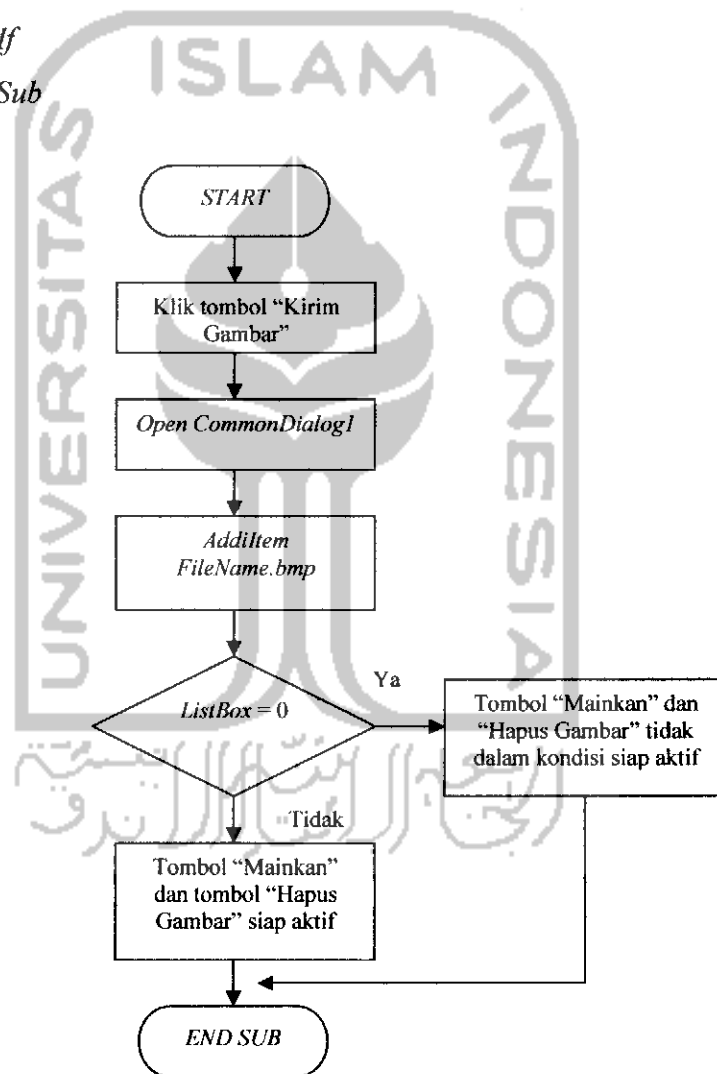
3.1.3.2 Prosedur *Private Sub Command1_Click ()*

Private Sub Command1_Click()

CommonDialog1.ShowOpen

List1.AddItem CommonDialog1.FileName

```
If List1.ListCount <> 0 Then  
Command2.Enabled = True  
Command3.Enabled = True  
Else  
Command2.Enabled = False  
Command3.Enabled = False  
End If  
End Sub
```



Gambar 3.6 Diagram alir aksi yang dilakukan tombol "Kirim Gambar" saat diklik

Prosedur akan menjalankan *event click* terhadap tombol *command1* (“Kirim Gambar”) artinya tombol *command1* akan jalan apabila ada aksi pengeklikkan terhadap tombol ini. Saat program pertama kali dijalankan, tombol “Kirim Gambar” dan “Tutup” (*command4*) telah dalam posisi siap aktif. Prosedur ini juga melakukan pemanggilan terhadap kotak dialog “open” untuk mencari data atau *frame* gambar berformat “.bmp” yang akan dikirimkan ke dalam ruang *listbox*. Pernyataan *if...then...else* yang digunakan bertujuan untuk mengaktifkan posisi tombol *command2* (“Mainkan”) dan tombol *command3* (“Hapus Gambar”). Jika data yang dikirim ke *listbox* tidak sama dengan nol artinya pengaktifan posisi pada kedua tombol tersebut akan terjadi jika ada daftar *frame* gambar yang masuk ke *listbox*. Sebaliknya, kedua tombol ini tidak akan aktif jika tidak ada daftar *frame* gambar yang masuk ke *listbox*.

3.1.3.3 Prosedur *Private Sub Command2_Click ()*

```

Private Sub Command2_Click()
If Command2.Caption = "Mainkan" Then
Text4.Text = "on"
Timer1.Enabled = True
Command2.Caption = "Berhenti"
Command1.Enabled = False
Command3.Enabled = False
ElseIf Command2.Caption = "Berhenti" Then
Text4 = "off"
Command2.Caption = "Mainkan"

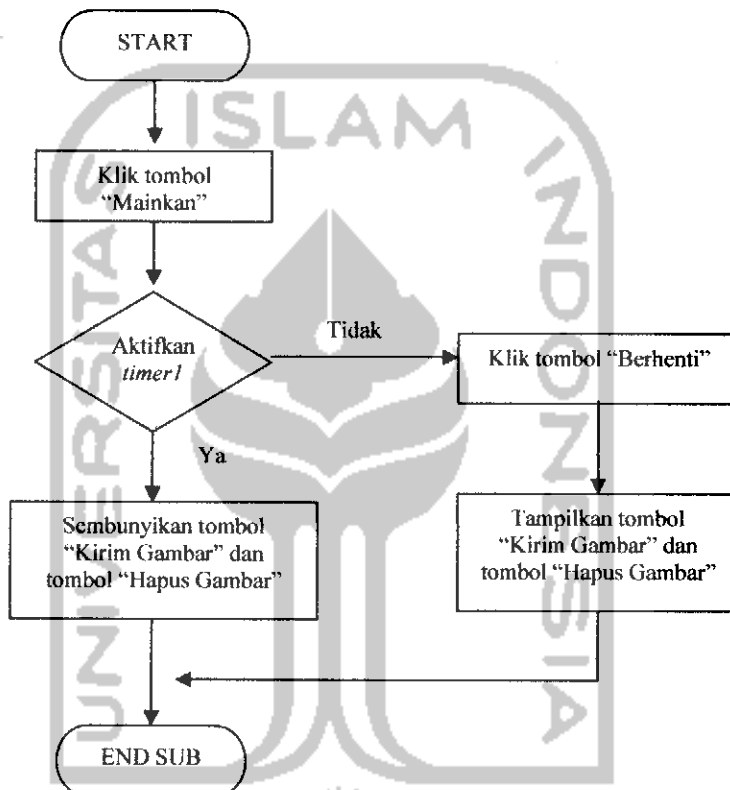
```

Command1.Enabled = True

Command3.Enabled = True

End If

End Sub



Gambar 3.7 Diagram alir aksi yang terjadi saat tombol “Mainkan” diklik

Tombol *command2* (“Mainkan”) akan aktif setelah dilakukan aksi pengeklikkan. Pada prosedur ini, ketika tombol “Mainkan” diklik, teks pada *text4* akan berubah menjadi kondisi “on” yang sebelumnya kondisi awal teks pada *text4* ini adalah “off”. Kontrol *text4* berfungsi sebagai indikator pengaktifan atau penonaktifan *timer1*, maksudnya adalah saat tombol “Mainkan” diklik, kontrol

text4 berubah menjadi kondisi “on” dan *timer1* menjadi aktif, begitu juga sebaliknya. Saat program menjalankan animasi (setelah tombol “Mainkan” diklik), tombol *command2* tersebut berubah fungsi dari tombol kontrol “Mainkan” menjadi tombol kontrol “Berhenti” yang masih dalam satu sistem pengoperasian kontrol *command2*. Aksi yang dilakukan saat tombol kontrol *command2* dalam posisi tombol “Berhenti” adalah merubah kondisi kontrol *text4* menjadi posisi “off” sehingga *timer1* menjadi tidak aktif dan proses penganimasian pada program yang sedang dijalankan akan berhenti. Selanjutnya tombol “Kirim Gambar” dan “Hapus Gambar” kembali dalam posisi siap aktif.

3.1.3.4 Prosedur *Private Sub Command3_Click ()*

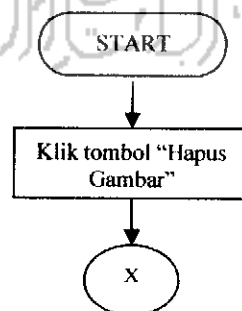
```
Private Sub Command3_Click()
```

```
List1.Clear
```

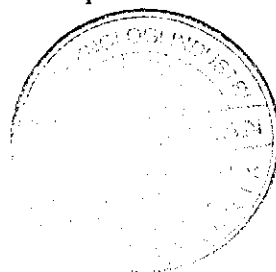
```
Command2.Enabled = False
```

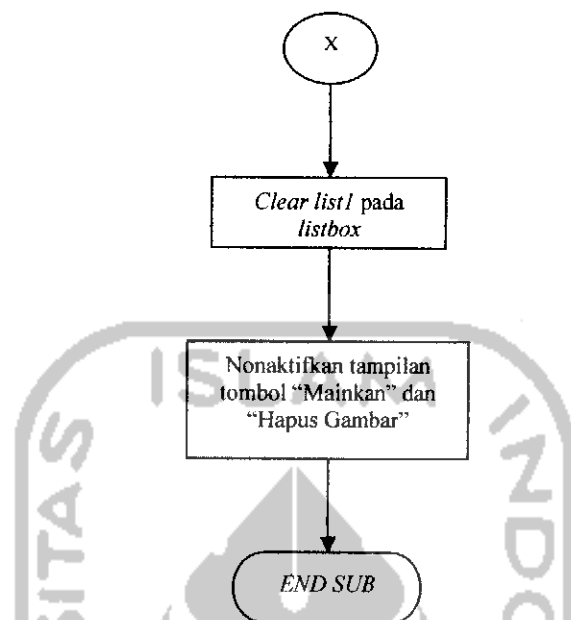
```
Command3.Enabled = False
```

```
End Sub
```



Gambar 3.8 Diagram alir aksi yang terjadi saat tombol “Hapus Gambar” diklik





Gambar 3.8 Diagram alir aksi yang terjadi saat tombol “Hapus Gambar” diklik

Prosedur pada *command3* melakukan pengaktifan terhadap tombol “Hapus Gambar” setelah aksi pengeklikkan. Saat tombol ini diklik, data (*frame* gambar) yang ada pada *listbox* akan dihilangkan seluruhnya dengan sekaligus. Selanjutnya, disaat yang bersamaan tombol “Mainkan” beserta tombol ini akan disembunyikan (dalam kondisi tidak siap aktif) atau tampilan antarmuka akan sama seperti tampilan awal program dijalankan pertama kali (*listbox* = 0)

3.1.3.5 Prosedur *Private Sub Command4_Click ()*

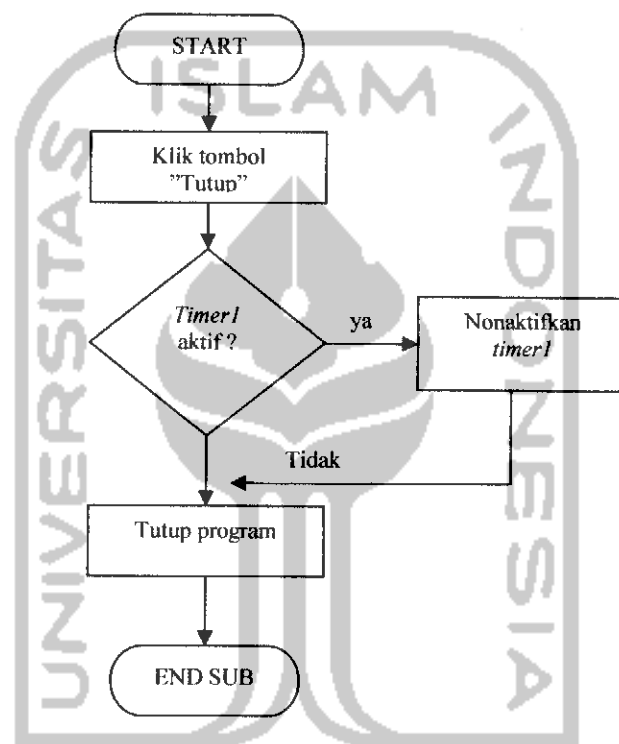
```

Private Sub Command4_Click()
  If Timer1.Enabled = False Then
    Unload Me
  Else
  
```

Text4 = "off"

End If

End Sub



Gambar 3.9 Diagram alir aksi yang terjadi saat tombol “Tutup” diklik

Sama seperti prosedur-prosedur sebelumnya, kontrol ini akan aktif ketika dilakukan aksi pengeklikkan. Saat tombol ini pertama kali diklik ketika program sedang memainkan animasi, *timer1* akan dinonaktifkan saja dan *form* tampilan program yang sedang dijalankan akan ditutup setelah dilakukan aksi pengeklikkan kedua. Kecuali, saat program sedang memainkan animasi sebelum tombol

“Tutup” diklik, tombol “Berhenti” diklik terlebih dahulu dengan tujuan untuk menonaktifkan *timer1*. Setelah itu pada tombol ”Tutup” cukup dilakukan satu kali pengeklikkan saja yaitu menutup program yang sedang dijalankan.

3.1.3.6 Prosedur *Private Sub Timer1_Timer ()*

```

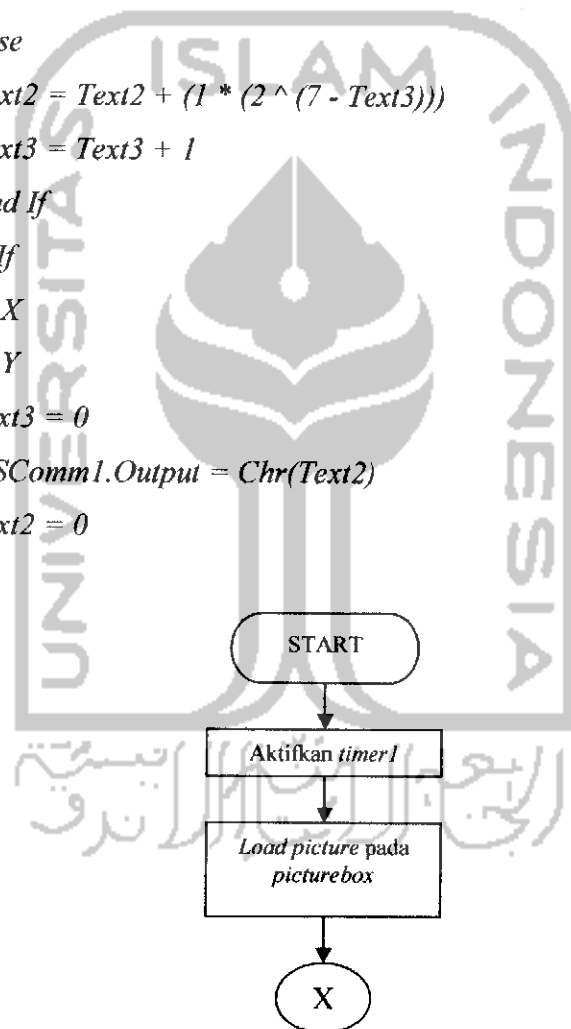
Private Sub Timer1_Timer()
    konstanta = List1.ListCount
    If konstanta <> Text1 Then
        Picture1.Picture = LoadPicture(List1.List(Text1))
        MSComm1.Output = Chr("65")
        For Y = 0 To 7
            For X = 0 To 159
                If Text3 <> 8 Then
                    If Picture1.Point(X, Y) = vbWhite Then
                        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
                        Text3 = Text3 + 1
                    ElseIf Picture1.Point(X, Y) = vbBlack Then
                        Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
                        Text3 = Text3 + 1
                    Else
                        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
                        Text3 = Text3 + 1
                    End If
                Else
                    Text3 = 0
                    MSComm1.Output = Chr(Text2)
                    Text2 = 0
                End If
            Next X
        Next Y
    End If
End Sub

```

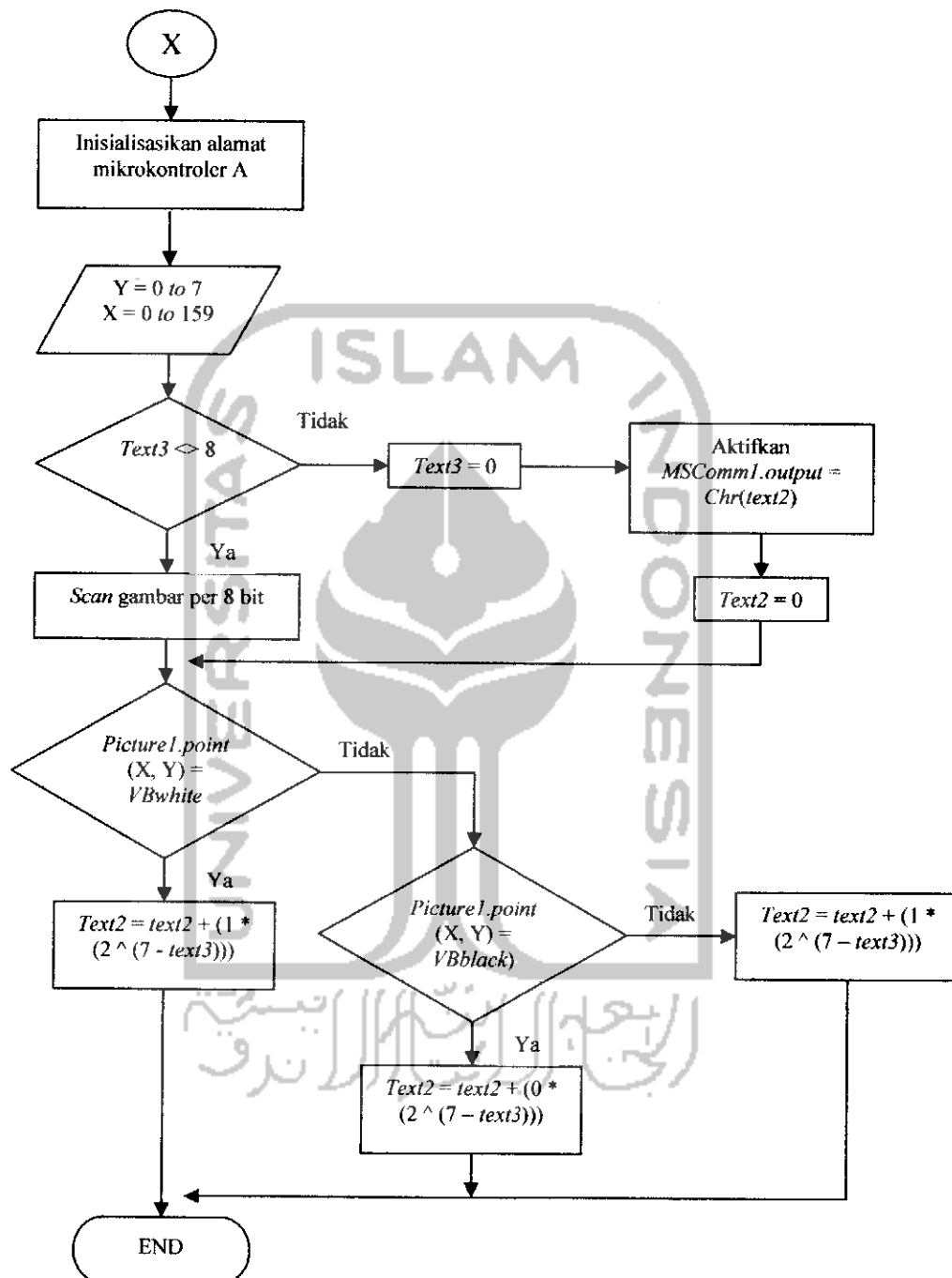
```

If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
End If
Next X
Next Y
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0

```



Gambar 3.10 Diagram alir aksi yang terjadi saat *timer1* aktif pada mikrokontroler “A”



Gambar 3.10 Diagram alir aksi yang terjadi saat *timer1* aktif pada mikrokontroler “A”

Prosedur diatas menunjukkan bahwa *event* yang terjadi saat kontrol *timer1* dijalankan. Dalam prosedur ini, jika konstanta (konstanta di sini berfungsi sebagai variabel yang memiliki nilai *list1.listcount*) tidak sama dengan *text1*, maka *text1* berfungsi sebagai pencacah (*counter*) dalam bentuk angka dari berapa banyak jumlah *frame* gambar yang akan ditampilkan di *picturebox* dan *text1* ini akan mulai mencacah saat *text1 = 0*. Selanjutnya, *MSComm* pertama kali melakukan inisialisasi terhadap alamat pada mikrokontroler pertama yang akan menjadi tujuan awal untuk dikirimkan sejumlah data (dalam satuan bit). Data yang dikirim ke mikrokontroler adalah data yang dikirimkan dengan cara mengirim data sebanyak 8 bit untuk tiap-tiap kolom (x) *dot matrix* (8x8) sebanyak 20 buah (160 bit atau $x = 0 \text{ to } 159$) yang disusun berjejer sebanyak satu baris. Tiap 160 bit kolom data yang dikirim mewakili satu baris (y) susunan nyala lampu *dot* (titik lampu LED *dot matrix*)) sebanyak 8 baris ($y = 0 \text{ to } 7$).

Pernyataan *if...elseif...else* yang digunakan pada prosedur bertujuan untuk bagaimana proses pengolahan data agar dapat diidentifikasi oleh mikrokontroler sebagai sinyal data aktif nyala atau tidak. Sedangkan proses identifikasi data yang dilakukan sendiri oleh perangkat lunak (*software*) ini adalah sistem *scanning* gambar (menguraikan gambar).

Pernyataan *if text3 <> 8*, menyatakan bahwa jika *text3* akan bernilai *error* apabila mencapai nilai batas 8 atau *text3* tidak sama dengan 8 maka data (gambar) diurai per 8 bit dalam dimensi dimana $x = 0-159$ dan $y = 0-7$ (untuk proses identifikasi awal), untuk 1 bit data mewakili 1 *pixel* gambar. *Text2* pada

kode program diatas berfungsi sebagai tempat untuk mempresentasikan jumlah data yang diolah secara biner agar terkonversi menjadi bilangan desimal (ASCII). Untuk algoritmanya yaitu $1*2^{(7-text3)}$ untuk *VBWhite* dan $0*2^{(7-text3)}$ untuk *VBBlack* adalah sebagai algoritma pengidentifikasi warna gambar untuk setiap 8 bit (8 pixel) yang dimulai dari mikrokontroler "A" *byte1*, yang mana proses pengidentifikasian warna pada *frame* gambar yang dikirim dimulai dari letak bit terakhir (*pixel* pertama) atau saat *text3* dimulai dari nilai 0 yaitu 2^7 , 2^6 , 2^5 dan seterusnya sampai 2^0 . Apabila diantara *range* 2^0 sampai 2^7 terdapat warna putih maka akan diberi logika 1 dan begitu juga untuk warna hitam diberi logika 0. Sehingga, setiap 8 bit *pixel frame* gambar akan didapat 8 digit bilangan biner yang dalam *visual basic* 8 digit bilangan biner ini didefinisikan langsung ke dalam bentuk bilangan desimal (ASCII).

