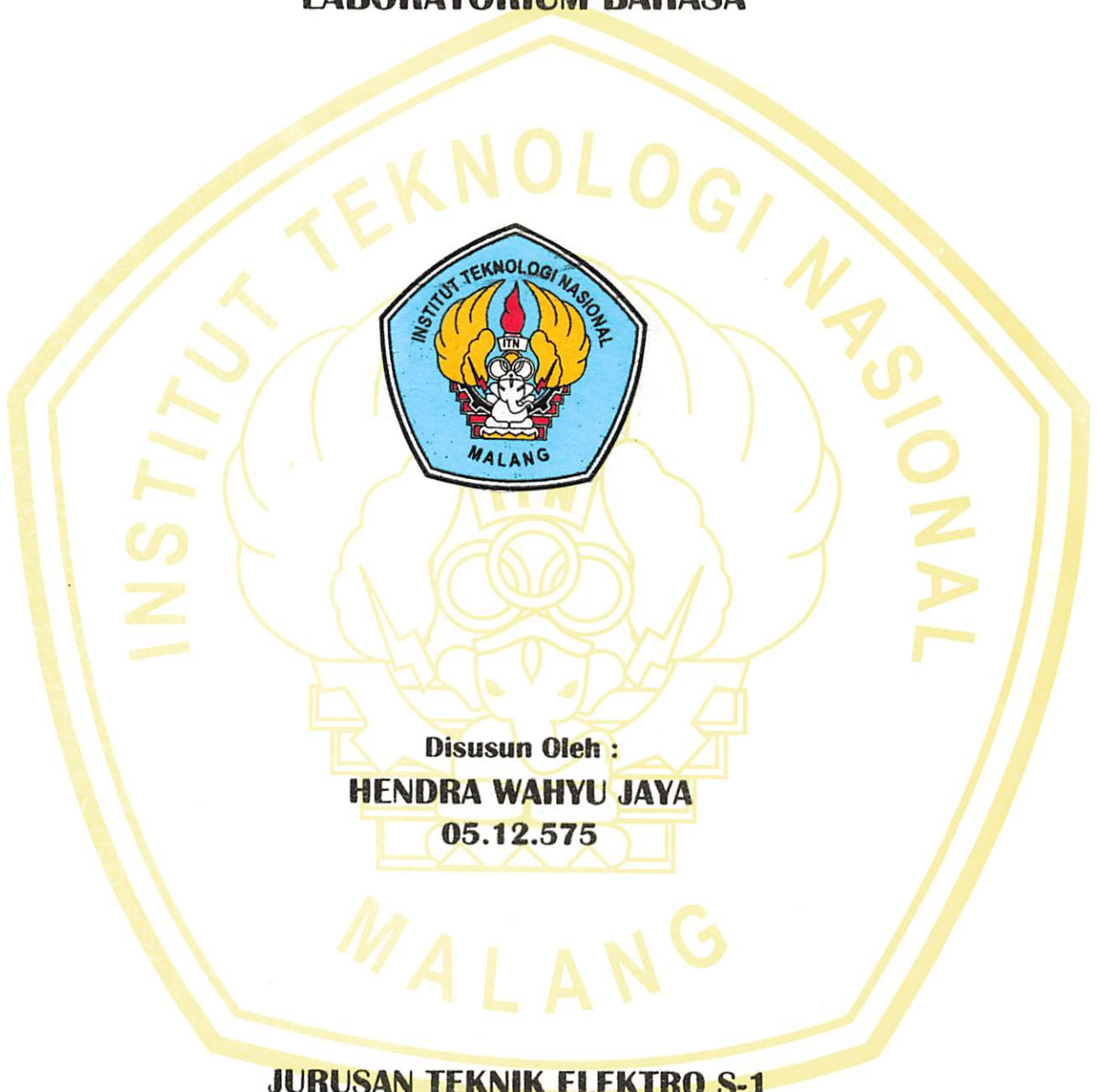


SKRIPSI

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK UNTUK APLIKASI MASTER KONTROL DIGITAL LABORATORIUM BAHASA



Disusun Oleh :
HENDRA WAHYU JAYA
05.12.575

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

1972

AMERICAN UNIVERSITY
WASHINGTON, D.C. 20004
OFFICE OF THE DEAN

1972-73
NON-GRANT ACCOUNT
\$72,21.00

1-0 OFFICE DEPARTMENT
AMERICAN UNIVERSITY
WASHINGTON, D.C. 20004
OFFICE OF THE DEAN
\$72,21.00

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUTAN PERANGKAT LUNAK UNTUK
APLIKASI MASTER KONTROL DIGITAL LABORATORIUM BAHASA
SKRIPSI**

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)*

Disusun Oleh :

**HENDRA WAHYU JAYA
NIM : 05.12.575**

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Dr. Eng. Aryanto Soetedjo, ST. MT
NIP.Y. 1030800417

Mengetahui

Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT.
NIP.Y.1039500274



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
UNTUK APLIKASI MASTER KONTROL DIGITAL
LABORATORIUM BAHASA**

**HENDRA WAHYU JAYA
(05.12.575)**

Jurusan Teknik Elektro S-1, Fakultas Teknik Komputer dan Informatika
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km.02 Malang, Indonesia
Email: hendra.wahyu0@gmail.com

Dosen Pembimbing : I. Ir. Yusuf Ismail Nakhoda, MT
II. Dr. Eng. Aryuanto Soetedjo, ST.MT

Abstrak

Saat ini Laboratorium Bahasa adalah suatu sarana dan prasarana yang sangat penting dalam pendidikan, Laboratorium Bahasa juga mengalami beberapa perkembangan mulai dari yang menggunakan sistem analog sampai dengan menggunakan sistem digital. Akan tetapi dari masing – masing sistem tersebut terdapat kekurangan dan kelebihan Oleh karena itu dibutuhkan suatu Laboratorium Bahasa bisa menggabungkan kelebihan antara laboratorium bahasa yang menggunakan sistem analog dan digital.

Untuk mewujudkan itu semua diperlukan suatu perancangan dan pembuatan perangkat lunak untuk aplikasi master kontrol digital laboratorium bahasa yang bisa menggabungkan kelebihan dari kedua sistem tersebut menjadi satu buah aplikasi master Kontrol yang lebih sederhana tanpa mengurangi fungsi dari Laboratorium Bahasa tersebut.

Dengan penggabungan kedua sistem tersebut akan didapatkan komunikasi audio yang jernih dan penambahan aplikasi pendukung seperti media player, absensi, rekording dan lain-lain.

Kata Kunci : Laboratorium Bahasa Analog, laboratorium Bahasa Digital, Master Kontrol digital

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan perangkat lunak untuk Aplikasi Master Kontrol Digital Laboratorium Bahasa” ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
3. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Dosen Pembimbing I.
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST. MT selaku Dosen Pembimbing II.
5. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do’a restu, dorongan, semangat, dan biaya.
6. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, September 2009

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5. Metode Penelitian	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Laboratorium Bahasa	6
2.1.1 Fungsi Laboratorium Bahasa	6
2.2 Bahasa Pemrograman Borland <i>Delphi</i>	9
2.2.1 Pengertian <i>Delphi</i>	9
2.2.2 Keunggulan Delphi	10
2.2.3 DE Integrated Development Environment	11

2.3	Microkontroller At89s51	15
2.3.1	RST	18
2.3.2	ALE	18
2.3.3	PSEN	19
2.3.4	EA	19
2.3.5	<i>XTAL1</i>	19
2.3.6	<i>XTAL2</i>	20
2.4	Microsoft Acces	20
2.5	PPI 8255	21
2.5.1	Register 8255	22
2.5.2	Programming 8255	22
2.6	RS232	28
2.6.1	Konfigurasi Null Modem DB9	30
2.6.2	Transmisi Data Pada RS232	32
2.6.3	Keuntungan Menggunakan Komunikasi Serial	32
BAB III PERANCANGAN SISTEM		34
3.1	Analisa sistem	34
3.2	Disain Sistem	32
3.3	Diagram Blok Sistem	33
3.4	Flow Chart	37
3.5	Perancangan Perangkat Lunak (Software)	39
3.5.1	Disain menu Aplikasi	39

3.5.2	Disain Visualisasi Aplikasi	42
3.5.3	Diagram Blok Aplikasi	43
3.5.4	Flow Chart	44
3.5.5	Disain Software mikrokontroler AT89s51	45
3.6	Perancangan Perangkat Keras (Hardware)	46
3.6.1	Diagram Blok Perangkat Keras	46
3.6.2	Rangkaian Mikrokontroler AT89s51	47
3.6.3	Rangkaian RS 232	48
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM		50
4.1	Implementasi Sistem	50
4.2	Cara menjalankan aplikasi	50
4.2.1	Menu Aplikasi	51
4.2	Pengujian Hardware	54
4.3	Pengujian Aplikasi	55
4.1.1	Pemilihan Port Komputer	55
4.1.2	Panel Siswa	56
4.1.3	Audio Input	57
4.1.4	Menu komunikasi	58
4.1.5	Menu Pilihan Tambahan	59
4.1.6	Audio Output	59
4.1.7	Media Player	60
4.1.8	Absensi	62

BAB V PENUTUP	65
5.1. Kesimpulan	65
5.2. Saran	66
DAFTAR PUSTAKA	67
LAMPIRAN	68

DAFTAR GAMBAR

Gambar 2.1 Tampilan IDE <i>Delphi</i>	11
Gambar 2.2 Menu bar	12
Gambar 2.3 Tool bar	13
Gambar 2.4 Komponent pallette	13
Gambar 2.5 Object Inspector	14
Gambar 2.6 Form	14
Gambar 2.7 Code editor	15
Gambar 2.8. Konfigurasi Kaki Mikrokomputer AT89S52	17
Gambar 2.9 Skema konektor PPI 8255	21
Gambar 2.10 Control Word A	23
Gambar 2.11 Control Word B	24
Gambar 2.12	25
Gambar 2.13	26
Gambar 2.14	26
Gambar 2.15	27
Gambar 2.16	27
Gambar 2.17 Perbedaan Tegangan RS232 dan TTL	28
Gambar 2.18 Konektor DB 9	29
Gambar 2.19 Koneksi Kaki- kaki DB 9	31
Gambar 3.1 Diagram Blok Sistem	35
Gambar 3.2 Flow Chart Switching penguat mic	36

Gambar 3.3 Flow Chart Tombol Call	37
Gambar 3.4 Disain Visual Aplikasi	41
Gambar 3.5 Diagram Blok Aplikasi	42
Gambar 3.6 Flowchart Aplikasi	43
Gambar 3.6 Diagram Blok Perangkat Keras	45
Gambar 3.7 Gambar Rangkaian Minimum	46
Gambar 3.8 Rangkaian RS232	48
Gambar 4.3 Tampilan Pertama Kali Program dijalankan	49
Gambar 4.4 Panel Siswa	50
Gambar 4.5 Menu Audio Input	51
Gambar 4.6 Menu Komunikasi	51
Gambar 4.7 Menu tambahan	51
Gambar 4.8 Menu Audio Output	52
Gambar 4.9 Media Palyer	52
Gambar 5.10 Tombol Pengganti Panel Nama Siswa	53
Gambar 4.11 Form Absensi	53
Gambar 4.12 Menu pilihan port computer	54
Gambar 4.13 Memilih port computer	55
Gambar 4.14 Kondisi jika koneksi berhasil	55
Gambar 4.15 Jika koneksi tidak berhasil	55
Gambar 4.16 Tombol panel siswa ketika belum ditekan	56
Gambar 4.17 Tombol panel siswa ketika sudah ditekan	56
Gambar 4.18 Menu audio input ketika sudah aktif	57

Gambar 4.19 Menu komunikasi	57
Gambar 4.20 Menu pilihan tambahan	57
Gambar 4.21 Menu audio output	58
Gambar 4.22 Form Yang muncul ketika Tombol Open File Di click	60
Gambar 4.23 Memutar Video menggunakan Media Placer	60
Gambar 4.24 Form Absensi	61
Gambar 4.24 Form Tambah Kelas	61
Gambar 4.25 Form Tambah Data	62
Gambar 4.26 Form Konfirmasi hapus data	62
Gambar 4.27 Form Ubah data	63
Gambar 4.28 Contoh Panel Siswa Menggunakan Nama Absensi	63

DAFTAR TABEL

Tabel 2.1	Keterangan Menu Bar	12
Tabel 2.2	Register 8255	22
Tabel 2.2	Konfigurasi Pin	30
Table 4.1	Pengujian Hardware.....	54
Tabel 4.2	Panel Siswa	56
Table 4.1	Kondisi tombol audio input	57
Table 4.2	Kondisi tombol menu komunikasi	58
Table 4.3	Kondisi Tombol Audio Output	59

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Seiring dengan berjalannya waktu dan perkembangan teknologi yang sangat cepat, penggunaan komputer sangatlah berperan aktif, khususnya dalam bidang pemerintahan, pendidikan, kesehatan dan bisnis. Semakin hari kemajuan teknologi komputer, baik dibidang piranti lunak maupun perangkat keras berkembang sangat cepat, disisi lain juga berkembang kearah yang sangat mudah dari segi penggunaan dan murah dalam biaya. Solusi untuk bidang kerja apapun akan ada cara untuk dapat dilakukan melalui media komputer , termasuk dalam bidang pendidikan.

Saat ini Laboratorium Bahasa adalah suatu sarana dan prasarana yang sangat penting dalam pendidikan, bahkan saat ini semua sekolah baik dari sekolah dasar sampai perguruan tinggi diharuskan memiliki laboratorium bahasa sebagai sarana dalam proses pendidikan.

Laboratorium Bahasa mengalami beberapa perkembangan mulai dari yang menggunakan sistem *analog* sampai dengan menggunakan sistem *digital*. Sistem *analog* yang dimaksud adalah Laboratorium tersebut mempunyai menu yang *standard* dan peralatan yang digunakan terdiri dari beberapa *device* elektro (*Hardware*). Laboratorium Bahasa kini telah mengalami perkembangan dari sisi *interface* dan *device* dengan menggunakan sistem *digital* berbasis PC (*Personal Komputer*) yang lebih praktis dan efisien dalam penggunaannya.

Laboratorium bahasa yang menggunakan sistem *analog* mempunyai kelebihan dalam hal komunikasi *audio*, sedangkan untuk laboratorium bahasa *digital* lebih dalam hal peyediaan fasilitas pendukung seperti media player, absensi, kuis dan lain-lain.

Dari latar belakang di atas maka penulis ingin membuat skripsi yang berjudul “Perancangan Dan Pembuatan Perangkat Lunak Untuk Aplikasi Master Kontrol Digital Laboratorium Bahasa” yang nantinya diharapkan bisa menggabungkan kelebihan antara laboratorium bahasa yang menggunakan sistem *analog* dan *digital*. menjadi satu buah aplikasi master Kontrol yang lebih sederhana tanpa mengurangi fungsi dari Laboratorium Bahasa tersebut

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana membangun sistem Laboratorium Bahasa yang bisa menggabungkan kelebihan dari sistem *analog* dan *digital*
2. Menambah menu-menu dari Laboratorium Bahasa *Analog* yang sudah ada.

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam pelaksanaan Skripsi ini adalah merancang dan membuat perangkat lunak untuk aplikasi master kontrol digital Laboratorium Bahasa yang bisa menggabungkan kelebihan antara sistem *analog* dan *digital* sehingga kekurangan dari masing-masing sistem tersebut bisa diminimalkan. Dan juga menyempurnakan Laboratorium Bahasa Analog yang sudah ada, dengan cara

menambah menu-menu yang penting untuk kelangsungan belajar dan mengajar antara lain Absensi siswa, *Media Player* dan *Pairing* (komunikasi kelompok).

1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan skripsi ini hanya membahas perancangan dan pembuatan perangkat lunak dan *source code* yang digunakan pada *microkontroler*.

1.5. Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Studi literatur, yaitu tinjauan pustaka untuk mempelajari teori-teori yang terkait melalui literatur yang ada, yang berhubungan dengan permasalahan.

2. Pengumpulan Data

Sebagai acuan untuk merancang Sistem Informasi ini, data diperoleh dengan cara observasi ke Sekolah dan universitas yang tersedia Laboratorium Bahasa dan konsultasi dengan ahli elektro.

3. Analisa Data

Data-data yang diperoleh akan diaplikasikan menjadi suatu kesatuan sistem Laboratorium Bahasa Digital

4. Desain Sistem dan Pembuatan

Merancang dan mengembangkan system master kontrol Laboratorium Bahasa sesuai dengan data – data yang diperoleh.

5. Kesimpulan

Mengambil kesimpulan dari hasil perancangan dan pembuatan Laboratorium Bahasa Digital.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Perumusan Masalah, Tujuan dan Manfaat Penelitian, Pembatasan Permasalahan, Metodologi Penelitian dan Sistematika Penulisan.

Bab II : Landasan Teori

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan pembahasan yang dilakukan, yang di dalamnya memuat teori-teori tentang penertian laboratorium bahasa, bahasa pemrograman Borland delphi, database *Microsoft Access* dan Rangkaian mikrokontroller.

Bab III : Perancangan Sistem

Dalam bab ini berisi mengenai metode perancangan sistem yang digunakan pada aplikasi ini, serta analisis *software* dan *hardware* yang diperlukan untuk menjalankan aplikasi ini.

Bab IV : Implementasi dan Pengujian Sistem

Berisi tentang pembuatan master kontrol Laboratorium Bahasa Digital menggunakan bahasa pemrograman *Delphi* dan cara kerja alat.

Bab V : Kesimpulan dan saran

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran mengenai hasil penelitian yang telah dilaksanakan peneliti.

BAB II

TINJAUAN PUSTAKA

2.1 Laboratorium Bahasa

Laboratorium bahasa adalah kumpulan peralatan yang bisa dimanfaatkan untuk menunjang paktek dan proses belajar mengajar bahasa asing ataupun bahasa daerah. Dalam pelaksanaannya diharapkan siswa dapat berlatih mendengar melalui media audio yang dikontrol oleh instruktur melalui komputer. Pembelajaran bahasa asing melalui laboratorium bahasa dibagi menjadi 3 fungsi dasar yaitu, percakapan (*conversation*), mendengarkan (*listening*) dan fungsi manajemen instruktur dalam mengatur kegiatan belajar mengajar. Agar tercapai tingkat sangat mahir maka suatu perangkat laboratorium bahasa harus mempunyai fasilitas yang sangat lengkap dan lebih dari fungsi lab bahasa sebagai sarana pembelajaran.

2.1.1 Fungsi Laboratorium Bahasa

Fungsi laboratorium bahasa sebagai sarana pembelajaran yang umum digunakan adalah sebagai berikut:

a. Listening.

Siswa mendengarkan melalui bantuan perangkat elektronik yang tersedia.

- ***Listening all.*** Semua siswa dapat langsung diarahkan untuk mendengarkan materi pelajaran yang diberikan oleh guru pada salah satu *Channel*.
- ***Individu.*** Siswa dapat dibagi menjadi beberapa kelompok secara acak maupun tetap (tergantung desain laboratorium bahasa) kemudian setiap kelompok dapat diarahkan untuk mendengarkan 2 materi yang berbeda pada setiap *Channel*.

b. Conversation

Siswa melakukan berbagai jenis percakapan.

- **Pair Row dan Pair Coloumn.** Siswa secara berpasangan bercakap-cakap. Percakapan dapat dilakukan dengan teman semeja atau teman di belakang/depan meja.
- **Fix Group.** Siswa dikelompokkan menjadi beberapa kelompok. Ada desain laboratorium bahasa yang mengelompokkan menjadi 2 kelompok saja ada yang lebih. Intinya percakapan siswa dilakukan secara berkelompok.

c. Attention.

Instruktur menyampaikan materi atau pengumuman.

- **Attention all.** Instruktur menyampaikan pengumuman atau materi pembelajaran kepada semua siswa dan melalui *speaker* ruangan
- **Attentioin channel.** Instruktur menyampaikan materi kepada siswa yang berada dalam kelompok tertentu.

d. Monitoring.

Instruktur menjalankan fungsi pengawasan terhadap kegiatan siswa.

- **Monitoring Channel.** Instruktur memperhatikan percakapan sekelompok siswa dalam *group* tertentu..
- **Monitoring Individual.** Instruktur memperhatikan pembicaraan seorang siswa secara khusus. Baik ketika siswa tersebut berada dalam kelompok atau saat sedang berpasangan.

e. *Intercom.*

Siswa dapat melakukan panggilan untuk percakapan dengan guru demikian pula sebaliknya.

- ***Intercom Group.*** Instruktur melakukan percakapan dengan kelompok siswa yang dihubungi.
- ***Intercom Individu.*** Seorang siswa menghubungi instruktur untuk melakukan percakapan dengannya.

f. *Multimedia Control.*

Perangkat lunak juga menyediakan kendali khusus untuk operasional file-file *audio/video* yang dapat digunakan untuk memberikan pelajaran khusus berbasis multimedia.

g. *Audio Record.*

Guru dapat merekam suara dalam bentuk file-file *audio* untuk keperluan soal, pengumuman atau hal-hal lain yang membutuhkan *file audio*.

h. *Audio Control.*

Guru dapat mengatur materi pelajaran atau suara apa yang masuk melalui empat Channel suara yang ada.

i. *Data base.*

Setiap kelas yang akan menggunakan Laboratorium Bahasa harus diisikan database-nya terlebih dahulu dalam komputer. Data yang diisikan mulai dari NIS, Nama, dan kelas siswa, jadi setiap kelas akan memiliki databasenya sendiri-sendiri.

2.2 Bahasa Pemrograman Borland *Delphi*

2.2.1 Pengertian *Delphi*

Delphi merupakan salah satu bahasa yang bekerja dibawah lingkup sistem operasi windows, dimana *Delphi* memberikan fasilitas- fasilitas pembuatan aplikasi visual seperti *Visual Basic*.

Dengan menggunakan *Delphi* kita dapat membuat aplikasi berbasis windows dengan beberapa keunggulan, yaitu produktifitas, kualitas, pengembangan perangkat lunak, kecepatan *compiler*, pola disain yang menarik serta diperkuat dengan bahasa yang terstruktur dalam struktur bahasa *object pascal*.

Kemampuan *Delphi* dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows.

Khusus untuk database, *borland Delphi* menyediakan fasilitas *object* yang kuat dan lengkap yang memudahkan programmer membuat program untuk aplikasi database.

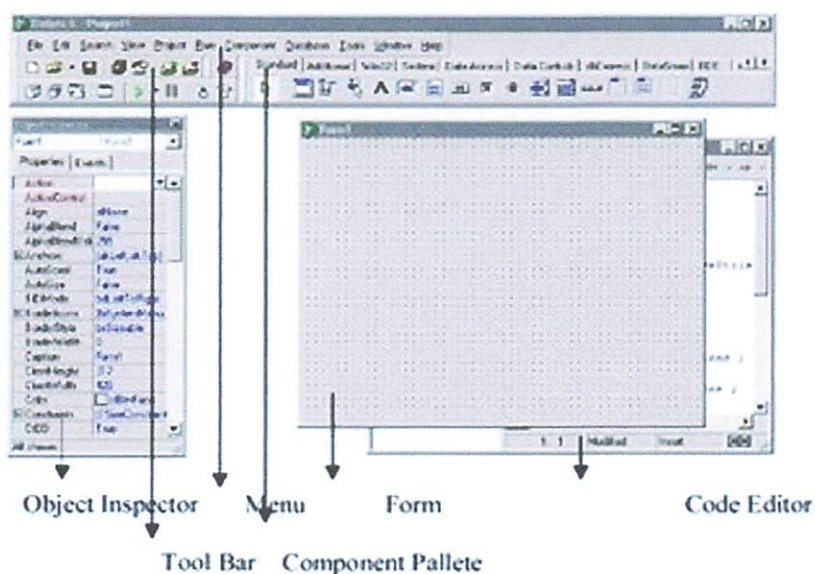
2.2.2 Keunggulan *Delphi*

- a. IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan *delphi*, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
- b. Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada *Delphi*, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.

- c. Mudah digunakan, *source kode* delphi yang merupakan turunan dari *pascal*, sehingga tidak diperlukan suatu penyesuaian lagi.
- d. Bersifat *multi purpose*, artinya bahasa pemrograman *Delphi* dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.

2.2.3 IDE Integrated Development Environment

Tampilan bidang kerja yang disebut dengan IDE (Integrated Development Environment) *Delphi* bisa dilihat pada gambar di bawah ini. IDE ini secara garis besar terdiri atas tiga bagian utama, yaitu Window Utama, *Object Inspector* dan *Editor*.

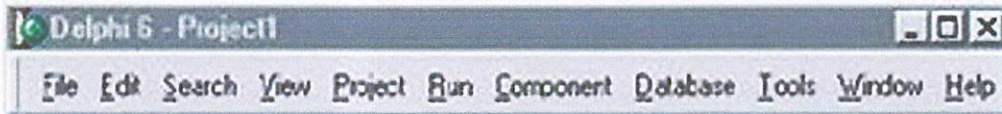


Gambar 2.1 Tampilan IDE *Delphi*

Window utama terdiri atas *Menu Bar*, *Tool Bar* dan *Komponen Palette*. *Object Inspector* menyediakan dua kelompok pengaturan komponen, yaitu *Properties* dan *Event*. Editor yang disediakan ada dua buah yaitu *Form Editor* dan *Code Editor*.

a. Menu Bar

Menu Bar menyediakan kelompok perintah yang digolongkan dalam sebelas menu. Menu tersebut antara lain *File, Edit, Search, View, Project, Run, Komponent, Database, Tools, Options,* dan *Help.*



Gambar 2.2 Menu bar

Tabel 2.1 Keterangan Menu Bar

Nama	Keterangan
File	Kelompok perintah yang berfungsi untuk pengaturan suatu file. Misalnya New, Save, Open, dan sebagainya.
Edit	Menyimpan perintah-perintah untuk pengeditan. Mulai dari pengeditan objek, pengeditan komponen maupun pengeditan kode pada Code Editor. Contohnya Cut, Paste, Align.
Search	Kelompok perintah yang berfungsi untuk melakukan proses pencarian, baik objek, kata, kalimat dan sebagainya.
View	Menampung perintah-perintah untuk mengaktifkan bagian pendukung Integrated Development Environment.
Project	Kelompok perintah yang berfungsi untuk manajemen proyek berikut bagian-bagian pendukungnya.
Run	Kelompok perintah untuk meruntangi proses kompilasi program seperti Run, Build, Step Over, Debug dan seterusnya.
Component	Mengatur suatu komponen.
Database	Mengandung perintah-perintah untuk pengaturan aplikasi database.
Tools	Kelompok perintah yang berfungsi sebagai penyedia perlengkapan tambahan yang diperlukan dalam penyusunan program seperti Image Editor, Database Engine dan lain-lain.
Options	Mengatur komposisi bidang kerja Delphi.
Help	Memberikan informasi yang bersifat menolong pemakai dalam menggunakan Delphi.

b. Tool Bar

Tool Bar adalah bagian dari *Delphi* yang menyediakan tombol-tombol *speed*. Tombol speed fungsinya sama dengan perintah yang ada pada menu. Tombol ini disediakan dengan tujuan meringkas atau mempercepat pekerjaan kita.

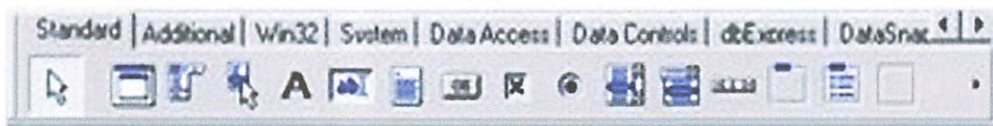
Untuk mengakses suatu perintah dengan menggunakan menu, langkah yang kita lakukan adalah klik menu lalu klik *submenu* lantas klik perintah tersebut. Sedangkan untuk mengakses perintah dengan menggunakan tombol *speed*, langkah yang kita lakukan hanya mengklik tombol yang kita inginkan.



Gambar 2.3 Tool bar

c. Komponen Palette

Komponen palette menyediakan berbagai komponen yang bisa kita pasang pada *form* sesuai keperluan kita. Telah tersedia berbagai komponen yang dikelompokkan dalam sebelas kelompok seperti tampak pada gambar.

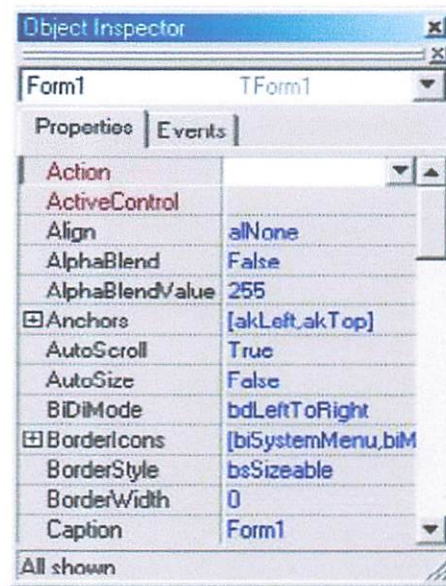


Gambar 2.4 Komponen palette

d. Object Inspector

Object inspector adalah sarana pengaturan objek yang kita pasang pada *form*, atau *form* itu sendiri. Dua hal penting yang bisa kita setel pada komponen adalah Properti dan *event*. Properti adalah yang terkait dengan sifat komponen seperti ukuran, warna dan sebagainya. Sedangkan *event* adalah kejadian atau peristiwa yang kita inginkan terpasang pada komponen tersebut kaitannya dengan

proses pemakaian. Contoh *event* misalnya klik, klik ganda, *drag* (geser), *drop* dan sebagainya.

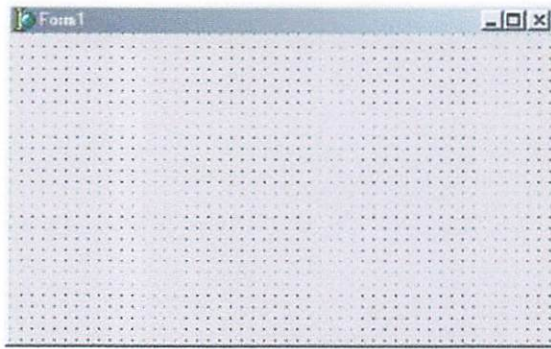


Gambar 2.5 Object Inspector

f. Form

Form adalah bahan dasar yang akan menjadi jendela aplikasi kita. Pada *form* terdapat tiga tombol kontrol, yaitu *Minimize*, *Maximize/Restore* dan *Close*.

Terdapat juga *caption bar* tempat kita menempatkan judul *Form* (yang kelak menjadi judul window) dan *icon*. Pembatas *form* juga bisa diubah ukurannya dengan cara *drag* (geser)-*drop*. Pada *form* kita bisa meletakkan komponen-komponen yang kita perlukan dalam suatu *User Interface*.

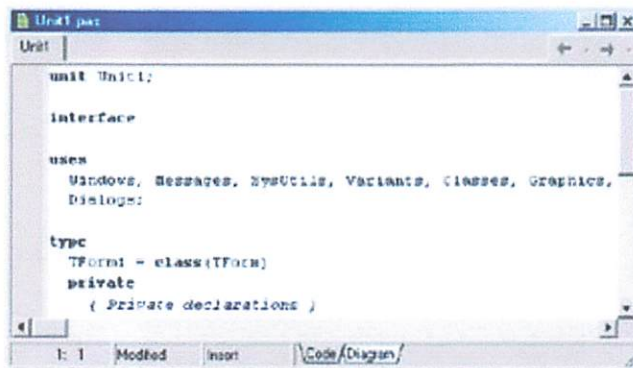


Gambar 2.6 Form

g. Code Editor

Code Editor adalah tempat kita menuliskan program dalam bahasa *Object Pascal*.

Secara *default Code Editor* ini terletak di belakang *Form Editor*.



Gambar 2.7 Code editor

Untuk menuliskan kode yang kita pasang pada suatu komponen, klik ganda komponen tersebut, *Code Editor* akan otomatis aktif dan menempatkan kursor di lokasi penulisan kode.

2.3 Microkontroller *At89s51*

Microkontroller merupakan suatu sistem Komputer yang seluruh atau bagian besar elemennya dikemas dalam suatu *chip IC*, sehingga sering disebut *single chip microkontroller*

AT89S51 memberikan fitur-fitur standar sebagai berikut: 4K *byte Flash*, 128 *byte RAM*, 32 jalur I/O, Timer *Wachtdog*, dua *data pointer*, dua 16 bit *timer/ counter*, lima vektor interupsi dua level, sebuah *port serial full duplex*, *oscilator internal*, dan rangkaian *clock*. Selain itu *AT89S51* didisain dengan logika statis untuk operasi dengan frekuensi sampai 0 Hz dan didukung dengan mode penghematan daya. Pada mode *idle* akan menghentikan CPU sementara RAM, *timer/ counter*, *serial port* dan sistem interupsi tetap berfungsi. *Mode Power Down* akan tetap menyimpan isi dari RAM tetapi akan membekukan *osilator*, menggagalkan semua fungsi *chip* sampai interupsi eksternal atau *reset hardware*.

AT89S51 mempunyai konsumsi daya rendah, mikrokontroller 8-bit CMOS dengan 4K *byte* memori Flash ISP (*in sistem programmable/* dapat diprogram didalam sistem). Divais ini dibuat dengan teknologi memori nonvolatile kerapatan tinggi dan kompatibel dengan standart industri 8051, set instruksi dan pin keluaran. Flash yang berada didalam *chip* memungkinkan memori program untuk diprogram ulang pada saat chip didalam sistem atau dengan menggunakan Programmer memori *nonvolatile konvensional*. Dengan mengkombinasikan CPU 8 bit yang serbaguna dengan flash ISP pada chip, *ATMEL 89S51* merupakan mikrokontroller yang luar biasa yang memberikan fleksibilitas yang tinggi dan penyelesaian biaya yang efektif untuk beberapa aplikasi kontrol.

byte-byte kode saat pemrograman Flash dan mengeluarkan *byte* kode saat verifikasi.

Pull-up eksternal diperlukan saat memverifikasi program.

❖ Port 1

Port 1 adalah port dua arah masukan/keluaran 8-bit dengan *pull-up internal*. Sebagai tambahan, P1.0 dan P1.1 dapat diatur sebagai pewaktu/ pencacah-2 eksternal masukan pencacah (P1.0/T2) dan pewaktu/pencacah-2 masukan pemicu (P1.1/T2EX). Port 1 juga menerima *byte-byte* alamat saat pemrograman dan verifikasi flash.

❖ Port 2

Port 2 adalah port masukan/keluaran dua arah 8-bit dengan *internal pull-up*. Port 2 juga menerima bit-bit alamat dan beberapa sinyal kendali saat pemrograman dan verifikasi flash.

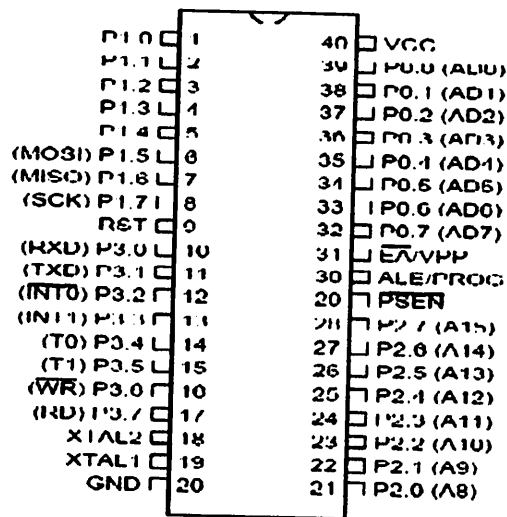
❖ Port 3

Port 3 adalah port masukan/keluaran dua arah 8-bit dengan *internal pull-up*. Port 3 juga menyediakan fasilitas berbagai fungsi khusus dari *AT89C51*. Port 2 juga menerima beberapa sinyal kendali saat pemrograman dan verifikasi flash.

2.3.1 RST

Masukan reset. Masukan tinggi pada kaki ini selama dua siklus instruksi mesin akan me-*reset* perangkat. *ALE/ PROG Address Latch Enable* (ALE) adalah pulsa keluaran untuk mengunci bit rendah dari alamat saat mengakses memori eksternal. Kaki

Mikrokontroler keluarga *MCS 51* memiliki port-port yang lebih banyak (40 port I/O) dengan fungsi yang bisa saling menggantikan sehingga mikrokontroler jenis ini menjadi sangat digemari karena hanya dalam sebuah chip sudah bisa mengkafer untuk banyak kebutuhan. Konfigurasi dan Deskripsi kaki-kaki mikrokomputer *AT89×5x* adalah sebagai berikut:



Gambar 2.8. Konfigurasi Kaki Mikrokomputer AT89S52.

❖ Port 0

Port 0 adalah port dua arah masukan/keluaran 8-bit saluran terbuka. Sebagai port keluaran, tiap kaki dapat menerima masukan TTL. Ketika logika 1 dimasukkan ke kaki-kaki port 0, kaki-kaki dapat digunakan sebagai masukan impedansi tinggi. Port 0 juga dapat diatur sebagai bus alamat/data saat mengakses program dan data dari memori luar. Pada mode ini port 0 memiliki *pull-up internal*. Port 0 juga menerima

ini juga digunakan sebagai masukan pulsa (PROG) saat pemrograman Flash. Pada operasi biasa,

2.3.2 ALE

ALE mengeluarkan rata-rata 1/6 kali frekuensi osilator dan mungkin digunakan sebagai pewaktu atau denyut. Catatan, satu pulsa ALE diabaikan saat setiap pengaksesan data memori eksternal. Jika diinginkan, operasi ALE dapat di-disable dengan menseting bit 0 dari SFR pada lokasi 8EH. Dengan bit yang diset, ALE aktif hanya saat menjalankan perintah MOVX dan MOVC. Selain itu, kaki ini dapat juga di-pull tinggi. Setting bit ALE-disable tidak berpengaruh jika mikrokomputer pada mode eksekusi eksternal.

2.3.3 PSEN

Program Store Enable (PSEN) adalah strobe pembacaan program pada memori eksternal. Ketika AT89C52 melakukan eksekusi program dari memori eksternal, PSEN diaktifkan dua kali setiap siklus instruksi mesin, kecuali bahwa dua aktivasi PSEN diabaikan setiap mengakses data memori eksternal.

2.3.4 EA

EA / Vpp *External Access Enable*. EA harus dihubungkan ke GND supaya memfungsikan perangkat untuk mengambil kode program dari lokasi memori eksternal dimulai dari 0000H hingga FFFFH. Catatan, jika *lock-bit* diprogram, EA akan dikunci secara internal pada saat reset. EA harus dihubungkan dengan Vcc untuk eksekusi

program internal. Kaki ini juga menerima tegangan yang memungkinkan pemrograman 12 Volt saat memprogram flash bila pemrograman 12 Volt dipilih.

2.3.5 XTAL1

XTAL1 Masukan inverting (pembalikan) penguat *osilator* dan masukan untuk operasi rangkaian denyut internal.

2.3.6 XTAL2

XTAL2 Keluaran dari inverting (pembalikan) penguat *osilator*.

2.4 Microsoft Acces

Microsoft Access (atau *Microsoft Office Access*) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi *Microsoft Office*, selain tentunya *Microsoft Word*, *Microsoft Excel*, dan *Microsoft PowerPoint*. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna

Microsoft Access dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/ *programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para *programmer* yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana.

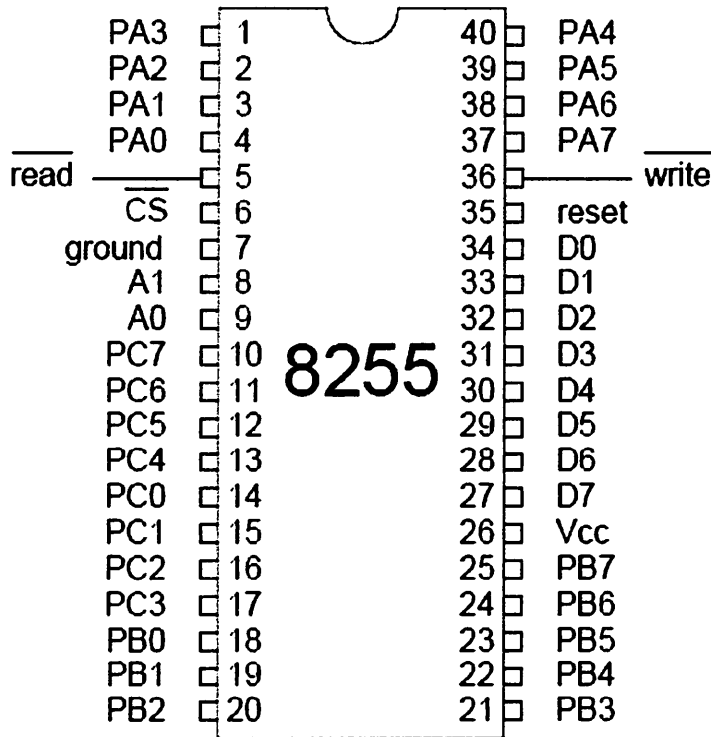
Access juga mendukung teknik-teknik berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu berorientasi objek.

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para *programmer* untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan *Microsoft Internet Information Services* (IIS) dan menggunakan *Microsoft Active Server Pages* (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada *Microsoft SQL Server* yang memiliki kemampuan yang lebih tinggi.

2.5 PPI 8255

PPI 8255 memiliki 3 *latch* yang dapat diprogram sesuka kita, apakah ingin dibuat sebagai input atau sebagai output.

Skema konektor dari IC PPI 8255 adalah sebagai berikut :



Gambar 2.9 Skema konektor PPI 8255

Terlihat pada skema diatas, PPI 8255 :

a. Memiliki 3 port :

- port A (pin 1-4 dan 37-40),
- port B (pin 18-25),
- port C (pin 10-17)

b. Masing-masing port tersebut dapat diprogram menjadi *port input* maupun *sebagai port output* atau keduanya. Dan jika dibutuhkan, port C dapat diprogram sebagai port kontrol

c. *Address line* A0 (pin 8) dan A1 (pin 9) digunakan untuk memilih salah satu port yg ingin diakses.

2.5.1 Register 8255

8255 menyediakan 4 buah register 8-bit :

Tabel 2.2 Register 8255

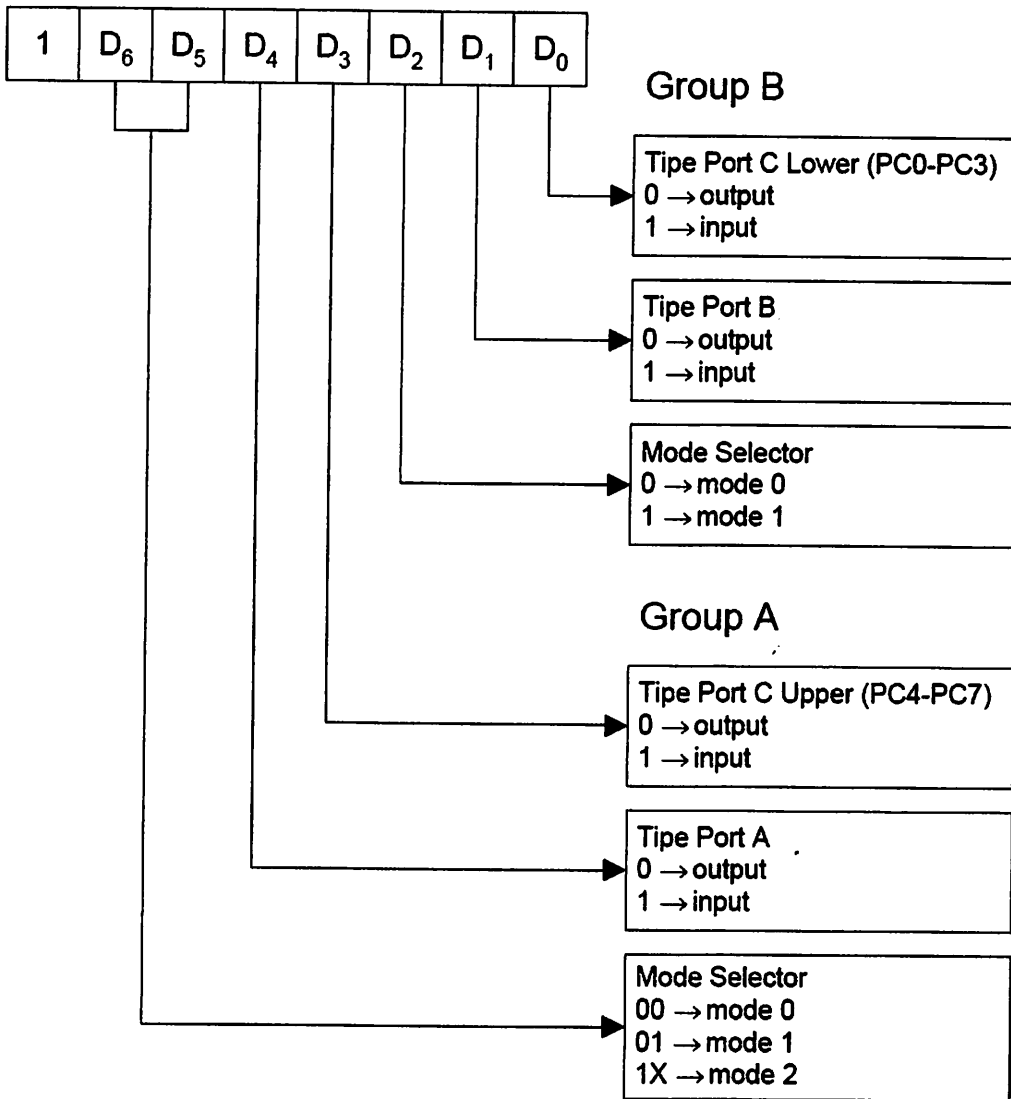
	diakses saat		keterangan
	A0	A1	
Register data A	0	0	register ini menyimpan data dari/untuk port A
Register data B	0	1	register ini menyimpan data dari/untuk port B
Register data C	1	0	register ini menyimpan data dari/untuk port C
Register kontrol	1	1	register untuk mengontrol kerja 3 register lainnya

Register data A, register data B, dan register kontrol hanya dapat diakses dalam format 8-bit. Register data C dapat diakses dalam 2 nibble (2 x 4 bit) yaitu Lower C (C0 – C3) dan Upper C (C4 – C7)

2.5.2 Programming 8255

Dengan mengirim suatu informasi selebar 1 byte ke register kontrol, kita dapat memprogram kerja dari port A, B, dan C. Informasi ini disebut juga dengan *Control Word*. *Control Word A* berfungsi untuk memprogram kerja setiap port, sedangkan *Control Word B* berfungsi untuk menSet atau meReset salah satu bit di Port C.

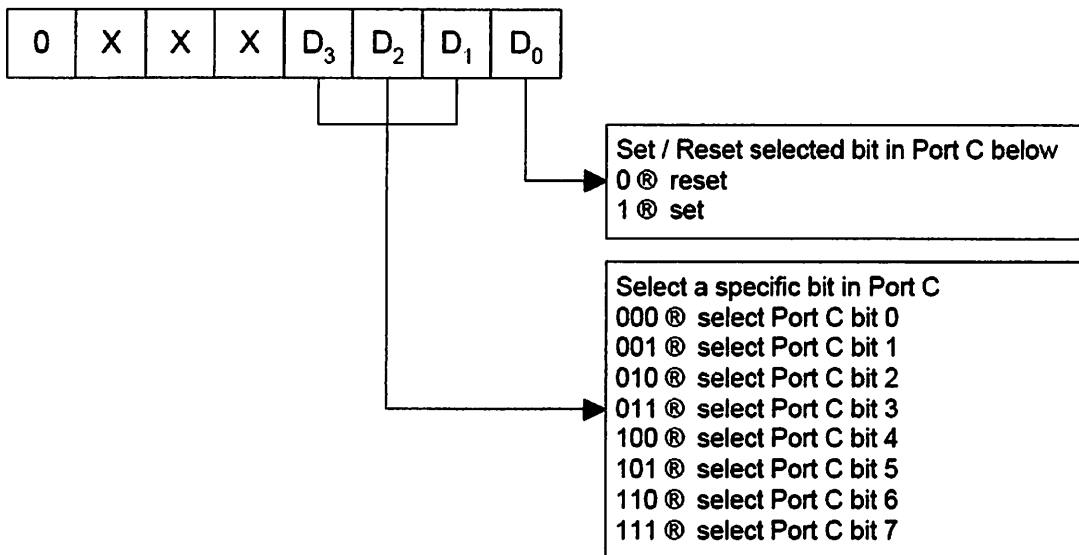
Control Word A : untuk memprogram kerja port



Gambar 2.10 Control Word A

Control Word B :

menSet/Reset salah satu bit di Port C saat 8255 bekerja pada mode

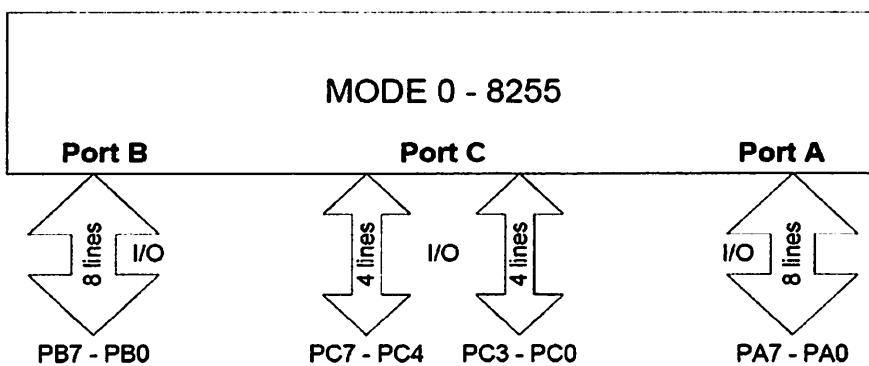


Gambar 2.11 Control Word B

Karena masing-masing port dapat diprogram sebagai *buffer input* maupun *output*, maka untuk memudahkan, 8255 didesain khusus untuk menerima 3 mode kerja :

a. Mode 0 : Basic I/O

pada mode ini setiap port bekerja secara independen. masing-masing dapat bekerja sebagai input atau output.



Control Word A :

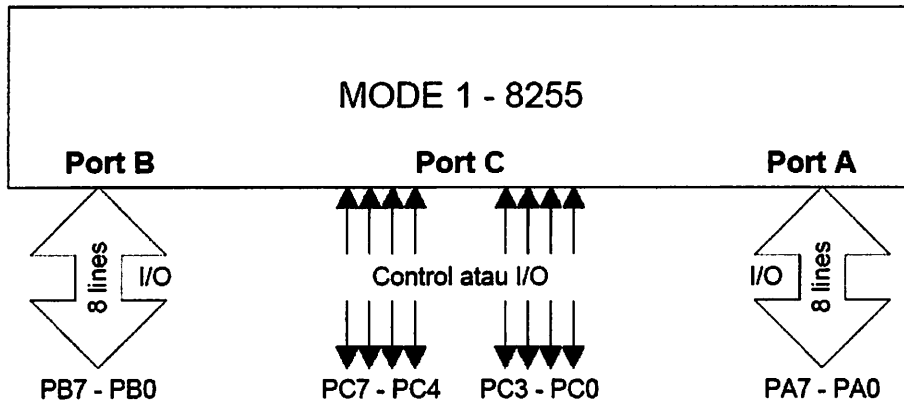
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	A	CU	0	B	CL

Gambar 2.12 Mode I/O

A = 0 (port A sbg Output); B = 0 (port B sbg Output); CU = 0 (Port C Upper sbg output);

CL = 0 (Port C Lower sbg Output)

b. Mode 1 : Strobe I/O



Control Word A :

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	1	A	CU	1	B	CL

Gambar 2.13 Mode Strobe I/O control Word A

A = 0 (port A sbg Output);

B = 0 (port B sbg Output);

CU = 0 (Port C Upper sbg output);

CL = 0 (Port C Lower sbg Output)

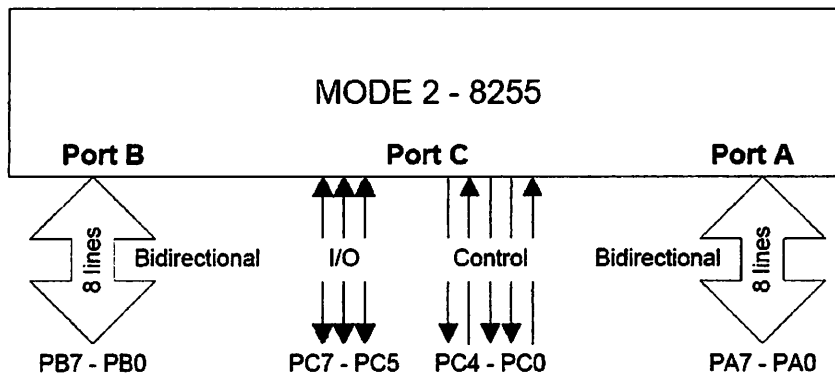
Control Word B :

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	don't care			a	b	c	0/1

a	b	c	bit di register C yg diakses
0	0	0	C ₀
0	0	1	C ₁
0	1	0	C ₂
0	1	1	C ₃
1	0	0	C ₄
1	0	1	C ₅
1	1	0	C ₆
1	1	1	C ₇

Gambar 2.14 Mode Strobe I/O control Word B

c. Mode 2 : Bidirectional I/O



Control Word A :

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	1	X	A	CU	1	B	CL

Gambar 2.15 Mode Bidirectional I/O control word A

A = 0 (port A sbg Output);

B = 0 (port B sbg Output);

CU = 0 (Port C Upper sbg output);

CL = 0 (Port C Lower sbg Output)

Control Word B :

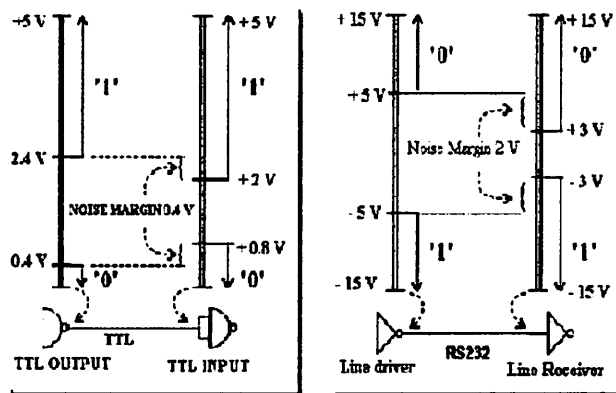
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	don't care			a	b	c	0/1

a	b	c	bit di register C yg diakses
0	0	0	C ₀
0	0	1	C ₁
0	1	0	C ₂
0	1	1	C ₃
1	0	0	C ₄
1	0	1	C ₅
1	1	0	C ₆
1	1	1	C ₇

Gambar 2.16 Mode Bidirectional I/O control word B

2.6 RS232

RS232 dibuat pada tahun 1962, jauh sebelum IC TTL populer, oleh karena itu level tegangan yang ditentukan untuk RS232 tidak ada hubungannya dengan level tegangan TTL, bahkan dapat dikatakan jauh berbeda. Berikut perbedaan antara level tegangan RS232 dan TTL :

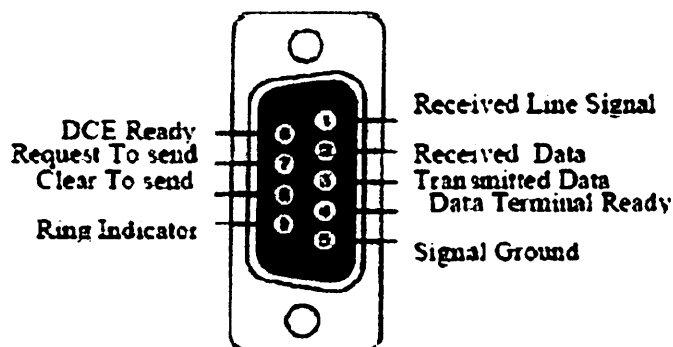


Gambar 2.17 Perbedaan Tegangan RS232 dan TTL

a. Jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki- kaki di konektor. Beberapa parameter yang ditetapkan EIA (*Electronics Industry Association*) antara lain:

- Sebuah ‘spasi’ (logika 0) antara tegangan +3 s/d +25 volt
- Sebuah ‘tanda’ (logika 1) antara tegangan -3 s/d -25 volt
- Daerah tegangan antara +3 s/d -3 volt tidak didefinisikan

b. Tegangan rangkaian terbuka tidak boleh lebih dari 25 volt (dengan acuan ground). Arus hubung singkat rangkaian tidak boleh lebih dari 500 mA. Sebuah penggerak (*driver*) harus mampu menangani arus ini tanpa mengalami kerusakan. Selain mendeskripsikan level tegangan seperti yang dibahas di atas, standard RS232 menentukan pula jenis-jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE dan DCE, semuanya terdapat 24 jenis sinyal tapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakai pun ditentukan dalam standard RS232, untuk sinyal yang lengkap dipakai konektor DB25, sedangkan konektor DB9 hanya bisa dipakai untuk 9 sinyal yang umum dipakai.