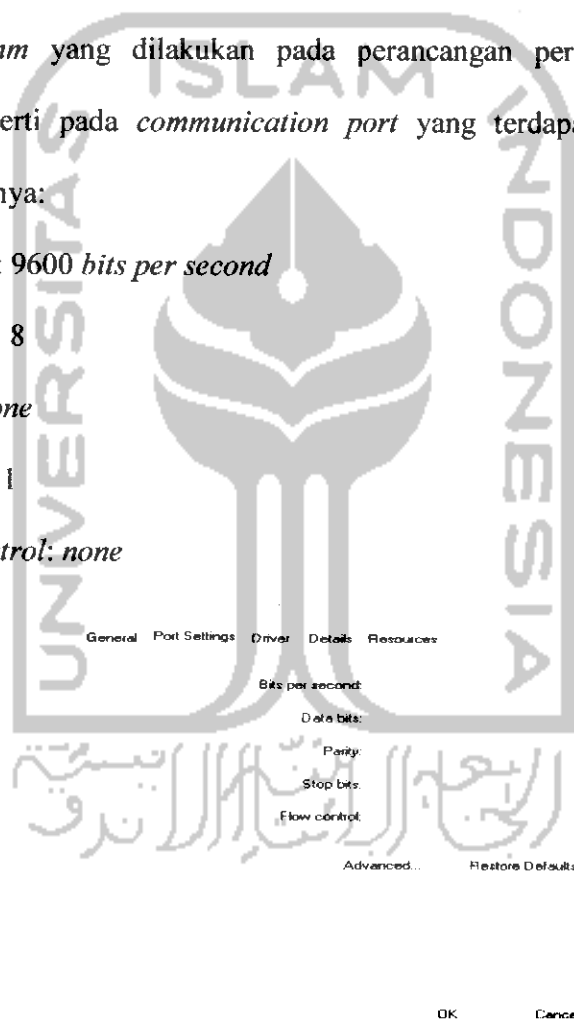
Berikutnya adalah pernyataan *else* yang berisi proses pengiriman data (ASCII) yang akan dikirimkan ke *serial port* lewat kontrol *MSComm1*. *MSComm1.Output = Chr(Text2)* menyatakan bahwa data yang dikirimkan lewat *MSComm* adalah hasil manipulasi fungsi *string* untuk mengkonversi suatu kode ASCII menjadi karakter (gambar, symbol, huruf dan sebagainya) tertentu. Pada bagian ini juga terdapat proses *scanning* gambar dengan cara yang sama seperti sebelumnya, hanya saja pada bagian ini, *MSComm* mulai mengirimkan data ke *serial port* saat *text3* bernilai 0 dan akan mulai melakukan proses *scanning* gambar lagi saat *text2* bernilai 0.

Selanjutnya, proses pengidentifikasian pada prosedur di atas akan diulang terus sampai variabel y mencapai 48 to 55.

3.2 Pengaturan Properti *Serial Port* Pada PC (*Personal Computer*)

Pengaturan ini dilakukan agar terjadi sinkronisasi antara pengaturan properti *MSComm* yang dilakukan pada perancangan perangkat lunak dengan pengaturan properti pada *communication port* yang terdapat di PC. Pengaturan tersebut diantaranya:

1. *Baudrate: 9600 bits per second*
2. *Data bits: 8*
3. *Parity: none*
4. *Stop bits: 1*
5. *Flow Control: none*

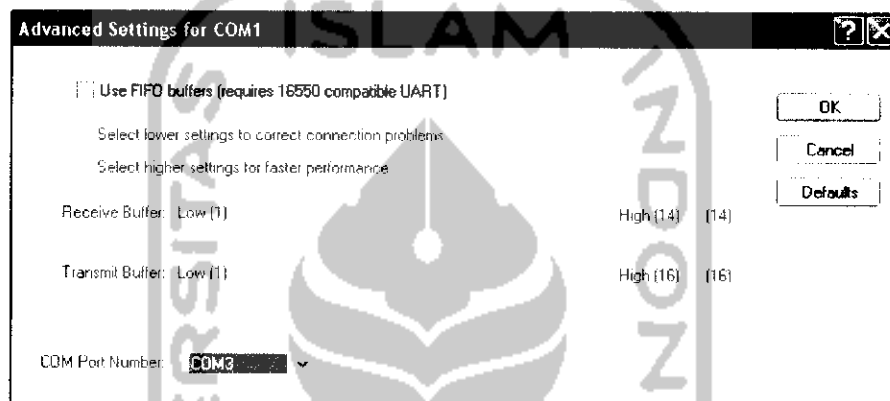


Gambar 3.11 Pengaturan properti *communication port* pada PC

(*Personal Computer*)



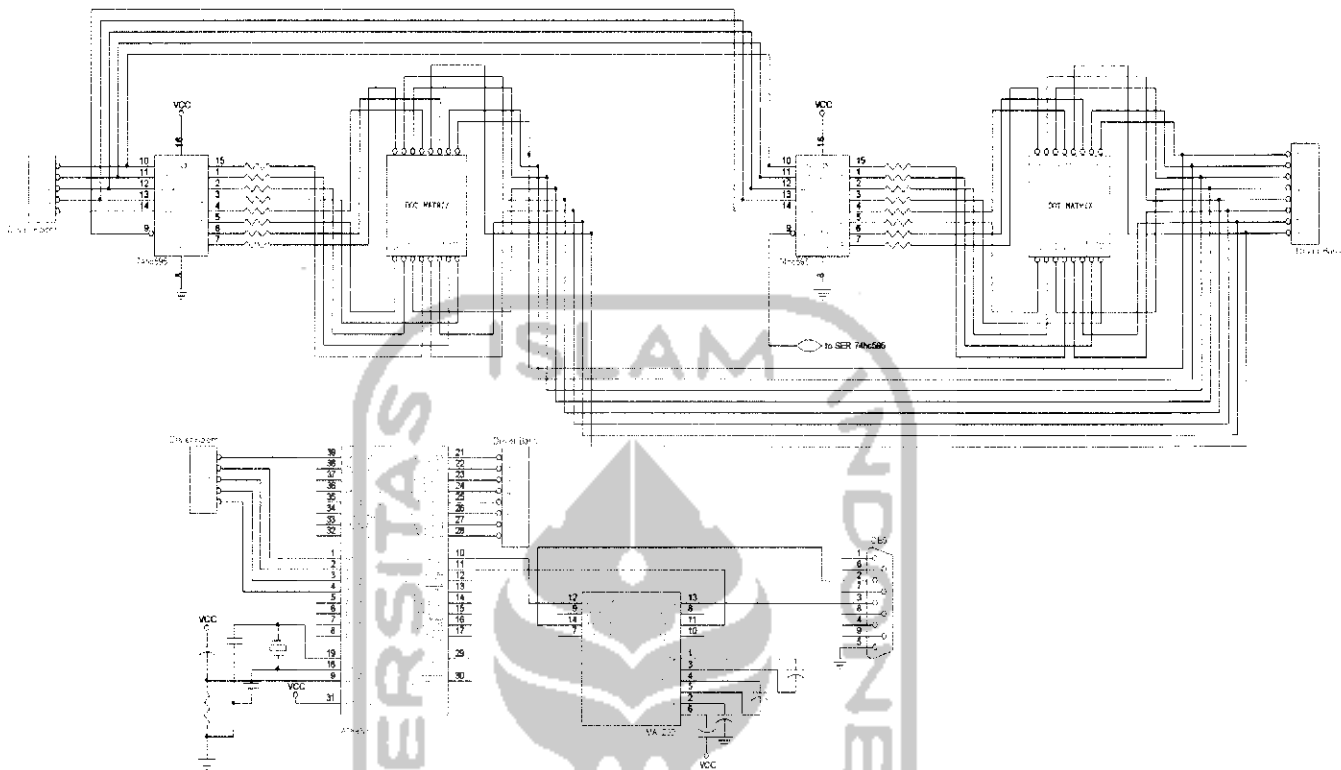
Sedangkan untuk pengaturan COM *port* yang ada pada PC harus disesuaikan juga dengan *commport* yang ada pada kontrol *MSComm visual basic*. Misalnya, pada kontrol *MSComm* diberi *commport* = 3, maka pada PC juga diberi COM *port number* = 3.



Gambar 3.12 Tampilan pengaturan COM *port number* pada PC
(*Personal Computer*)

3.3 Rancangan Perangkat Keras (*Hardware*)

Skema rangkaian perangkat keras *dot matrix display* yang digunakan disini adalah skema rangkaian dari perangkat keras *dot matrix display* yang sudah dibangun atau dirancang sebelum judul penelitian ini diangkat menjadi penelitian tugas akhir mahasiswa.

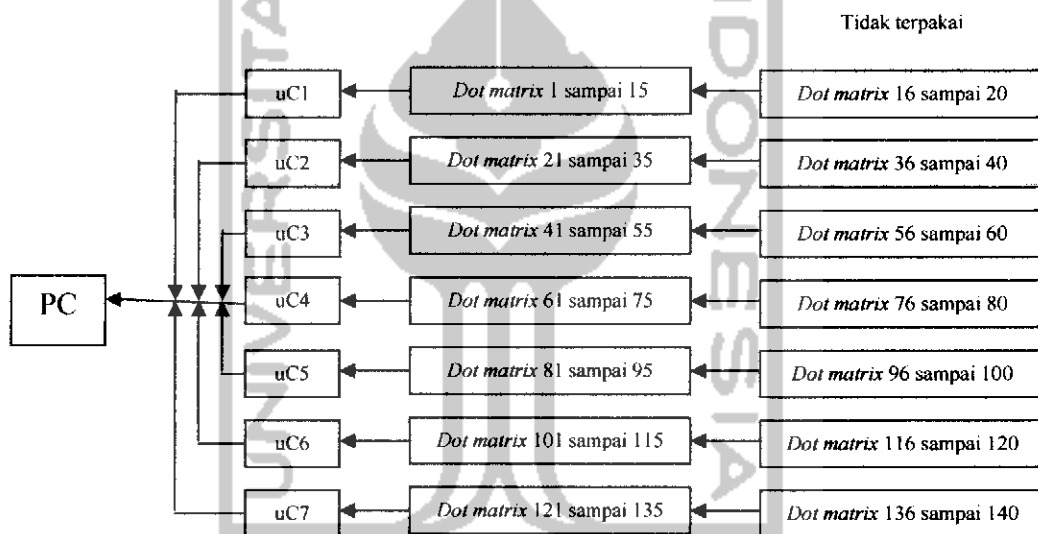


Gambar 3.13 Skema rangkaian *dot matrix display* untuk satu rangkaian sistem minimum dan beberapa buah *dot matrix*

Rangkaian sistem minimum pada mikrokontroler AT89S52 menempatkan dua buah *portnya* sebagai *driver* baris untuk *port2* dan *driver* kolom untuk *port1*. *Driver* baris pada *port2* berfungsi sebagai pengatur penyalan LED *dot matrix* untuk setiap baris *dot matrix* 8x8 (sebanyak 8 baris) yang disusun berjejer (berbaris) sebanyak 15 keping, sedangkan *driver* kolom pada *port1* berfungsi sebagai pengatur penyalan LED *dot matrix* untuk setiap kolom (sebanyak 120 kolom). Untuk satu

penyalan LED *dot matrix* untuk setiap kolom (sebanyak 120 kolom). Untuk satu rangkaian sistem minimum mikrokontroler AT89S52 mengendalikan satu baris susunan *dot matrix* 8x8 sebanyak 15 keping.

Perangkat keras *dot matrix display* ini terdiri atas 7 baris susunan *dot matrix* 8x8 yang setiap barisnya dikendalikan oleh satu rangkaian sistem minimum mikrokontroler.



Gambar 3.14 Blok diagram rancangan *dot matrix display*

BAB IV

PENGUJIAN, ANALISIS DAN PEMBAHASAN SISTEM

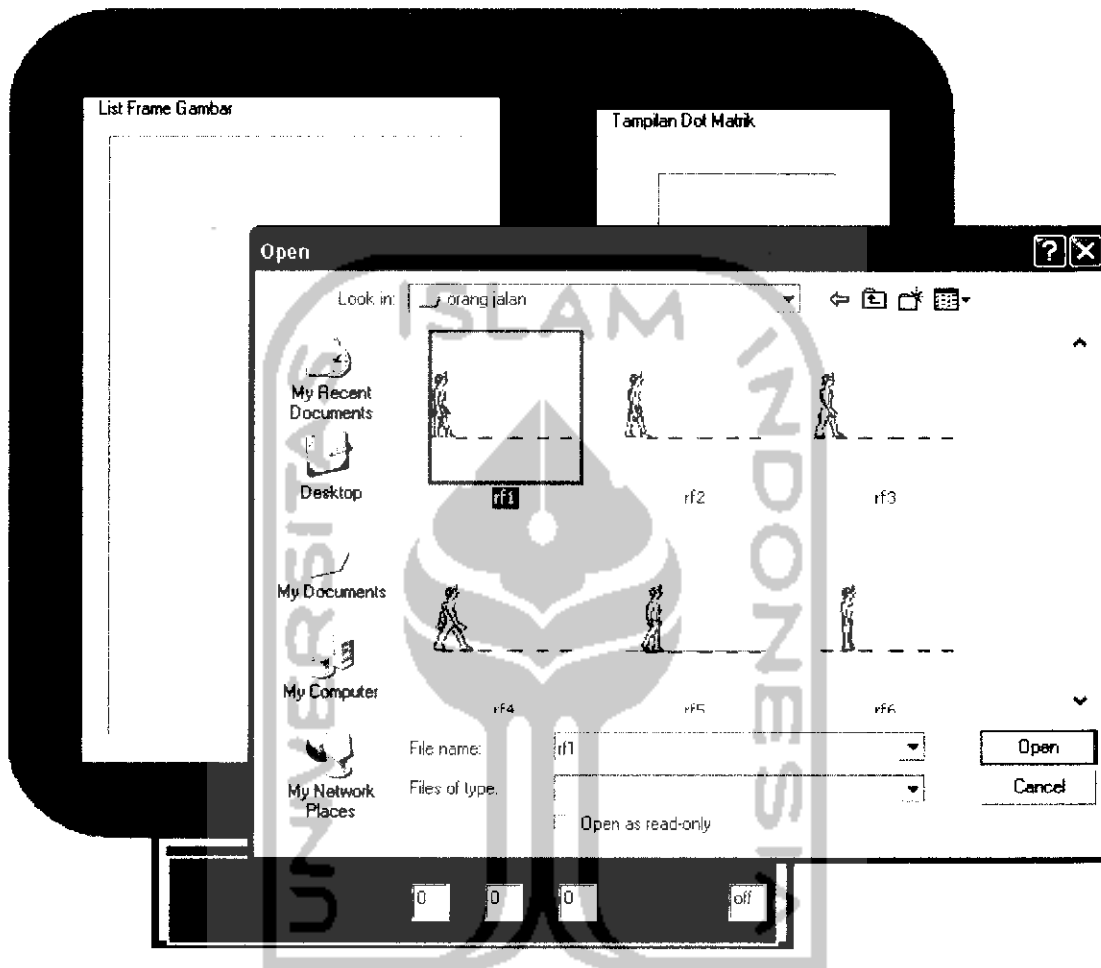
Perangkat lunak (*software*) untuk menampilkan animasi pada *dot matrix display* ini terdiri atas beberapa tombol kontrol utama yang dipasang pada antarmuka dalam menjalankan animasi yang akan ditampilkan pada perangkat keras *dot matrix display* diantaranya yaitu tombol *command1* (“Kirim Gambar”), tombol *command2* (“Mainkan” dan “Berhenti”), tombol *command3* (“Hapus Gambar”), dan tombol *command4* (“Tutup”).

4.1 Pengujian User Interface/Antarmuka Pada Perangkat Lunak (*Software*)

Sebelum menjalankan antarmuka program aplikasi yang telah dirancang, pastikan terlebih dahulu *port* komunikasi antara *MSComm* dengan *COM number* yang ada di PC (*Personal Computer*) pengaturannya sudah sama.

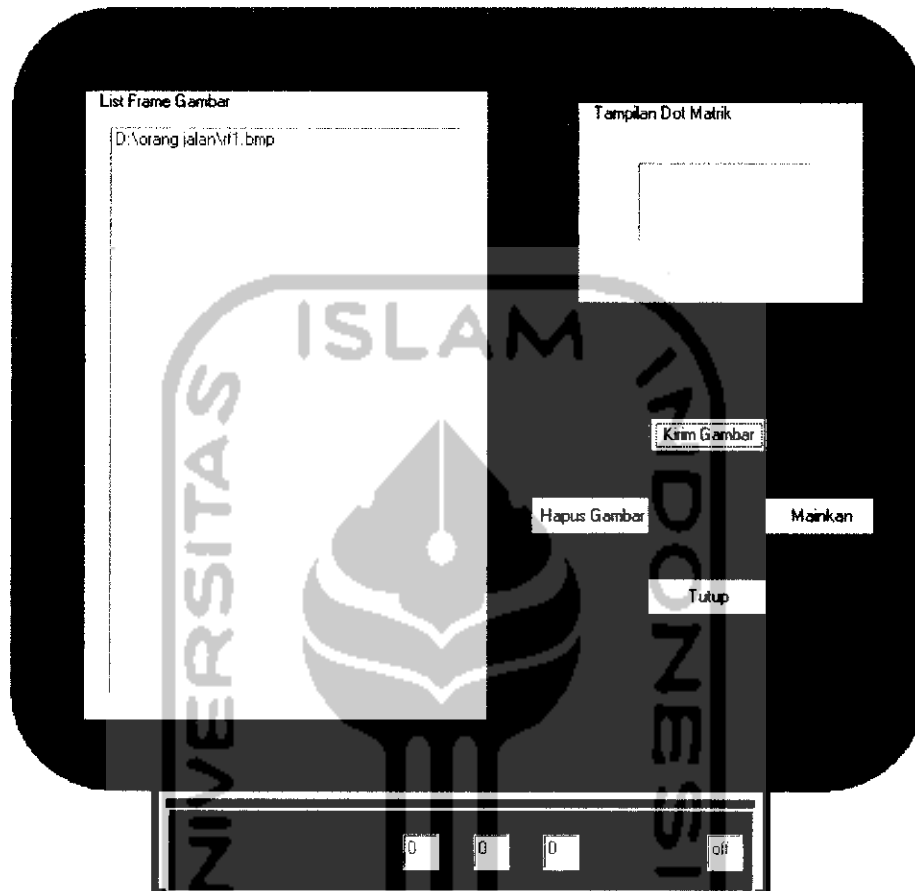
4.1.1 Pengujian Tombol *Command1* (“Kirim Gambar”)

Tombol perintah “Kirim Gambar” pada saat setelah melakukan pengeklikkan, melakukan pemanggilan terhadap kotak dialog “*Open*” untuk mengirimkan data berformat *.bmp* atau berupa *frame* gambar dengan jenis warna yang terdefenisikan hitam dan putih untuk dikumpulkan pada kotak “*List Frame Gambar*” sesuai dengan alur animasi gambar yang dirancang. Pengiriman *frame* gambar dilakukan secara satu per satu setiap dilakukan pemanggilan terhadap kotak dialog “*open*”.



Gambar 4.1 Tampilan antarmuka saat tombol “Kirim Gambar” diklik

Setiap *frame* gambar yang dikirim ke kotak “*List Frame Gambar*” akan diberi nama sesuai dengan *source path* (saluran sumber data) lokasi tempat *frame* gambar disimpan.

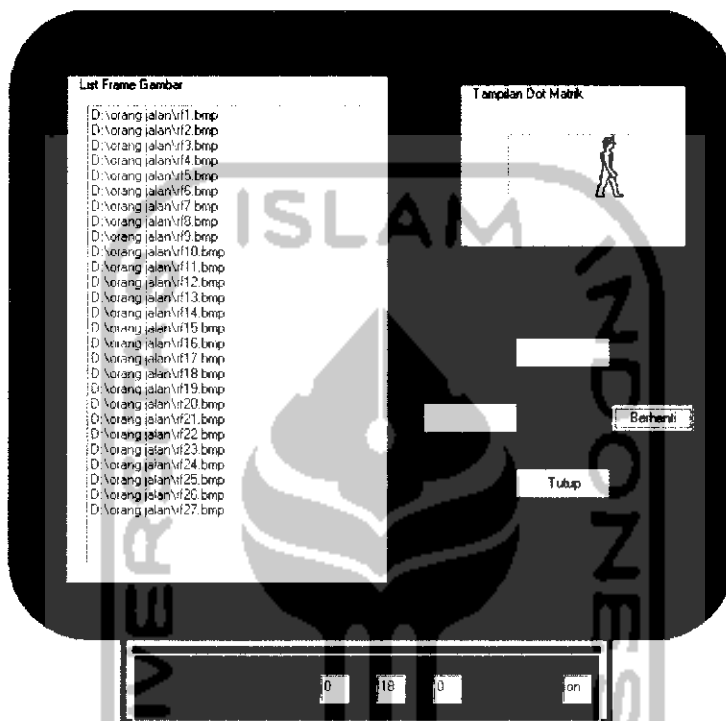


Gambar 4.2 Tampilan antarmuka ketika *frame* gambar telah dikirim ke *listbox*

4.1.2 Pengujian Tombol *Command2* (“Mainkan” dan “Berhenti”)

Tombol perintah “Mainkan” akan menjalankan animasi gambar yang dibangun sesuai dengan susunan rancangan *frame* gambar yang telah dibuat sebelumnya. Proses animasi ini terjadi karena *frame-frame* gambar yang ada di *listbox* “*List Frame* gambar” ditampilkan satu persatu pada *picturebox* (“Tampilan Dot Matrik”) sesuai *interval* waktu yang telah diatur pada pengaturan kecepatan

pengiriman data (*baudrate*) yaitu sebesar 9600 *bits per second* yang ada di properti *MSComm* dan *COM number*.

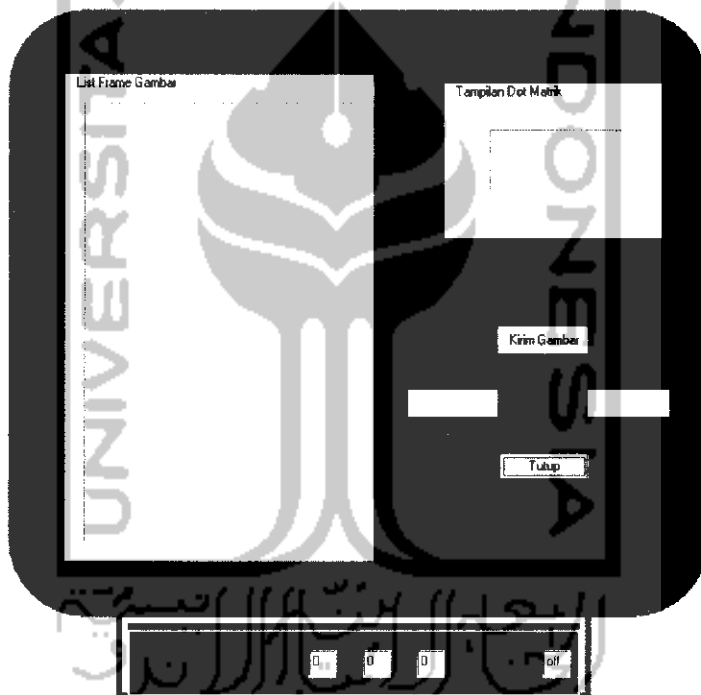


Gambar 4.3 Tampilan antarmuka saat proses penganimasian gambar sedang berlangsung

Proses penganimasian gambar ini akan berlangsung terus menerus secara berulang-ulang sampai tombol perintah “Berhenti” ditekan. Saat tombol perintah “Berhenti” ditekan kemudian tombol perintah “Mainkan” ditekan kembali maka penganimasian gambar akan dimulai dari *frame* gambar terakhir saat proses penganimasian gambar dihentikan.