Gambar 2.18 Konektor DB 9

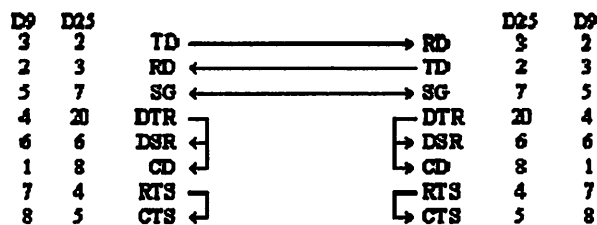
Singkatan	Keterangan	Fungsi
TD	<i>Transmit Data</i>	Untuk pengiriman data serial (TDX)
RD	<i>Receve Data</i>	Untuk penerimaan data serial (RDX)
RTS	<i>Request To Send</i>	Sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran data
CTS	<i>Clear To Send</i>	Digunakan untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data
DSR	<i>Data Set Ready</i>	Memberitahukan UART bahwa modem siap untuk melakukan pertukaran data
CD	<i>Carrier Detect</i>	Saat modem mendeteksi suatu 'carrier' dari modem lain maka sinyal ini akan diaktifkan
DTR	<i>Data Terminal Ready</i>	Kebalikan dari DSR untuk memberitahukan bahwa UAT siap melakukan hubungan komunikasi
RI	<i>Ring Indicator</i>	Akan aktif jika modem mendeteksi adanya sinyal dering dari saluran telepon

Tabel 2.2 Konfigurasi Pin

Sinyal-sinyal tersebut ada yang menuju ke DCE ada pula yang berasal dari DCE. Bagi sinyal yang menuju ke DCE artinya DTE berfungsi sebagai output dan DCE berfungsi sebagai input, misalnya sinyal TD, pada sisi DTE kaki TD adalah output, dan kaki ini dihubungkan ke kaki TD pada DCE yang berfungsi sebagai input. Kebalikan sinyal TD adalah RD, sinyal ini berasal dari DCE dan dihubungkan ke kaki RD pada DTE yang berfungsi sebagai output.

2.6.1 Konfigurasi Null Modem DB9

Konfigurasi Null Modem digunakan untuk menghubungkan dua DTE dengan diagram pengkabelan yang dapat dilihat pada gambar dibawah. Dalam hal ini hanya dibutuhkan tiga kabel antar DTE, yakni untuk TxD, RxD dan Gnd. Cara kerjanya adalah bagaimana membuat komputer agar berpikir bahwa Komputer berkomunikasi dengan modem (DCE) bukan dengan komputer lainnya.



Gambar 2.19 Koneksi Kaki- kaki DB 9

Pada gambar diatas terlihat bahwa kaki DTR (*Data Terminal Ready*) dihubungkan ke DSR (*Data Set Ready*) dan juga ke CD (*Carrier Detect*) pada masing masing komputer, sehingga pada saat sinyal DTR diaktifkan maka sinyal DSR dan CD juga ikut aktif (konsep Modem Semu atau *Virtual Modem*). Karena Komputer dalam hal ini melakukan pengiriman data dengan kecepatan yang sama, maka kontrol aliran (*flow control*) belum dibutuhkan sehingga RTS (*Request To Send*) dan CTS (*Clear to Send*) pada masing masing komputer saling dihubungkan.

2.6.2 Transmisi Data Pada RS232

Komunikasi pada RS-232 dengan PC adalah komunikasi asinkron. Dimana sinyal clocknya tidak dikirim bersamaan dengan data. Masing-masing data disinkronkan menggunakan *clock* internal pada tiap-tiap sisinya. Gambar 2.19 Format transmisi satu *byte* pada RS232 Data yang ditransmisikan pada format diatas adalah 8 bit, sebelum data tersebut ditransmisikan maka akan diawali oleh *start bit* dengan logik 0 (0 Volt), kemudian 8 bit data dan diakhiri oleh satu *stop bit* dengan logik 1 (5 Volt).

2.6.3 Keuntungan Menggunakan Komunikasi Serial

Antar muka komunikasi serial menawarkan beberapa kelebihan dibandingkan dengan komunikasi paralel, diantaranya:

a. Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel.

Data-data dalam komunikasi serial dikirimkan untuk logika '1' sebagai tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibanding dengan paralel.

b. Jumlah kabel serial lebih sedikit.

Dua perangkat komputer yang berjauhan dengan hanya tiga kabel untuk konfigurasi *null modem*, yakni TxD (saluran kirim), RxD (saluran terima) dan *Ground*, akan tetapi jika menggunakan komunikasi paralel akan terdapat dua puluh hingga dua puluh lima kabel.

c. Komunikasi serial dapat menggunakan udara bebas sebagai media transmisi.

Pada komunikasi serial hanya satu bit yang ditransmisikan pada satu waktu sehingga apabila transmisi menggunakan media udara bebas (*free space*) maka dibagian penerima tidak akan muncul kesulitan untuk menyusun kembali bit bit yang ditransmisikan.

d. Komunikasi serial dapat diterapkan untuk berkomunikasi dengan mikrokontroler.

Hanya dibutuhkan dua pin utama TxD dan RxD (diluar acuan *ground*).

BAB III

PERANCANGAN SISTEM

3.1 Analisa sistem

Perancangan ini dibagi meenjadi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak yang dapat mendukung sistem kerja alat ini.

Pada perancangan perangkat lunak meliputi perancangan diagram blok sitem, perancangan *interface*, data base sistem dan penambahan menu-menu tambahan pada aplikasi. Sedangkan perancangan perangkat keras terdiri dari microkontroller, RS232 dan IC PPI 8255.

3.2 Disain Sistem

Sistem yang akan dibangun pada skripsi ini adalah sebuah aplikasi yang terhubung dengan mikrokontroler untuk mengendalikan rangkaian elektronika(*Hardware*), yang nantinya bisa difungsikan sebagai laboratorium bahasa. Untuk disain sistem yang akan dibuat adalah sebagai berikut.

a. Master Kontrol

Master kontrol pada skripsi ini berfungsi sebagai pusat pengendali dan juga sebagai aplikasi multimedia. Nantinya master Kontrol ini akan mengirimkan dan menerima perintah dari mikrokontroler, dan nantinya dalam aplikasi master Kontrol inilah seluruh pusat pengolahan data di lakukan. Aplikasi master Kontrol juga di lengkapi dengan media player yang nantinya akan digunakan sebagai

pembelajaran media *visual* dan *learning* bagi siswa, data base untuk media penyimpanan data absensi siswa..

b. Minimum Sistem

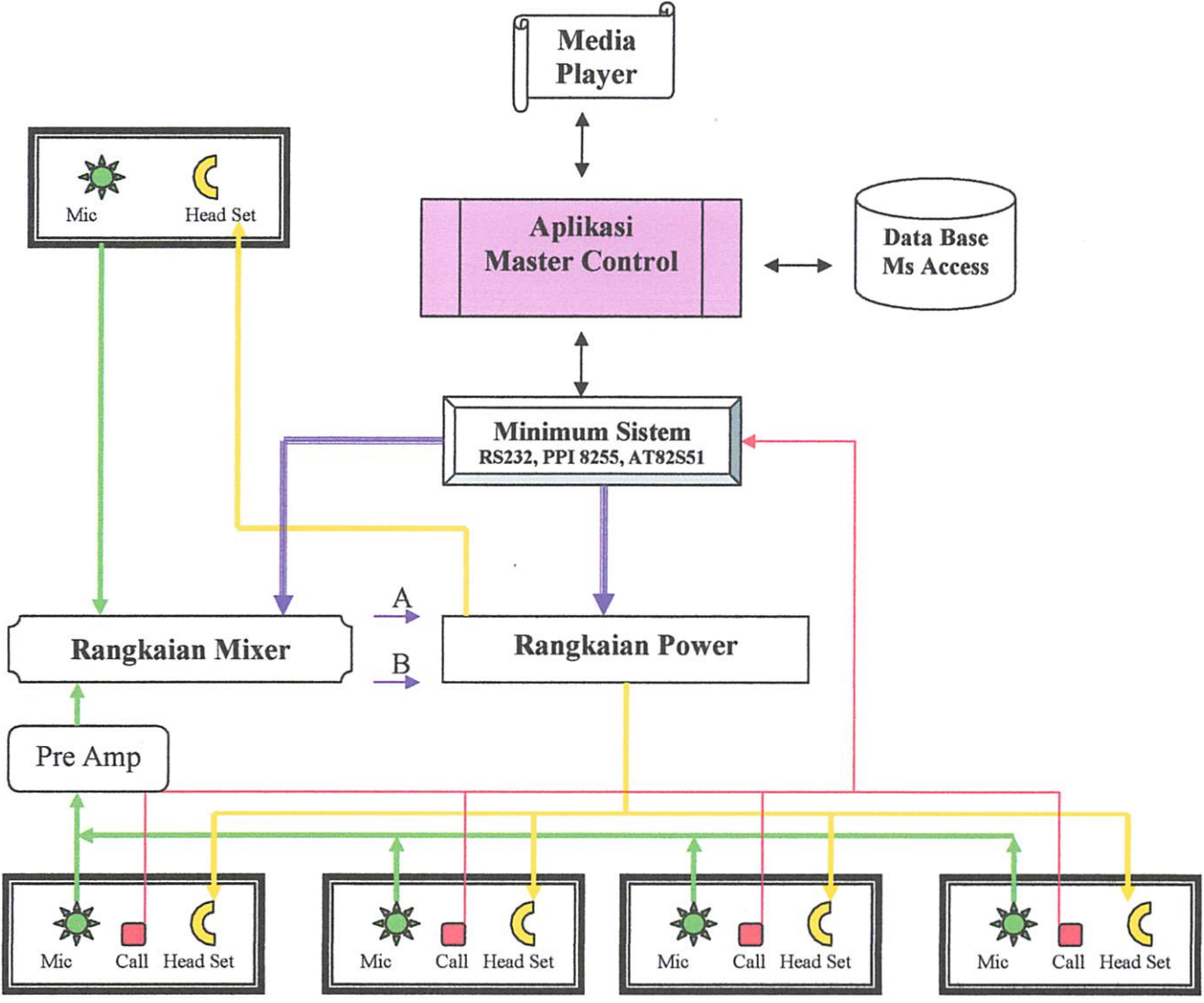
Microkontroller ini difungsikan sebagai *smart swich* yang dikendalikan oleh aplikasi master kontrol. Perintah yang diberikan dari master kontrol nantinya akan diterjemahkan menjadi tindakan switching pada panel penguat mic siswa. Microcontroller ini juga akan menerima perintah dari panel siswa berupa tekanan penekanan tombol panggilan (*call*) dan akan meneruskannya ke aplikasi master kontrol yang nantinya akan di fungsikan sebaga *call* siswa.

c. Rangkaian Elektro

Penguat mic siswa adalah rangkaian komponen siswa yang nantinya akan mengolah sinyal audio *analog* sehingga suara yang dihasilkan bisa di dengar oleh semua siswa dengan lancar dan jernih.

3.3 Diagram Blok Sistem

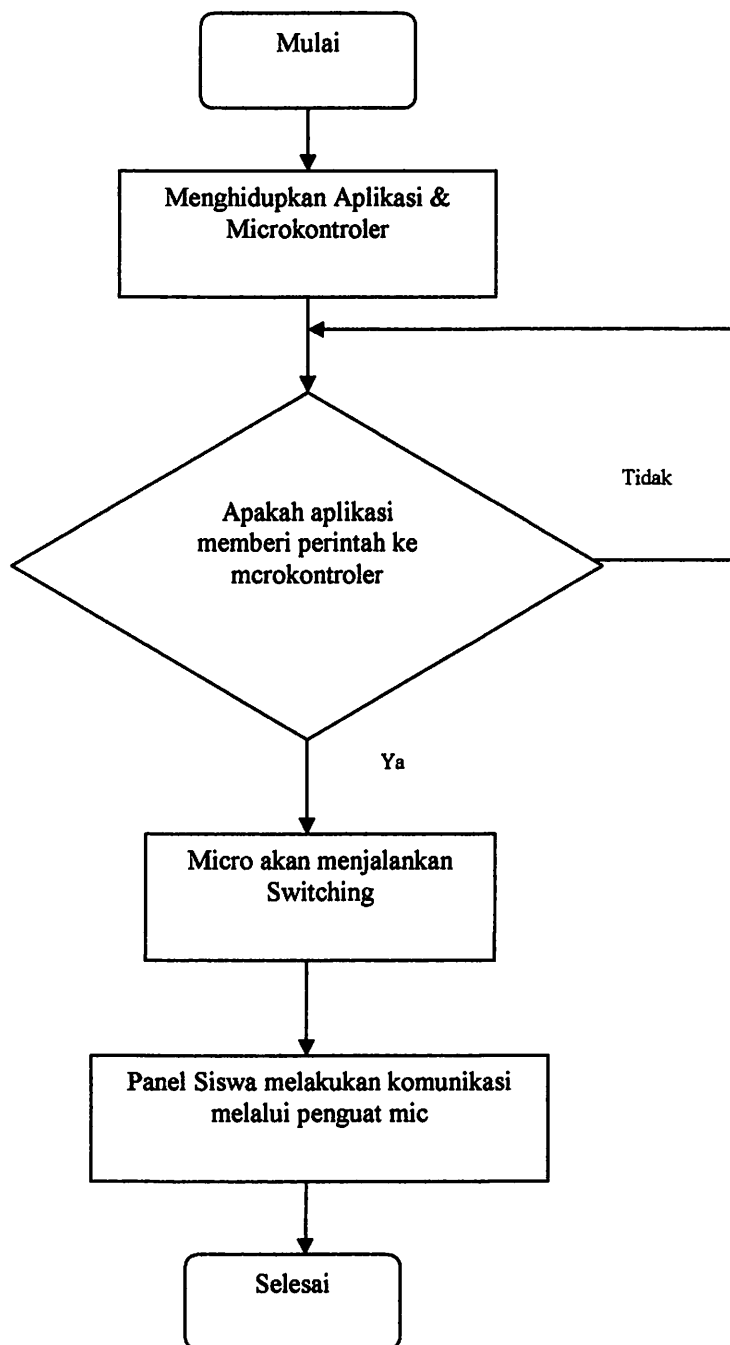
Berikut ini adalah diagram blok dari sistem yang akan dibuat, diagram blok ini menggambarkan interaksi antar komponen-komponen yang terlibat dalam pembuatan sistem.



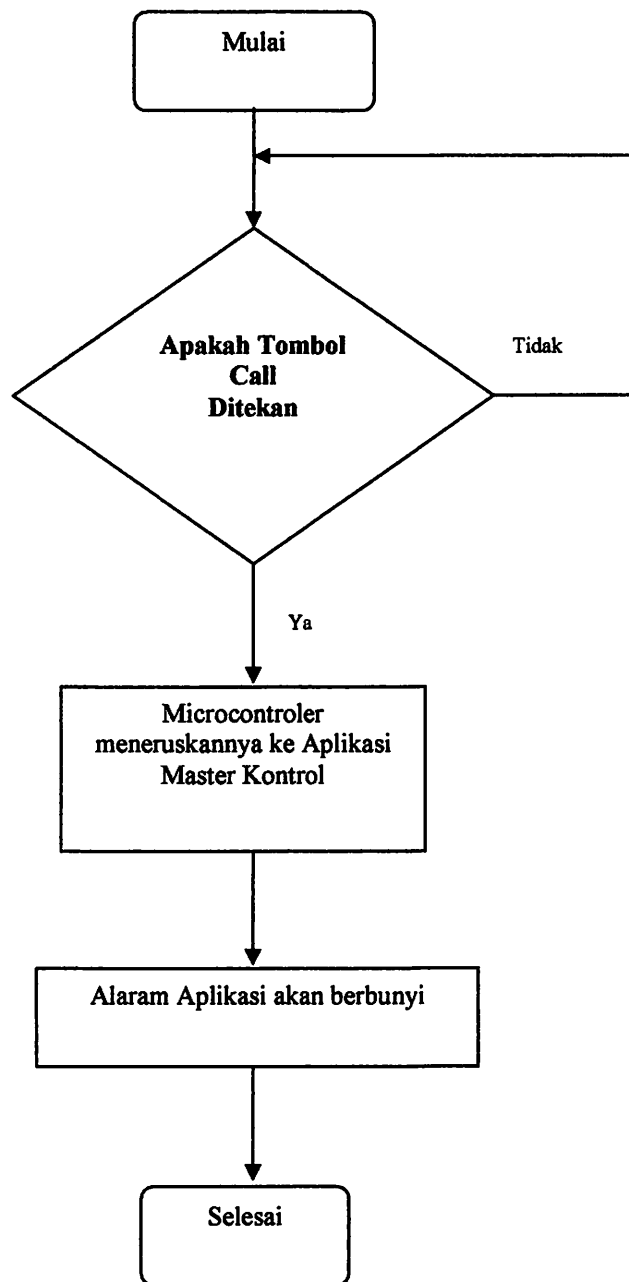
Gambar 3.1 Diagram Blok Sistem

3.3 Flow Chart

Sesuai dengan desain sistem yang dibuat, dapat dibuat diagram alir sebagai berikut :



Gambar 3.2 Flow Chart Switching penguat mic



Gambar 3.3 Flow Chart Tombol Call

3.5 Perancangan Perangkat Lunak (Software)

Perancangan *software* disini menggunakan bahasa pemrograman *Borlan Delphi*, yang nantinya akan digunakan untuk membuat *interface* berupa aplikasi master Kontrol digital laboratorium bahasa. Dan untuk mengendalikan mikrokontroller digunakan bahasa *Assembler* .Aplikasi ini akan mengirimkan dan menerima perintah dari mikrokontroler

3.5.1 Disain menu Aplikasi

Rancangan menu yang dibuat, disesuaikan dengan sistem yang telah dirancang.

a. Panel Siswa

Panel siswa digunakan untuk memilih dengan siapa guru akan berkomunikasi

b. Menu Audio Input

▪ Line In

Line in di gunakan untuk inputan *audio* dari luar misalnya VCD MP3Player atau TV

▪ Tape

Tape adalah Inputan *Audio* dari pemutar musik/ *tape recorder*

c. Menu Komunikasi

• A

Tombol A pada menu komunikasi digunakan untuk berkomunikasi dengan siswa yang berada pada panel bernomor Ganjil

- **B**

Tombol B pada menu komunikasi digunakan untuk berkomunikasi dengan siswa yang berada pada panel bernomor Genap

- **All**

Tombol All pada menu komunikasi digunakan untuk berkomunikasi dengan semua siswa (gabungan antara A dan B)

- **Individu**

Tombol Individu pada menu komunikasi digunakan untuk berkomunikasi dengan salah satu atau sebagian siswa saja tanpa di dengar oleh siswa lain.

c. Menu Pilihan Tambahan

- **Pairing**

Tombol *Pairing* pada menu tambahan digunakan untuk mengelompokkan komunikasi siswa

Contoh :

ketika Tombol pairing dalam keadaan *on* maka panel siswa no 1 dan 2 dapat berkomunikasi, panel no 3 dan 4 dapat berkomunikasi. Akan tetapi antara kelompok panel siswa no 1 dan 2 tidak akan di dengar oleh kelompok siswa no 3 dan 4.

d. Audio Output

- **ROOM**

Tombol ROOM pada Audio Output berfungsi untuk menghasilkan keluaran suara pada Speaker ruangan sehingga percakapan ataupun aktifitas komunikasi lab dapat didengar oleh semua *audience* / peserta.

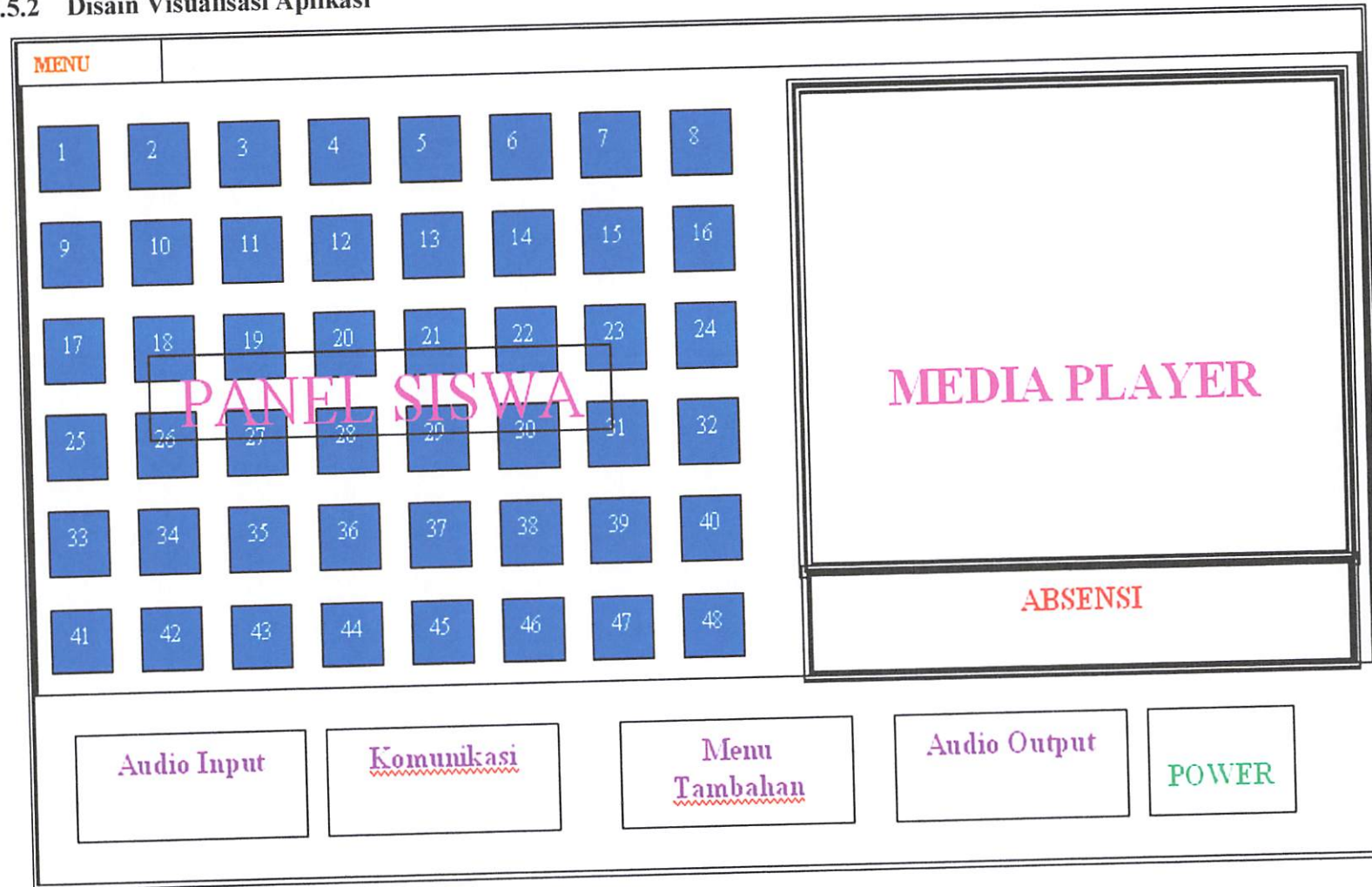
e. Media Player

Untuk memutar file *audio* dan *video* yang ingin digunakan sebagai materi di kelas.

f. ABSENSI

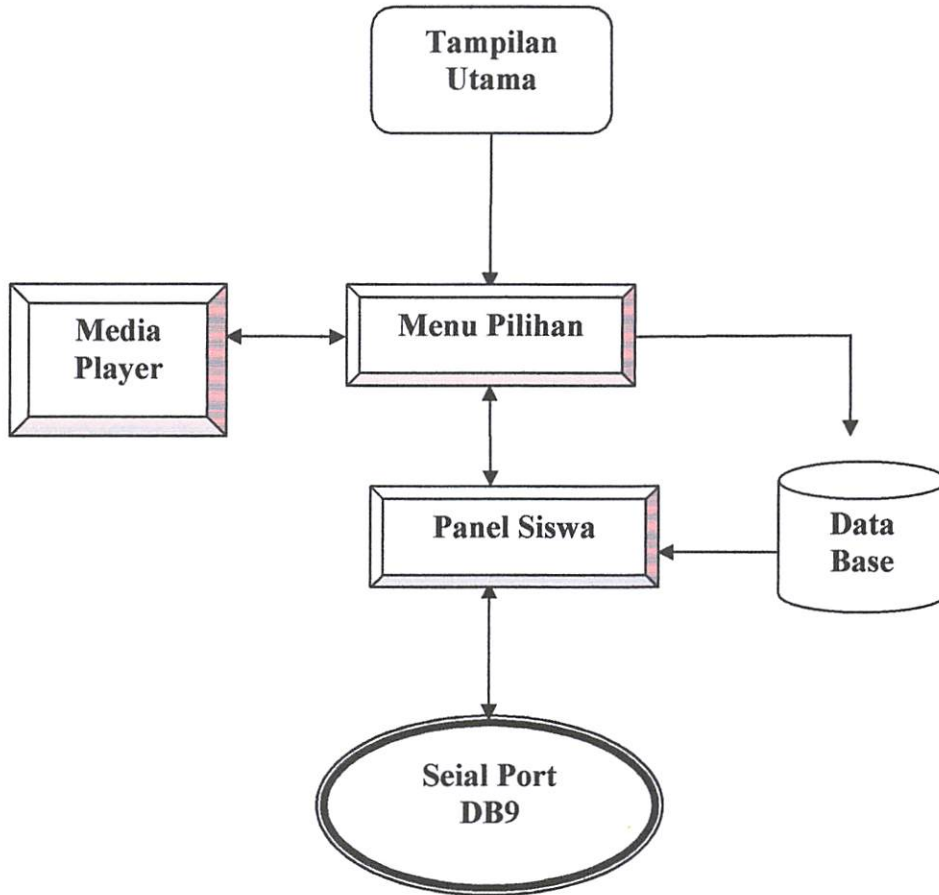
Untuk mengganti dan menambah absensi yang digunakan untuk mengganti nama di tombol – tombol panel siswa

3.5.2 Disain Visualisasi Aplikasi



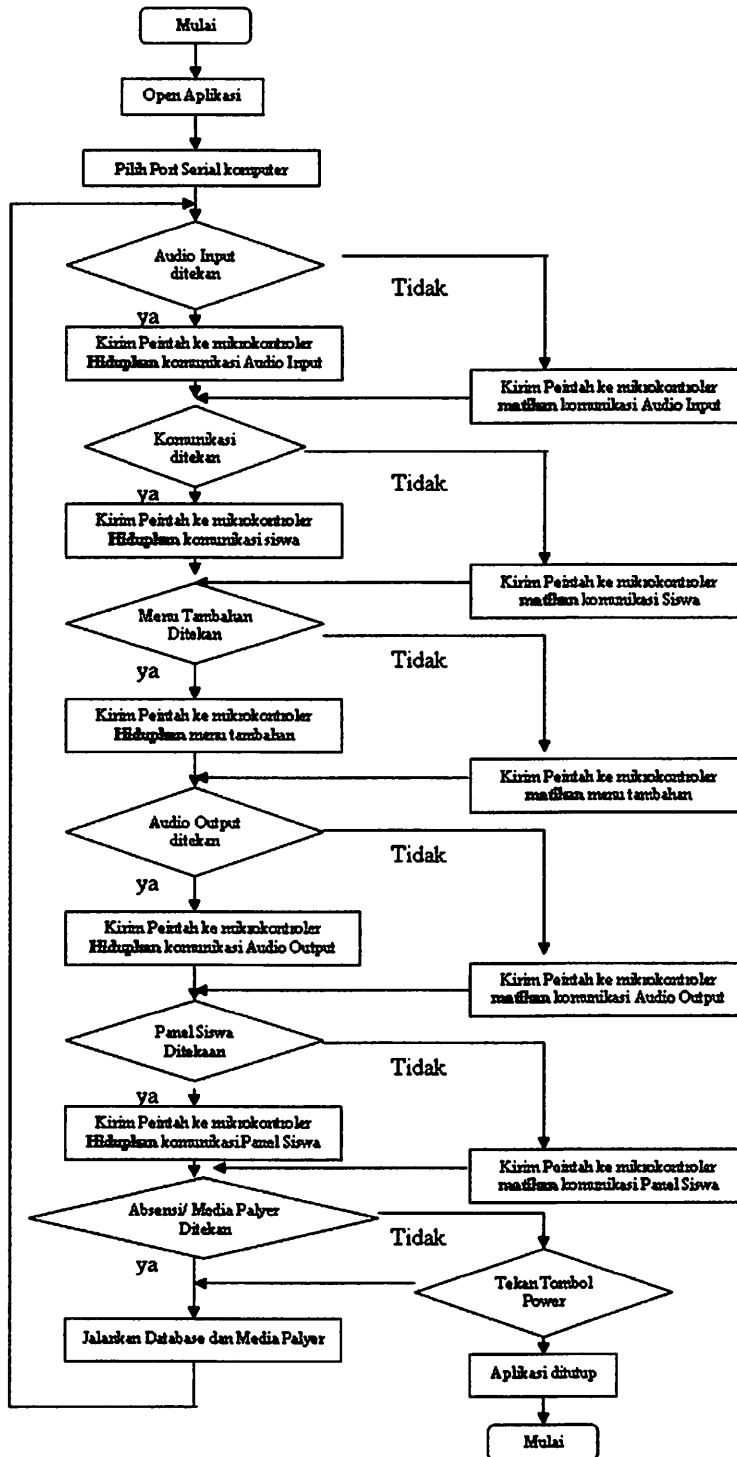
Gambar 3.4 Disain Visual Aplikasi

3.5.3 Diagram Blok Aplikasi



Gambar 3.5 Diagram Blok Aplikasi

3.5.4 Flow Chart Aplikasi



Gambar 3.6 Flow Chart Aplikasi

3.5.5 Disain Software mikrokontroler AT89s51

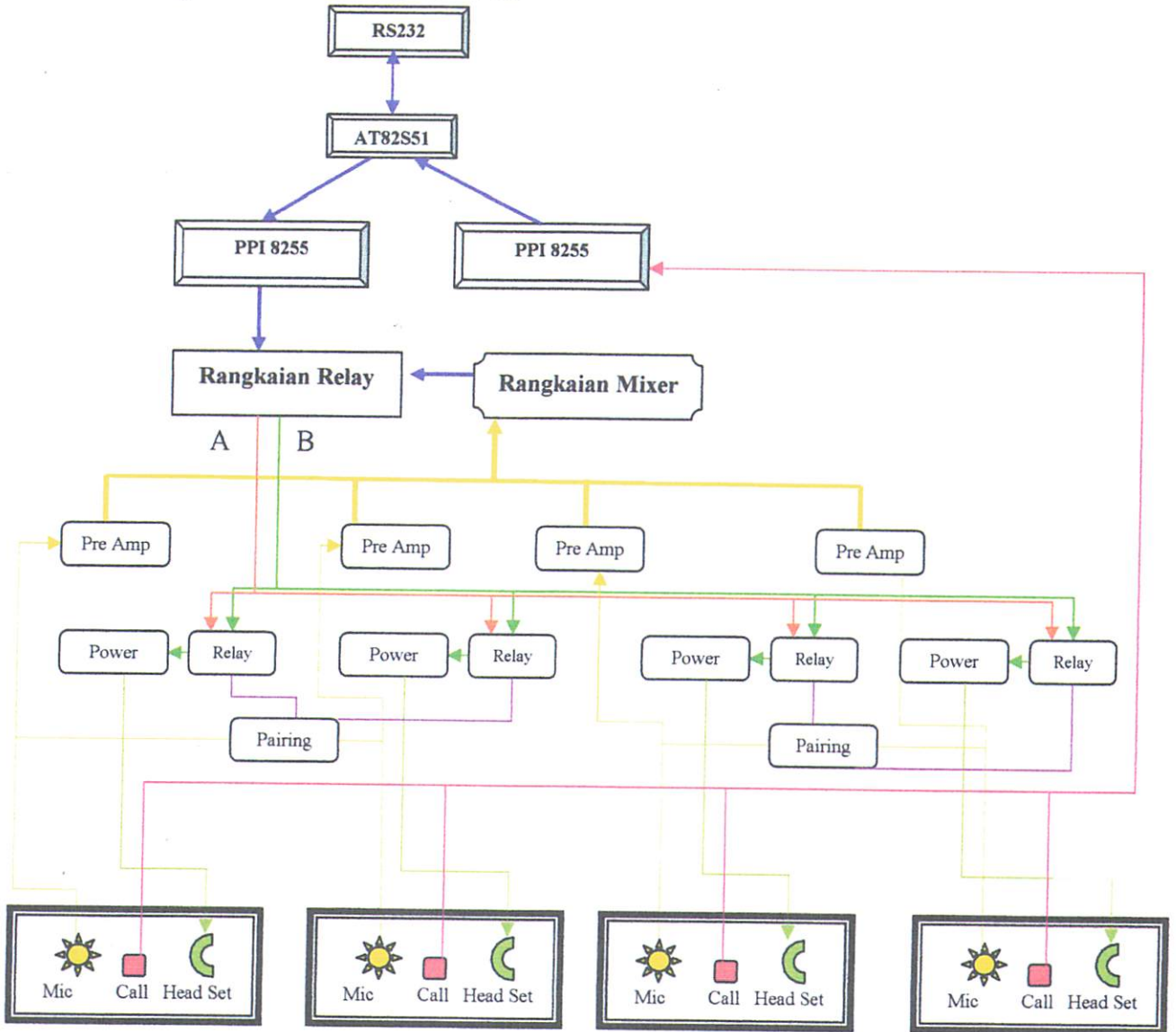
Supaya mikrokontroler dapat bekerja sesuai dengan fungsinya, maka perlu dibuat sebuah program yang mengendalikan setiap proses yang akan dilakukan oleh mikrokontroler. Program *software* ditulis dengan pengolah kata (teks) dan disimpan dengan *H51* atau *ASM*.

Setelah program selesai dituliskan, maka program tersebut harus di *compile* supaya berubah menjadi bahasa mesin yang kemudian dimasukan pada ROM Mikrokontroler.

Salah satu *compiler* program ini adalah *Roeltech* menjadi file *heksa decimal* (berekstensi HEX). File inilah yang akan dimasukan ke *ROM* mikro kontroler dengan menggunakan EPROM programmer.

3.6 Perancangan Perangkat Keras (Hardware)

3.6.1 Diagram Blok Perangkat Keras

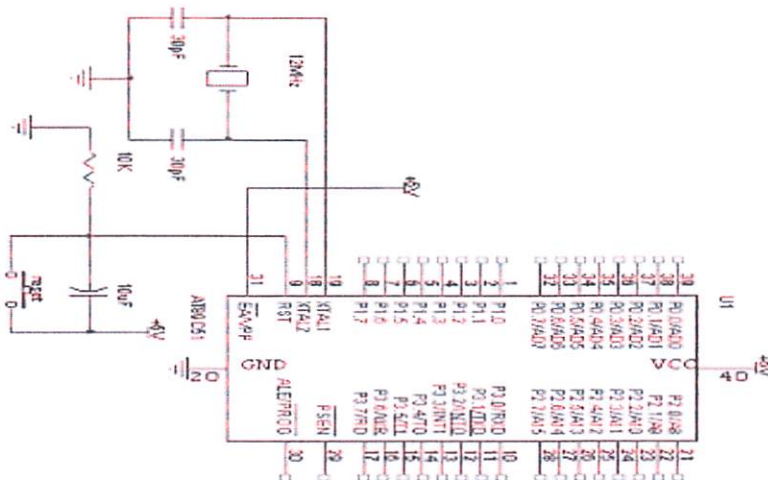


Gambar 3.6 Diagram Blok Perangkat Keras

3.6.2 Rangkaian Mikrokontroler AT89s51

Sebuah rangkaian yang terdiri dari sedikit komponen yang sudah dapat berjalan atau bekerja disebut dengan rangkaian minimum. Rangkaian minimum dari mikrokontroler AT89S51 terdiri dari 3 Kapasitor, 1 IC Mikrokontroler, dan 1 Resistor, 1 Kristal dapat dilihat pada gambar 3.1 dengan rangkaian yang sederhana ini dapat dibuat bermacam- macam alat dengan menambahkan sedikit komponen tambahan lainnya.

Dalam perancangan ini Dari rangkaian minimum ini nantinya akan digunakan sebagai pusat pengendalali untuk rangkain IC PPI.



Gambar 3.7 Gambar Rangkaian Minimum

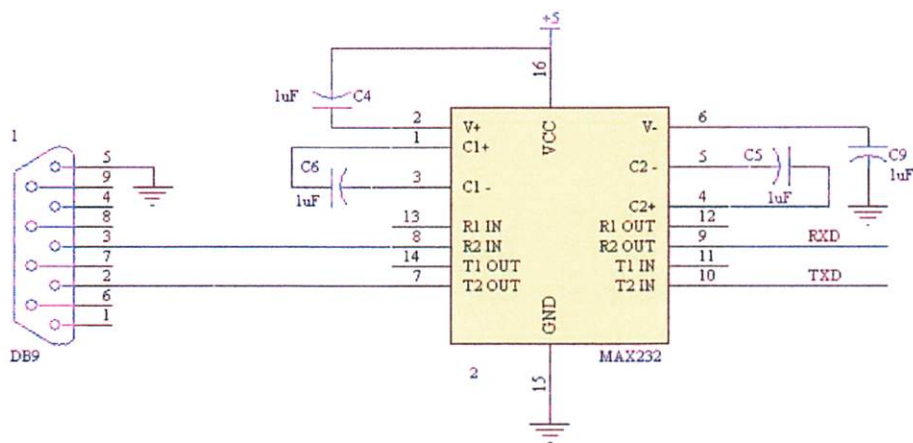
3.6.3 Rangkaian RS 232

Mikrokontroler AT89s51 dilengkapi dengan port serial, yang memungkinkan kita mengirimkan data dalam format serial. Pin 10 (P3.0/RXD) dan pin 11 (P3.1/TXD) dipakai untuk komunikasi serial antara minimum sistem AT89s51 dengan Komputer. Apabila akan menghubungkan mikrokontroler AT89S51 dengan PC melalui serial Port, maka tegangan TTL dari mikrokontroler AT89s51 harus diubah kedalam standart RS-232 yang menggunakan level – 10 Volt untuk logika “1” dan +10 untuk Logika “0”.

Sebelum di imputkan ke Komputer dibutuhkan rangkaian converter tegangan. Microcontroler mempunyai output logika *high* dihasilkan dari tegangan 5 volt dan logika *low* sebesar 0 Volt. Tegangan ini akan sering mengakibatkan terjadinya kesalahan di dalam pengiriman dan penerimaan data dikarenakan rugi-rugi dari kabel. RS-232 berfungsi untuk memperlebar range tegangan karena berada di kisaran +10V dan -10V, dengan range yang lebar ini kesalahan karena rugi-rugi sistem komunikasi dari mikrokontroler ke Komputer tidak mempengaruhi nilai data yang dikirim.

Untuk menghubungkan antara rangkaian penerima data dengan PC (Personal Komputer) digunakan *RS 232* yang menggunakan *IC MAX 232*, gambar rangkaian dapa dilihat pada gambar 3.2 dimana *IC* ini mempunyai (3) mode kerja, yaitu :

- a.. Voltage converter (pengubah tegangan)
- b.. Trasmitter (pemancar)
- c. Receiver (penerima)



Gambar 3.8 Rangkaian RS232

Kabel penghubung dengan Komputer digunakan DB9. DB 9 pada mikrokontroler Pin 5 dihubungkan dengan ground, pin 2 dihubungkan dengan pin 8 dari RS 232 dan pin 3 dihubungkan dengan pin 7. DB 9 dihubungkan dengan Komputer adalah pin 2 dihubungkan dengan pin 3 dari DB9 mikroKontroler dan sebaliknya pin 2 dihubungkan dengan pin 3, ini dilakukan supaya pengiriman data dari mikrokontroler di terima di penerima dari Komputer atau sebaliknya.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Aplikasi Master kontrol digital ini di implementasikan pada komputer yang terhubung dengan rangkaian hardware. Dalam pengujiannya aplikasi ini diinstal pada 1 komputer dan terhubung dengan rangkaian *hardware* yang terhubung dengan konektor DB9, *Hardware* tersebut telah dilengkapi dengan 4 sample *Headset* dan *mikrofon*.

4.2 Cara menjalankan aplikasi

Untuk menjalankan aplikasi yang telah diinstal maka pada *desktop* pilih *start menu*, kemudian pilih *All Program*, kemudian pilih Lab bahasa dan click LAB BAHASA.



Gambar 4.3 Tampilan Pertama Kali Program dijalankan

Pada kondisi seperti pada gambar 4.3 Tersebut merupakan kondisi dimana program tersebut baru pertama kali dijalankan.

4.2.1 Menu Aplikasi

Pada aplikasi master kontrol ini memiliki beberapa menu seperti :

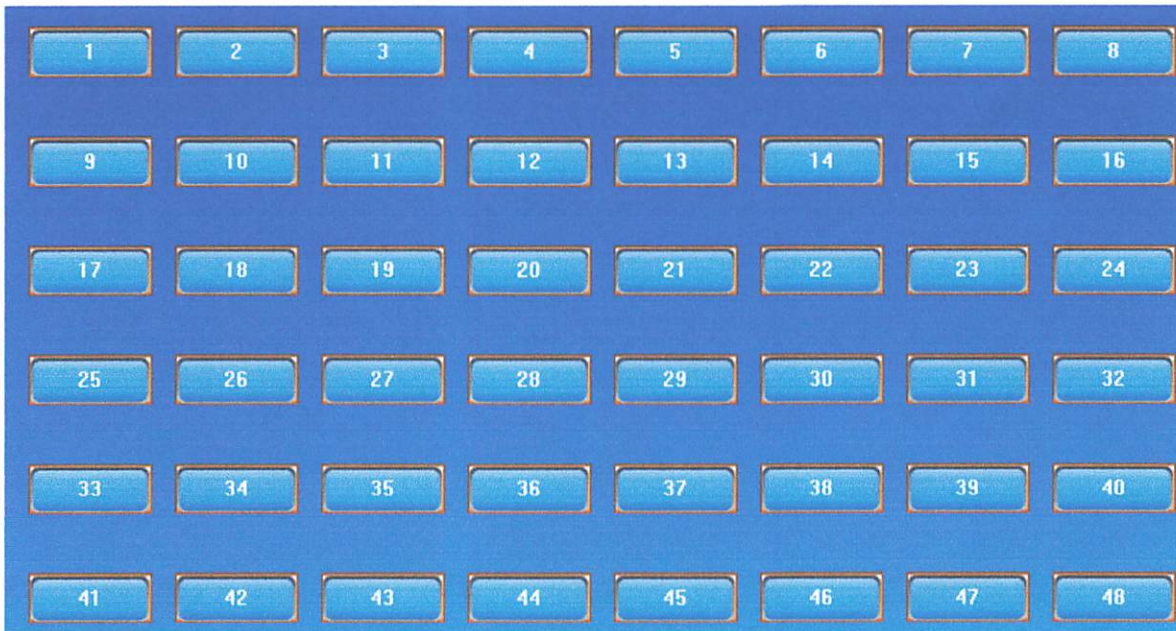
a. Panel Siswa

Pada tombol panel siswa ini terdapat instruksi untuk mengirimkan data ke dalam rangkaian mikrokontroler. Untuk mengakses serial port komputer digunakan component tambahan yaitu *Vacomm* yang terdapat pada paket *component TMs Async32*



Source code untuke mengaktifakan komunikasi yaitu :

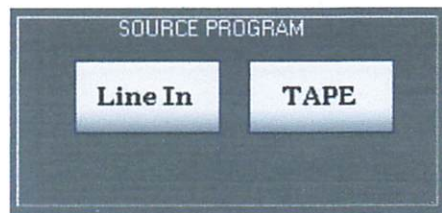
```
Vacomm1.WriteChar(char($5B));
```



Gambar 4.4 Panel Siswa

b. Audio Input

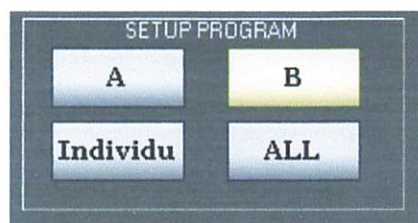
Pada menu ini inputan audio hanya bisa dipilih salah satu saja, jadi apabila salah satu tombol sudah terpilih dan tombol lain di tekan maka otomatis tombol tadi akan berganti.



Gambar 4.5 Menu Audio Input

c. Menu Komunikasi

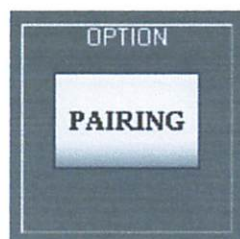
Digunakan untuk memilih cara berkomunikasi dengan siswa.



Gambar 4.6 Menu Komunikasi

d. Menu Pilihan Tambahan

Digunakan untuk memilih mode *Audio mono* atau *stereo* dan memilih mode komunikasi pengelompokan siswa.



Gambar 4.7 Menu tambahan

e. **Menu *Audio Output***

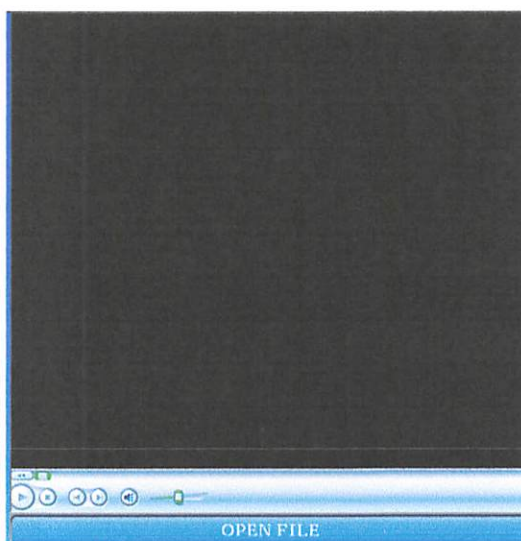
Menu ini berfungsi untuk memilih mode keluaran *Audio. Room* untuk memperdengarkan suara ke ruangan kelas.



Gambar 4.8 Menu *Audio Output*

f. **Media player**

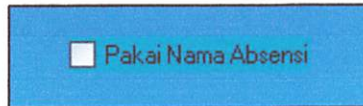
Media player ini mengambil *component activex control* dari windows. Fungsi dan fasilitas yang dimiliki media player ini sama persis seperti *media player windows*, dan untuk file video yang bisa diputar dalam media player ini tergantung pada banyaknya *codec* yang dimiliki windows.



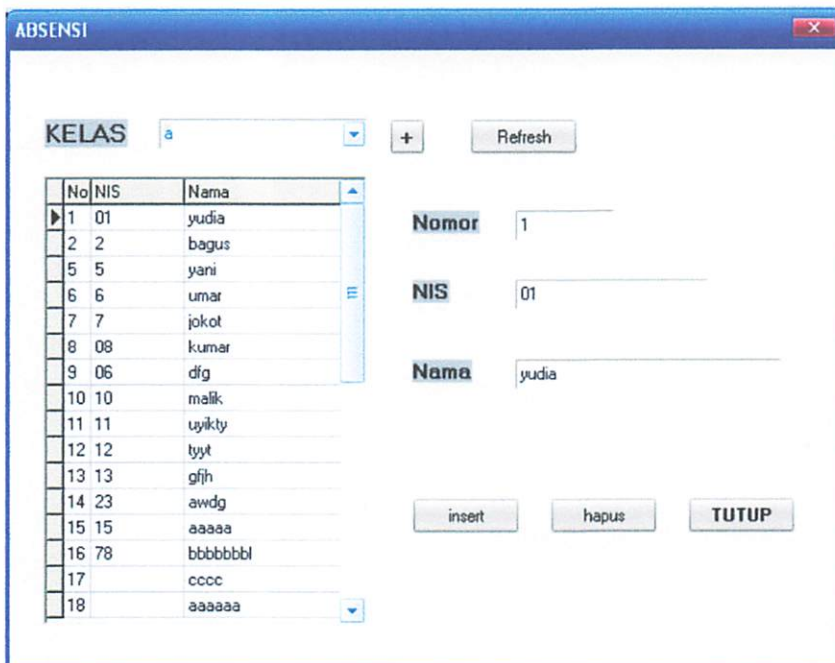
Gambar 4.9 *Media Palyer*

g. Absensi

Absensi ini nantinya akan digunakan untuk mengganti nama – nama pada panel siswa sehingga guru akan mudah mengetahui dengan siapa akan berkomunikasi.



Gambar 5.10 Tombol Pengganti Panel Nama Siswa



No	NIS	Nama
1	01	yudia
2	2	bagus
5	5	yani
6	6	umar
7	7	jokot
8	08	kumar
9	06	dfg
10	10	malik
11	11	uyikty
12	12	tyyt
13	13	gfjh
14	23	awdg
15	15	aaaaa
16	78	bbbbbbbl
17		cccc
18		aaaaaa

Gambar 4.11 Form Absensi

4.3 Pengujian Hardware

Pada pengujian ini bertujuan untuk mengetahui kinerja dari *hardware* Laboratorium Bahasa. Cara pengujian yaitu dengan mengirimkan perintah atau instruksi ke rangkaian mikrokontroler untuk mengaktifkan rangkaian *power Audio* dan penguat *mic*.

Table 4.1 Pengujian *Hardware*

No	Tombol	Perintah	Relay	Kondisi	Headset
1.	Panel 1	Vacomm1.WriteChar(char(\$5B));	Aktif	Terhubung	Bersuara
2.	Panel 1	Vacomm1.WriteChar(char(\$12));	Tidak Aktif	Tidak Tehubung	Diam
3.	Panel 2	Vacomm1.WriteChar(char(\$5A));	Aktif	Terhubung	Bersuara
4.	Panel 2	Vacomm1.WriteChar(char(\$11));	Tidak Aktif	Tidak Tehubung	Diam
5.	Panel 3	Vacomm1.WriteChar(char(\$59));	Aktif	Terhubung	Bersuara
6.	Panel 3	Vacomm1.WriteChar(char(\$10));	Tidak Aktif	Tidak Tehubung	Diam

4.4 Pengujian Aplikasi

Pengujian ini dilakukan untuk mengetahui kinerja dari *software* aplikasi master kontrol laboratorium bahasa.

4.4.1 Pemilihan Port Komputer

Untuk menghubungkan aplikasi ini pertama kali harus memilih *port* yang sudah aktif atau dipilih pada *combo box*. Setelah menentukan *port* yang sudah aktif tekan tombol hubungkan, jika koneksi antara aplikasi dan *port* komputer sukses maka tombol akan menyatakan “terhubung”.



Gambar 4.12 Menu pilihan port komputer



Gambar 4.13 Memilih port komputer



Gambar 4.14 Kondisi jika koneksi berhasil



Gambar 4.15 Jika koneksi tidak berhasil

4.4.2 Panel Siswa

Tombol ini berfungsi untuk mengaktifkan komunikasi antara siswa dan guru, ketika tombol ini ditekan maka aplikasi akan mengirimkan perintah untuk menghubungkan komunikasi antara guru dengan siswa, yang berupa data kepada rangkaian mikro kontroler.

```
Vacomm1.WriteChar(char($5B));
```



Gambar 4.16 Tombol panel siswa ketika belum ditekan



Gambar 4.17 Tombol panel siswa ketika sudah ditekan

Ketika tombol panel siswa berwarna kuning di tekan maka perintah untuk memutus komunikasi akan dikirim kerangkaian mikro kontroler.

```
Vacomm1.WriteChar(char($I3));
```

Tabel 4.2 Panel Siswa

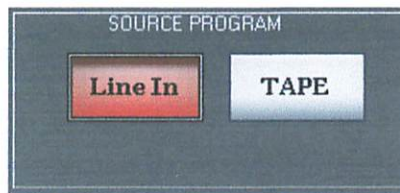
No	Tombol Panel 1	Perintah	Relay	Headset
1	On	Hubungkan komunikasi siswa no 1	Terhubung/ aktif	Bisa berkomunikasi
2	Off	Putuskan komunikasi siswa no 1	Putus/ tidak aktif	Tidak bisa berkomunikasi

4.4.3 Audio Input

Untuk memilih sumber audio kita harus mengaktifkan salah satu dari pilihan dari tombol menu audio input. secara otomatis tombol menu akan bergantian aktif karena untuk menu ini kita tidak bisa mengaktifkan keseluruhan tombol.

Table 4.3 Kondisi tombol audio input

No	Line In	Tape	Perintah	Relay	Headset
1	Aktif	Tidak Aktif	Terhubung dengan <i>Audio</i> dari komputer	<i>Line in</i> = aktif	Suara dari komputer
2	Tidak Aktif	Aktif	Terhubung dari <i>Audio Tape</i>	<i>Tape</i> = aktif	Suara dari <i>Audio Tape</i>



Gambar 4.18 Menu audio input ketika sudah aktif

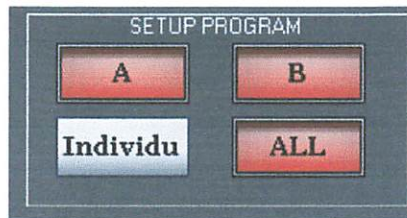
4.4.4 Menu komunikasi

Pada menu ini terdapat tombol 'A', 'B', 'ALL', dan 'INDIVIDU' dimana keempat tombol ini memiliki fungsi dan kegunaan yang berbeda.

Table 4.4 Kondisi tombol menu komunikasi

No	A	B	ALL	Perintah	Relay	Headset
1	Aktif	Tidak Aktif	Tidak Aktif	Sambungkan dengan siswa bernomor ganjil	Ganjil terhubung	Suara dari siswa nomor ganjil
2	Tidak Aktif	Aktif	Tidak Aktif	Sambungkan dengan siswa bernomor genap	Genap terhubung	Suara dari siswa nomor genap
3	Aktif	Aktif	Aktif	Sambungkan dengan siswa bernomor ganjil dan genap	Semua siswa terhubung	Suara dari keseluruhan siswa

Untuk menu tombol individu hanya digunakan untuk berkomunikasi dengan nomor siswa yang dipilih tanpa di dengar oleh siswa lainnya.



Gambar 4.19 Menu komunikasi

4.4.5 Menu Pilihan Tambahan

Pengujian pada menu ini, ditujukan untuk mengetahui fungsi dan kinerja dari tombol pairing. Tombol Pairing pada menu tambahan digunakan untuk mengelompokkan komunikasi siswa

Contoh :

ketika Tombol pairing dalam keadaan on maka panel siswa no 1 dan 2 dapat berkomunikasi, panel no 3 dan 4 dapat berkomunikasi. Akan tetapi antara kelompok panel siswa no 1 dan 2 tidak akan di dengar oleh kelompok siswa no 3 dan 4.



Gambar 4.20 Menu pilihan tambahan

4.4.6 Audio Output

Pengujian pada menu ini bertujuan untuk mengetahui apakah aplikasi memberikan perintah kepada mikrokontroler untuk mengaktifkan audio yang ingin dikeluarkan.