4.1.3 Pengujian Tombol *Command3* (“Hapus Gambar”)

Tombol ini akan aktif setelah adanya pengeklikan pada tombol “Berhenti” saat proses penganimasian gambar sedang berlangsung, apabila tombol “Hapus Gambar” ini diklik maka *frame* gambar yang ada di *listbox* akan dihapus seluruhnya secara sekaligus. Hal ini disebabkan karena tombol ini melakukan aksi menjadikan *listbox* bernilai 0.



Gambar 4.4 Tampilan antarmuka setelah dilakukan pengeklikan pada tombol “Hapus Gambar”

Tampilan ini sama seperti tampilan awal antarmuka saat program pertama kali dijalankan, yaitu saat belum adanya *frame* gambar yang dikirimkan ke *listbox*.

4.1.4 Pengujian Tombol *Command4* (“Tutup”)

Tombol ini menutup program aplikasi yang sedang dijalankan sesaat setelah penganimasian gambar telah dihentikan. Apabila tombol ini diklik sewaktu proses penganimasian gambar sedang berlangsung, maka tombol ini akan berubah fungsi menjadi fungsi tombol “Berhenti” untuk menghentikan proses penganimasian gambar terlebih dahulu dan setelah itu barulah program akan ditutup setelah dilakukan pengeklikkan kedua.

4.2 Pengujian Perangkat Lunak Terhadap Perangkat Keras *Dot Matrix Display*

4.2.1 Pengujian Nilai *Baudrate*

Program aplikasi dari perangkat lunak yang dijalankan dihubungkan dengan menggunakan kabel serial sebagai media pengiriman data ke perangkat keras *dot matrix display*. Karena banyak data yang dikirimkan untuk sekali pengiriman ke perangkat keras *dot matrix display* adalah per 8 bit (1 *byte*), maka untuk setiap pengiriman *frame* gambar membutuhkan waktu sesuai dengan perhitungan di bawah ini:

<i>Baudrate</i>	=	9600 bps
Banyak data yang dikirimkan	=	8 bit (1 <i>byte</i>)
Ukuran dimensi <i>dot matrix display</i>	=	8960 <i>pixel</i> (1 <i>pixel</i> mewakili 1 bit data)

Maka, waktu yang dibutuhkan untuk sekali menampilkan satu *frame* gambar ke *dot matrix display* adalah:

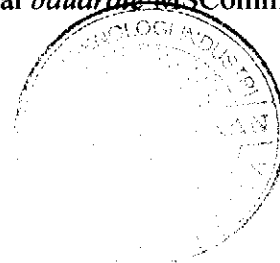
$$\frac{8960}{9600} \times 1 \text{ detik} = 0,933 \text{ detik}$$

Dapat diketahui bahwa semakin besar nilai *baudrate* yang digunakan semakin cepat waktu yang dibutuhkan untuk sekali pengiriman *frame* gambar ke *dot matrix display*. Karena pada perangkat keras *dot matrix display* ini menggunakan *oscillator* sebesar 11.0592 MHz, maka ada beberapa pilihan penggunaan *baudrate* yang dapat dikonfigurasi pada program assembly didalam mikrokontrolernya.

Tabel 4.1 Konfigurasi *baudrate* dalam bahasa *assembly*

Baudrate (bps)	Frekuensi <i>Oscillator</i> (MHz)					SMOD
	11.0592	12	14.7456	16	20	
150	40h	30h	00h			0
300	A0h	98h	80h	75h	52h	0
600	D0h	CCh	C0h	BBh	A9h	0
1200	E8h	E6h	E0h	DEh	D5h	0
2400	F4h	F3h	F0h	EFh	EAh	0
4800		F3h	EFh	EFh		1
4800	FAh		F8h		F5h	0
9600	FDh		FCh			0
9600					F5h	1
19200	FDh		FCh			1
38400			FEh			
76800			FFh			

SMOD merupakan bit yang ada pada *register* PCON yang mana bila nilainya diset 1, maka kecepatan data (*baudrate*) akan dikalikan dua. Setelah dilakukan uji coba pengubahan nilai *baudrate* dari 9600 bps ke 19200 bps pada mikrokontrolernya yang diikuti dengan menyesuaikan nilai *baudrate* MScComm dan



COM pada PC maka dapat dianalisa bahwa perubahan yang terjadi pada nyala LED *dot matrix* dapat dilihat pada tabel berikut:

Tabel 4.2 Uji coba pengaruh peningkatan nilai *baudrate* terhadap LED *dot matrix*

<i>Baudrate</i> (bps)	Kondisi LED <i>Dot Matrix</i>
9600	Nyala terang
19200	<i>error</i>

Baudrate dipengaruhi oleh besarnya nilai TH1 (*timer high*). *Timer* yang dipakai pada program *assembly* pengendalian nyala LED *dot matrix* ini adalah jenis *timer1 mode2* yang dipengaruhi oleh SMOD yang merupakan bit yang ada pada *register* PCON. Apabila SMOD diset aktif (SMOD = 1) pada *register* ini maka nilai *baudrate* digandakan menjadi 19200 bps persamaan yang dipakai untuk menghitung nilai TH1 adalah:

$$TH1 = 256 - (\text{frekuensi osilator} / (12 \cdot 16)) / \text{baudrate}$$

(12*32) di sini adalah untuk 12 merupakan pembagi frekuensi waktu sistem (*system clock frequency*) sedangkan 32 merupakan frekuensi yang melebihi batas *baudrate* yang digunakan dari pewaktu yang dijalankan (*timer overflow frequency*). Dengan demikian dari persamaan di atas didapat nilai TH1 adalah 253 (0FDh). Sedangkan SMOD diset nonaktif (SMOD = 0) maka persamaan yang digunakan untuk menghitung nilai TH1 adalah

$$TH1 = 256 - (\text{frekuensi osilator} / (12 \cdot 32)) / \text{baudrate}$$

Karena batasan masalah yang sudah ditentukan dalam penelitian tugas akhir ini yaitu *baudrate* yang diisikan ke mikrokontroler lewat program menggunakan bahasa *assembly* adalah *9600 bits per second* maka waktu yang diperlukan untuk satu kali menampilkan *frame* gambar adalah 0,9 detik.

Waktu yang dihasilkan oleh *baudrate* 9600 ini dinilai kurang cepat untuk melakukan proses penganimasian gambar. maka dari itu perlu dilakukan sedikit uji coba perubahan *delay* (tunda) waktu yang terdapat pada program *assembly* di dalam mikrokontrolernya.

Waktu tampil tiap 1 *frame* gambar = 1 detik / (TH1 * waktu tunda)

Tabel 4.3 Uji coba pengaruh pengubahan nilai waktu tunda

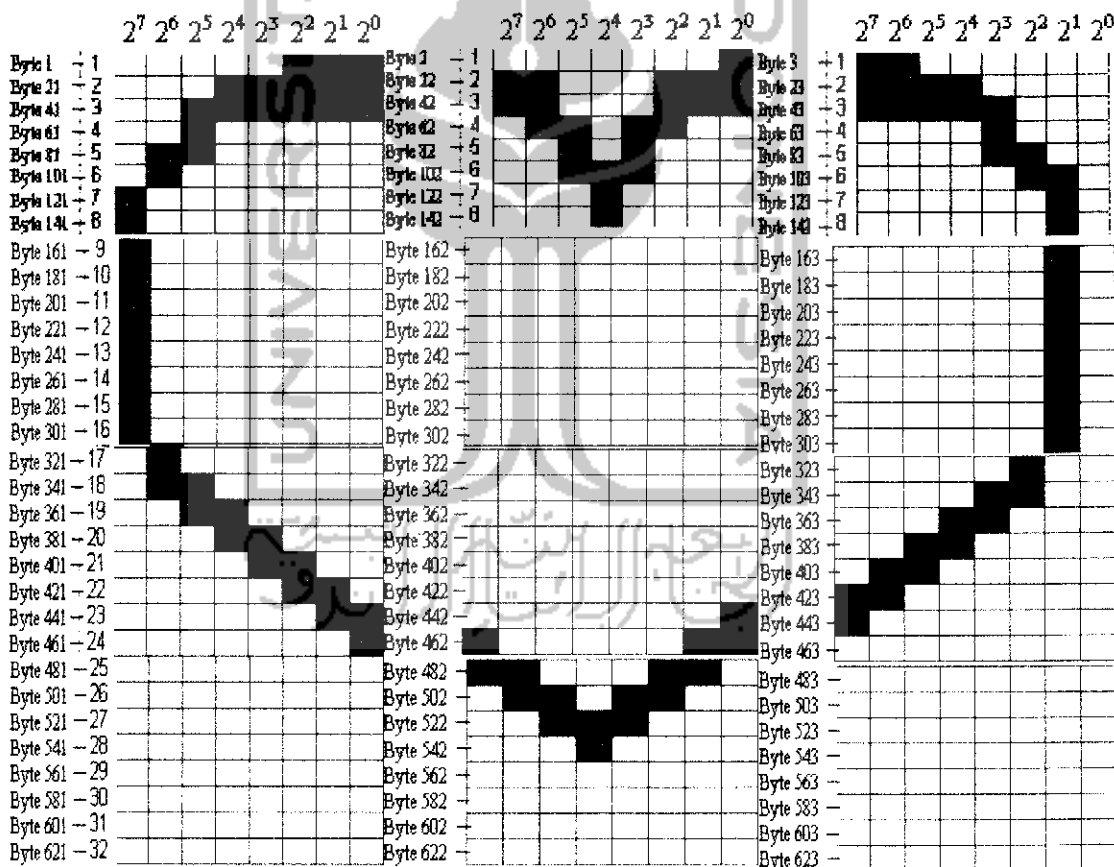
Waktu Tunda	Waktu tampil tiap 1 <i>frame</i> gambar	Satuan
-1000	0,004	mS
-2000	0,002	mS
-6000	0,6	uS

4.2.2 Pengujian Pengiriman *Frame* Gambar

Frame gambar yang dikirimkan dari antarmuka perangkat lunak dipilah menjadi 8 bit penyalaan LED *dot matrix* (1 bit mewakili nyala 1 LED) yang dimulai dari *byte* 1, *byte* 2, *byte* 3 dan seterusnya sesuai dengan urutan alur *scanning* penyalaan LED dari kiri paling atas (*byte*1) hingga kanan paling bawah (*byte* 1120). Karena pada perangkat keras *display dot matrix* ini hanya terpasang 15 keping *dot matrix* untuk panjang *display* yang seharusnya 20 keping *dot matrix* (sesuai dengan program *assembly* yang diisikan pada setiap mikrokontrolernya)

maka, untuk penyalan LED dari *byte* 16 sampai *byte* 20 hingga seterusnya sejajar sampai kebawah tidak dapat dilakukan. Akan tetapi, alur penyalan LED *dot matrix* tetap diproses dari *byte* 1 sampai *byte* 20 dan seterusnya. Berikut adalah contoh proses pengiriman sebuah *frame* gambar yang dinyalakan pada *dot matrix display*:

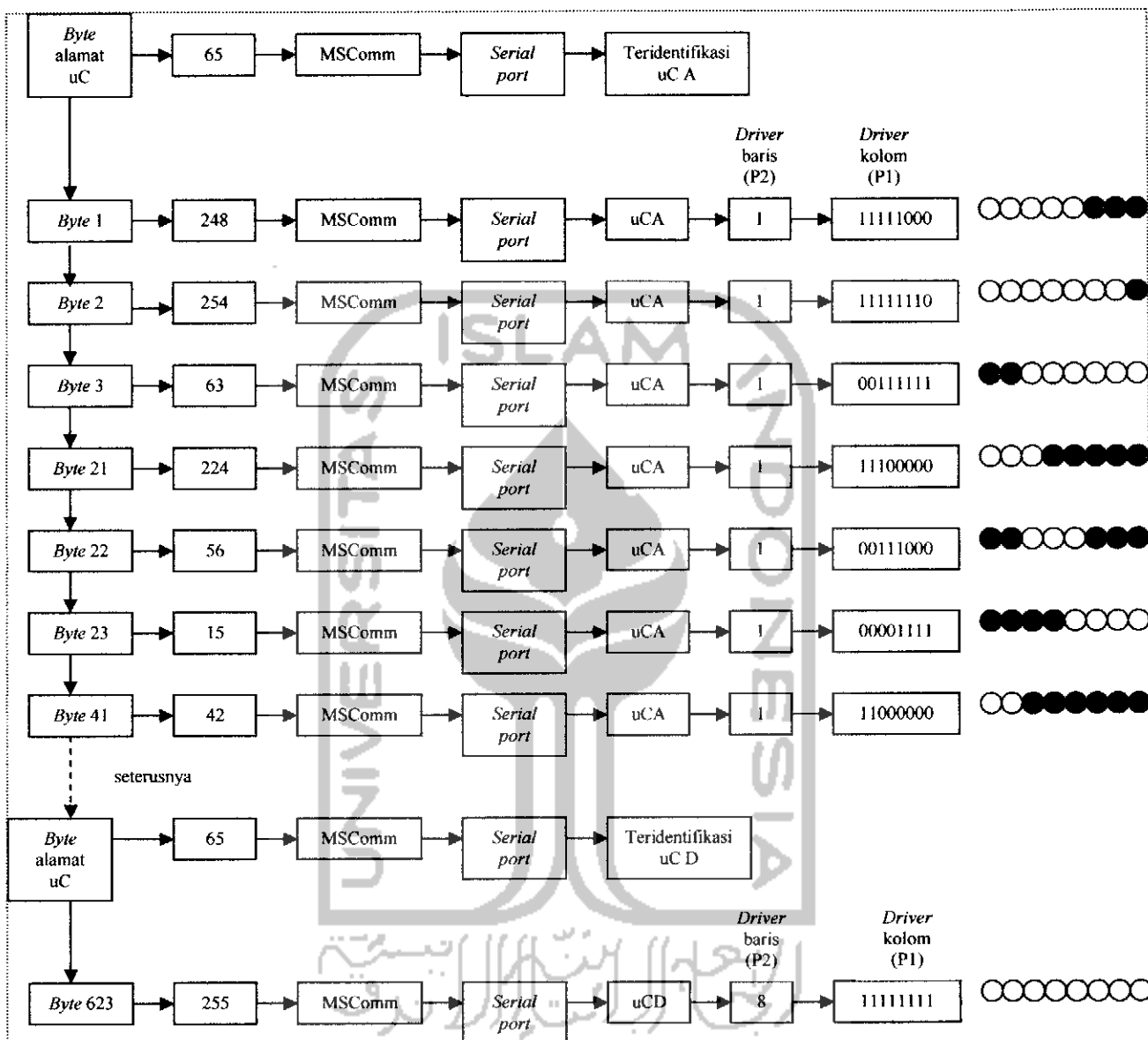
Gambar 4.5 Love.bmp



Gambar 4.6 Alur proses *scanning* penyalan LED *dot matrix*

Tabel 4.4 Proses pengolahan data dari bilangan biner ke bentuk bilangan ASCII

Byte	Biner	ASCII	Byte	Biner	ASCII	Byte	Biner	ASCII
uC A								
1	11111000	248	2	11111110	254	3	00111111	63
21	11100000	224	22	00111000	56	23	00001111	15
41	11000000	192	42	00111000	56	43	00000111	7
61	11011111	223	62	10010011	147	63	11110111	247
81	10011111	159	82	11010111	215	83	11110011	243
101	10111111	191	102	11000111	199	103	11111001	249
121	01111111	127	122	11101111	239	123	11111101	253
141	01111111	127	142	11101111	239	143	11111101	253
uC B								
161	01111111	127	162	11111111	255	163	11111101	253
181	01111111	127	182	11111111	255	183	11111101	253
201	01111111	127	202	11111111	255	203	11111101	253
221	01111111	127	222	11111111	255	223	11111101	253
241	01111111	127	242	11111111	255	243	11111101	253
261	01111111	127	262	11111111	255	263	11111101	253
281	01111111	127	282	11111111	255	283	11111101	253
301	01111111	127	302	11111111	255	303	11111101	253
uC C								
321	10111111	191	322	11111111	255	323	11111011	251
341	10011111	159	342	11111111	255	343	11110011	243
361	11001111	207	362	11111111	255	363	11100111	231
381	11100111	231	382	11111111	255	383	11001111	207
401	11110011	243	402	11111111	255	403	10011111	159
421	11111001	249	422	11111111	255	423	00111111	63
441	11111100	252	442	11111110	254	443	01111111	127
461	11111110	254	462	01111100	124	463	11111111	255
uC D								
481	11111111	255	482	00111001	57	483	11111111	255
501	11111111	255	502	10010011	147	503	11111111	255
521	11111111	255	522	11000111	199	523	11111111	255
541	11111111	255	542	11101111	239	543	11111111	255
561	11111111	255	562	11111111	255	563	11111111	255
581	11111111	255	582	11111111	255	583	11111111	255
601	11111111	255	602	11111111	255	603	11111111	255
621	11111111	255	622	11111111	255	623	11111111	255



Gambar 4.7 Blok diagram alur proses penyalan LED *dot matrix*

Kode-kode ASCII diatas akan dikirimkan lewat MSComm, karena pengolahan data yang dilakukan di perangkat lunak hanya mengerti tipe data yang

berformat ASCII. *Serial port* yang mengkonversi kode-kode ASCII tersebut kedalam bentuk bilangan biner yang kemudian akan dibaca oleh mikrokontroler.

Alur proses penyalan LED *dot matrix* pada blok diagram menggambarkan proses pengiriman data ASCII dari antarmuka *visual basic* sampai pada penyalan LED *dot matrix* diperangkat kerasnya. Untuk gambar “love” pengiriman data ASCII digambarkan hanya sampai pada alamat mikrokontroler “D”, sedangkan untuk mikrokontroler “E” sampai “G” kondisinya adalah sama yaitu berlogika 1.



Gambar 4.8 Tampilan gambar “love” pada *dot matrix display*

Perancangan gambar animasi yang sudah diujikan di sini adalah gambar animasi orang yang sedang berjalan. Gambar animasi ini dibentuk oleh 27 *frame* gambar yang menyusun pergerakan orang yang tampak sedang berjalan. Pengujian gambar animasi ini dilakukan sebanyak tiga kali yaitu berdasarkan perubahan nilai waktu tunda pada program *assembly* yang diisikan ke dalam mikrokontrolernya.



Gambar 4.9 Tampilan gambar animasi orang sedang berjalan



BAB V

PENUTUP

5.1 Kesimpulan

Atas dasar hasil pengujian dan pembahasan perangkat lunak untuk tampilan animasi pada susunan *dot matrix* 8x8, maka dapat diambil kesimpulan sebagai berikut:

1. *Dot matrix display* merupakan alat penampil informasi berupa karakter dan gambar, yang dibangun atas susunan beberapa buah *dot matrix*.
2. Pada perangkat lunak, pengendalian dilakukan lewat sebuah tampilan antarmuka yang berfungsi sebagai tempat simulasi awal gambar animasi yang akan ditampilkan di perangkat keras *dot matrix display*.
3. Pada perangkat keras, pengendalian dilakukan oleh mikrokontroler lewat fungsi masing-masing *portnya* yaitu sebagai *driver* baris untuk *port2* dan sebagai *driver* kolom untuk *port1*.
4. Data yang diolah pada *visual basic* adalah tipe data ASCII yang dikirim ke *serial port* melalui sebuah kontrol MSComm. Sedangkan data yang dibaca di mikrokontrolernya adalah data dalam bentuk bilangan biner hasil pengkonversian data ASCII yang telah dikonversi di *serial portnya*.

5. Animasi gambar dilakukan dengan proses pengiriman tiap-tiap *frame* gambar dari antarmuka *visual basic* ke perangkat keras *dot matrix display* yang dijalankan satu per satu sesuai dengan urutan langkah-langkah skenario pembentukan gerak animasi gambar yang dirancang.

5.2 Saran

- 1 Diperlukan adanya tambahan tampilan antarmuka untuk pengiriman karakter langsung lewat *keyboard* untuk menampilkan karakter secara cepat.
- 2 Penambahan IC memori pada perangkat keras dapat membantu menyimpan data animasi gambar yang telah dirancang dan dijalankan untuk ditampilkan lagi secara mandiri tanpa dibutuhkan adanya proses pengolahan animasi gambar yang dilakukan di PC.
- 3 Untuk menambah kecepatan gerak animasi gambar dibutuhkan pemakaian *oscillator* yang lebih besar pada rangkaian mikrokontrolernya agar nilai *baudrate* yang digunakan dapat dimaksimalkan.

DAFTAR PUSTAKA

- Prasetia, Retna. & Edi Widodo, Catur. 2004.** *Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0.* **Yogyakarta: ANDI.**
- MADCOMS. 2006.** *Panduan Pemrograman dan Referensi Kamus Visual Basic 6.0.* **Yogyakarta: ANDI.**
- Kadir, Abdul. 2008.** *Mudah Menjadi Programmer Visual Basic 6.0.* **Yogyakarta: YesCom.**
- Ir. Pamungkas. 2000.** *Tip dan Trik Microsoft Visual Basic 6.0.* **Jakarta: PT Elex Media Komputindo.**
- Satriyo Basuki, Hari. 2006.** *Perancangan dan Implementasi Penampil Nama Stasiun Kereta Api Lokal Berbasis Mikrokontroler Dengan Display Dot Matrix.* **Bandung: Pusat Penelitian Informatika Lembaga Ilmu Pengetahuan Indonesia.**
- Pramardianto, Adisatya. 2007.** *Penampil Informasi Dot Matrix Dengan Animasi Terprogram Berbasis Mikrokontroler AT89C52.* **Yogyakarta: Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.**
- Heru Supriyono, Jatmiko. 2008.** *Pengembangan Tulisan Berjalan (Running Text) Pada Dot Matrix Dengan Pengisian Karakter Berbasis Layanan Short Message Service (SMS) Jaringan GSM.* **Surakarta: Jurusan Teknik Elektro Universitas Muhammadiyah Semarang.**

Widayanto, Wahyu. 2008. *Pengendalian Running Text Dengan SMS Menggunakan Mikrokontroler AT89S52*. Yogyakarta: Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Shato. 2008. *Penampil Dot Matrix*. Shato Media Online.