No	Tombol Room	Perintah	Relay	Spiker Ruangan	Headset
1	Aktif	Komunikasi / <i>Audio</i> terdengar di <i>speaker</i> ruangan	Terhubung dengan spiker	Mengeluarkan suara	Tidak mengeluarkan suara
2	Tidak Aktif	Komunikasi / <i>Audio</i> terdengar di <i>head set</i>	Terhubung dengan head set	Tidak mengeluarkan suara	Mengeluarkan suara

Table 4.5 Kondisi Tombol *Audio Output*



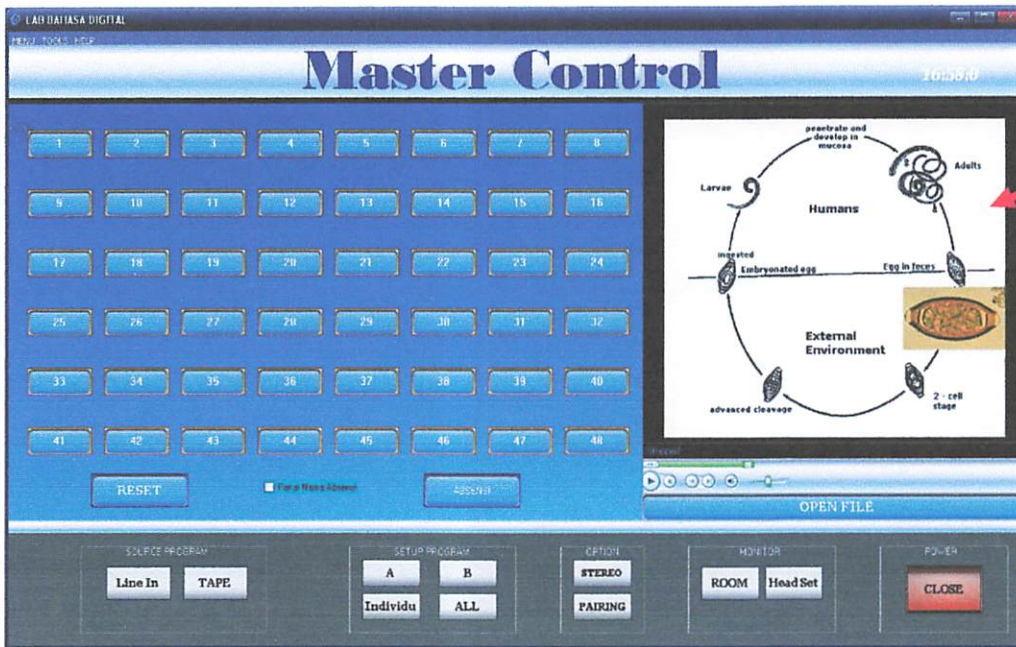
Gambar 4.21 Menu audio output

4.4.7 Media Player

Untuk memutar *file Audio* dan *video* yang ingin digunakan sebagai materi kelas, tekan tombol *Open File*. Kemudian akan muncul dialog untuk memilih file yang akan dibuka. Untuk menampilkan Video dalam *format full screen*, click 2x pada *screen video* yang sedang berjalan



Gambar 4.22 Form Yang muncul ketika Tombol Open File Di click



Gambar 4.23 Memutar *Video* menggunakan *Media Player*

4.4.8 Absensi

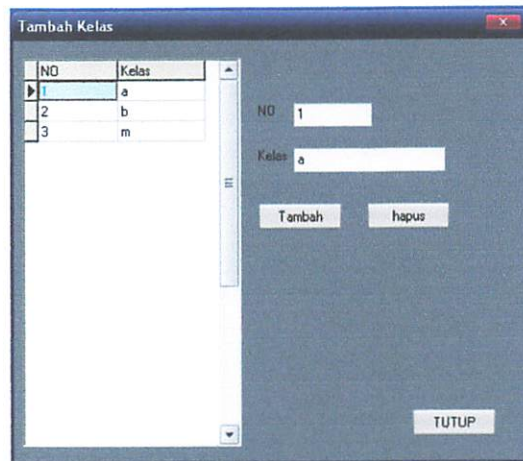
Untuk mengganti dan menambah absensi yang digunakan untuk mengganti tombol – tombol panel siswa, klik tombol absensi pada aplikasi laboratorium bahasa Digital maka akan keluar form Absensi.

No	NIS	Nama
6	6	umar
7	7	rudi
8	08	kumar
9	06	made
10	10	malik
11	11	holim
12	12	panjaya
13	13	arafah
14	23	jupri
15	15	manis
16	78	mulyono
17	987	agung
18	345	sajak
19	345	kukman
20	34	supri
3	3	maksum

Gambar 4.24 Form Absensi

a. **Tombol + (plus)**

Pada form Absensi digunakan untuk menambah kelas atau kelompok belajar baru.



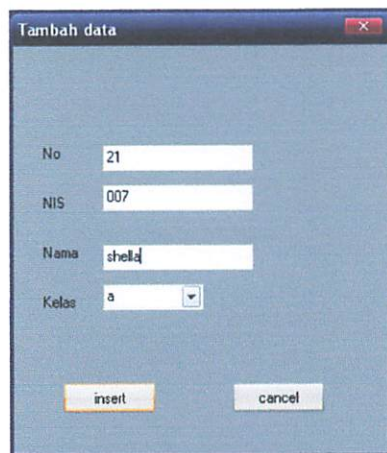
Gambar 4.24 Form Tambah Kelas

b. **Refresh**

Tombol *Refresh* di gunakan setiap kali kita selesai meng edit/menambah dan menghapus data agar data siswa yang ditampilkan sesuai dengan data terbaru.

c. **Insert**

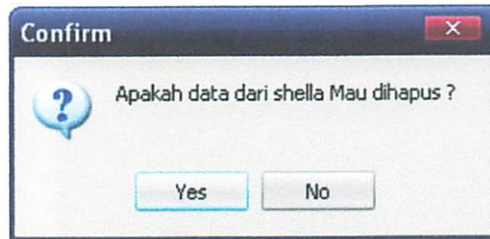
Tombol *Insert* di gunakan untuk menambah data siswa baru



Gambar 4.25 Form Tambah Data

d. **Delete**

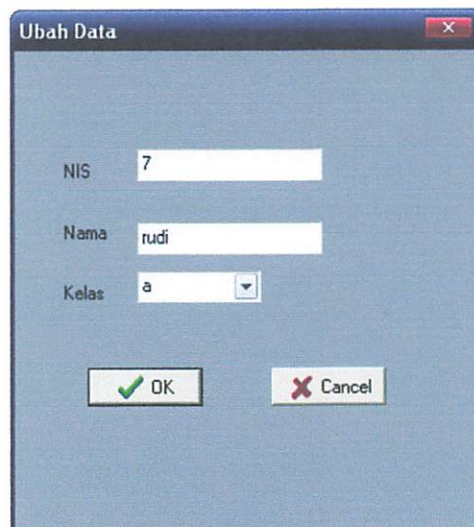
Tombol Delete digunakan untuk menghapus *record* data siswa.



Gambar 4.26 Form Konfirmasi hapus data

e. **Edit Data**

Untuk mengubah/ meng edit data siswa cukup click 2x pada grid yang berisi data siswa Maka akan keluar form ubah data.



Gambar 4.27 Form Ubah data

f. **Penggunaan Nama Absensi Pada Panel Siswa**

Untuk mengubah nama pada tombol – tombol panel siswa, centang *option* Pakai Nama *Absensi*, maka tombol - tombol tersebut akan berubah menjadi nama – nama siswa yang ada pada colom absensi.



Gambar 4.28 Contoh Panel Siswa Menggunakan Nama Absensi

BAB V

PENUTUP

5.1. Kesimpulan.

1. Dari pengujian hardware diketahui bahwa ketika mikrokontroler di beri perintah oleh aplikasi master control, maka mikrokontroler akan melakukan switching relay untuk menghubungkan komunikasi siswa sesuai dengan tombol panel yang dituju.
2. Laboratorium Bahasa ini bisa menyederhanakan perangkat kerja menjadi satu buah aplikasi masterkontrol saja.
3. Dengan sistem absensi memudahkan operator untuk mengetahui dengan siapa akan berkomunikasi.
4. Laboratorium Bahasa ini dilengkapi dengan sistem *Pairing* yang berfungsi untuk diskusi kelompok siswa.

5.2 Saran

Berikut ini adalah saran yang diberikan untuk pengembangan selanjutnya.

- 1 Untuk pengembangan selanjutnya bisa ditambahkan menu kuis untuk menambah fasilitas laboratorium bahasa.
- 2 Untuk hardware bisa menggunakan jenis IC lain yang bisa menghemat dan meminimalisir penggunaan kabel dan PCB.
- 3 Nantinya diharapkan Volume dan Tone Kontrol bisa di aplikasikan dalam bentuk digital.

DAFTAR PUSTAKA

- [1] Microcontroler Dasar, 28 Mei 2009, www.mytutorialcafe.com
- [2] Microsoft Access.30 Mei 2009, www.wikipedia.org
- [3]Akses Port.30 mei 2009. www.delphiindonesia.com
- [4] Budhi widodo, romy ; dedy irawan, joseph, interfacing pararel dan serial menggunakan Delphi.yogyakarta : graha ilmu, 2007
- [5] Microprosesor laboratorium tim, micricontroler AT89s51 dengan c++ dan assembler, yogyakarta,ANDI, 2007.
- [6].Wahana Komputer, "*Membuat Program Kreatif dan Profesional dengan DELPHI*", Elex Media Computindo, Jakarta, 2005.
- [7].Raharjo, Budi, *Langkah dan Proses tercepat menjadi programmer Kylix dan Delphi*, INFORMATIKA, 2005.
- [8].Cantu, Marco, *Mastering Delphi, Chapter 22: Graphics*, Cybex, Inc, Alameda, CA, WWW.CYBEX.COM
- [9]. Website komponen *Internet Direct (Indy)* [Http://www.indy-project.com](http://www.indy-project.com)



LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Hendra Wahyu Jaya
Nim : 05.12.575
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : **PERANCANGAN DAN PEMBUATAN PERANGKAT
LUNAK UNTUK APLIKASI MASTER KONTROL DIGITAL
LABORATORIUM BAHASA**

Dipertahankan di hadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Senin
Tanggal : 07 September 2009
Dengan Nilai : 83,5 (A) *Bay*



Ketua Majelis Penguji

Ir. H. Sidik Noertjahjono, MT
NIP.Y. 1028700163

PANITIA UJIAN SKRIPSI

Sekretaris Majelis Penguji

Ir. F. Yudi Limpraptono, MT
NIP Y. 1039500274

ANGGOTA PENGUJI

Penguji I

I Komang Somawirata, ST.MT
NIP. P.1030100361

Penguji II

Mohammad Ibrahim Ashari, ST.MT
NIP. P . 1030100358



FORMULIR PERBAIKAN SKRIPSI

Nama : Hendra Wahyu Jaya
Nim : 05.12.575
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Masa Bimbingan : 26 Mei 2009 s/d 26 November 2009
Judul Skripsi : **PERANCANGAN DAN PEMBUTAN PERANGKAT LUNAK UNTUK APLIKASI MASTER KONTROL DIGITAL LABORATORIUM BAHASA**

Tanggal	Uraian	Paraf
Penguji I 7 September 2009	* Perancangan Sistem Minimum At89s52 - Mapping Memory - Desain Schematic * Flow Chart - Program Mikrokontroler - Program Aplikasi * Demo Ulang	
Penguji II 7 September 2009	Perbaiki Alat	

Disetujui

Penguji I

I Komang Somawirata, ST.MT
NIP. P.1030100361

Penguji II

Mohammad Ibrahim Ashari, ST.MT
NIP. P. 1030100358

Mengetahui

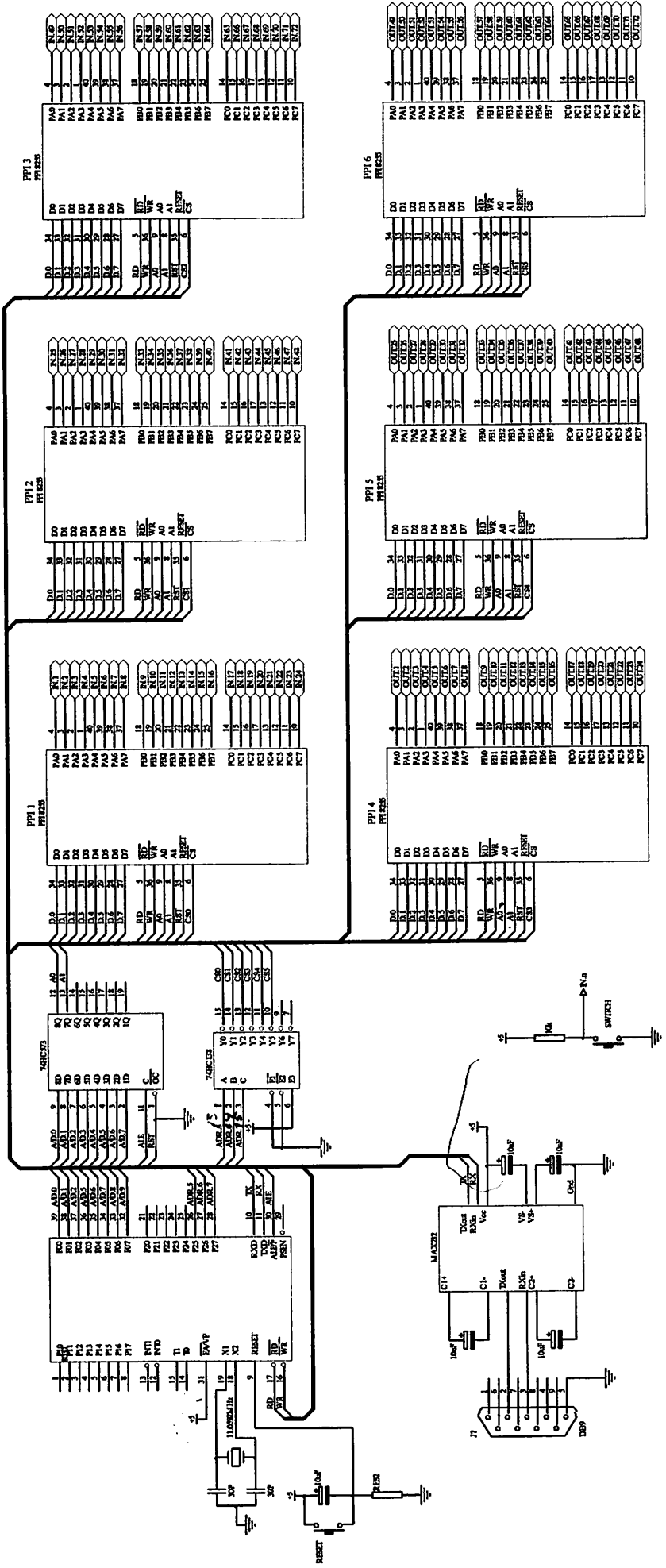
Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

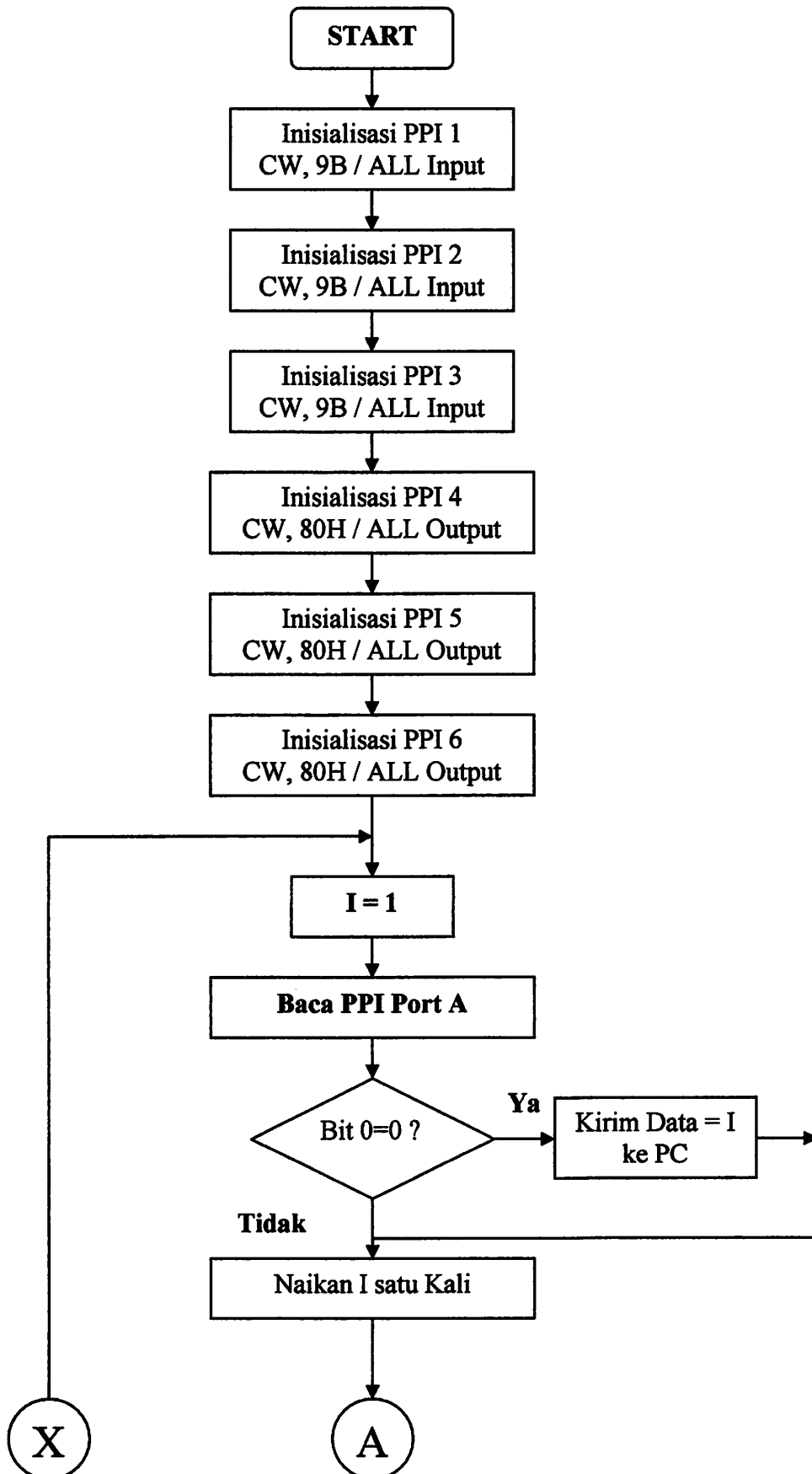
Dosen Pembimbing II

Dr. Eng. Aryunto Soetedjo, ST. MT
NIP.Y. 1030800417

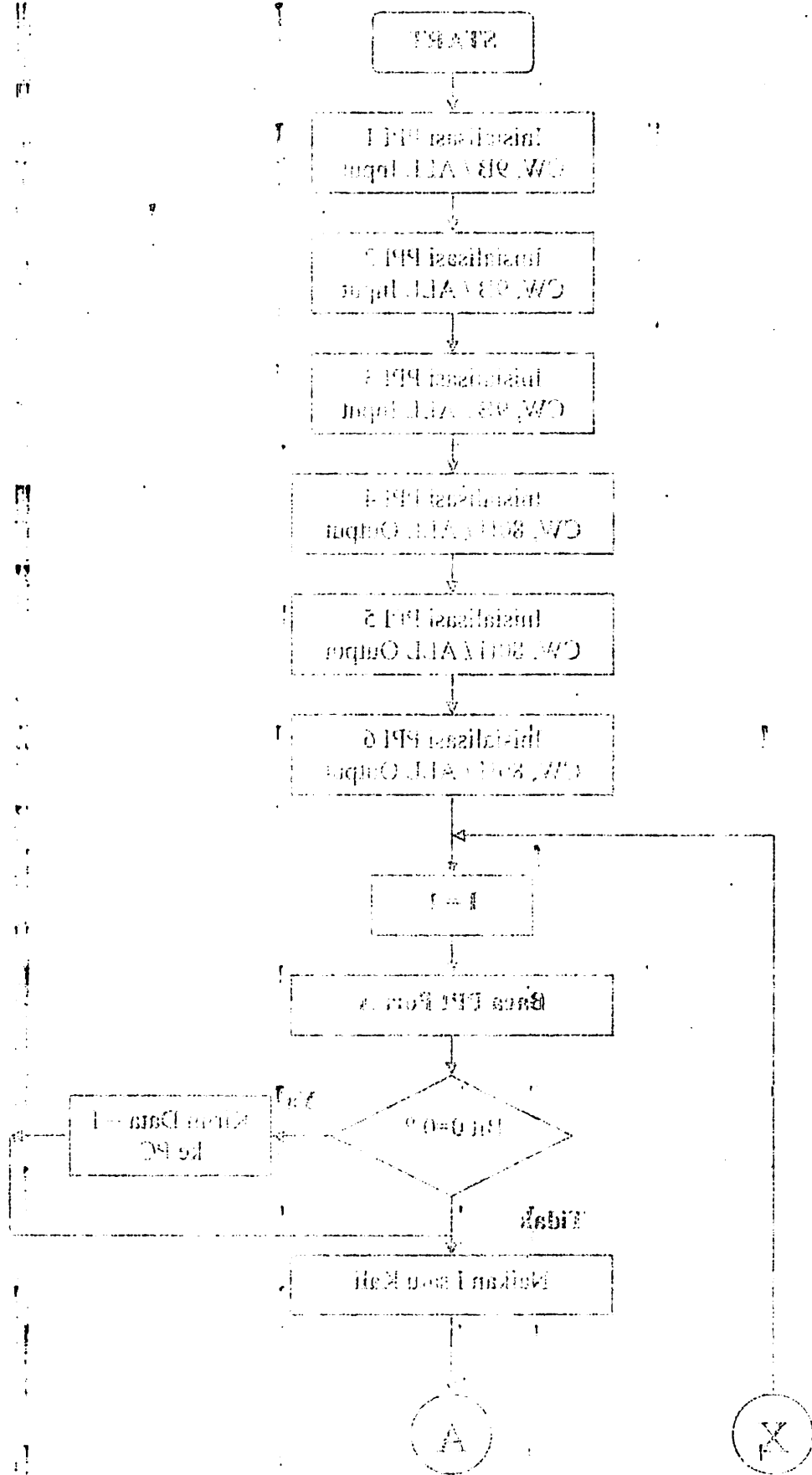
Rangkaian Control

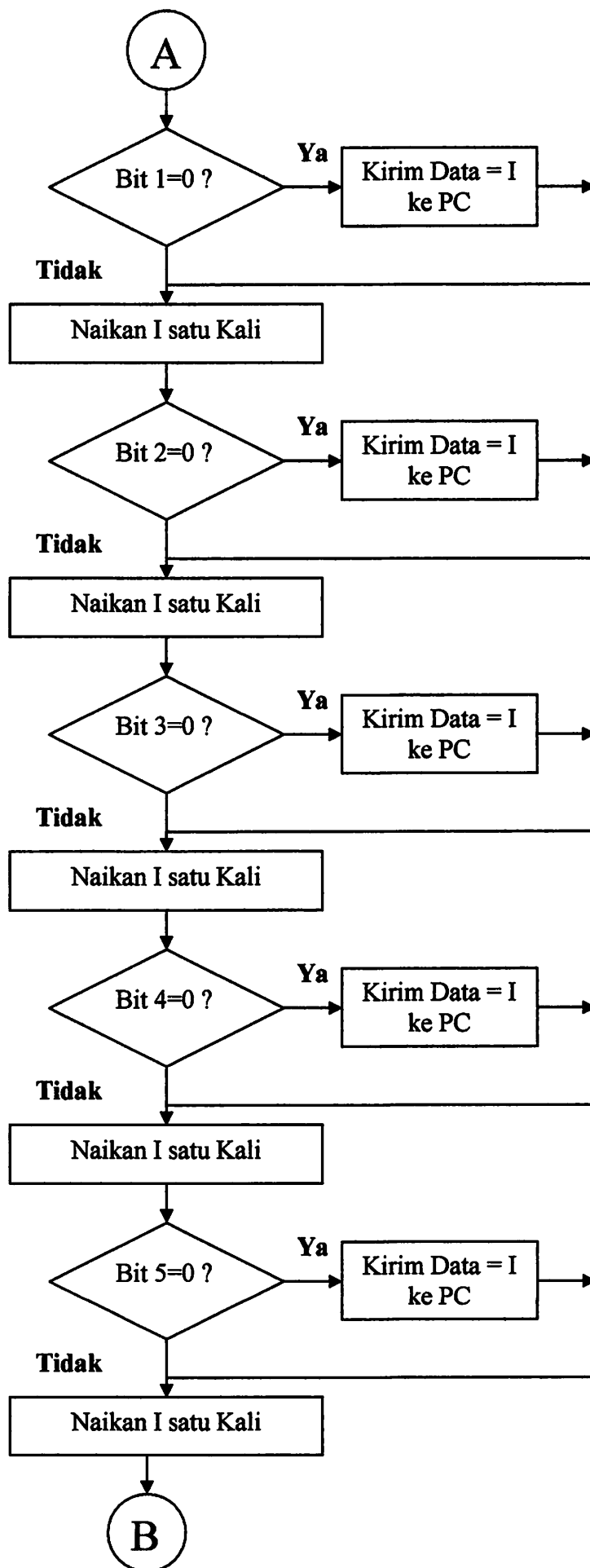


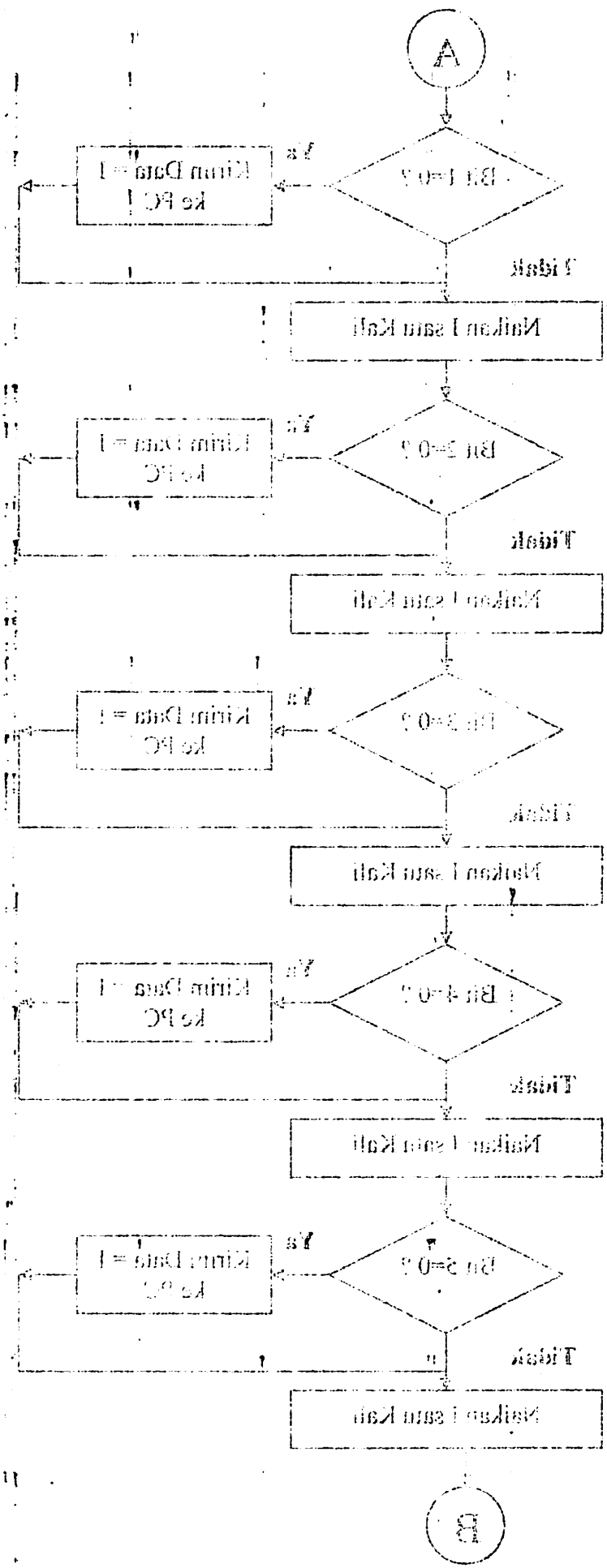
FLOW CHART PROGRAM MICROCONTROLER

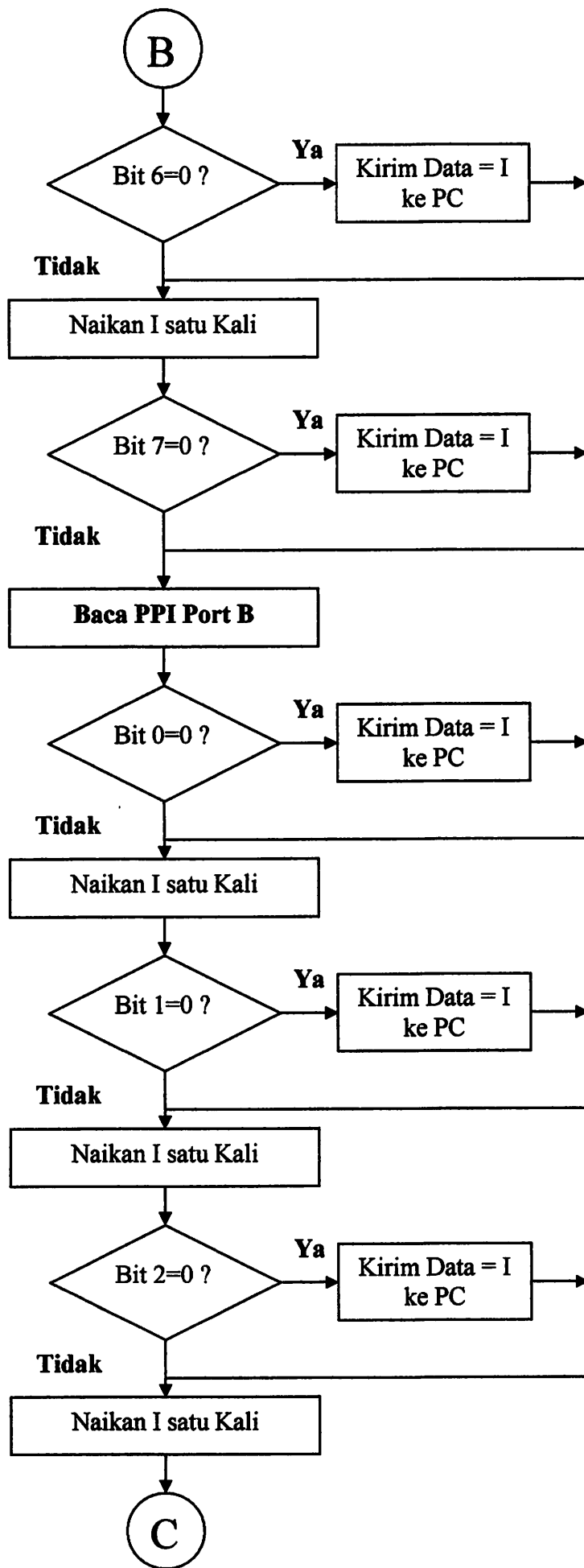


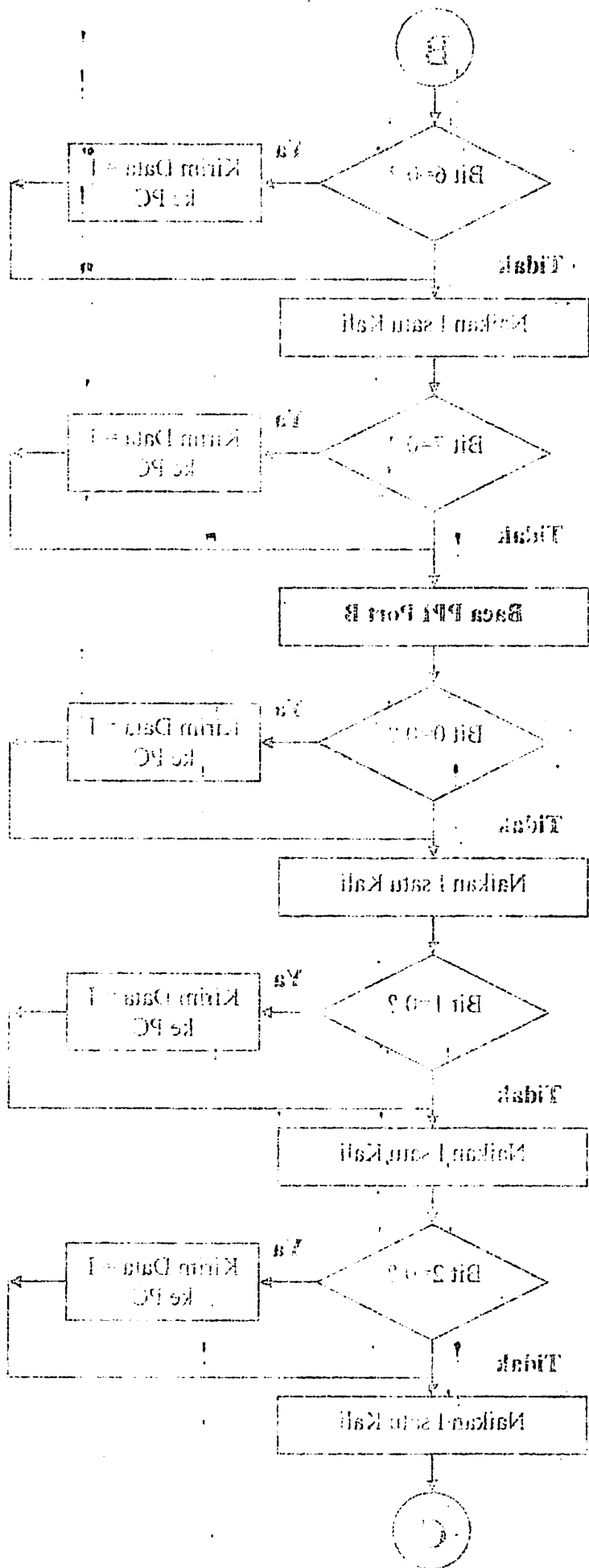
FLOW CHART PROGRAM WITH OPTIMIZATION

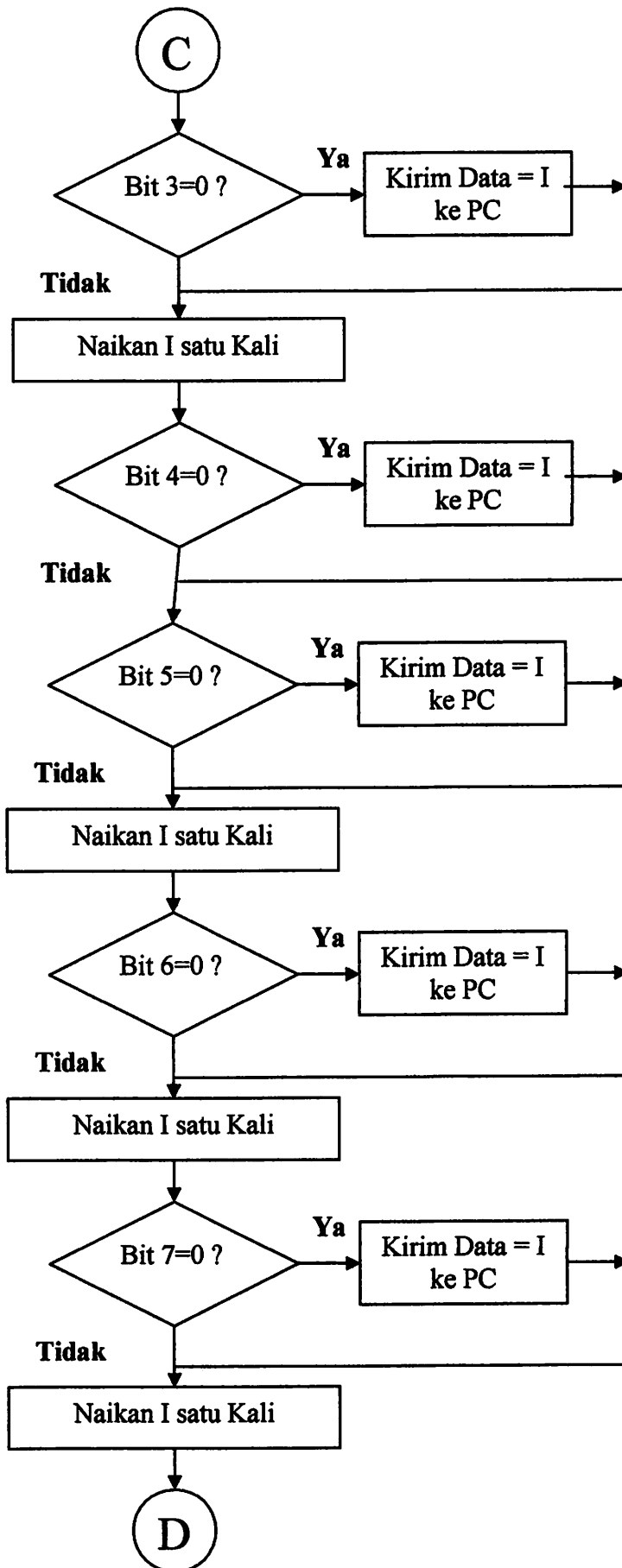


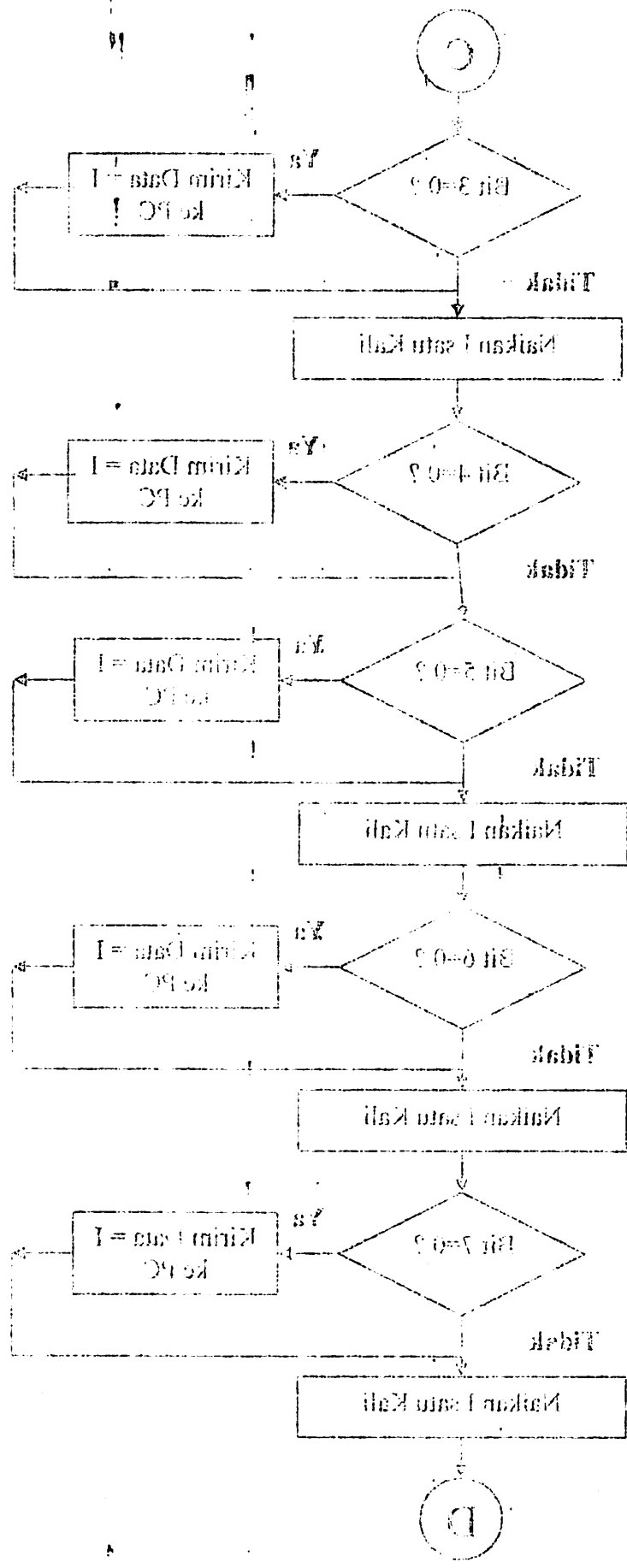


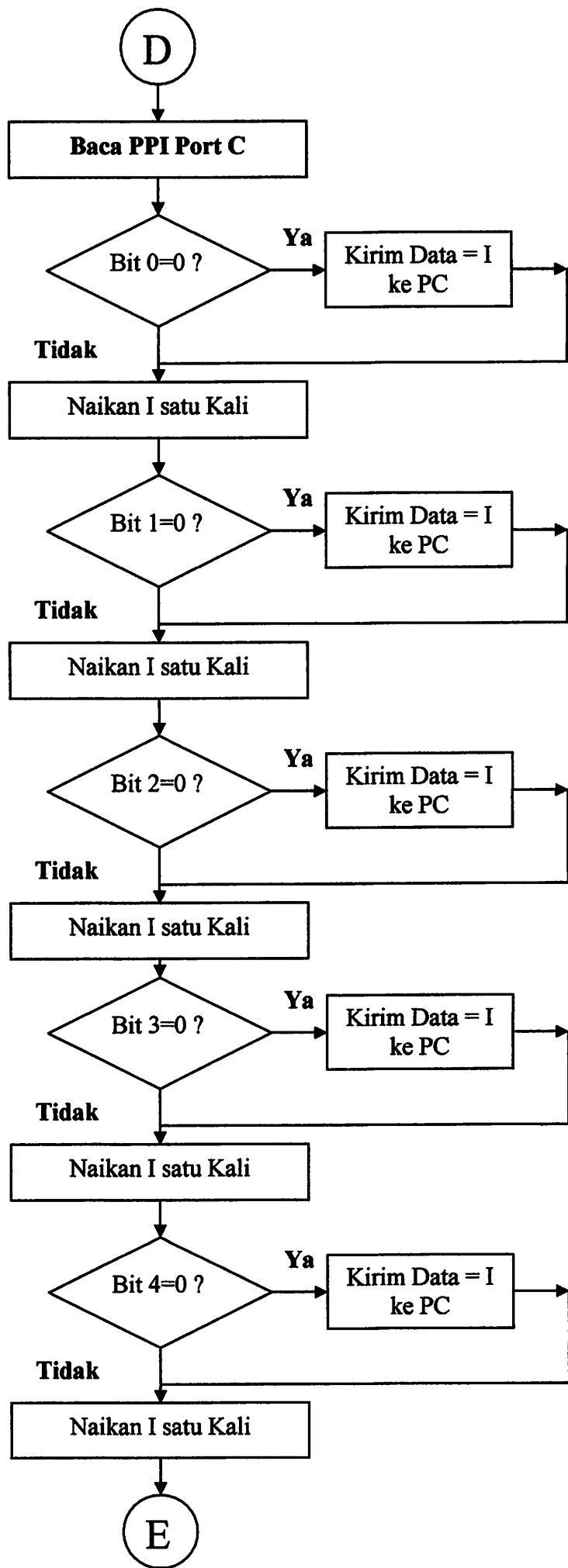


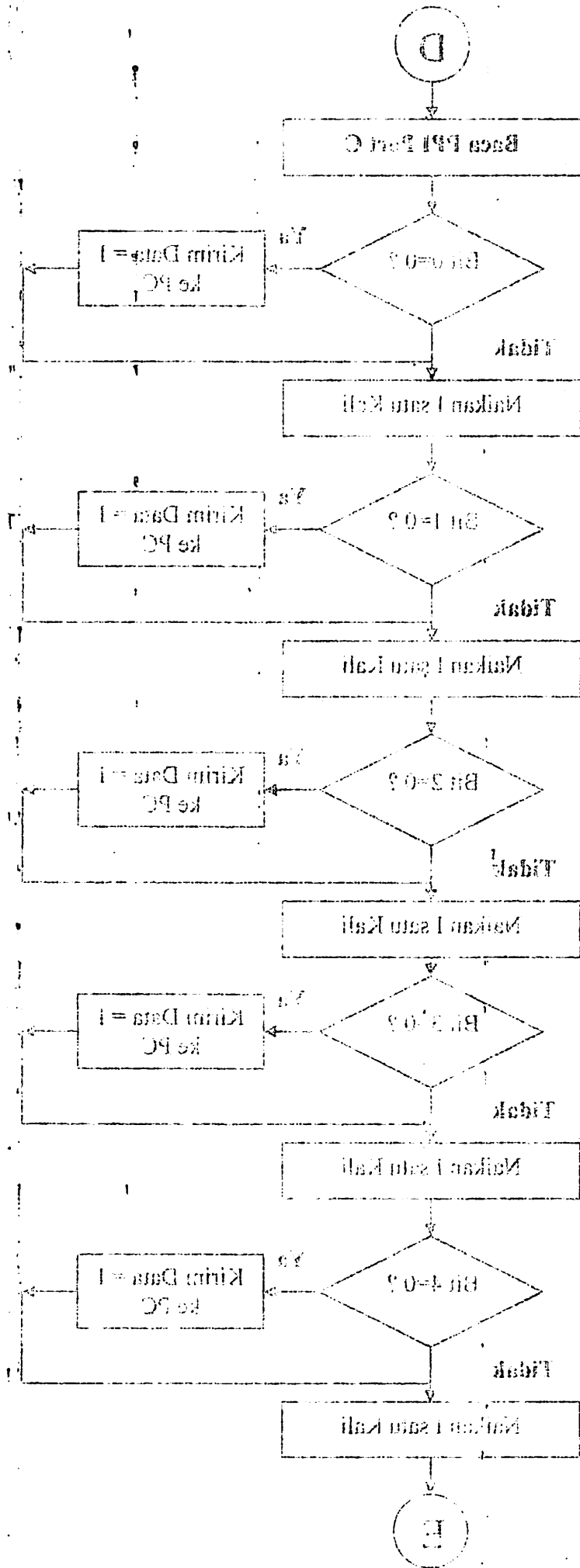


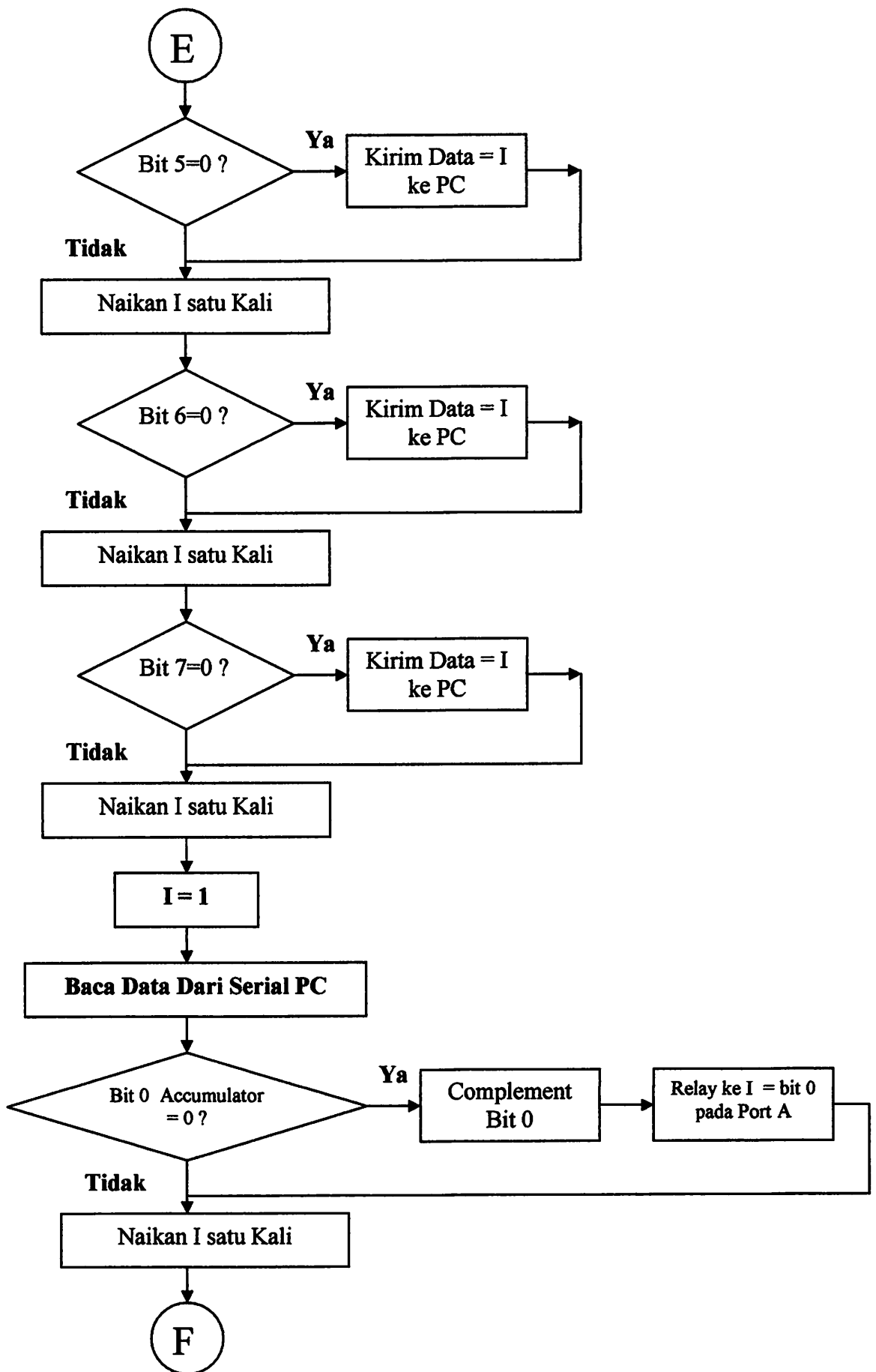


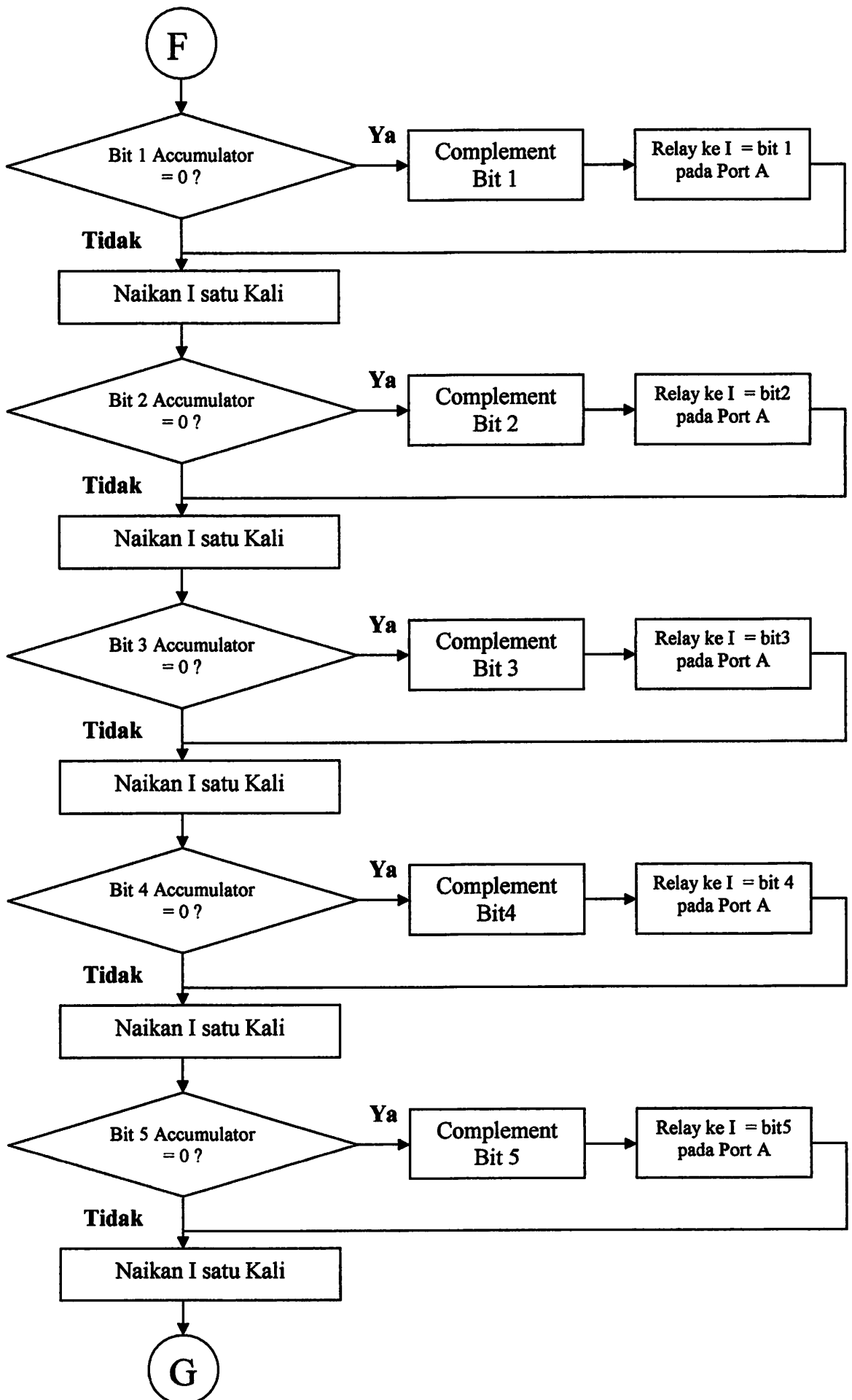




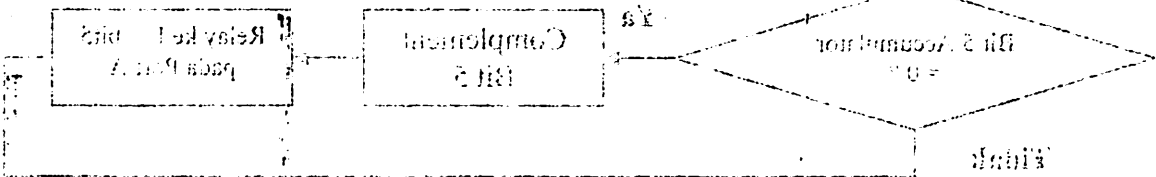
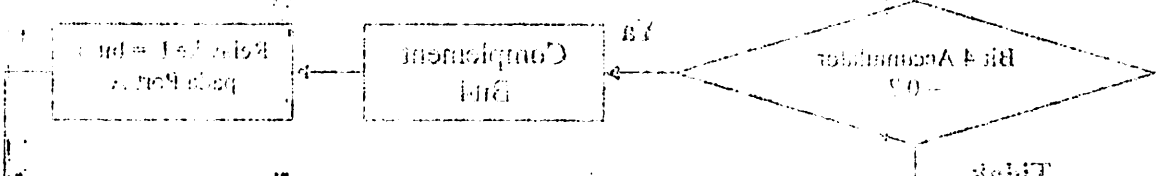
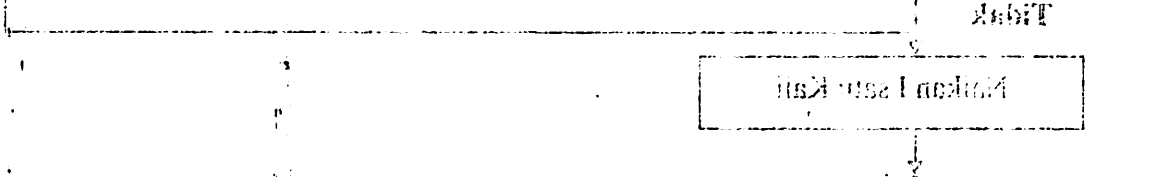
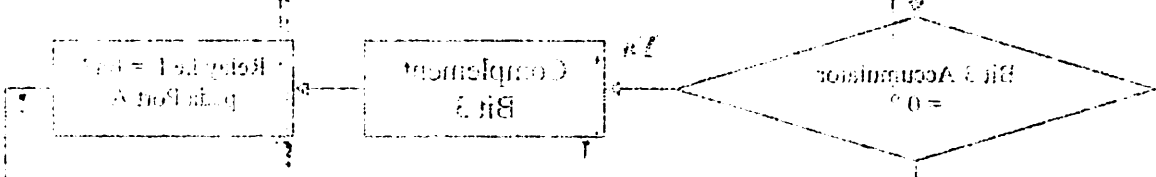
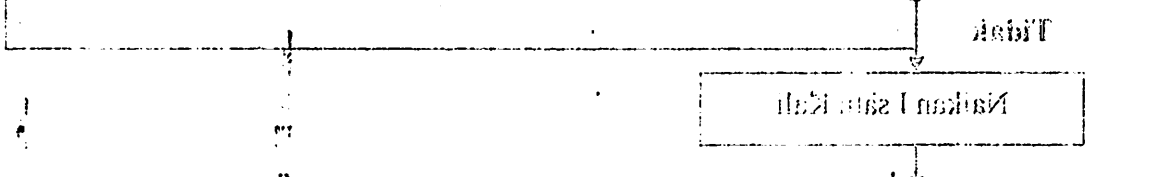
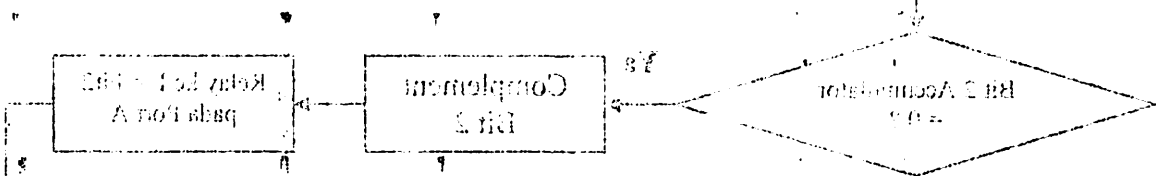
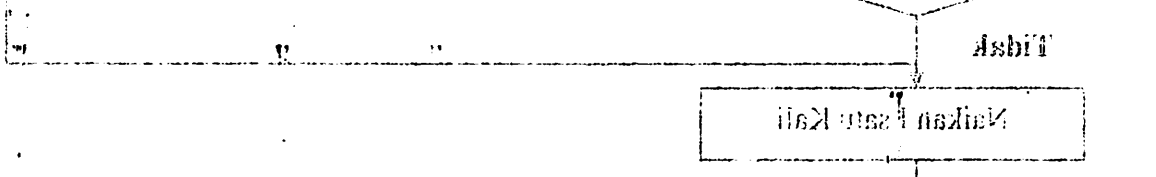
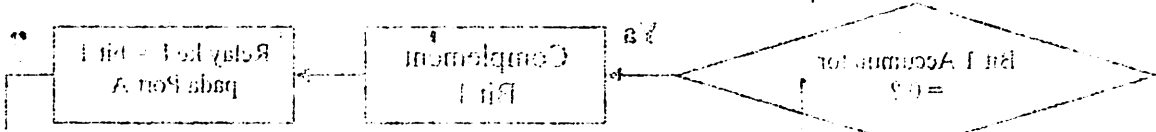