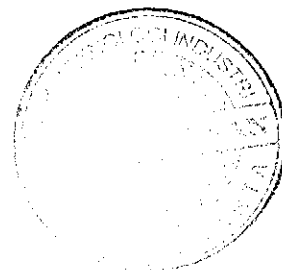
<http://www.8052.com>

<http://www.atmel.com>





LAMPIRAN



Kode Program Visual Basic 6.0

```
Private Sub Command1_Click()  
CommonDialog1.ShowOpen  
List1.AddItem CommonDialog1.FileName  
If List1.ListCount <> 0 Then  
Command2.Enabled = True  
Command3.Enabled = True  
Else  
Command2.Enabled = False  
Command3.Enabled = False  
End If  
End Sub
```

```
Private Sub Command2_Click()  
If Command2.Caption = "Mainkan" Then  
Text4.Text = "on"  
Timer1.Enabled = True  
Command2.Caption = "Berhenti"  
Command1.Enabled = False  
Command3.Enabled = False  
ElseIf Command2.Caption = "Berhenti" Then  
Text4 = "off"  
Command2.Caption = "Mainkan"  
Command1.Enabled = True  
Command3.Enabled = True  
End If  
End Sub
```

```
Private Sub Command3_Click()  
List1.Clear  
Command2.Enabled = False  
Command3.Enabled = False  
End Sub
```

```
Private Sub Command4_Click()  
If Timer1.Enabled = False Then  
Unload Me  
Else  
Text4 = "off"  
End If  
End Sub
```



```

Private Sub Form_Load()
MSComm1.CommPort = 1
On Error GoTo end_sub
MSComm1.PortOpen = True
end_sub:
End Sub

```

```

Private Sub Timer1_Timer()
konstanta = List1.ListCount
If konstanta <> Text1 Then
Picture1.Picture = LoadPicture(List1.List(Text1))
MSComm1.Output = Chr("65")
For Y = 0 To 7
For X = 0 To 159
If Text3 <> 8 Then
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
Else
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
End If
Next X
Next Y

```

```
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0
```

```
MSComm1.Output = Chr("66")
```

```
For Y = 8 To 15
```

```
For X = 0 To 159
```

```
If Text3 <> 8 Then
```

```
    If Picture1.Point(X, Y) = vbWhite Then
```

```
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    ElseIf Picture1.Point(X, Y) = vbBlack Then
```

```
        Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    Else
```

```
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    End If
```

```
Else
```

```
    Text3 = 0
```

```
    MSComm1.Output = Chr(Text2)
```

```
    Text2 = 0
```

```
    If Picture1.Point(X, Y) = vbWhite Then
```

```
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    ElseIf Picture1.Point(X, Y) = vbBlack Then
```

```
        Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    Else
```

```
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
```

```
        Text3 = Text3 + 1
```

```
    End If
```

```
End If
```

```
Next X
```

```
Next Y
```

```
    Text3 = 0
```

```
    MSComm1.Output = Chr(Text2)
```

```
    Text2 = 0
```

```
MSComm1.Output = Chr("67")
```

```
For Y = 16 To 23
```

```
For X = 0 To 159
```

```
If Text3 <> 8 Then
```

```

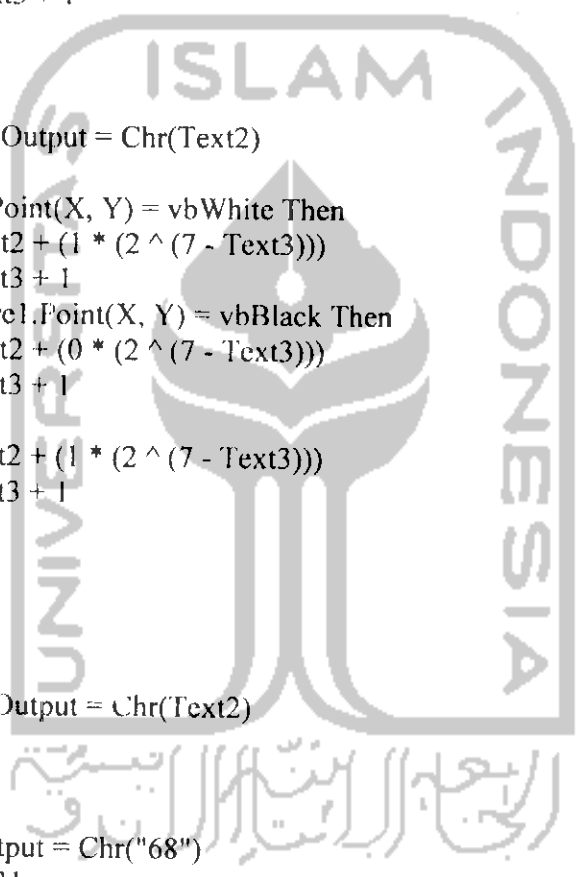
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
Else
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
End If
Next X
Next Y
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0

```

```

MSComm1.Output = Chr("68")
For Y = 24 To 31
For X = 0 To 159
If Text3 <> 8 Then
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else

```

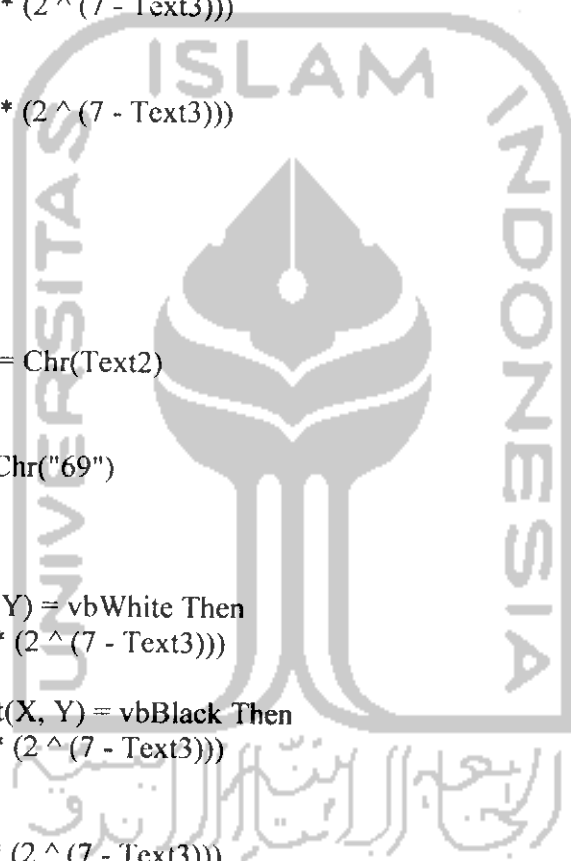



```

    Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
    Text3 = Text3 + 1
End If
Else
    Text3 = 0
    MSComm1.Output = Chr(Text2)
    Text2 = 0
    If Picture1.Point(X, Y) = vbWhite Then
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    ElseIf Picture1.Point(X, Y) = vbBlack Then
        Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    Else
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    End If
End If
Next X
Next Y
    Text3 = 0
    MSComm1.Output = Chr(Text2)
    Text2 = 0

MSComm1.Output = Chr("69")
For Y = 32 To 39
For X = 0 To 159
If Text3 <> 8 Then
    If Picture1.Point(X, Y) = vbWhite Then
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    ElseIf Picture1.Point(X, Y) = vbBlack Then
        Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    Else
        Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
        Text3 = Text3 + 1
    End If
Else
    Text3 = 0
    MSComm1.Output = Chr(Text2)
    Text2 = 0
    If Picture1.Point(X, Y) = vbWhite Then

```



```

Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
End If
Next X
Next Y
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0

```

```

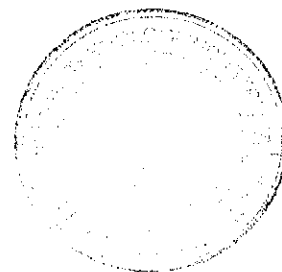
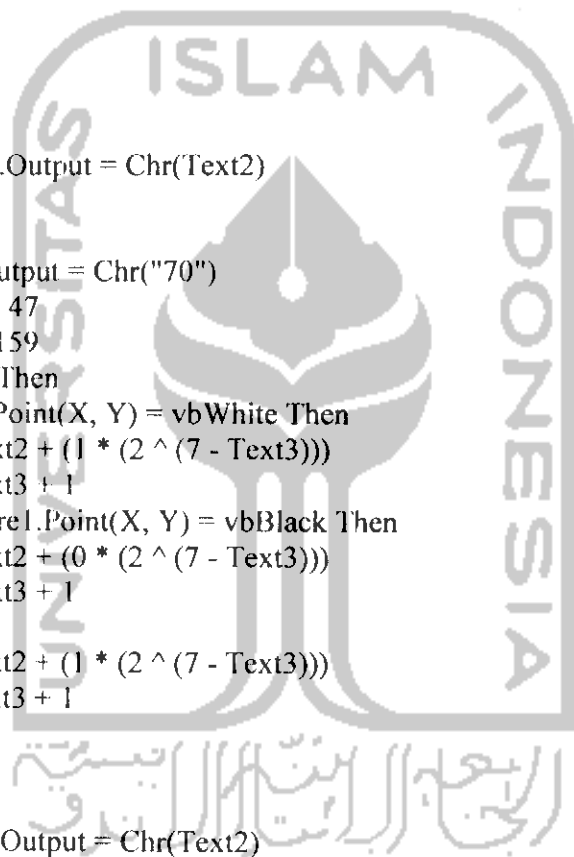
MSComm1.Output = Chr("70")
For Y = 40 To 47
For X = 0 To 159
If Text3 <> 8 Then
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If

```

```

Else
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1

```



```

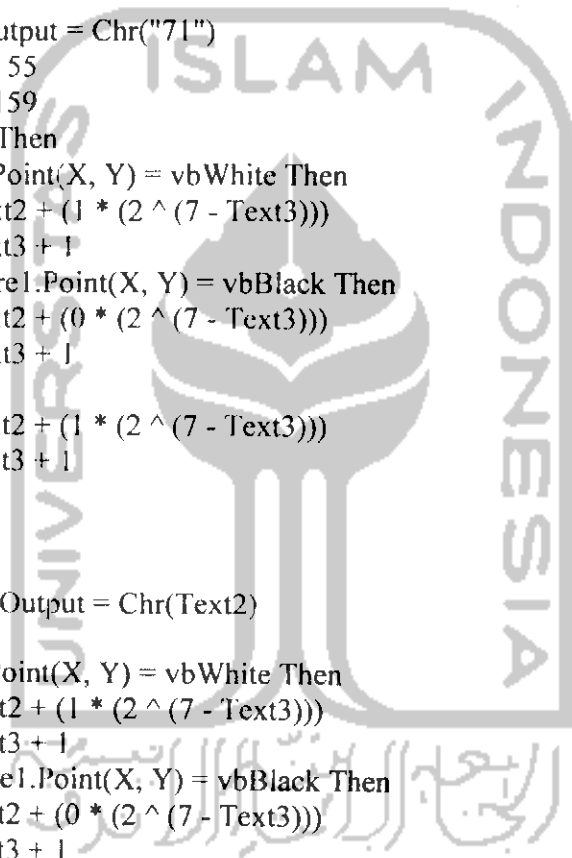
End If
End If
Next X
Next Y
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0

```

```

MSComm1.Output = Chr("71")
For Y = 48 To 55
For X = 0 To 159
If Text3 <> 8 Then
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
Else
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0
If Picture1.Point(X, Y) = vbWhite Then
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
ElseIf Picture1.Point(X, Y) = vbBlack Then
Text2 = Text2 + (0 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
Else
Text2 = Text2 + (1 * (2 ^ (7 - Text3)))
Text3 = Text3 + 1
End If
End If
Next X
Next Y
Text3 = 0
MSComm1.Output = Chr(Text2)
Text2 = 0

```



```
Text1 = Text1 + 1
```

```
Else
```

```
Text1 = 0
```

```
End If
```

```
If Text4 = "off" Then
```

```
Timer1.Enabled = False
```

```
Else
```

```
Timer1.Enabled = True
```

```
End If
```

```
End Sub
```



MIKROKONTROLER A

```
ENABLE    BIT    P2.0
RCLK      BIT    P2.1
SCLK      BIT    P2.2
CLEAR     BIT    P2.3
ENABLE2   BIT    P2.4
RCLK2     BIT    P2.5
SCLK2     BIT    P2.6
CLEAR2    BIT    P2.7
```

```
WAKTU     EQU    -1000
```

```
BARIS0    equ    10H
BARIS1    equ    2EH
BARIS2    equ    4CH
BARIS3    equ    6AH
BARIS4    equ    88H
BARIS5    equ    0A6H
BARIS6    equ    0C4H
BARIS7    equ    0E2H
```

```
ORG 00H
MOV R7,#0
SJMP MAIN ; lompat ke program utama
```

```
;-----
; ; ISI vektor interupsi port serial
```

```
ORG 23H ; lokasi vektor interupsi port serial
LJMP SERINT ; karena panjang lompat ke SERINT
```

```
;-----
ORG 030H ; program utama mulai di sini
MAIN: MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
      MOV TH1,#0FDH ; 9600 baud rate
      MOV SCON,#50H ; Mode serial: 8-bit UART
      SETB TR1 ; Jalankan Timer 1
```

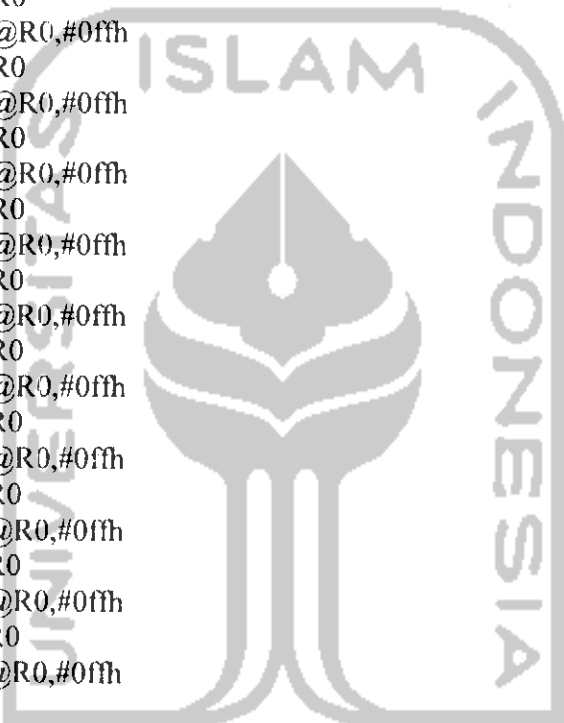
```
;-----
```



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

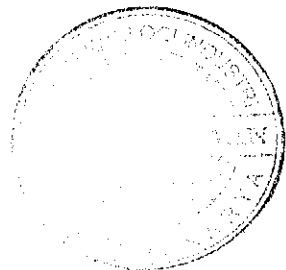
```



```

MOV R0,#38H ;BARIS2
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```




```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

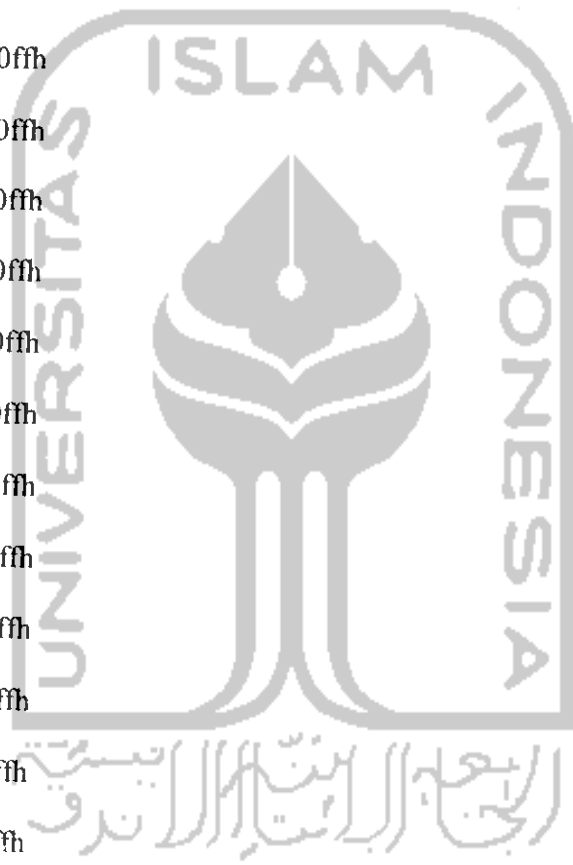
MOV R0,#60H                                     ;BARIS4
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

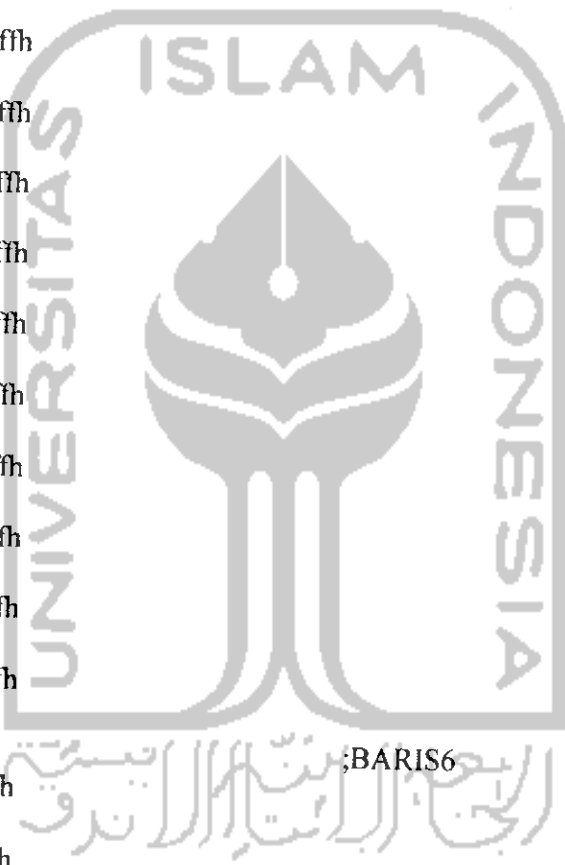
```

MOV R0,#741H                                     ;BARIS5

```



```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#88H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
```



```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#9CH;BARIS7
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
```



```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;-----
tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
```

```
MOV R0,#23H
```

```
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```

```
CJNE R0,#0FH,LANJUT_BARIS0
```

```
ACALL NYALA0
```

ISIBARIS1:

```
MOV P2,#10001000B
```

```
MOV R0,#37H
```

```
MOV A,@R0
```

LANJUT_BARIS1:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```

```
CJNE R0,#23H,LANJUT_BARIS1
```

```
ACALL NYALA1
```

ISIBARIS2:

```
MOV P2,#10001000B
```

```
MOV R0,#4BH
```

```
MOV A,@R0
```

LANJUT_BARIS2:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2
```

ISIBARIS3:

```
MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0
```

LANJUT_BARIS3:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3
```

ISIBARIS4:

```
MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0
```

LANJUT_BARIS4:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4
```

ISIBARIS5:

```
MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0
```

LANJUT_BARIS5:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5
```

ISIBARIS6:

```
MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0
```

LANJUT_BARIS6:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#87H,LANJUT_BARIS6
ACALL NYALA6
```

ISIBARIS7:

```
MOV P2,#10001000B
MOV R0,#0AFH
MOV A,@R0
```

LANJUT_BARIS7:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#9BH,LANJUT_BARIS7
ACALL NYALA7
```

JMP TAMPIL

NYALA0:

```
MOV P1,#11111110B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA1:

```
MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA2:

```
MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA3:



```
MOV P1,#11110111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA4:

```
MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA5:

```
MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA6:

```
MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA7:

```
MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

ISIBARIS:

```
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
```

```

RR    A
MOV  P0,A
MOV  P2,#1001100B
MOV  P2,#10001000B
RR    A
MOV  P0,A
MOV  P2,#11001100B
MOV  P2,#10001000B
RR    A
MOV  P0,A
MOV  P2,#11001100B
MOV  P2,#10001000B
RR    A
MOV  P0,A
MOV  P2,#11001100B
MOV  P2,#10001000B
RR    A
MOV  P0,A
MOV  P2,#11001100B
MOV  P2,#10001000B
RR    A
MOV  P0,A
MOV  P2,#11001100B
MOV  P2,#10001000B

```

RET

TUNDA:

```

MOV  TH0,#HIGH WAKTU
MOV  TL0,#LOW WAKTU
SETB TR0
JNB  TF0,$
CLR  TR0
CLR  TF0

```

RET

SERINT:

```

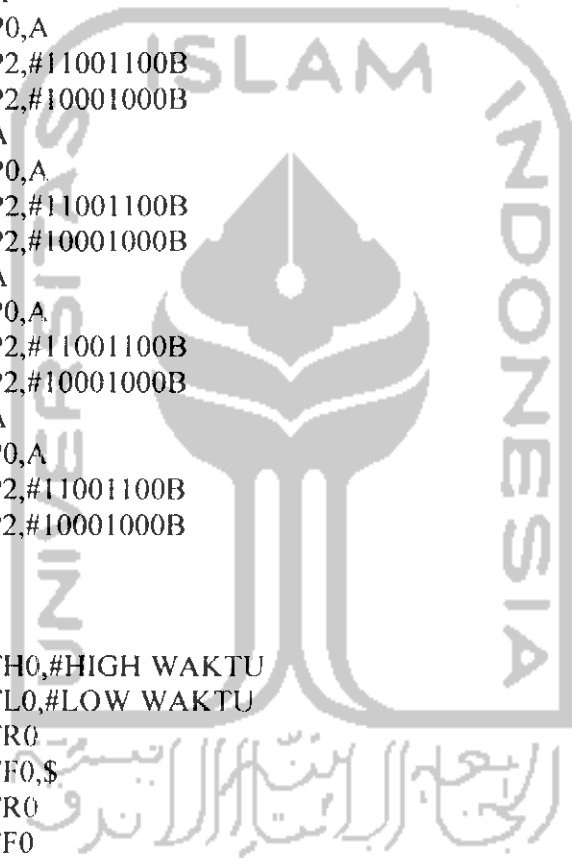
JB    RI,AMBIL_DATA
CLR  TI

```

RETI

AMBIL_DATA:

CEK_1ST:




```
CJNE R7,#0,DATA_160
MOV R7,#1
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,'#A',LANJUT
CLR RI
JMP LANJUT
```

DATA_160:

```
CJNE R5,'#A',SALAH_DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR RI
MOV @R1,A
CLR TI
INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H
```

LANJUT:

```
MOV A,R3
RETI
```

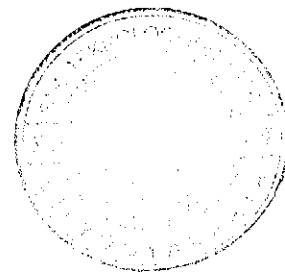
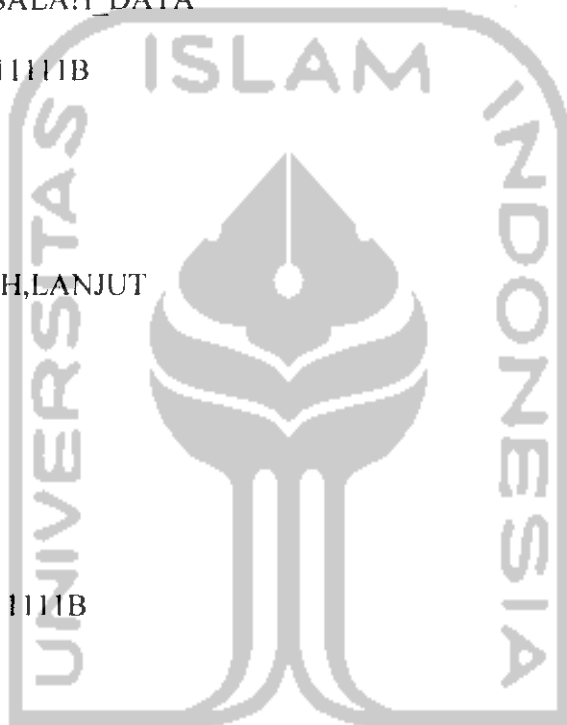
SALAH_DATA:

```
MOV P1,#11111111B
MOV B,SBUF
CLR RI
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
```