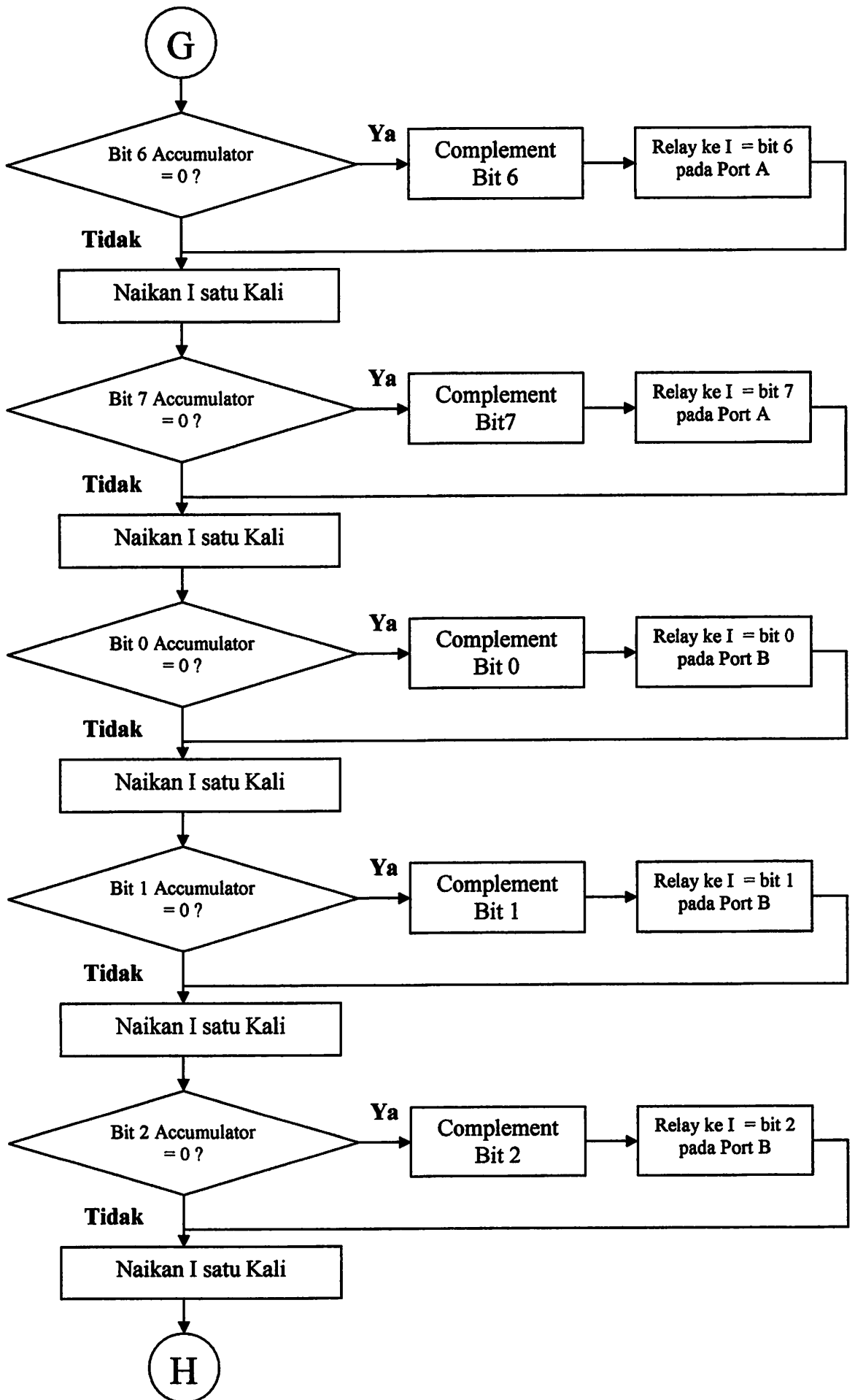


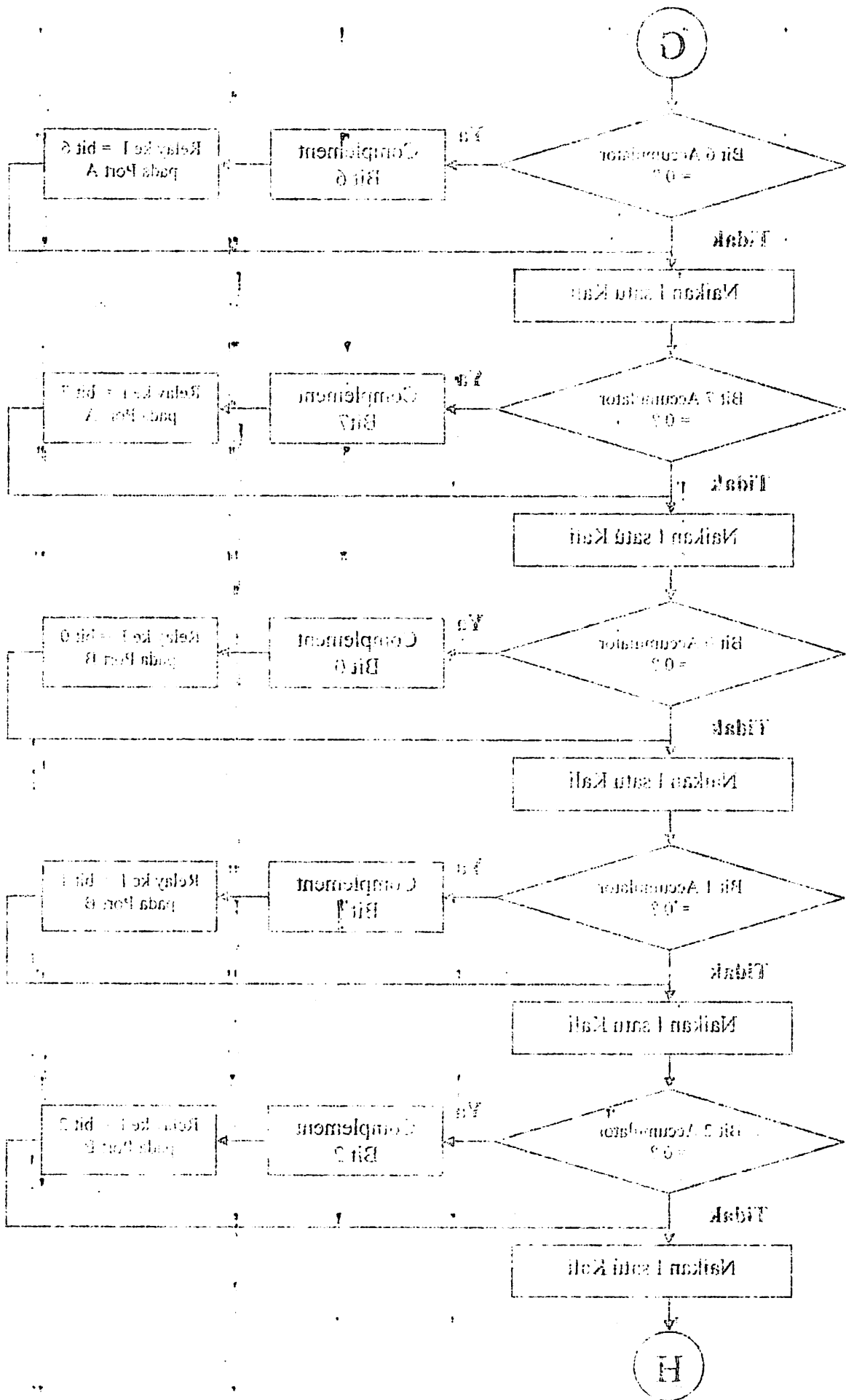


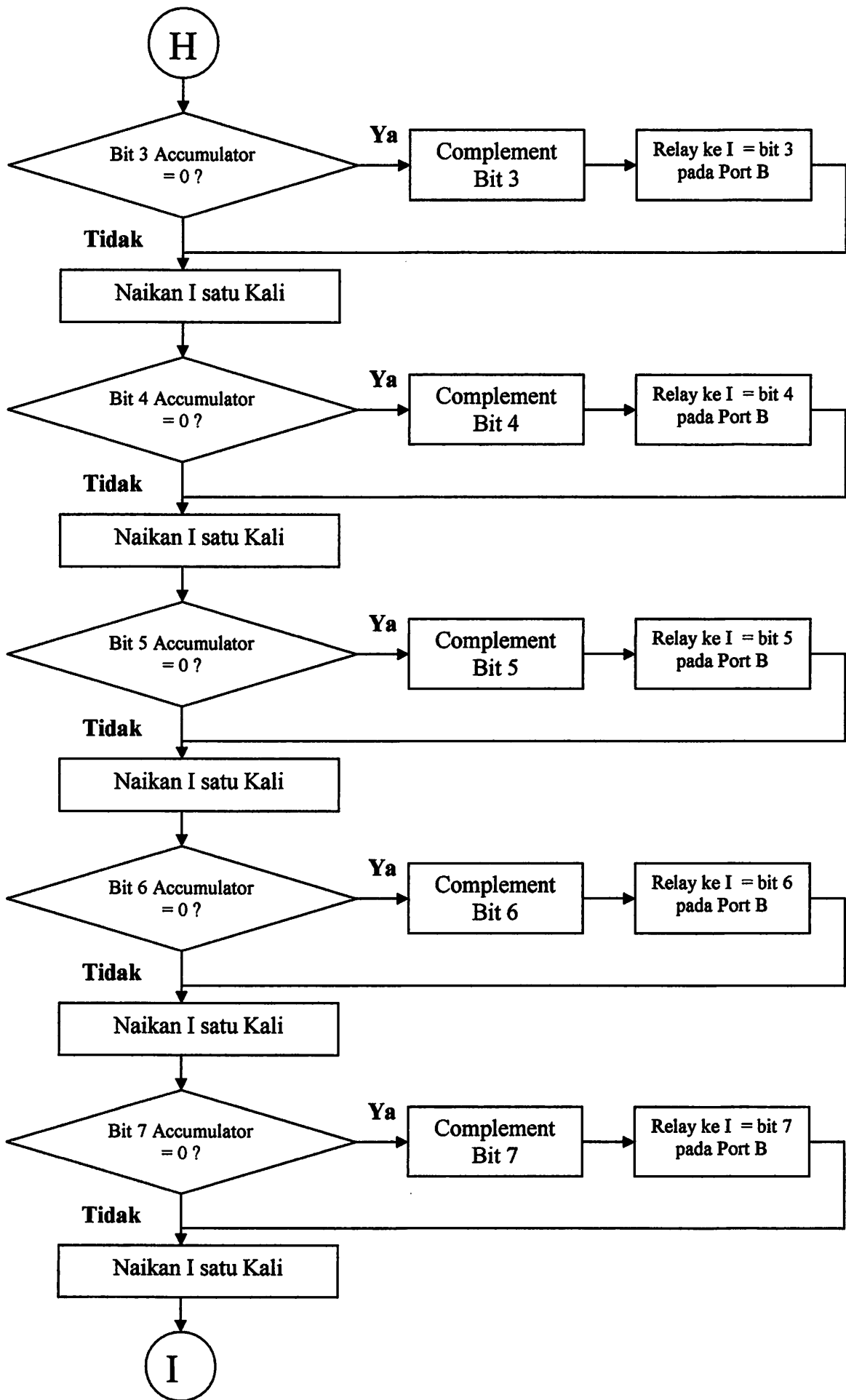
F

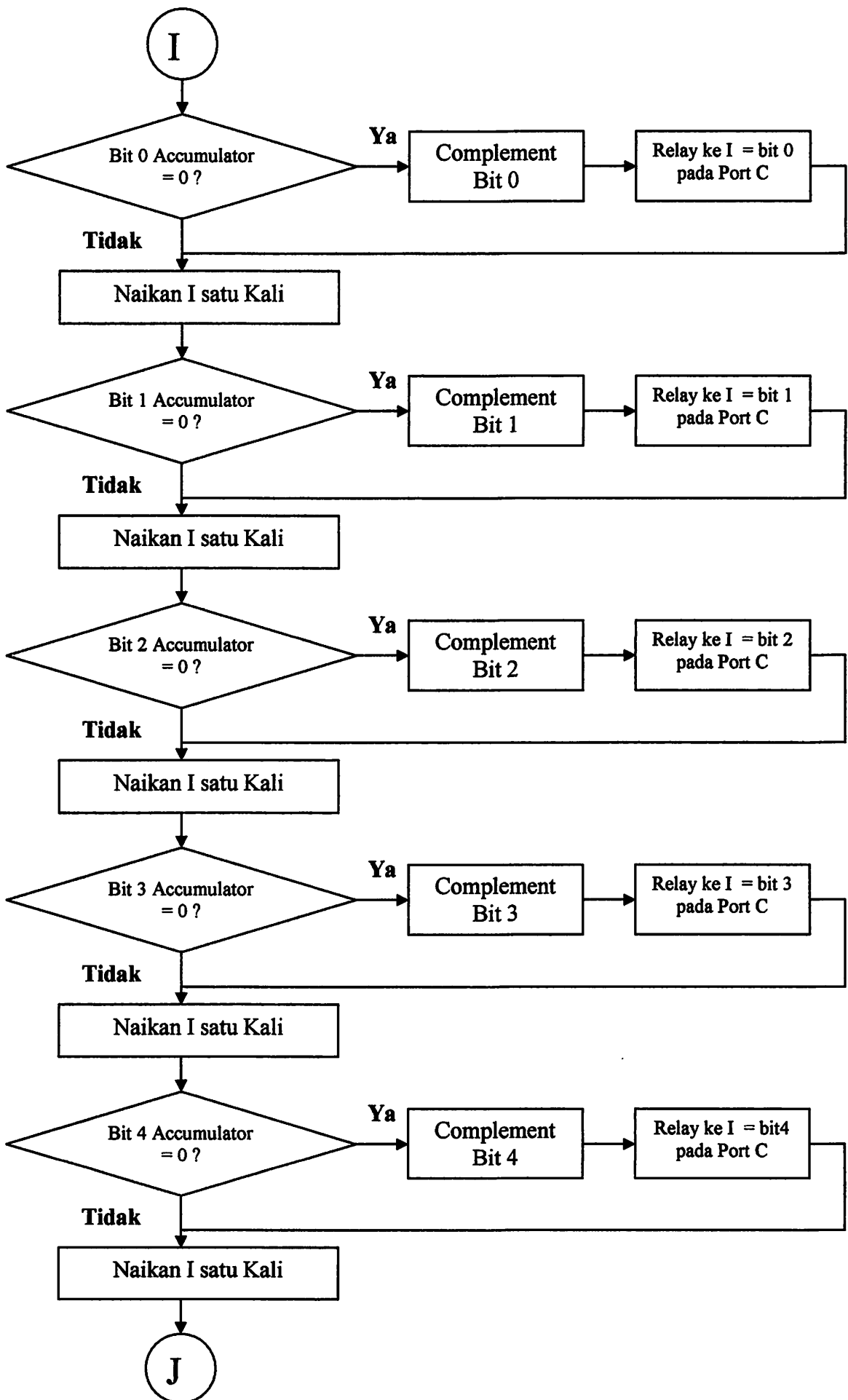


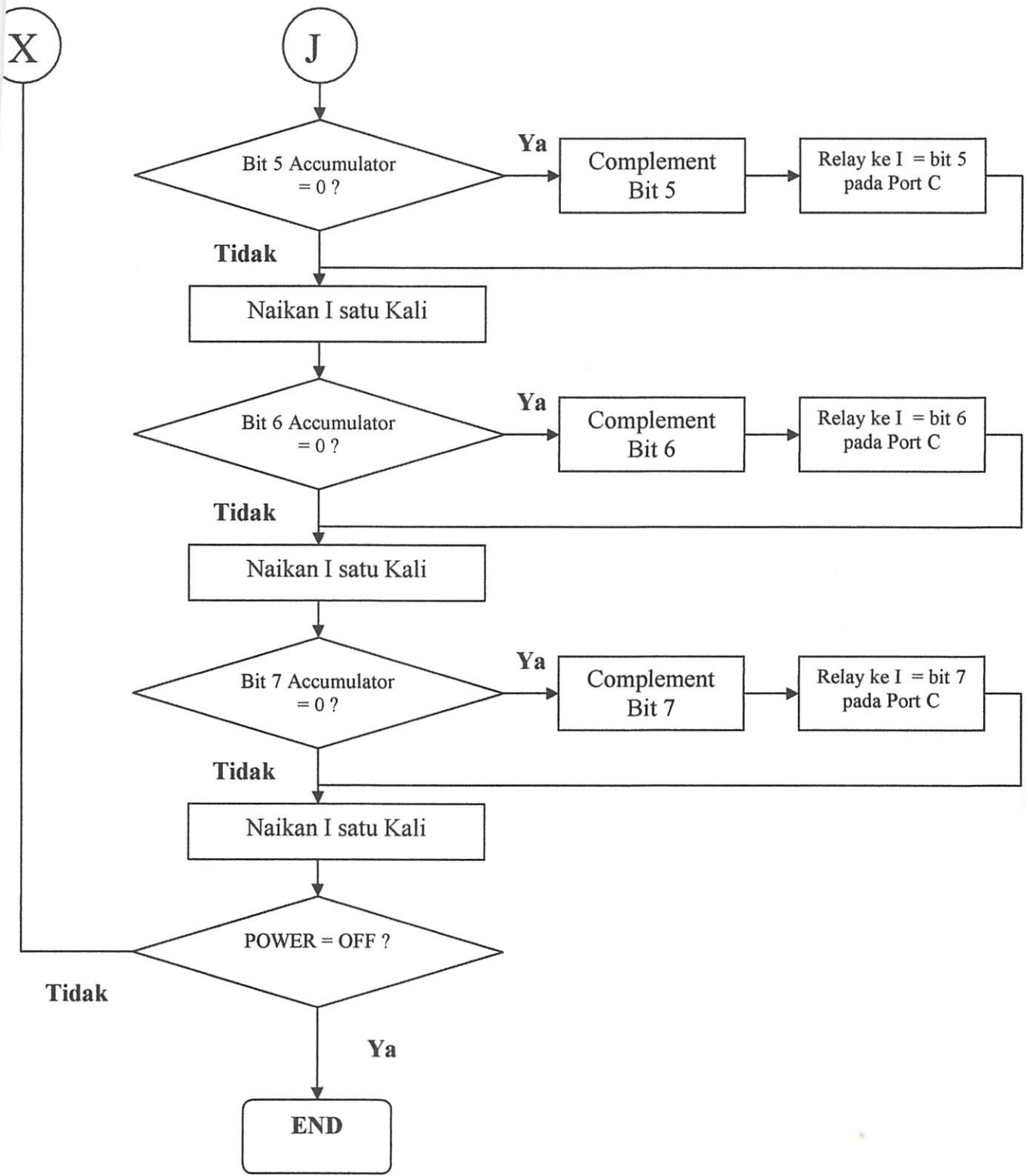
G

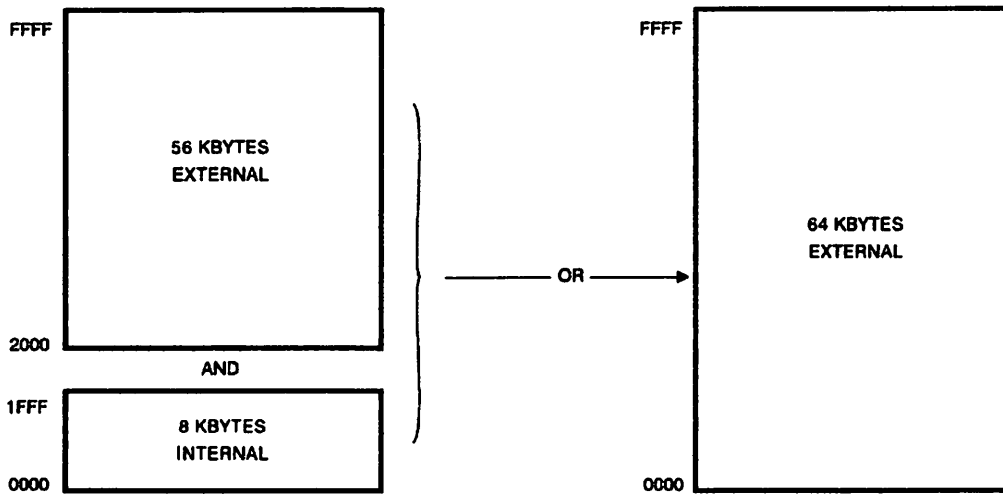




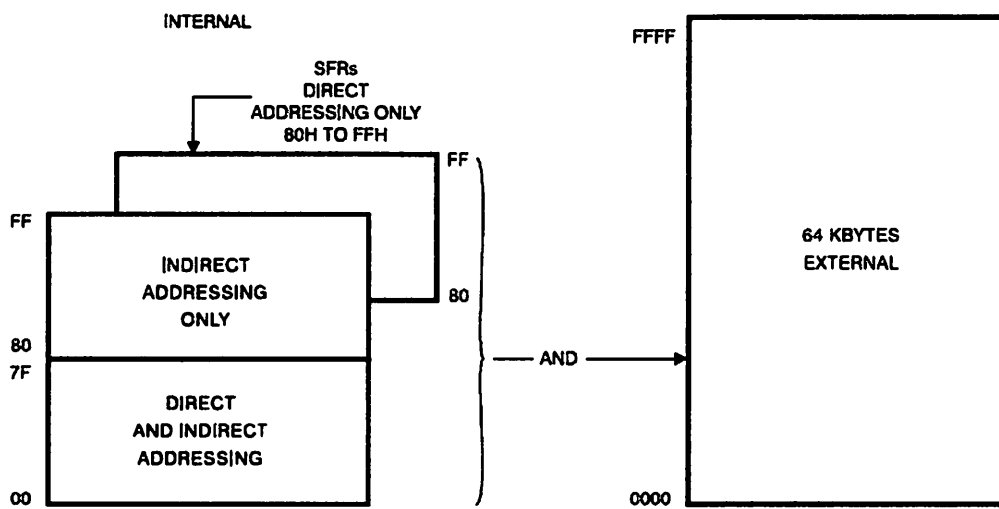




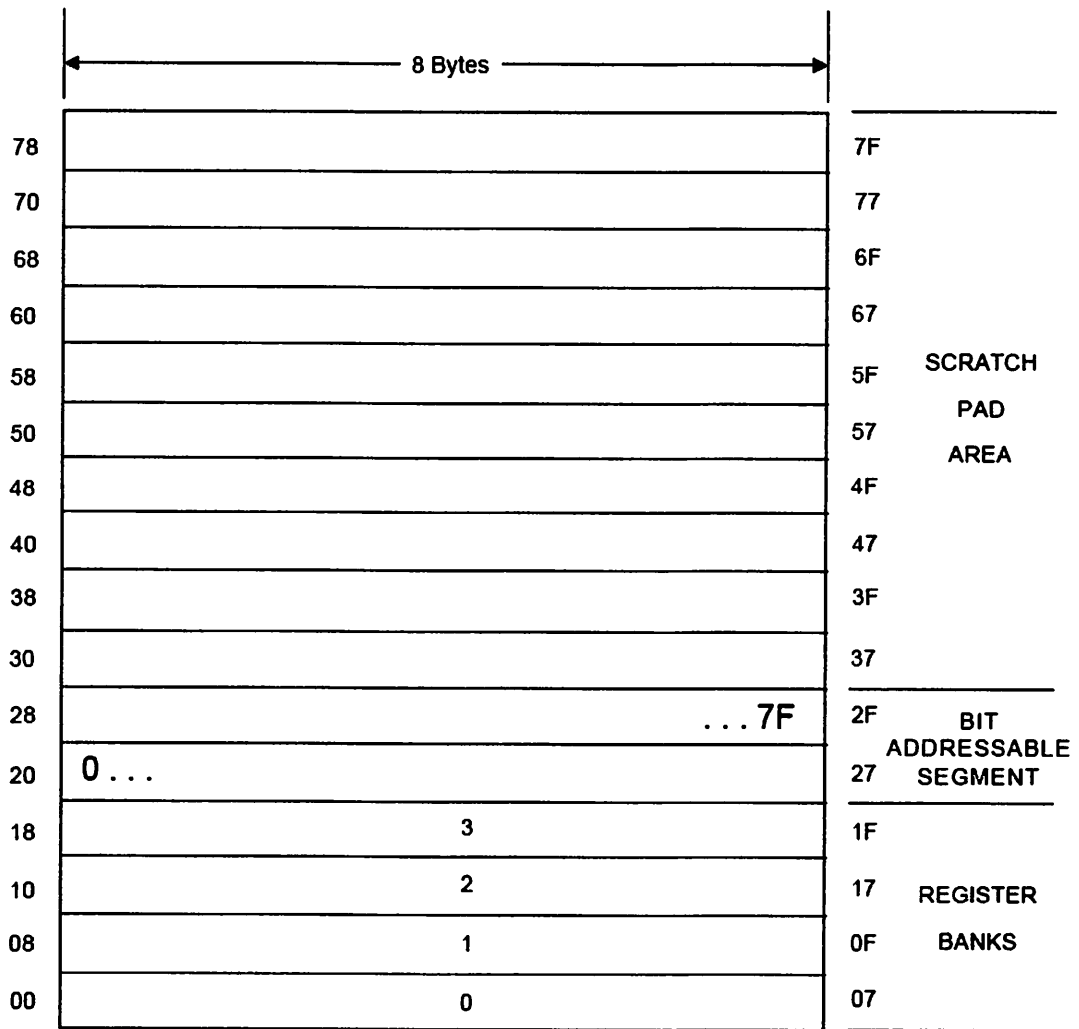




Gambar Memory program



Gambar Memory data MCU AT89S52



Mapping Area RAM Area MCU AT89S52

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H

Memory map untuk Special function register (SFR)

Table 1. Special Function Registers

Symbol	Name	Address
ACC ⁽¹⁾	Accumulator	0E0H
B ⁽¹⁾	B Register	0F0H
PSW ⁽¹⁾	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
P0 ⁽¹⁾	Port 0	80H
P1 ⁽¹⁾	Port 1	90H
P2 ⁽¹⁾	Port 2	0A0H
P3 ⁽¹⁾	Port 3	0B0H
IP ⁽¹⁾	Interrupt Priority Control	0B8H
IE ⁽¹⁾	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON ⁽¹⁾	Timer/Counter Control	88H
T2CON ⁽¹⁾⁽²⁾	Timer/Counter 2 Control	0C8H
T2MOD ⁽²⁾	Timer/Counter 2 Mode Control	0C9H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
TH2 ⁽²⁾	Timer/Counter 2 High Byte	0CDH
TL2 ⁽²⁾	Timer/Counter 2 Low Byte	0CCH
RCAP2H ⁽²⁾	T/C 2 Capture Reg. High Byte	0CBH
RCAP2L ⁽²⁾	T/C 2 Capture Reg. Low Byte	0CAH
SCON ⁽¹⁾	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