CONTI:

```
RETI
```

END



MIKROKONTROLER B

ENABLE	BIT	P2.0
RCLK	BIT	P2.1
SCLK	BIT	P2.2
CLEAR	BIT	P2.3
ENABLE2	BIT	P2.4
RCLK2	BIT	P2.5
SCLK2	BIT	P2.6
CLEAR2	BIT	P2.7

WAKTU EQU -1000

BARIS0	equ	10H
BARIS1	equ	2EH
BARIS2	equ	4CH
BARIS3	equ	6AH
BARIS4	equ	88H
BARIS5	equ	0A6H
BARIS6	equ	0C4H
BARIS7	equ	0E2H

```
ORG 00H
MOV R7,#0
SJMP MAIN ; lompat ke program utama
```

```
; ISI vektor interupsi port serial
```

```
ORG 23H ; lokasi vektor interupsi port serial
LJMP SERINT ; karena panjang lompat ke SERINT
```

```
ORG 030H ; program utama mulai di sini
MAIN: MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
MOV TH1,#0FDH ; 9600 baud rate
MOV SCON,#50H ; Mode serial: 8-bit UART
SETB TR1 ; Jalankan Timer 1
```

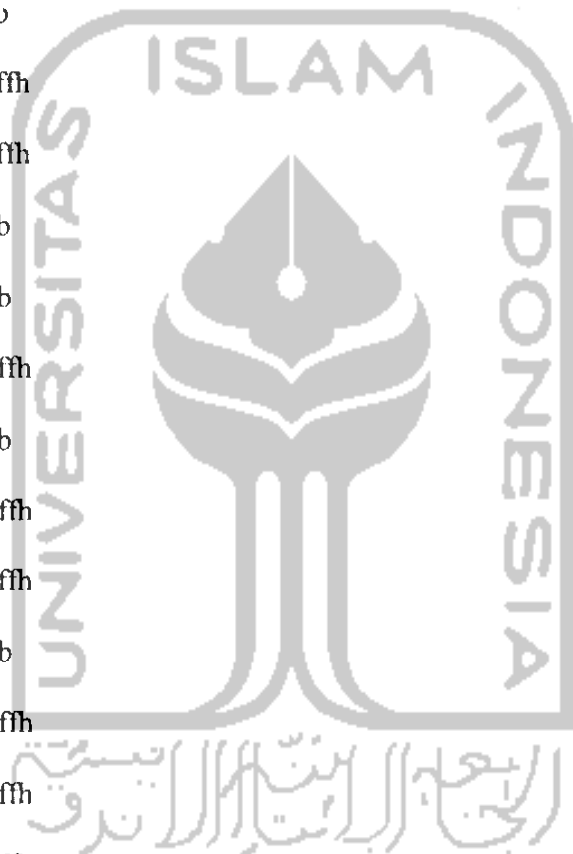
```
SETB ES
SETB EA
MOV R1,#10H
MOV R4,#10H
```

; ALAMAT MEMORY

START:

```
MOV R0,#10H ;BARIS0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#24H ;BARIS1
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
```



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```



;BARIS2

الجامعة الإسلامية
 الربيعية
 الاندونيسية

```

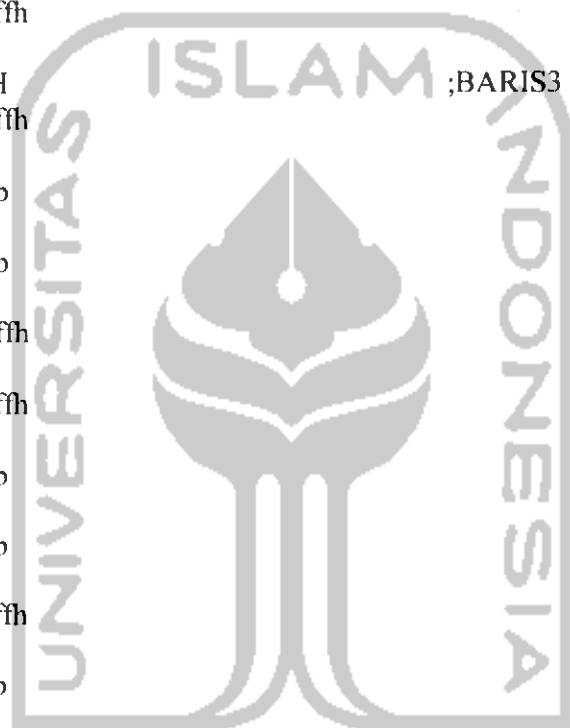
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

MOV R0,#4CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```



```

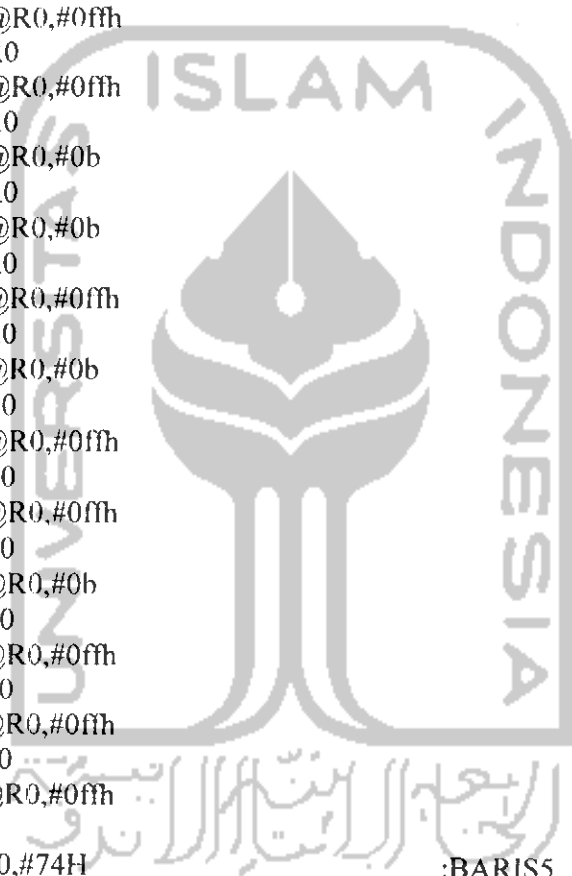
MOV R0,#60H ;BARIS4
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

```

```

MOV R0,#74H ;BARIS5
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

```



```

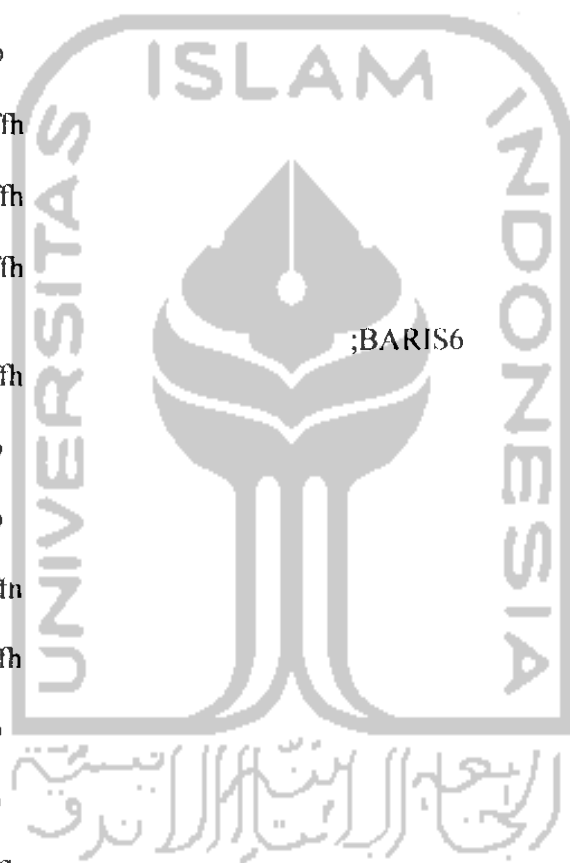
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

MOV R0,#88H
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

```



;BARIS6

```

MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

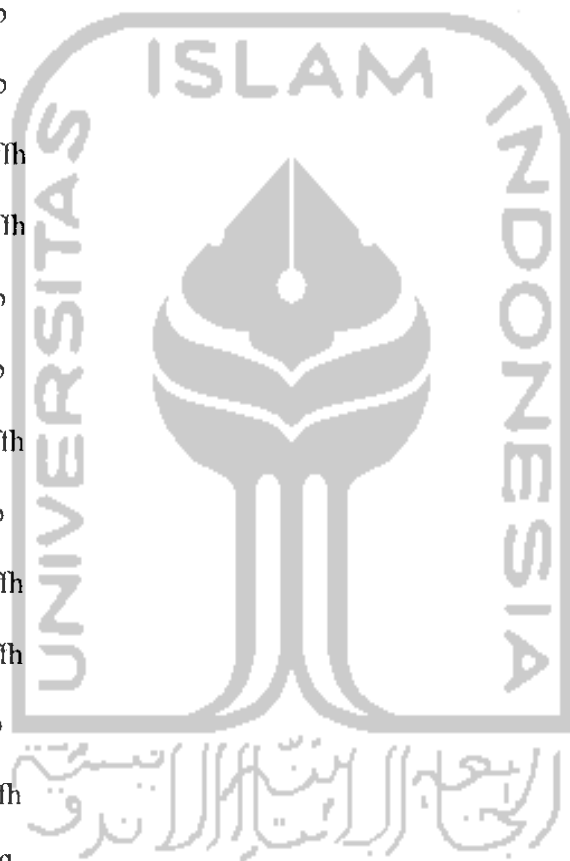
```

```

MOV R0,#9CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```
;BARIS7
```



tampil:

SIAPKAN:

```
MOV P2,#0
```


ISIBARIS0:

MOV P2,#10001000B
MOV R0,#23H
MOV A,@R0

LANJUT_BARIS0:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#0FH,LANJUT_BARIS0
ACALL NYALA0

ISIBARIS1:

MOV P2,#10001000B
MOV R0,#37H
MOV A,@R0

LANJUT_BARIS1:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#23H,LANJUT_BARIS1
ACALL NYALA1

ISIBARIS2:

MOV P2,#10001000B
MOV R0,#4BH
MOV A,@R0

LANJUT_BARIS2:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2

ISIBARIS3:

MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0

LANJUT_BARIS3:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3

ISIBARIS4:

MOV P2,#10001000B

MOV R0,#73H

MOV A,@R0

LANJUT_BARIS4:

ACALL ISIBARIS

DEC R0

MOV A,@R0

CJNE R0,#5FH,LANJUT_BARIS4

ACALL NYALA4

ISIBARIS5:

MOV P2,#10001000B

MOV R0,#87H

MOV A,@R0

LANJUT_BARIS5:

ACALL ISIBARIS

DEC R0

MOV A,@R0

CJNE R0,#73H,LANJUT_BARIS5

ACALL NYALA5

ISIBARIS6:

MOV P2,#10001000B

MOV R0,#9BH

MOV A,@R0

LANJUT_BARIS6:

ACALL ISIBARIS

DEC R0

MOV A,@R0

CJNE R0,#87H,LANJUT_BARIS6

ACALL NYALA6

ISIBARIS7:

MOV P2,#10001000B

MOV R0,#0A1'H

MOV A,@R0

LANJUT_BARIS7:

ACALL ISIBARIS

DEC R0

MOV A,@R0

CJNE R0,#9BH,LANJUT_BARIS7

ACALL NYALA7

JMP TAMPIL

NYALA0:

MOV P1,#11111110B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA1:

MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA2:

MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA3:

MOV P1,#11110111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

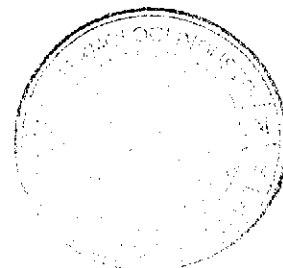
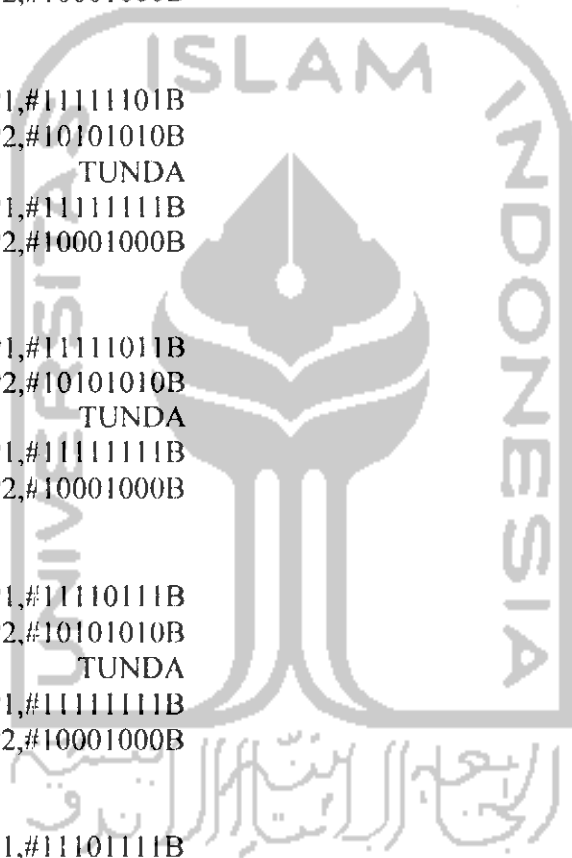
NYALA4:

MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA5:

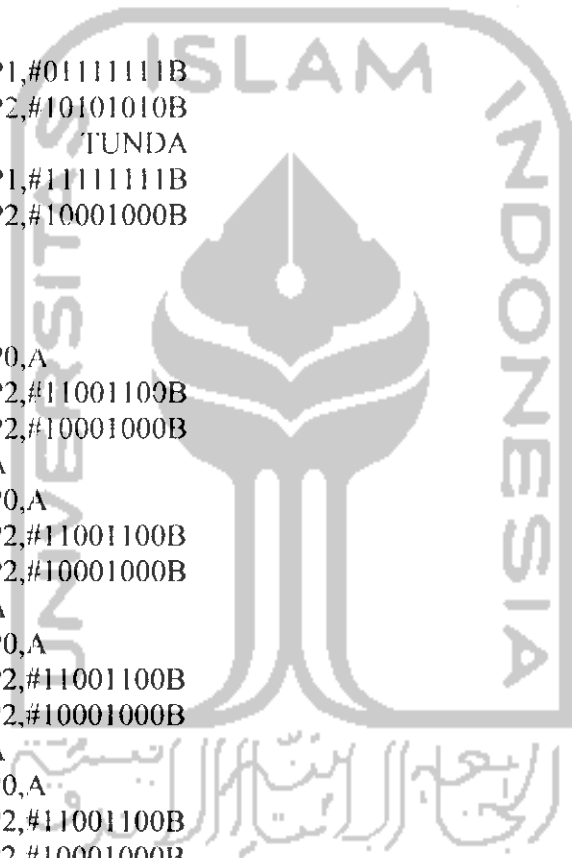
MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B



```
MOV P2,#10001000B
RET
NYALA6:
MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

```
RET
NYALA7:
MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

```
RET
ISIBARIS:
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
```



```
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
```

RET

TUNDA:

```
MOV TH0,#HIGH WAKTU
MOV TL0,#LOW WAKTU
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
```

RET

SERINT:

```
JB RI,AMBIL_DATA
CLR TI
```

RETI

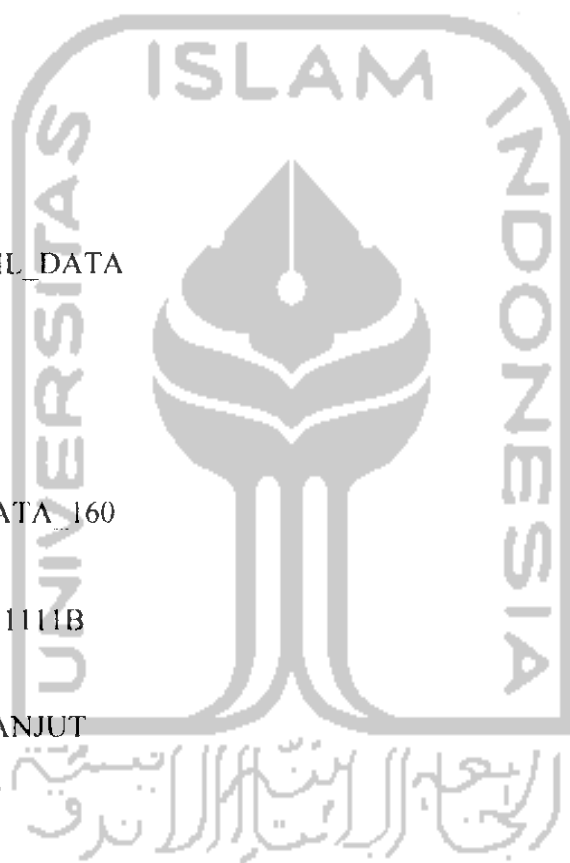
AMBIL_DATA:

CEK_1ST:

```
CJNE R7,#0,DATA_160
MOV R7,#1
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,#'B',LANJUT
CLR RI
JMP LANJUT
```

DATA_160:

```
CJNE R5,#'B',SALAH_DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR RI
MOV @R1,A
CLR TI
```



```

INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H

```

```

LANJUT:
MOV A,R3
RETI

```

```

SALAH_DATA:
MOV P1,#11111111B
MOV B,SBUF
CLR RI
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H

```

```

CONTI:
RETI

```

```

END

```

```

ENABLE BIT P2.0
RCLK BIT P2.1
SCLK BIT P2.2
CLEAR BIT P2.3
ENABLE2 BIT P2.4
RCLK2 BIT P2.5
SCLK2 BIT P2.6
CLEAR2 BIT P2.7

```

```

WAKTU EQU -1000

```

```

BARIS0 equ 10H
BARIS1 equ 2EH
BARIS2 equ 4CH
BARIS3 equ 6AH

```



MIKROKONTROLER C

```

BARIS4      equ    88H
BARIS5      equ    0A6H
BARIS6      equ    0C4H
BARIS7      equ    0E2H

```

```

ORG 00H
MOV R7,#0
SJMP MAIN      ; lompat ke program utama

```

```

;-----
;                               ; ISI vektor interupsi port serial

```

```

ORG 23H      ; lokasi vektor interupsi port serial
LJMP SERINT  ; karena panjang lompat ke SERINT

```

```

;-----
ORG 030H      ; program utama mulai di sini
MAIN:        MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
             MOV TH1,#0FDH      ; 9600 baud rate
             MOV SCON,#50H      ; Mode serial: 8-bit UART
             SETB TR1           ; Jalankan Timer 1

```

```

;-----
SETB ES
SETB EA
MOV R1,#10H  ; ALAMAT MEMORY
MOV R4,#101H

```

```

START:
MOV R0,#10H ;BARIS0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#24H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

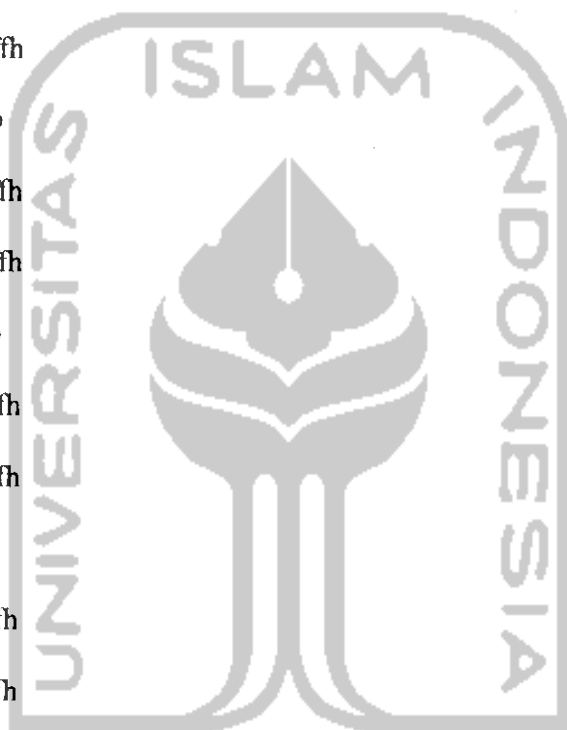


BARIS1


```
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#38H ;BARIS2
```

```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

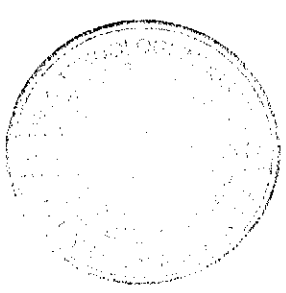
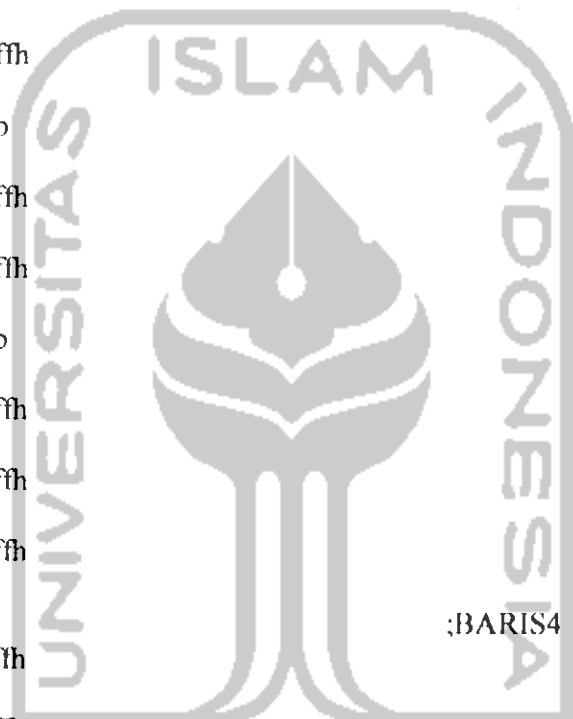


```
MOV R0,#4CH ;BARIS3
MOV @R0,#0ffh
INC R0
```

```

MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#60H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

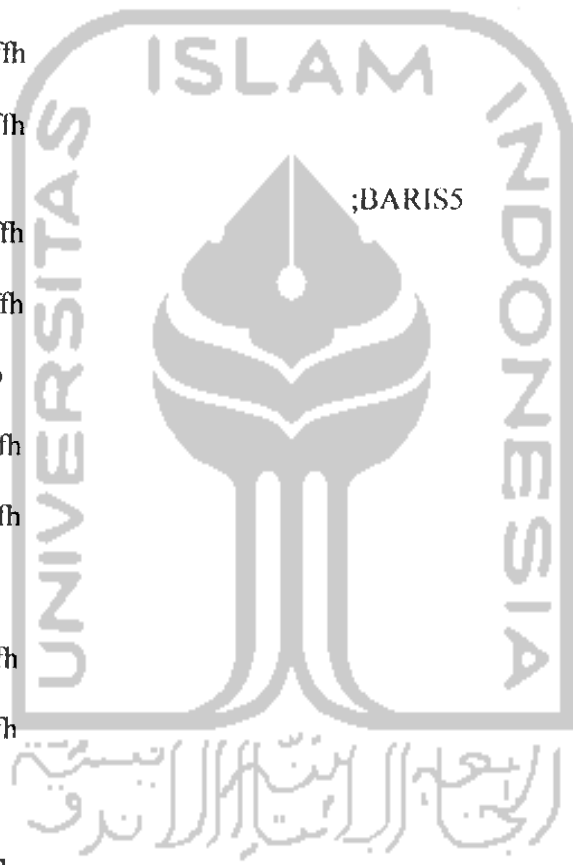
```



```

MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#74H ;BARIS5
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0

```



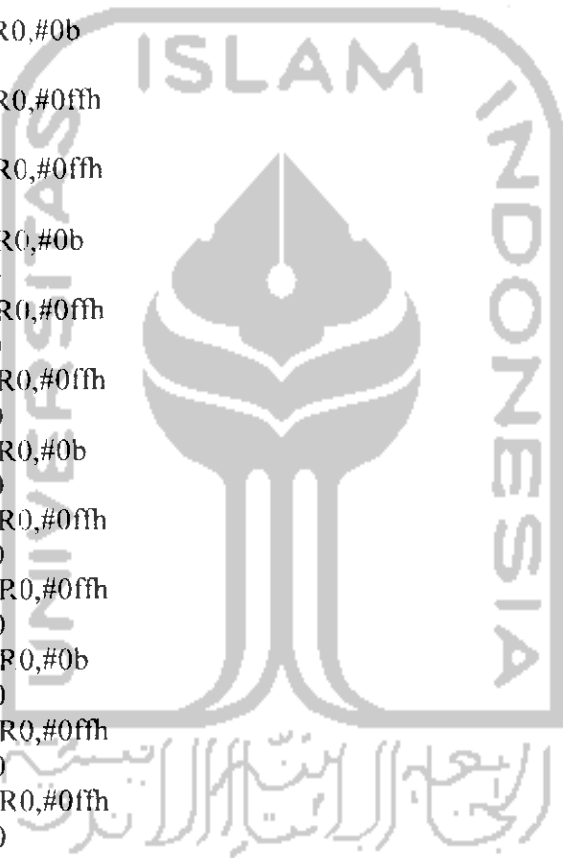
```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#88H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;BARIS6

```
MOV R0,#9CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
```

;BARIS7



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

;-----
tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
```

```
MOV R0,#23H
```

```
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```

```
CJNE R0,#0FH,LANJUT_BARIS0
```

```
ACALL NYALA0
```

ISIBARIS1:

```
MOV P2,#10001000B
```

```
MOV R0,#37H
```

```
MOV A,@R0
```

LANJUT_BARIS1:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#23H,LANJUT_BARIS1
ACALL NYALA1

ISIBARIS2:

MOV P2,#10001000B
MOV R0,#4BH
MOV A,@R0

LANJUT_BARIS2:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2

ISIBARIS3:

MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0

LANJUT_BARIS3:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3

ISIBARIS4:

MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0

LANJUT_BARIS4:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4

ISIBARIS5:

MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0

LANJUT_BARIS5:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5

ISIBARIS6:

MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0

LANJUT_BARIS6:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#87H,LANJUT_BARIS6
ACALL NYALA6

ISIBARIS7:

MOV P2,#10001000B
MOV R0,#0AFH
MOV A,@R0

LANJUT_BARIS7:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#9BH,LANJUT_BARIS7
ACALL NYALA7

JMP TAMPIL

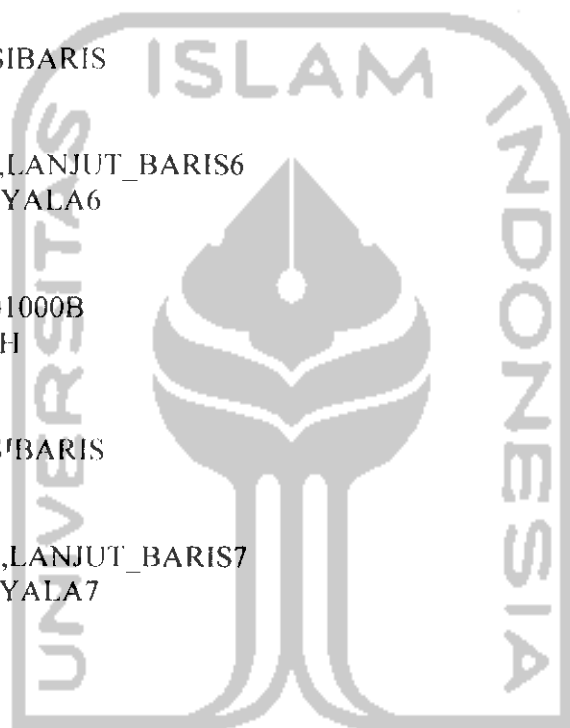
NYALA0:

MOV P1,#1111110B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#1111111B
MOV P2,#10001000B

RET

NYALA1:

MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA



MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA2:

MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA3:

MOV P1,#11110111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA4:

MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA5:

MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

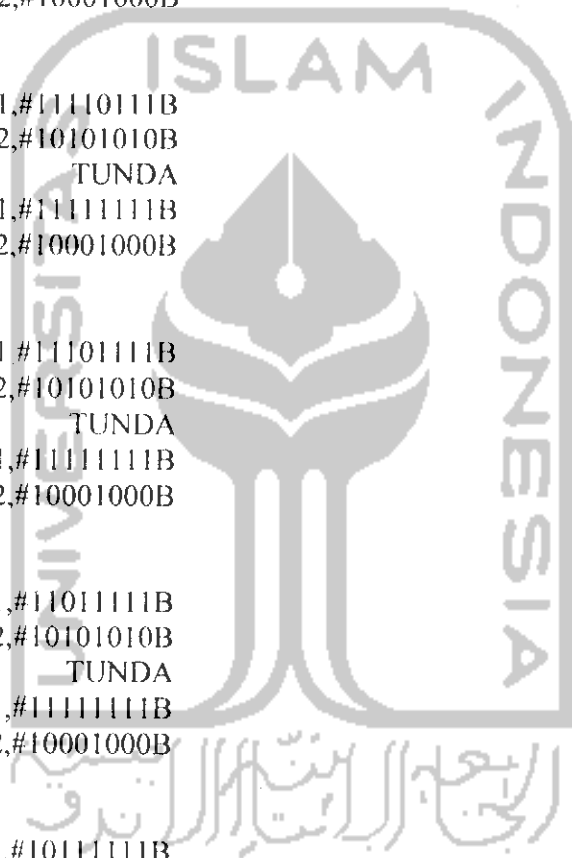
NYALA6:

MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA7:

MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B




```
CLR TF0  
RET
```

```
SERINT:  
JB RI,AMBIL_DATA  
CLR TI  
RETI
```

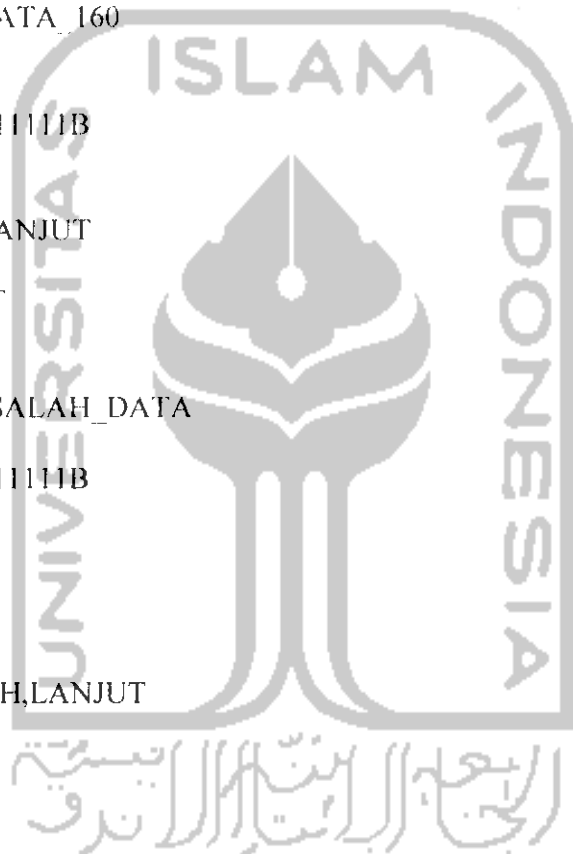
```
AMBIL_DATA:  
CEK_1ST:
```

```
CJNE R7,#0,DATA_160  
MOV R7,#1  
MOV R3,A  
MOV P1,#11111111B  
MOV A,SBUF  
MOV R5,A  
CJNE A,#'C',LANJUT  
CLR RI  
JMP LANJUT
```

```
DATA_160:  
CJNE R5,#'C',SALAH_DATA  
MOV R3,A  
MOV P1,#11111111B  
MOV A,SBUF  
CLR RI  
MOV @R1,A  
CLR TI  
INC R1  
CJNE R1,#0B0H,LANJUT  
MOV R7,#0  
MOV R1,#10H
```

```
LANJUT:  
MOV A,R3  
RETI
```

```
SALAH_DATA:  
MOV P1,#11111111B  
MOV B,SBUF  
CLR RI  
CLR TI
```

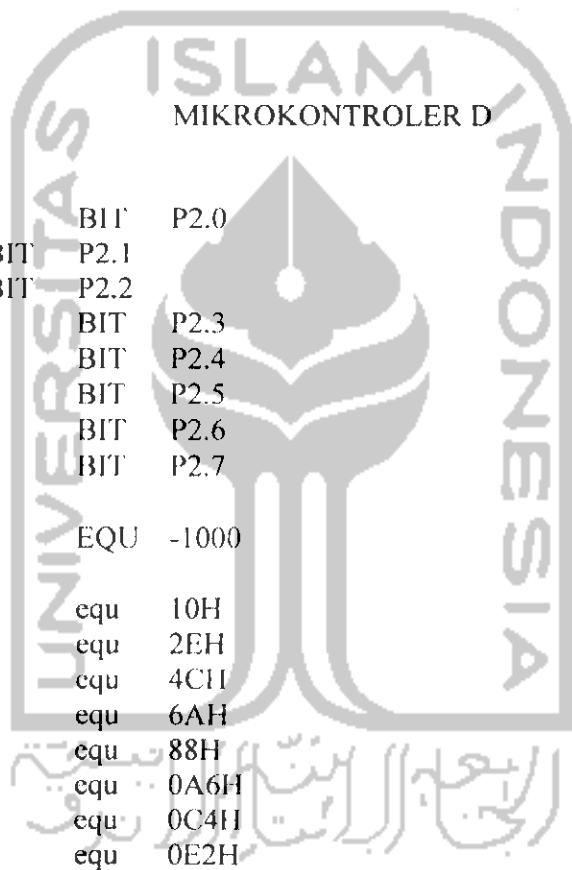


```

INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
CONTI:
RETI

END

```



```

ENABLE BIT P2.0
RCLK BIT P2.1
SCLK BIT P2.2
CLEAR BIT P2.3
ENABLE2 BIT P2.4
RCLK2 BIT P2.5
SCLK2 BIT P2.6
CLEAR2 BIT P2.7

WAKTU EQU -1000

BARIS0 equ 10H
BARIS1 equ 2EH
BARIS2 equ 4CH
BARIS3 equ 6AH
BARIS4 equ 88H
BARIS5 equ 0A6H
BARIS6 equ 0C4H
BARIS7 equ 0E2H

```

```

ORG 00H
MOV R7,#0
SJMP MAIN ; lompat ke program utama

```

```

; ISI vektor interupsi port serial

```

```
ORG 23H ; lokasi vektor interupsi port serial
LJMP SERINT ; karena panjang lompat ke SERINT
```

```
-----
ORG 0301H ; program utama mulai di sini
MAIN: MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
MOV TH1,#0FDH ; 9600 baud rate
MOV SCON,#50H ; Mode serial: 8-bit UART
SETB TR1 ; Jalankan Timer 1
```

```
-----
SETB ES
SETB EA
MOV R1,#10H
MOV R4,#10H
```

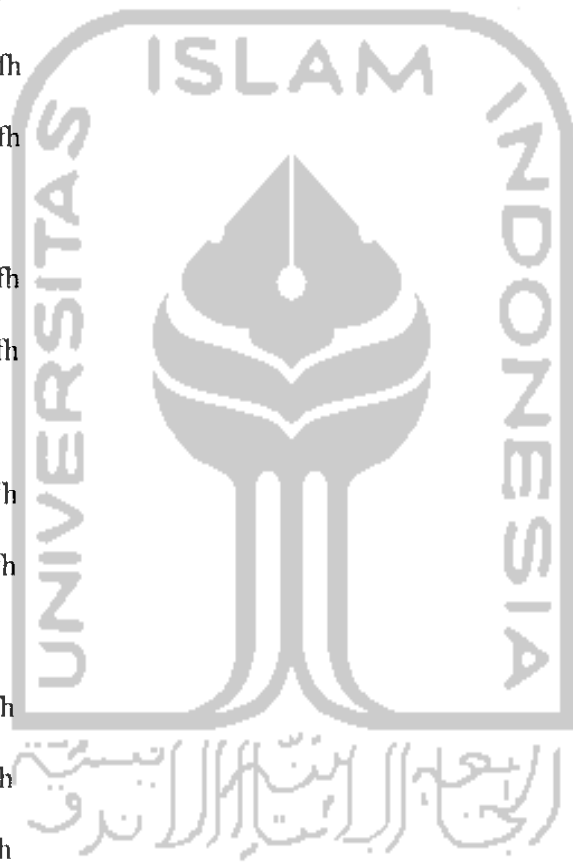
START:

```
MOV R0,#10H ;BARISO
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
```

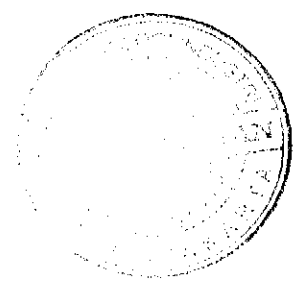


```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#24H ;BARIS1
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```



```
MOV R0,#38H ;BARIS2
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
```



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```



:BARIS3

```

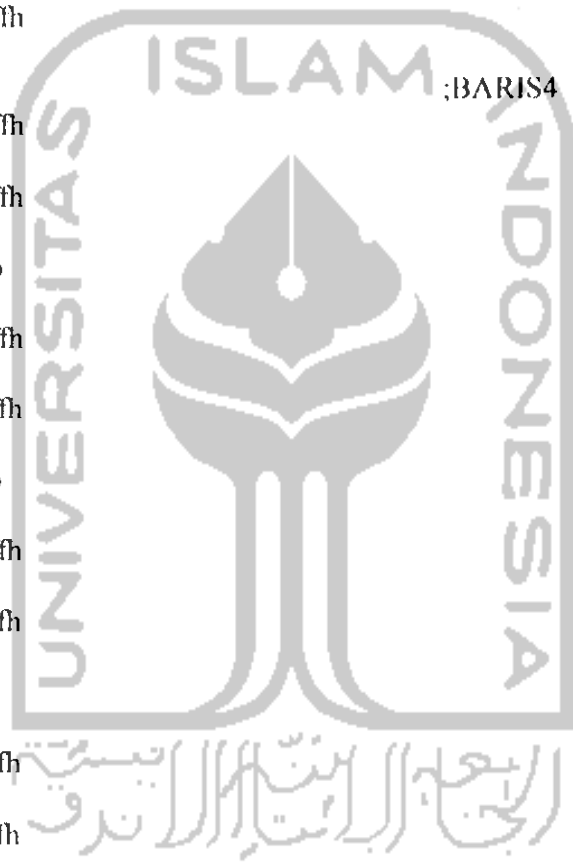
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

MOV R0,#6011
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

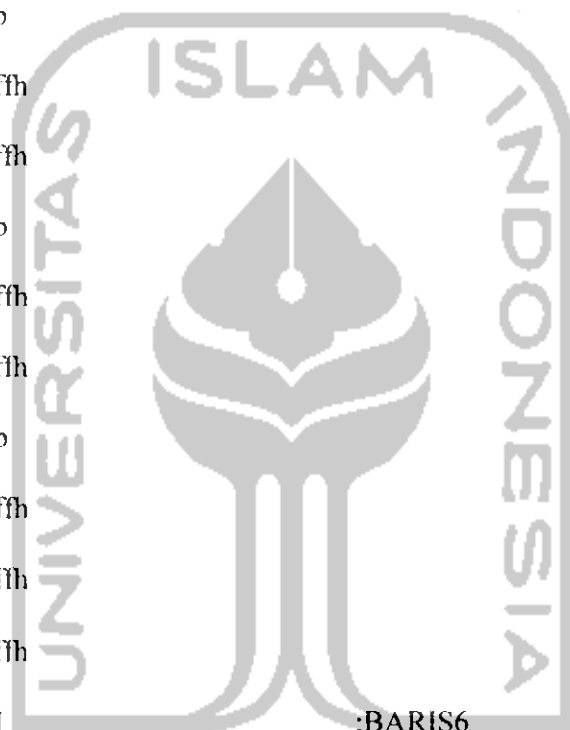


```

MOV R0,#74H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

;BARIS5



```

MOV R0,#88H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```

;BARIS6

الجامعة الإسلامية
 الربيعية
 لا اله الا الله
 لا شريك له
 محمد
 عبده
 ورسوله


```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#9CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```



الجامعة الإسلامية
 الرابحة الاستاذة الاندوف

```
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;-----
tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
MOV R0,#23H
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#0FH,LANJUT_BARIS0
ACALL NYALA0
```

ISIBARIS1:

```
MOV P2,#10001000B
MOV R0,#37H
MOV A,@R0
```

LANJUT_BARIS1:

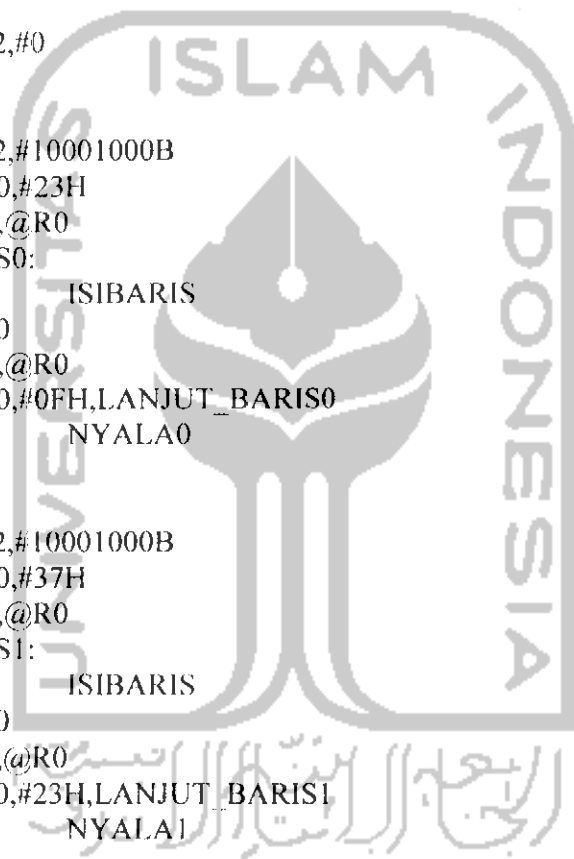
```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#23H,LANJUT_BARIS1
ACALL NYALA1
```

ISIBARIS2:

```
MOV P2,#10001000B
MOV R0,#4BH
MOV A,@R0
```

LANJUT_BARIS2:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2
```



ISIBARIS3:

MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0

LANJUT_BARIS3:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3

ISIBARIS4:

MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0

LANJUT_BARIS4:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4

ISIBARIS5:

MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0

LANJUT_BARIS5:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5

ISIBARIS6:

MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0

LANJUT_BARIS6:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#87H,LANJUT_BARIS6
ACALL NYALA6

ISIBARIS7:

MOV P2,#10001000B

MOV R0,#0AFH

MOV A,@R0

LANJUT_BARIS7:

ACALL ISIBARIS

DEC R0

MOV A,@R0

CJNE R0,#9BH,LANJUT_BARIS7

ACALL NYALA7

JMP TAMPIL

NYALA0:

MOV P1,#11111110B

MOV P2,#10101010B

ACALL TUNDA

MOV P1,#11111111B

MOV P2,#10001000B

RET

NYALAI:

MOV P1,#11111101B

MOV P2,#10101010B

ACALL TUNDA

MOV P1,#11111111B

MOV P2,#10001000B

RET

NYALA2:

MOV P1,#11111011B

MOV P2,#10101010B

ACALL TUNDA

MOV P1,#11111111B

MOV P2,#10001000B

RET

NYALA3:

MOV P1,#11110111B

MOV P2,#10101010B

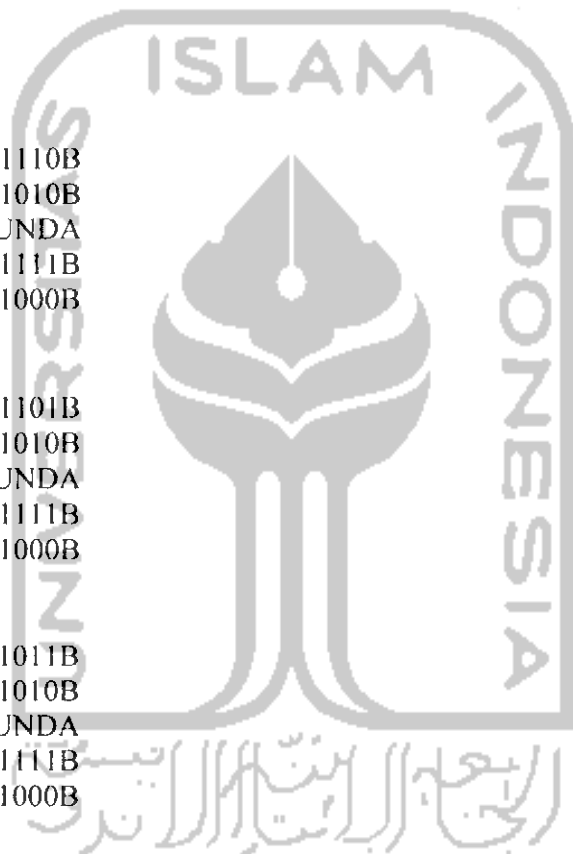
ACALL TUNDA

MOV P1,#11111111B

MOV P2,#10001000B

RET

NYALA4:



MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA5:

MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA6:

MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

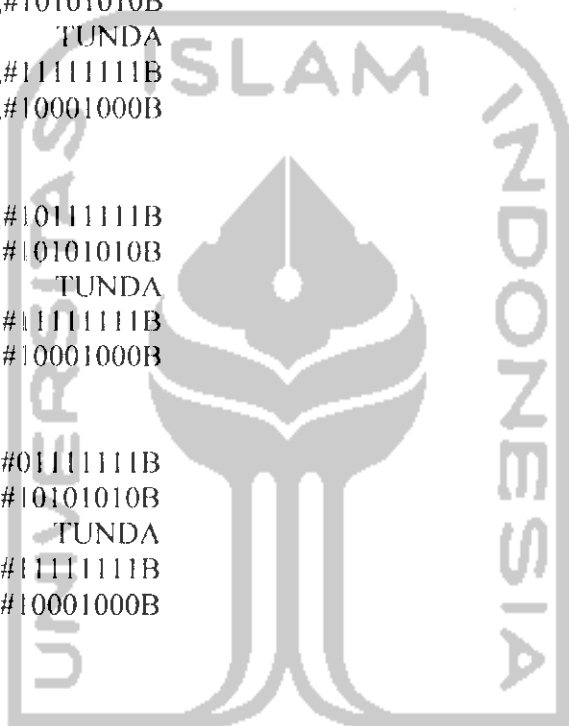
NYALA7:

MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

ISIBARIS:

MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B



```

MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B

```

RET

TUNDA:

```

MOV TH0,#HIGH WAKTU
MOV TL0,#LOW WAKTU
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0

```

RET

SERINT:

```

JB RI,AMBIL_DATA
CLR TI

```

RETI

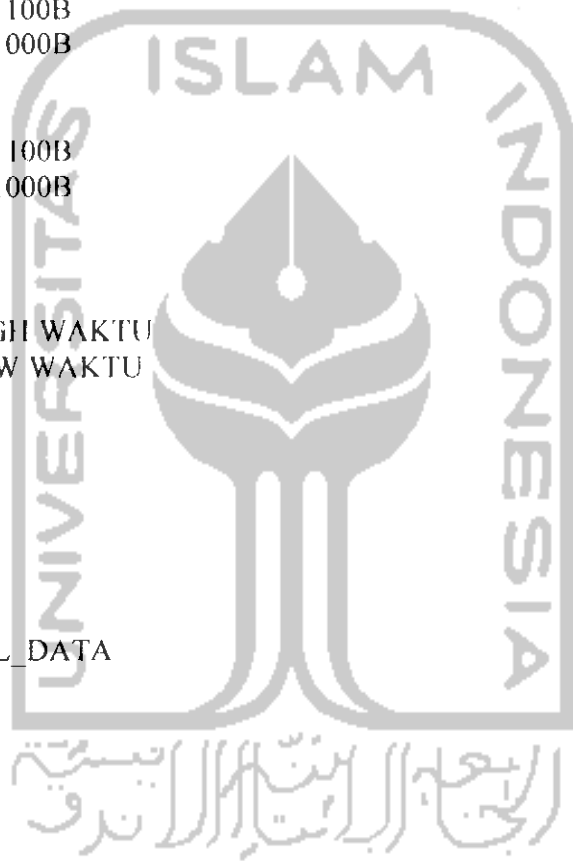
AMBIL_DATA:

CEK_1ST:

```

CJNE R7,#0,DATA_160
MOV R7,#1
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,#'D',LANJUT

```



```
CLR R1
JMP LANJUT
```

DATA_160:

```
CJNE R5,#'D',SALAH_DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR R1
MOV @R1,A
CLR TI
INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H
```

LANJUT:

```
MOV A,R3
RETI
```

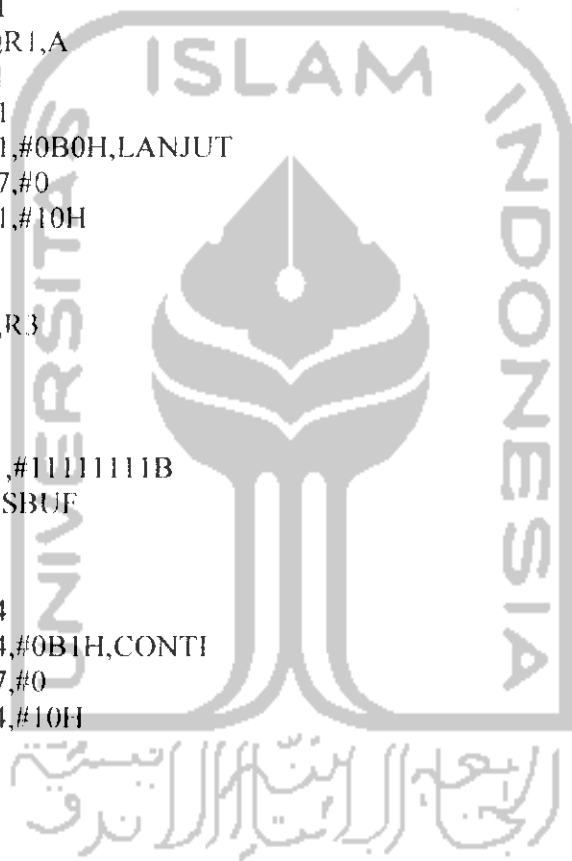
SALAH_DATA:

```
MOV P1,#11111111B
MOV B,SBUF
CLR R1
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
```

CONTI:

```
RETI
```

END



MIKROKONTROLER E

```
ENABLE      BIT    P2.0
RCLK        BIT    P2.1
SCLK        BIT    P2.2
```

```

CLEAR          BIT    P2.3
ENABLE2        BIT    P2.4
RCLK2          BIT    P2.5
SCLK2          BIT    P2.6
CLEAR2         BIT    P2.7

```

```

WAKTU          EQU    -1000

```

```

BARIS0         equ    10H
BARIS1         equ    2EH
BARIS2         equ    4CH
BARIS3         equ    6AH
BARIS4         equ    88H
BARIS5         equ    0A6H
BARIS6         equ    0C4H
BARIS7         equ    0E2H

```

```

ORG 00H
MOV R7,#0
SJMP MAIN      ; lompat ke program utama

```

```

;-----
; ISI vektor interupsi port serial

```

```

ORG 23H      ; lokasi vektor interupsi port serial
LJMP SERINT  ; karena panjang lompat ke SERINT

```

```

;-----
ORG 030H      ; program utama mulai di sini
MAIN: MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
      MOV TH1,#0FDH      ; 9600 baud rate
      MOV SCON,#50H      ; Mode serial: 8-bit UART
      SETB TR1           ; Jalankan Timer 1

```

```

;-----
SETB ES
SETB EA
MOV R1,#10H      ; ALAMAT MEMORY
MOV R4,#10H

```

```

START:

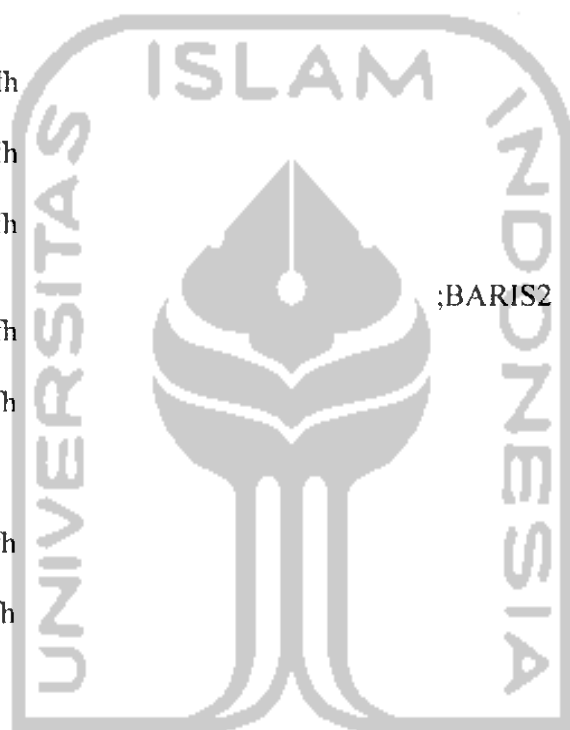
```



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV R0,#38H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```



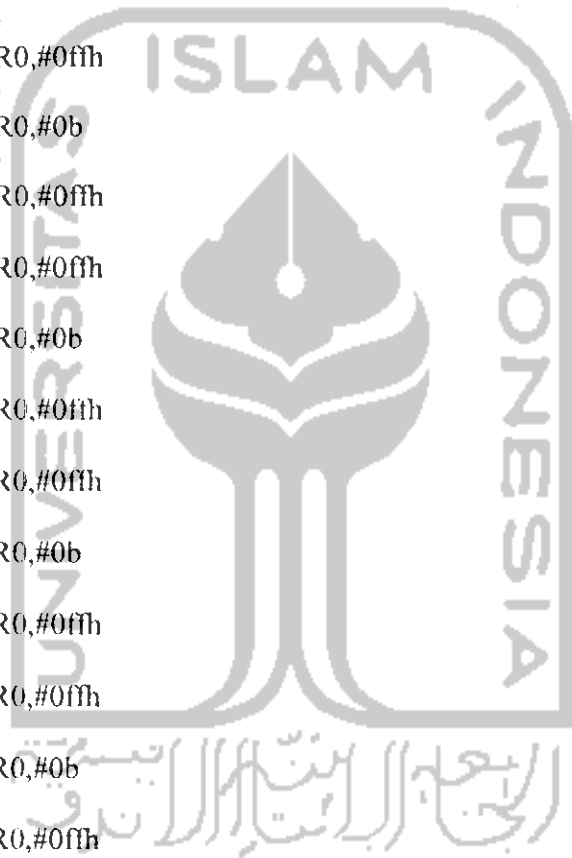
;BARIS2

الجامعة الإسلامية الاندونيسية

```
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#4CH ;BARIS3
```

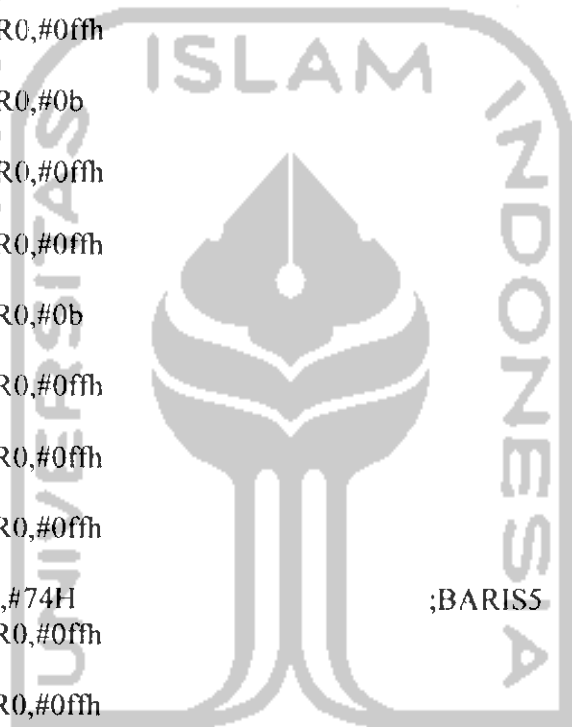
```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
MOV R0,#60H ;BARIS4
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
```



```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#74H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b

```



;BARISS-

الجامعة الإسلامية
 الربيعية
 في
 بريس

```

INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

```

MOV R0,#88H;BARIS6
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```



```

MOV R0,#9CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

```

;BARIS7

tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
```

```
MOV R0,#23H
```

```
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```



```
CJNE R0,#0FH,LANJUT_BARIS0
ACALL NYALA0
```

ISIBARIS1:

```
MOV P2,#10001000B
MOV R0,#37H
MOV A,@R0
```

LANJUT_BARIS1:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#23H,LANJUT_BARIS1
ACALL NYALA1
```

ISIBARIS2:

```
MOV P2,#10001000B
MOV R0,#4BH
MOV A,@R0
```

LANJUT_BARIS2:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2
```

ISIBARIS3:

```
MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0
```

LANJUT_BARIS3:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3
```

ISIBARIS4:

```
MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0
```

LANJUT_BARIS4:

```
ACALL ISIBARIS
DEC R0
```

```
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4
```

ISIBARIS5:

```
MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0
```

LANJUT_BARIS5:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5
```

ISIBARIS6:

```
MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0
```

LANJUT_BARIS6:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#87H,LANJUT_BARIS6
ACALL NYALA6
```

ISIBARIS7:

```
MOV P2,#10001000B
MOV R0,#0AFH
MOV A,@R0
```

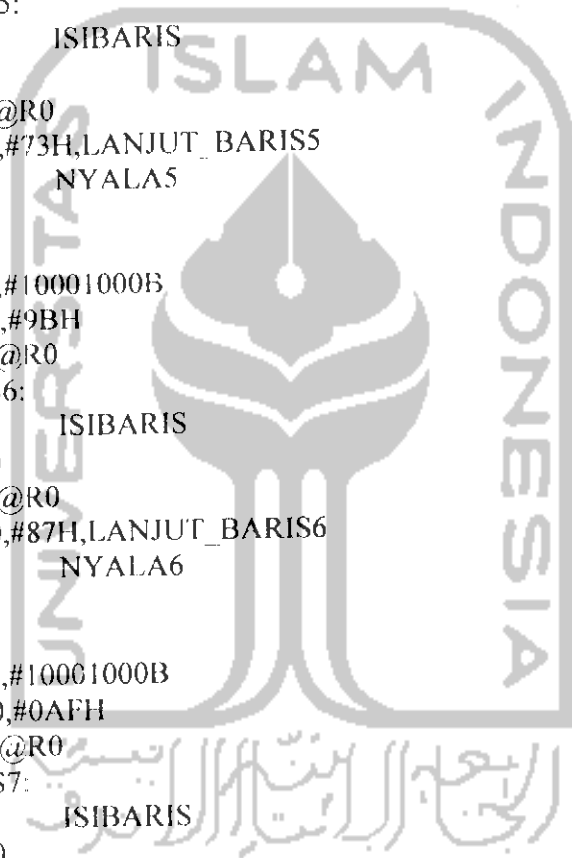
LANJUT_BARIS7:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#9BH,LANJUT_BARIS7
ACALL NYALA7
```

JMP TAMPIL

NYALA0:

```
MOV P1,#1111110B
MOV P2,#10101010B
ACALL TUNDA
```



MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA1:

MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA2:

MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA3:

MOV P1,#11110111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA4:

MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

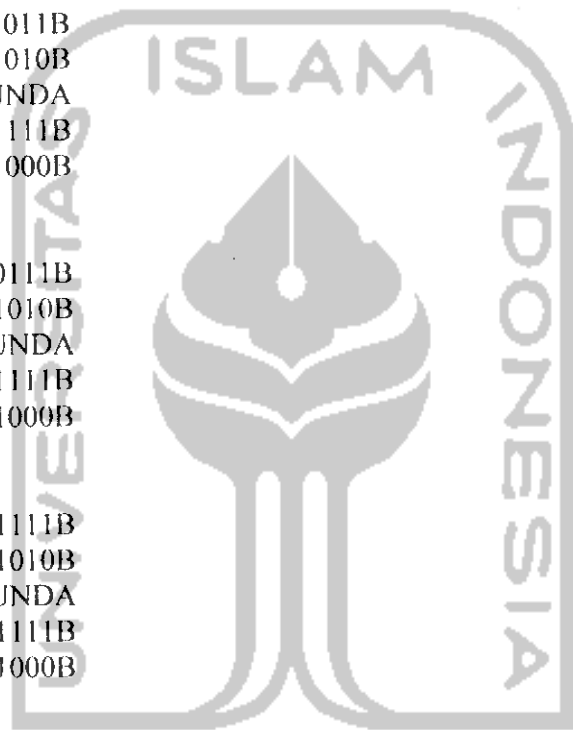
NYALA5:

MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA6:

MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B



الجامعة الإسلامية
INDONESIA

MOV P2,#10001000B

RET

NYALA7:

MOV P1,#01111111B

MOV P2,#10101010B

ACALL TUNDA

MOV P1,#11111111B

MOV P2,#10001000B

RET

ISIBARIS:

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

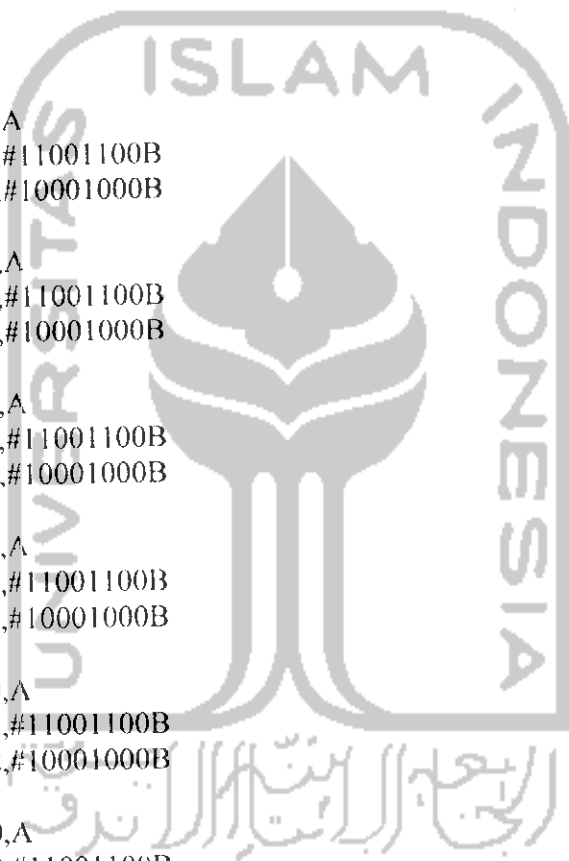
RR A

MOV P0,A

MOV P2,#11001100B

MOV P2,#10001000B

RET



TUNDA:

```
MOV TH0,#HIGH WAKTU
MOV TL0,#LOW WAKTU
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
```

RET

SERINT:

```
JB RI,AMBIL_DATA
CLR TI
```

RETI

AMBIL_DATA:
CEK_1ST:

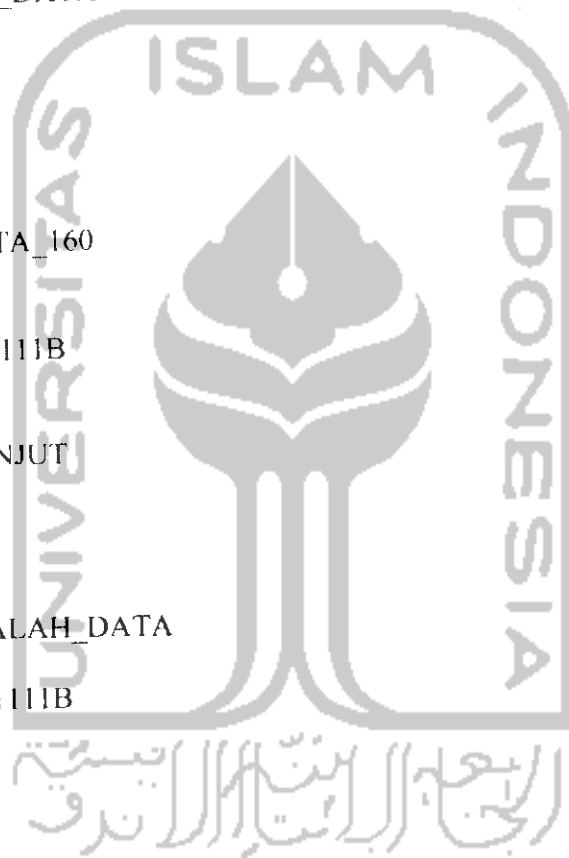
```
CJNE R7,#0,DATA_160
MOV R7,#1
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,#'E',LANJUT
CLR RI
JMP LANJUT
```

DATA_160:

```
CJNE R5,#'E',SALAH_DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR RI
MOV @R1,A
CLR TI
INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H
```

LANJUT:

```
MOV A,R3
RETI
```



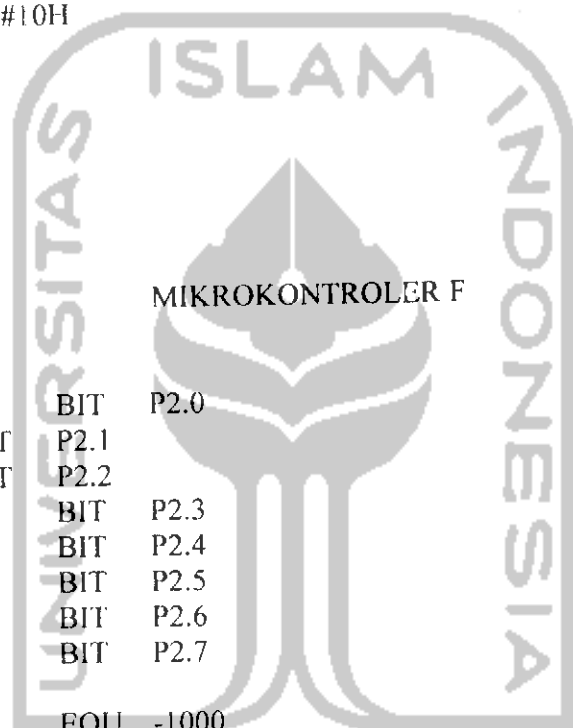
SALAH_DATA:

```
MOV P1,#11111111B
MOV B,SBUF
CLR RI
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
```

CONTI:

```
RETI
```

END



ENABLE	BIT	P2.0
RCLK	BIT	P2.1
SCLK	BIT	P2.2
CLEAR	BIT	P2.3
ENABLE2	BIT	P2.4
RCLK2	BIT	P2.5
SCLK2	BIT	P2.6
CLEAR2	BIT	P2.7

WAKTU EQU -1000

BARIS0	equ	10H
BARIS1	equ	2EH
BARIS2	equ	4CH
BARIS3	equ	6AH
BARIS4	equ	88H
BARIS5	equ	0A6H
BARIS6	equ	0C4H
BARIS7	equ	0E2H

```
ORG 0011
MOV R7,#0
SJMP MAIN
```

; lompat ke program utama

```

;-----
; ISI vektor interupsi port serial

ORG 23H ; lokasi vektor interupsi port serial
LJMP SERINT ; karena panjang lompat ke SERINT

```

```

;-----
ORG 030H ; program utama mulai di sini
MAIN: MOV TMOD,#00100001B ; timer 1 mode 2 (8-bit, ISI-ulang)
MOV TH1,#0FDH ; 9600 baud rate
MOV SCON,#50H ; Mode serial: 8-bit UART
SETB TR1 ; Jalankan Timer 1

```

```

;-----
SETB ES
SETB EA
MOV R1,#10H ; ALAMAT MEMORY
MOV R4,#10H

```

```

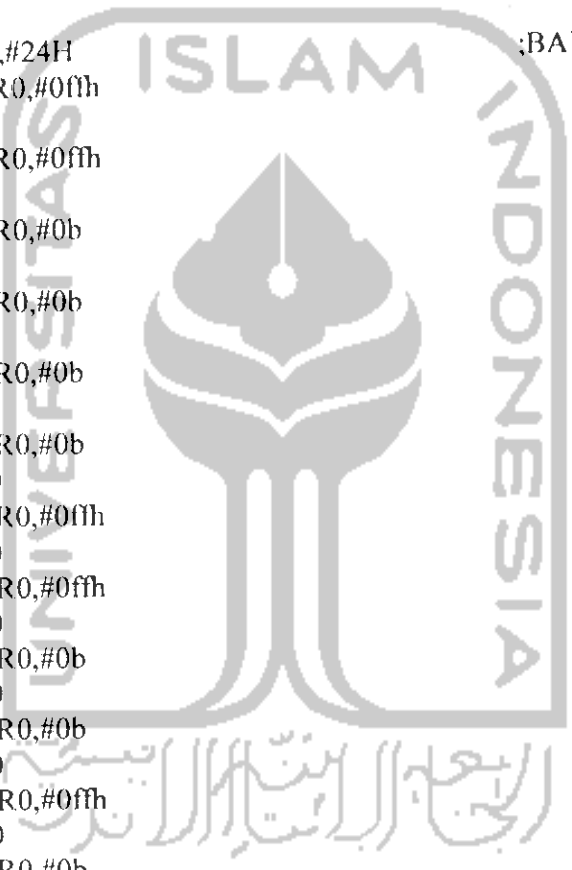
START:
MOV R0,#10H ;BARISO
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh

```

```
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#24H ;BARIS1
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#38H ;BARIS2
MOV @R0,#0ffh
INC R0
```



```

MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#4CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

```



```

MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

MOV R0,#60H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0

```




```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

```
MOV R0,#74H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;BARIS5



```
MOV R0,#88H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
```

;BARIS6

```

INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0fh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh

MOV R0,#9CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b

```



;BARIS7

```
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0b
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;-----
tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
```

```
MOV R0,#23H
```

```
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```

```
CJNE R0,#0FH,LANJUT_BARIS0
```

```
ACALL NYALAO
```

ISIBARIS1:

```
MOV P2,#10001000B
```

```
MOV R0,#37H
```

```
MOV A,@R0
```

LANJUT_BARIS1:

```
ACALL ISIBARIS
```

```
DEC R0
```

```
MOV A,@R0
```

```
CJNE R0,#23H,LANJUT_BARIS1
```

```
ACALL NYALAI
```

ISIBARIS2:

```
MOV P2,#10001000B
```

```
MOV R0,#4BH
```

```
MOV A,@R0
```

LANJUT_BARIS2:

```
ACALL ISIBARIS
```

```
DEC R0
MOV A,@R0
CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2
```

ISIBARIS3:

```
MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0
```

LANJUT_BARIS3:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3
```

ISIBARIS4:

```
MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0
```

LANJUT_BARIS4:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4
```

ISIBARIS5:

```
MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0
```

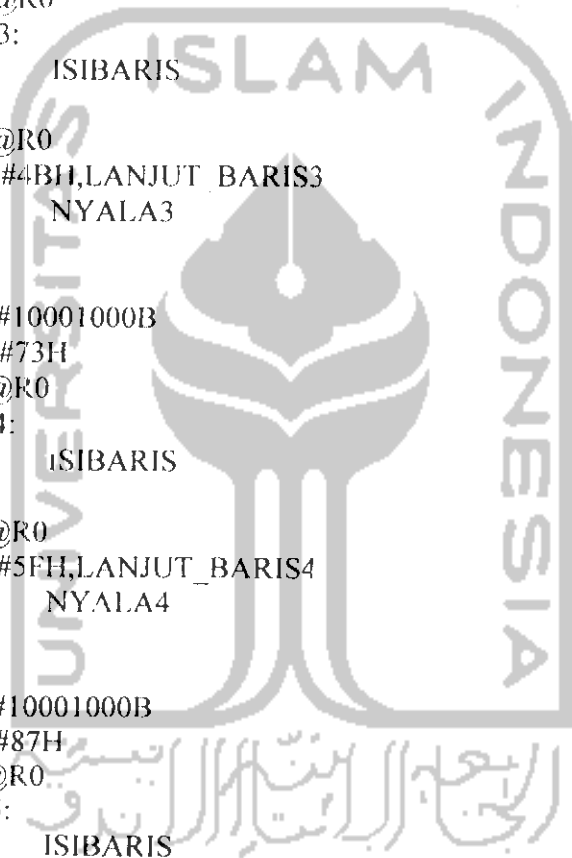
LANJUT_BARIS5:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5
```

ISIBARIS6:

```
MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0
```

LANJUT_BARIS6:



```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#87H,LANJUT BARIS6
ACALL NYALA6
```

ISIBARIS7:

```
MOV P2,#10001000B
MOV R0,#0AFH
MOV A,@R0
```

LANJUT BARIS7:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#9BH,LANJUT BARIS7
ACALL NYALA7
```

JMP TAMPIL

NYALA0:

```
MOV P1,#1111110B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#1111111B
MOV P2,#10001000B
```

RET

NYALA1:

```
MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#1111111B
MOV P2,#10001000B
```

RET

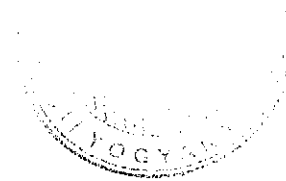
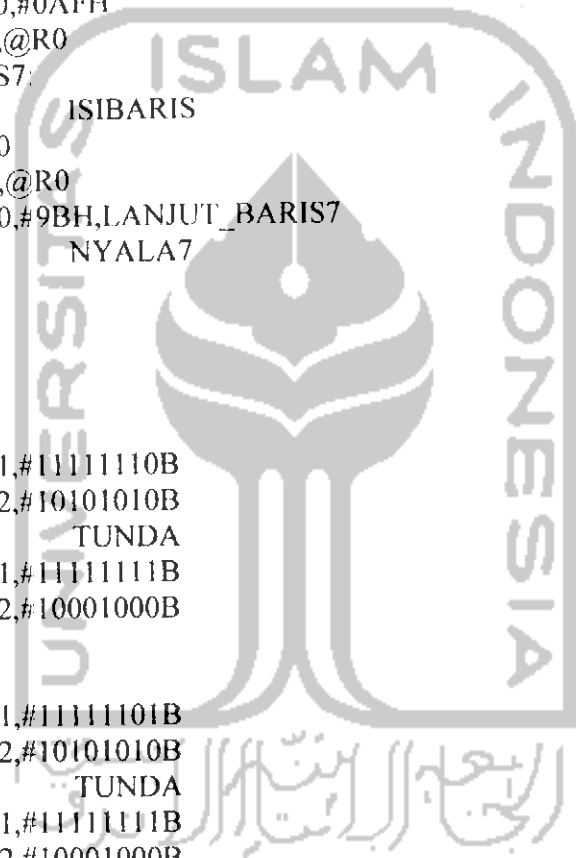
NYALA2:

```
MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#1111111B
MOV P2,#10001000B
```

RET

NYALA3:

```
MOV P1,#11110111B
```



MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA4:

MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA5:

MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA6:

MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

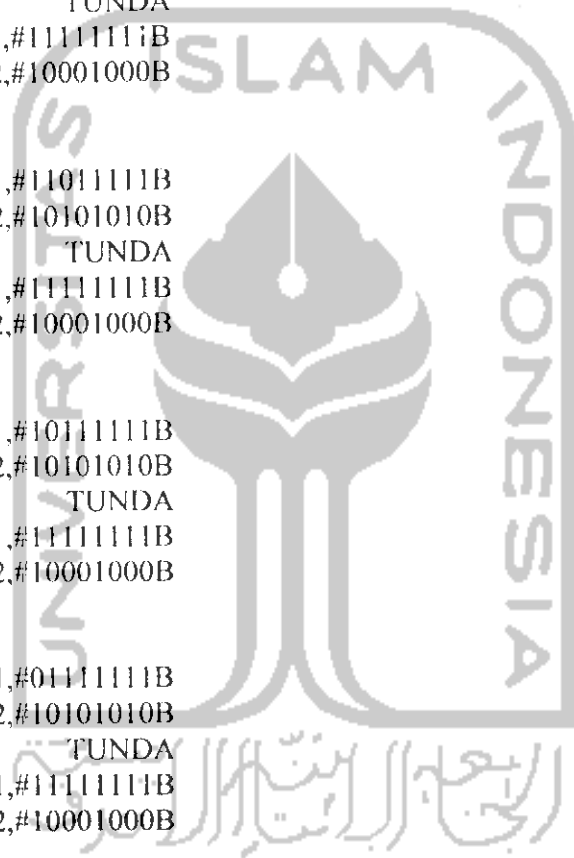
NYALA7:

MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

ISIBARIS:

MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A



```
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
```

RET

TUNDA:

```
MOV TH0,#HIGH WAKTU
MOV TL0,#LOW WAKTU
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
```

RET

SERINT:

```
JB RI,AMBIL_DATA
CLR TI
```

RETI

AMBIL_DATA:

CEK_1ST:

```
CJNE R7,#0,DATA 160
MOV R7,#1
```

```
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,#F',LANJUT
CLR RI
JMP LANJUT
```

DATA_160:

```
CJNE R5,#F',SALAH DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR RI
MOV @R1,A
CLR TI
INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H
```

LANJUT:

```
MOV A,R3
RETI
```

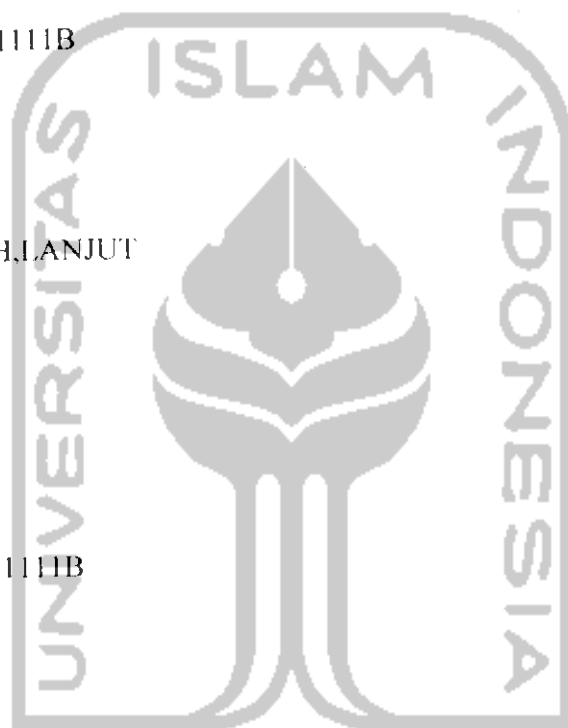
SALAH_DATA:

```
MOV P1,#11111111B
MOV B,SBUF
CLR RI
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
```

CONTI:

```
RETI
```

END



ENABLE		BIT	P2.0
RCLK	BIT		P2.1
SCLK	BIT		P2.2
CLEAR		BIT	P2.3
ENABLE2		BIT	P2.4
RCLK2		BIT	P2.5
SCLK2		BIT	P2.6
CLEAR2		BIT	P2.7

WAKTU EQU -1000

BARIS0	equ	10H
BARIS1	equ	2EH
BARIS2	equ	4CH
BARIS3	equ	6AH
BARIS4	equ	88H
BARIS5	equ	0A6H
BARIS6	equ	0C4H
BARIS7	equ	0E2H

```
ORG 00H
MOV R7,#0
SJMP MAIN ; lompat ke program utama
```

```
-----
; ; ISI vektor interupsi port serial
```

```
ORG 23H ; lokasi vektor interupsi port serial
LJMP SERINT ; karena panjang lompat ke SERINT
```

```
-----
ORG 030H ; program utama mulai di sini
MAIN: MOV TMOD,#0010000!B ; timer 1 mode 2 (8-bit, ISI-ulang)
MOV TH1,#0FDH ; 9600 baud rate
MOV SCON,#50H ; Mode serial: 8-bit UART
SETB TR1 ; Jalankan Timer 1
```

```
-----
SETB ES
SETB EA
MOV R1,#10H ; ALAMAT MEMORY
```



```

MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV R0,#38H
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0

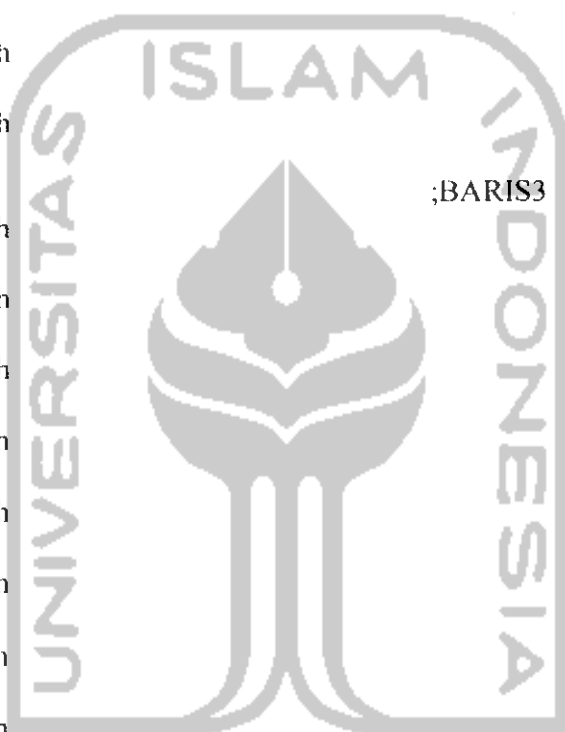
```



الجامعة الإسلامية
البريد الإلكتروني

```
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
```

```
MOV R0,#4CH
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
```



الجامعة الإسلامية
Baris3


```
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
INC R0
MOV @R0,#0ffh
```

;-----
tampil:

SIAPKAN:

```
MOV P2,#0
```

ISIBARIS0:

```
MOV P2,#10001000B
MOV R0,#23H
MOV A,@R0
```

LANJUT_BARIS0:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#0FH,LANJUT_BARIS0
ACALL NYALAO
```

ISIBARIS1:

```
MOV P2,#10001000B
MOV R0,#37H
MOV A,@R0
```

LANJUT_BARIS1:

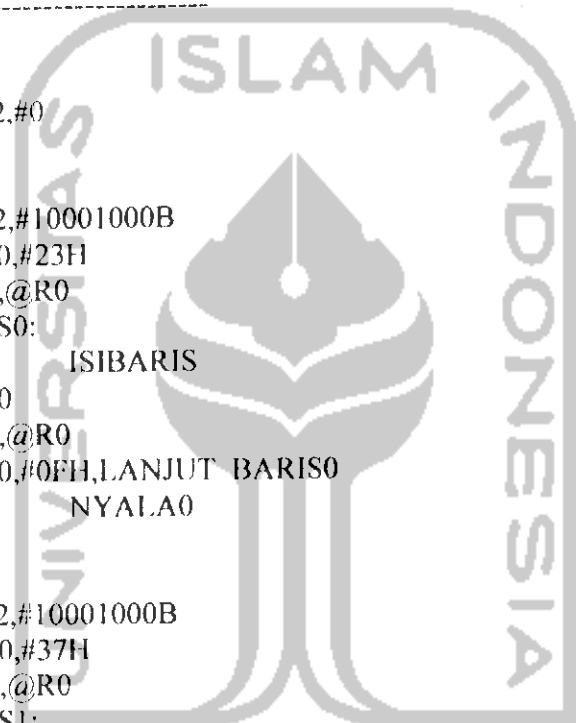
```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#23H,LANJUT_BARIS1
ACALL NYALAI
```

ISIBARIS2:

```
MOV P2,#10001000B
MOV R0,#4BH
MOV A,@R0
```

LANJUT_BARIS2:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
```



CJNE R0,#37H,LANJUT_BARIS2
ACALL NYALA2

ISIBARIS3:

MOV P2,#10001000B
MOV R0,#5FH
MOV A,@R0

LANJUT_BARIS3:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#4BH,LANJUT_BARIS3
ACALL NYALA3

ISIBARIS4:

MOV P2,#10001000B
MOV R0,#73H
MOV A,@R0

LANJUT_BARIS4:

ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#5FH,LANJUT_BARIS4
ACALL NYALA4

ISIBARIS5:

MOV P2,#10001000B
MOV R0,#87H
MOV A,@R0

LANJUT_BARIS5:

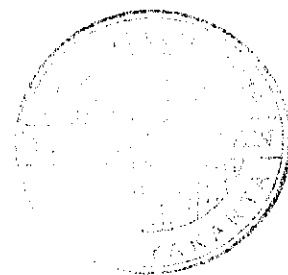
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#73H,LANJUT_BARIS5
ACALL NYALA5

ISIBARIS6:

MOV P2,#10001000B
MOV R0,#9BH
MOV A,@R0

LANJUT_BARIS6:

ACALL ISIBARIS
DEC R0



```
MOV A,@R0
CJNE R0,#87H,LANJUT_BARIS6
ACALL NYALA6
```

ISIBARIS7:

```
MOV P2,#10001000B
MOV R0,#0AFH
MOV A,@R0
```

LANJUT_BARIS7:

```
ACALL ISIBARIS
DEC R0
MOV A,@R0
CJNE R0,#9BH,LANJUT_BARIS7
ACALL NYALA7
```

JMP TAMPIL

NYALA0:

```
MOV P1,#11111110B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA1:

```
MOV P1,#11111101B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

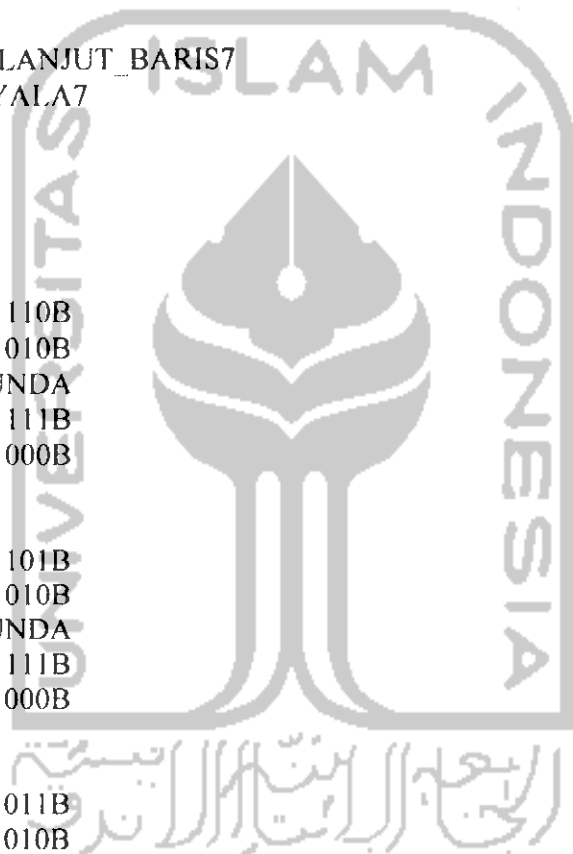
NYALA2:

```
MOV P1,#11111011B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B
```

RET

NYALA3:

```
MOV P1,#11110111B
MOV P2,#10101010B
ACALL TUNDA
```



MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA4:

MOV P1,#11101111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA5:

MOV P1,#11011111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

NYALA6:

MOV P1,#10111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

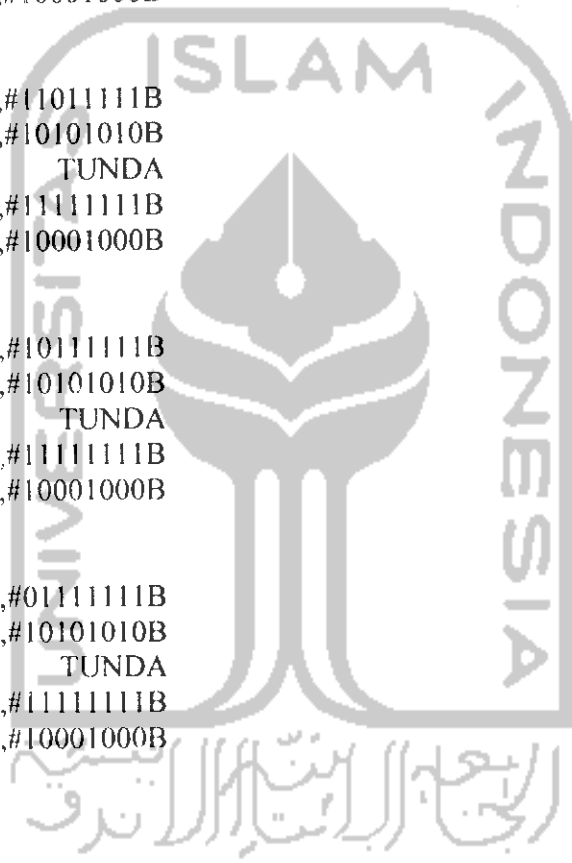
NYALA7:

MOV P1,#01111111B
MOV P2,#10101010B
ACALL TUNDA
MOV P1,#11111111B
MOV P2,#10001000B

RET

ISIBARIS:

MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B



```

MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B
RR A
MOV P0,A
MOV P2,#11001100B
MOV P2,#10001000B

```

RET

TUNDA:

```

MOV TH0,#HIGH WAKTU
MOV TL0,#LOW WAKTU
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0

```

RET

SERINT:

```

JB RI,AMBIL_DATA
CLR TI

```

RETI

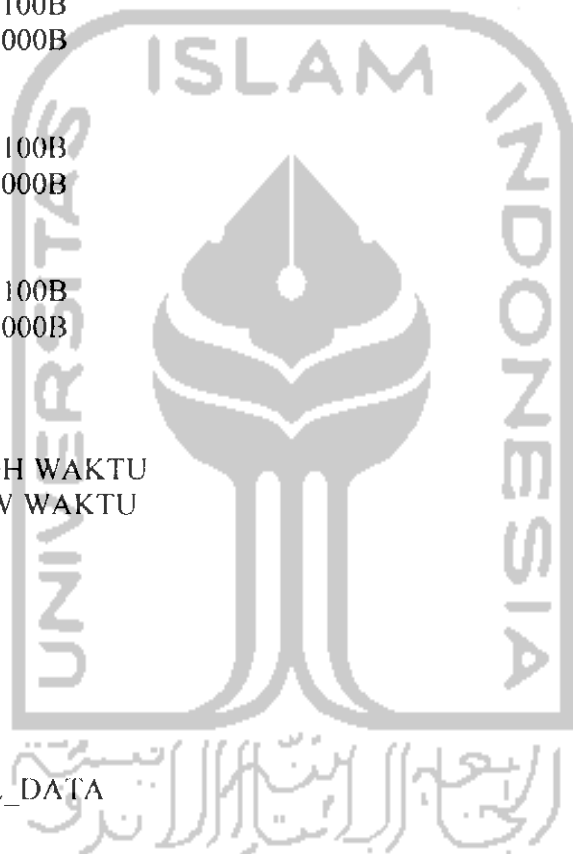
AMBIL_DATA:

CEK_IST:

```

CJNE R7,#0,DATA_160
MOV R7,#1
MOV R3,A

```



```
MOV P1,#11111111B
MOV A,SBUF
MOV R5,A
CJNE A,'#G',LANJUT
CLR RI
JMP LANJUT
```

DATA_160:

```
CJNE R5,'#G',SALAH_DATA
MOV R3,A
MOV P1,#11111111B
MOV A,SBUF
CLR RI
MOV @R1,A
CLR TI
INC R1
CJNE R1,#0B0H,LANJUT
MOV R7,#0
MOV R1,#10H
```

LANJUT:

```
MOV A,R3
RETI
```

SALAH_DATA:

```
MOV P1,#11111111B
MOV B,SBUF
CLR RI
CLR TI
INC R4
CJNE R4,#0B1H,CONTI
MOV R7,#0
MOV R4,#10H
```

CONTI:

```
RETI
```

END