Notes: 1. Bit addressable
2. AT89C52 only

SOURCE CODE APLIKASI

```
{SR *.dfm}
procedure TForm1.Delay(Lama : Longint);
var
  ref : Longint;
begin
  ref := GetTickCount;
  repeat
    Application.ProcessMessages;
  Until ((GetTickCount - Ref)>= Lama) ;
end;

procedure TForm1.open1Click(Sender: TObject);
begin
  if opendialog1.Execute then
  begin
    WindowsMediaPlayer1.URL := opendialog1.filename;
  end;
end;
procedure TForm1.FormCreate(Sender: TObject);
var dir: String;
begin
  SXSkinButton110.Click;
  Button1.Click;
  SXSkinLibrary5.StoredSkin := GetStoredSkinByIndex(0);
  SXSkinLibrary5.Active := true;
  dir := GetCurrentDir +'/skin/CO_CO';
  With SXSkinLibrary2 DO
  Begin
    SkinDir:= dir;
    SkinFile:='skin.sxs';
    SkinFile2:='skin.sxs';
    Active:=True;
  begin
  dir := GetCurrentDir +'/skin/opera';
  With SXSkinLibrary3 DO
  Begin
    SkinDir:= dir;
    SkinFile:='skin.sxs';
    SkinFile2:='skin.sxs';
    Active:=True;
  begin
  dir := GetCurrentDir +'/skin/Wisp_WISPBLUE';
  With SXSkinLibrary1 DO
  Begin
    SkinDir:= dir;
    SkinFile:='skin.sxs';
    SkinFile2:='skin.sxs';
    Active:=True;
  begin
  dir := GetCurrentDir +'/skin/posh';
  With SXSkinLibrary4 DO
  Begin
    SkinDir:= dir;
    SkinFile:='skin.sxs';
    SkinFile2:='skin.sxs';
    Active:=True;
  end;
end;end;end;end;end;end;end;

procedure TForm1.SXSkinButton1Click(Sender: TObject); // 1
begin
Image10.Visible := false;
Vacomm1.WriteChar(char($5B));
SXSkinButton1.Visible := false;
SXSkinButton2.Visible := true;
```

```

end;

procedure TForm1.SXSkinButton2Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($13));
SXSkinButton1.Visible := true;
end;

procedure TForm1.SXSkinButton3Click(Sender: TObject); //2
begin
Image12.Visible := false;
Vacomm1.WriteChar(char($5A));
SXSkinButton3.Visible := false;
SXSkinButton4.Visible := true;

end;

procedure TForm1.SXSkinButton4Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($12));
SXSkinButton3.Visible := true;
end;

procedure TForm1.SXSkinButton5Click(Sender: TObject); //3
begin
Image13.Visible := false;
Vacomm1.WriteChar(char($59));
SXSkinButton5.Visible := false;
SXSkinButton6.Visible := true;

end;

procedure TForm1.SXSkinButton6Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($11));
SXSkinButton5.Visible := true;
end;

procedure TForm1.SXSkinButton7Click(Sender: TObject); //4
begin
Image14.Visible := false;
Vacomm1.WriteChar(char($50));
SXSkinButton7.Visible := false;
SXSkinButton8.Visible := true;

end;

procedure TForm1.SXSkinButton8Click(Sender: TObject);
begin
SXSkinButton7.Visible := true;
Vacomm1.WriteChar(char($08));
end;

procedure TForm1.SXSkinButton9Click(Sender: TObject); //5
begin
Vacomm1.WriteChar(char($03));
SXSkinButton9.Visible := false;
SXSkinButton10.Visible := true;

end;

procedure TForm1.SXSkinButton10Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($4B));
SXSkinButton9.Visible := true;

```

```

end;

procedure TForm1.SXSkinButton11Click(Sender: TObject); //6
begin
Vacomm1.WriteChar(char($02));
SXSkinButton11.Visible := false;
SXSkinButton12.Visible := true;
panel15.Visible := false;

end;

procedure TForm1.SXSkinButton12Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($4A));
SXSkinButton11.Visible := true;
end;

procedure TForm1.SXSkinButton13Click(Sender: TObject); //7
begin
Vacomm1.WriteChar(char($01));
SXSkinButton13.Visible := false;
SXSkinButton14.Visible := true;
panel16.Visible := false;

end;

procedure TForm1.SXSkinButton14Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($49));
SXSkinButton13.Visible := true;
end;

procedure TForm1.SXSkinButton15Click(Sender: TObject); //8
begin
Vacomm1.WriteChar(char($00));
SXSkinButton15.Visible := false;
SXSkinButton16.Visible := true;
panel17.Visible := false;

end;

procedure TForm1.SXSkinButton16Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($48));
SXSkinButton15.Visible := true;
end;

.
. SAMA DENGAN DIATAS
.

procedure TForm1.SXSkinButton94Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($69));
SXSkinButton93.Visible := true;
end;

procedure TForm1.SXSkinButton95Click(Sender: TObject); //48
begin
Vacomm1.WriteChar(char($20));
SXSkinButton95.Visible := false;
SXSkinButton96.Visible := true;
panel57.Visible := false;

end;

procedure TForm1.SXSkinButton96Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($68));
SXSkinButton95.Visible := true;
end;

```

```
procedure TForm1.SXSkinButton107Click(Sender: TObject);
begin
Port2.Click;
SXSkinCheckBox1.Checked := false;
CLOSE;
end;
```

```
procedure TForm1.SXSkinButton109Click(Sender: TObject);
begin
if opendialog1.Execute then
begin
WindowsMediaPlayer1.URL := opendialog1.filename;
end;
end;
procedure TForm1.SXSkinButton97Click(Sender: TObject);
begin
SXSkinButton118.Visible := false;
SXSkinButton99.Visible := true;
```

```
Vacomm1.WriteChar(char($09));
end;
```

```
procedure TForm1.SXSkinButton99Click(Sender: TObject);
begin
SXSkinButton99.Visible := false;
SXSkinButton118.Visible := true;
Vacomm1.WriteChar(char($51));
end;
```

```
procedure TForm1.SXSkinButton118Click(Sender: TObject);
begin
SXSkinButton118.Visible := false;
SXSkinButton99.Visible := true;
Vacomm1.WriteChar(char($09));
end;
```

```
procedure TForm1.SXSkinButton100Click(Sender: TObject);
begin
SXSkinButton118.Visible := false;
SXSkinButton99.Visible := true;
end;
```

```
procedure TForm1.SXSkinButton101Click(Sender: TObject);
begin
SXSkinButton101.Visible := false;
SXSkinButton102.Visible := FALSE;
SXSkinButton103.Visible := FALSE;
Vacomm1.WriteChar(char($56));
Vacomm1.WriteChar(char($57));
//SXSkinButton123.Click;
end;
```

```
procedure TForm1.SXSkinButton102Click(Sender: TObject);
begin
IF SXSkinButton103.Visible = true THEN
begin
SXSkinButton101.Visible := true;
SXSkinButton102.Visible := False;
SXSkinButton103.Visible := true;
Vacomm1.WriteChar(char($56));
//SXSkinButton123.Click;
end
else
IF SXSkinButton103.Visible = False THEN
begin
SXSkinButton101.Visible := true;
SXSkinButton102.Visible := False;
SXSkinButton103.Visible := true;
Vacomm1.WriteChar(char($56));
```

```
Vacomm1.WriteChar(char($0F));  
end  
end;
```

```
procedure TForm1.SXSkinButton103Click(Sender: TObject);  
begin  
IF SXSkinButton102.Visible = true THEN  
begin  
SXSkinButton103.Visible := false;  
SXSkinButton102.Visible := true;  
SXSkinButton101.Visible := true;  
Vacomm1.WriteChar(char($57));  
//SXSkinButton123.Click;  
END  
ELSE  
IF SXSkinButton102.Visible = FALSE THEN  
begin  
SXSkinButton103.Visible := false;  
SXSkinButton102.Visible := true;  
SXSkinButton101.Visible := true;  
Vacomm1.WriteChar(char($57));  
Vacomm1.WriteChar(char($0E));  
END  
end;
```

```
procedure TForm1.SXSkinButton104Click(Sender: TObject);  
begin  
SXSkinButton104.Visible := false;  
SXSkinButton123.Visible := true;  
Vacomm1.WriteChar(char($55));  
end;
```

```
procedure TForm1.SXSkinButton123Click(Sender: TObject);  
begin  
SXSkinButton123.Visible := false;  
SXSkinButton104.Visible := true;  
Vacomm1.WriteChar(char($0D));  
end;
```

```
procedure TForm1.SXSkinButton120Click(Sender: TObject);  
begin  
SXSkinButton101.Visible := true;  
SXSkinButton102.Visible := true;  
SXSkinButton103.Visible := true;  
Vacomm1.WriteChar(char($0E));  
Vacomm1.WriteChar(char($0F));  
end;
```

```
procedure TForm1.SXSkinButton121Click(Sender: TObject);  
begin  
IF SXSkinButton101.Visible = False THEN  
begin  
SXSkinButton102.Visible := False;  
SXSkinButton101.Visible := true;  
SXSkinButton103.Visible := true;  
Vacomm1.WriteChar(char($0f));  
end  
else  
IF SXSkinButton101.Visible = true THEN  
begin  
SXSkinButton102.Visible := true;  
SXSkinButton101.Visible := true;  
SXSkinButton103.Visible := true;  
Vacomm1.WriteChar(char($0E));  
end end;
```

```
procedure TForm1.SXSkinButton122Click(Sender: TObject);  
begin  
IF SXSkinButton101.Visible = False THEN  
begin
```

```

SXSkinButton103.Visible := False;
SXSkinButton102.Visible := true;
SXSkinButton101.Visible := true;
Vacomm1.WriteChar(char($0e));
end
else
IF SXSkinButton101.Visible = true THEN
begin
SXSkinButton103.Visible := true;
SXSkinButton102.Visible := true;
SXSkinButton101.Visible := true;
Vacomm1.WriteChar(char($0f));
end end;

procedure TForm1.SXSkinButton105Click(Sender: TObject);
begin
SXSkinButton105.Visible := false;
SXSkinButton124.Visible := true;
Vacomm1.WriteChar(char($54));
end;

procedure TForm1.SXSkinButton124Click(Sender: TObject);
begin
SXSkinButton124.Visible := false;
SXSkinButton105.Visible := true;
Vacomm1.WriteChar(char($0C));

end;

procedure TForm1.SXSkinButton106Click(Sender: TObject);
begin
SXSkinButton124.Visible := false;
SXSkinButton105.Visible := true;
Vacomm1.WriteChar(char($54));
Vacomm1.WriteChar(char($0B));
end;

procedure TForm1.SXSkinButton110Click(Sender: TObject); // RESET
begin
SXSkinButton1.Visible := true;
SXSkinButton3.Visible := true;
SXSkinButton5.Visible := true;
SXSkinButton7.Visible := true;
SXSkinButton9.Visible := true;
SXSkinButton11.Visible := true;
SXSkinButton13.Visible := true;
SXSkinButton15.Visible := true;
SXSkinButton17.Visible := true;
SXSkinButton19.Visible := true;
SXSkinButton21.Visible := true;
SXSkinButton23.Visible := true;
SXSkinButton25.Visible := true;
SXSkinButton27.Visible := true;
SXSkinButton29.Visible := true;
SXSkinButton31.Visible := true;
SXSkinButton33.Visible := true;
SXSkinButton35.Visible := true;
SXSkinButton37.Visible := true;
SXSkinButton39.Visible := true;
SXSkinButton41.Visible := true;
SXSkinButton43.Visible := true;
SXSkinButton45.Visible := true;
SXSkinButton47.Visible := true;
SXSkinButton49.Visible := true;
SXSkinButton51.Visible := true;
SXSkinButton53.Visible := true;
SXSkinButton55.Visible := true;
SXSkinButton57.Visible := true;
SXSkinButton59.Visible := true;
SXSkinButton61.Visible := true;

```

```

SXSkinButton63.Visible := true;
SXSkinButton65.Visible := true;
SXSkinButton67.Visible := true;
SXSkinButton69.Visible := true;
SXSkinButton71.Visible := true;
SXSkinButton73.Visible := true;
SXSkinButton75.Visible := true;
SXSkinButton77.Visible := true;
SXSkinButton79.Visible := true;
SXSkinButton81.Visible := true;
SXSkinButton83.Visible := true;
SXSkinButton85.Visible := true;
SXSkinButton87.Visible := true;
SXSkinButton89.Visible := true;
SXSkinButton91.Visible := true;
SXSkinButton93.Visible := true;
SXSkinButton95.Visible := true;
Vacomm1.WriteChar(char($10));
Vacomm1.WriteChar(char($11));
Vacomm1.WriteChar(char($12));
Vacomm1.WriteChar(char($13));
Vacomm1.WriteChar(char($0E));
Vacomm1.WriteChar(char($0F));
Vacomm1.WriteChar(char($0D));
Vacomm1.WriteChar(char($0A));
Vacomm1.WriteChar(char($09));
Vacomm1.WriteChar(char($08));
Vacomm1.WriteChar(char($0F));
Vacomm1.WriteChar(char($0E));
Vacomm1.WriteChar(char($0D));
Vacomm1.WriteChar(char($0C));
Vacomm1.WriteChar(char($0B));
end;

procedure TForm1.Timer1Timer(Sender: TObject);
VAR
    jam_menit_detik_milidetik : word;
begin
    decodetime(time,jam_menit_detik_milidetik);
    Label2.Caption := inttostr(jam) + ':' +
        inttostr(menit) + ':' +
        inttostr(detik);

end;

procedure TForm1.SXSkinButton111Click(Sender: TObject);
begin
    winexec('C:\program files\windows media player\mplayer2.exe ',Sw_normal);
end;

procedure TForm1.Reset1Click(Sender: TObject);
begin
    close;
end;

procedure TForm1.Panel8Click(Sender: TObject);
begin
    application.MessageBox('Panel,Tombol untuk menyambung dan memutus komunikasi dengan siswa'
        , 'Panel', MB_OK );
end;

procedure TForm1.Source1Click(Sender: TObject);
begin
    application.MessageBox('Pilih Inputan Suara Dari Line in, VCD, Tape, PC(computer)'
        , 'Source', MB_OK );
end;

procedure TForm1.Program1Click(Sender: TObject);
begin

```



```
application.MessageBox('(A) untuk berinteraksi dengan panel No Genap, (B) untuk berinteraksi dengan panel No Ganjil, (ALL) Untuk semua Panel (A+B), dan (IND) untuk Pribadi/ Prifasi '
, 'Program', MB_OK );
```

```
end;
```

```
procedure TForm1.Sound1Click(Sender: TObject);
```

```
begin
```

```
application.MessageBox(' (ROOM) untuk menampilkan Suara Ruangan/ Spiker, (Headset) untuk alat bantu dengar/ headphone '
, 'Sound', MB_OK );
```

```
end;
```

```
procedure TForm1.MediaPlayer1Click(Sender: TObject);
```

```
begin
```

```
application.MessageBox('(Media Player) untuk memutar file video pada program Lab Bahasa '
, 'MediaPlayer', MB_OK );
```

```
end;
```

```
procedure TForm1.WindowMplayer1Click(Sender: TObject);
```

```
begin
```

```
application.MessageBox('Untuk memanggil aplikasi tambahan Media Player dari windows '
, 'WindowMplayer', MB_OK );
```

```
end;
```

```
procedure TForm1.About1Click(Sender: TObject);
```

```
begin
```

```
form2.show;
```

```
end;
```

```
procedure TForm1.VaComm1RxChar(Sender: TObject; Count: Integer); // COOOLL
```

```
var i:string;
```

```
n:integer;
```

```
begin
```

```
i:=vacomm1.ReadText; //1
```

```
if (i=#5) then
```

```
begin
```

```
for n:=1 to 5 do
```

```
begin
```

```
Image10.Visible := false;
```

```
Delay(10);
```

```
Image10.Visible := true;
```

```
PlaySound('ding.wav',0,SND_SYNC);
```

```
Delay(200);
```

```
end
```

```
end;
```

```
//2
```

```
if (i=#6) then
```

```
begin
```

```
for n:=1 to 5 do
```

```
begin
```

```
Image12.Visible := true;
```

```
Delay(20);
```

```
PlaySound('ding.wav',0,SND_SYNC);
```

```
Image12.Visible := false;
```

```
Delay(100);
```

```
end
```

```
end;
```

```
//3
```

```
if (i=#7) then
```

```
begin
```

```
for n:=1 to 5 do
```

```
begin
```

```
Image13.Visible := true;
```

```
Delay(20);
```

```
PlaySound('ding.wav',0,SND_SYNC);
Image13.Visible := false;
Delay(100);
end
end;
```

//4

```
if (i=#8) then
begin
```

```
for n:=1 to 5 do
begin
Image14.Visible := true;
```

```
Delay(20);
PlaySound('ding.wav',0,SND_SYNC);
Image14.Visible := false;
Delay(100);
end;
end;
end;
```

```
procedure TForm1.Action5Execute(Sender: TObject);
begin
application.MessageBox('Pilih Inputan Suara Dari Line in, VCD, Tape, PC(computer)'
, 'Source', MB_OK );
end;
```

```
procedure TForm1.Action6Execute(Sender: TObject);
begin
application.MessageBox('(A) untuk berinteraksi dengan panel No Genap, (B) untuk berinteraksi dengan panel No Ganjil, (ALL) Untuk
semua Panel (A+B), dan (IND) untuk Pribadi/ Prifasi '
, 'Program', MB_OK );
end;
```

```
procedure TForm1.Action7Execute(Sender: TObject);
begin
application.MessageBox(' (ROOM) untuk menampilkan Suara Ruangan/ Spiker, (Headset) untuk alat bantu dengar/ headphone'
, 'Sound', MB_OK );
end;
```

```
procedure TForm1.Action9Execute(Sender: TObject);
begin
application.MessageBox('(Open Filles) untuk membuka atau mngambil file video pada program Lab Bahasa '
, 'Open Filles', MB_OK );
end;
```

```
procedure TForm1.Action10Execute(Sender: TObject);
begin
application.MessageBox('(Reset) untuk merestart/mengulang program Lab Bahasa dari awal'
, 'Reset', MB_OK );
end;
```

```
procedure TForm1.Action11Execute(Sender: TObject);
begin
application.MessageBox('(Full Screen) untuk menampilkan fidio dalam format full screen'
, 'Full Screen', MB_OK );
end;
```

```
procedure TForm1.Action12Execute(Sender: TObject);
begin
application.MessageBox('(Media Player) untuk memutar file video pada program Lab Bahasa '
, 'MediaPlayer', MB_OK );
end;
```

```
procedure TForm1.Action8Execute(Sender: TObject);
begin
```

```

form2.show;
end;

procedure TForm1.Action1Execute(Sender: TObject);
begin
SXSkinButton109.Click;
end;

procedure TForm1.Action2Execute(Sender: TObject);
begin
SXSkinButton110.Click;
end;

procedure TForm1.Action4Execute(Sender: TObject);
begin
close;
end;

procedure TForm1.SXSkinButton114Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($55));
end;

procedure TForm1.SXSkinButton115Click(Sender: TObject);
begin
SXSkinButton115.Visible := false;
SXSkinButton113.Visible := true;
Vacomm1.WriteChar(char($53));
end;

procedure TForm1.SXSkinButton112Click(Sender: TObject);
begin
Vacomm1.WriteChar(char($0D));
end;

procedure TForm1.SXSkinButton113Click(Sender: TObject);
begin
SXSkinButton113.Visible := false;
SXSkinButton115.Visible := true;
Vacomm1.WriteChar(char($0B));
end;

procedure TForm1.SXSkinButton126Click(Sender: TObject);
begin
Form3.ShowModal;
end;

procedure TForm1.SXSkinCheckBox1Change(Sender: TObject);
begin
if SXSkinCheckBox1.Checked = false then
begin
SXSkinCheckBox1.Caption := 'Pakai Nama Absensi';
SXSkinButton1.Caption := '1';
SXSkinButton2.Caption := '1';
SXSkinButton3.Caption := '2';
SXSkinButton4.Caption := '2';
SXSkinButton5.Caption := '3';
SXSkinButton6.Caption := '3';
SXSkinButton7.Caption := '4';
SXSkinButton8.Caption := '4';
SXSkinButton9.Caption := '5';
SXSkinButton10.Caption := '5';
SXSkinButton11.Caption := '6';
SXSkinButton12.Caption := '6';
SXSkinButton13.Caption := '7';
SXSkinButton14.Caption := '7';
SXSkinButton15.Caption := '8';
SXSkinButton16.Caption := '8';
SXSkinButton17.Caption := '9';

```

```

SXSkinButton88.Caption := '44';
SXSkinButton89.Caption := '45';
SXSkinButton90.Caption := '45';
SXSkinButton91.Caption := '46';
SXSkinButton92.Caption := '46';
SXSkinButton93.Caption := '47';
SXSkinButton94.Caption := '47';
SXSkinButton95.Caption := '48';
SXSkinButton96.Caption := '48';
end
else
begin
SXSkinCheckBox1.Caption := 'Matikan Nama Absensi';
Form3.ADOQuery1.Locate('Nomor', '1',[lopartialkey]);
IF form3.DBGrid2.Fields[0].Value = '1' THEN           // tombol 1
begin
SXSkinButton1.Caption := Form3.DBGrid2.Fields[2].AsString;
SXSkinButton2.Caption := Form3.DBGrid2.Fields[2].AsString;
end
else
begin
SXSkinButton1.Caption := '--';
SXSkinButton2.Caption := '--';
end;
begin
Form3.ADOQuery1.Locate('Nomor', '2',[lopartialkey]);
IF form3.DBGrid2.Fields[0].Value = '2' THEN           // tombol 2
begin
SXSkinButton3.Caption := Form3.DBGrid2.Fields[2].AsString;
SXSkinButton4.Caption := Form3.DBGrid2.Fields[2].AsString;
end
else
begin
SXSkinButton3.Caption := '--';
SXSkinButton4.Caption := '--';
end;
begin
Form3.ADOQuery1.Locate('Nomor', '3',[lopartialkey]);
IF form3.DBGrid2.Fields[0].Value = '3' THEN           // tombol 3
begin
SXSkinButton5.Caption := Form3.DBGrid2.Fields[2].AsString;
SXSkinButton6.Caption := Form3.DBGrid2.Fields[2].AsString;
end
else
begin
SXSkinButton5.Caption := '--';
SXSkinButton6.Caption := '--';
end;
begin
Form3.ADOQuery1.Locate('Nomor', '4',[lopartialkey]);
IF form3.DBGrid2.Fields[0].Value = '4' THEN           // tombol 4
begin
SXSkinButton7.Caption := Form3.DBGrid2.Fields[2].AsString;
SXSkinButton8.Caption := Form3.DBGrid2.Fields[2].AsString;
end
else
begin
SXSkinButton7.Caption := '--';
SXSkinButton8.Caption := '--';
end;
begin
Form3.ADOQuery1.Locate('Nomor', '5',[lopartialkey]);
IF form3.DBGrid2.Fields[0].Value = '5' THEN           // tombol 5
begin
SXSkinButton9.Caption := Form3.DBGrid2.Fields[2].AsString;
SXSkinButton10.Caption := Form3.DBGrid2.Fields[2].AsString;
end
else
begin
SXSkinButton9.Caption := '--';

```


SOURCE CODE MIKROKONTROLER

ADDR0	BIT	P2.7	SETB	ES
ADDR1	BIT	P2.6	MOV	PPI1PA,#0FFH
ADDR2	BIT	P2.5	MOV	PPI1PB,#0FFH
			MOV	PPI1PC,#0FFH
BIT0	BIT	ACC.0	MOV	PPI2PA,#0FFH
BIT1	BIT	ACC.1	MOV	PPI2PB,#0FFH
BIT2	BIT	ACC.2	MOV	PPI2PC,#0FFH
BIT3	BIT	ACC.3		
BIT4	BIT	ACC.4	MOV	PPI3PA,#0FFH
BIT5	BIT	ACC.5	MOV	PPI3PB,#0FFH
BIT6	BIT	ACC.6	MOV	PPI3PC,#0FFH
BIT7	BIT	ACC.7		
TOMBOL1	BIT	20H.0	CALL	DELAY
TOMBOL2	BIT	20H.1	CALL	DELAY
TOMBOL3	BIT	20H.2		
TOMBOL4	BIT	20H.3	WAIT:	MOV R5,#5
TOMBOL5	BIT	20H.4	CALL	ENABLE_PPI_1
TOMBOL6	BIT	20H.5	MOV	A,#INISIAL_OUTPUT
TOMBOL7	BIT	20H.6	MOV	R0,#3
TOMBOL8	BIT	20H.7	MOVX	@R0,A
LAMPU	BIT	21H.7	CALL	ENABLE_PPI_2
INISIAL_OUTPUT	EQU	10000000B	MOV	A,#INISIAL_OUTPUT
INISIAL_INPUT	EQU	10011011B	MOV	R0,#3
			MOVX	@R0,A
PPI1PA	EQU	30H	CALL	ENABLE_PPI_3
PPI1PB	EQU	31H	MOV	A,#INISIAL_OUTPUT
PPI1PC	EQU	32H	MOV	R0,#3
			MOVX	@R0,A
PPI2PA	EQU	33H	CALL	ENABLE_PPI_4
PPI2PB	EQU	34H	MOV	A,#INISIAL_INPUT
PPI2PC	EQU	35H	MOV	R0,#3
			MOVX	@R0,A
PPI3PA	EQU	36H	CALL	ENABLE_PPI_5
PPI3PB	EQU	37H	MOV	A,#INISIAL_INPUT
PPI3PC	EQU	38H	MOV	R0,#3
			MOVX	@R0,A
PPI4PA	EQU	39H	CALL	ENABLE_PPI_6
PPI4PB	EQU	3AH	MOV	A,#INISIAL_INPUT
PPI4PC	EQU	3BH	MOV	R0,#3
			MOVX	@R0,A
PPI5PA	EQU	3CH	DJNZ	R5,WAIT
PPI5PB	EQU	3DH	CALL	DELAY
PPI5PC	EQU	3EH	CALL	DELAY
PPI6PA	EQU	3FH		
PPI6PB	EQU	40H	CALL	ENABLE_PPI_1
PPI6PC	EQU	41H	MOV	A,#0FFH
TEMP	EQU	42H	CALL	TULIS_PORT_A
DPH1	EQU	43H	CALL	TULIS_PORT_B
DPL1	EQU	44H	CALL	TULIS_PORT_C
DATA_LAMPU	EQU	45H	CALL	TULIS_PORT_C
ORG	00H		CALL	ENABLE_PPI_2
LJMP	START		MOV	A,#0FFH
			CALL	TULIS_PORT_A
ORG	23H		CALL	TULIS_PORT_B
LJMP	SERIAL		CALL	TULIS_PORT_C
ORG	100H		CALL	ENABLE_PPI_3
START:SETB	EA		MOV	A,#0FFH
MOV	TMOD,#20H		CALL	TULIS_PORT_A
MOV	TH1,#0FDH		CALL	TULIS_PORT_B
;BAUD RATE 9600 BPS			CALL	TULIS_PORT_C
SETB	TR1		SETB	LAMPU
MOV	SCON,#50H			

```

#####
#####
; INPUT PORT A PPI 4
#####
#####

```

```

CEK_TOMBOL:
    JB     LAMPU,NOL
    MOV    A,DATA_LAMPU
    CALL   AKSES
    SETB   LAMPU
NOL:     CALL  ENABLE_PPI_4
        CALL  BACA_PORT_A
        CJNE  A,PPI4PA,SCAN1
        JMP   CEK_TOMBOL2
SCAN1:   MOV   PPI4PA,A
        MOV   20H,A
TBL1:    JB   TOMBOL1,TBL2
        MOV   A,#1
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL2:    JB   TOMBOL2,TBL3
        MOV   A,#2
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL3:    JB   TOMBOL3,TBL4
        MOV   A,#3
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL4:    JB   TOMBOL4,TBL5
        MOV   A,#4
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL5:    JB   TOMBOL5,TBL6
        MOV   A,#5
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL6:    JB   TOMBOL6,TBL7
        MOV   A,#6
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL7:    JB   TOMBOL7,TBL8
        MOV   A,#7
        CALL  SEND
        JMP   CEK_TOMBOL2
TBL8:    JB   TOMBOL8,CEK_TOMBOL2
        MOV   A,#8
        CALL  SEND

```

```

#####
#####
; INPUT PORT B PPI 4
#####
#####

```

```

CEK_TOMBOL2:
    CALL   ENABLE_PPI_4
    CALL   BACA_PORT_B
    CJNE  A,PPI4PB,SCAN2
    JMP   CEK_TOMBOL3
SCAN2:   MOV   PPI4PB,A
        MOV   20H,A
TBL1B:   JB   TOMBOL1,TBL2B
        MOV   A,#9
        CALL  SEND
        JMP   CEK_TOMBOL3

```

```

TBL2B:   JB   TOMBOL2,TBL3B
        MOV   A,#10
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL3B:   JB   TOMBOL3,TBL4B
        MOV   A,#11
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL4B:   JB   TOMBOL4,TBL5B
        MOV   A,#12
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL5B:   JB   TOMBOL5,TBL6B
        MOV   A,#13
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL6B:   JB   TOMBOL6,TBL7B
        MOV   A,#14
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL7B:   JB   TOMBOL7,TBL8B
        MOV   A,#15
        CALL  SEND
        JMP   CEK_TOMBOL3
TBL8B:   JB   TOMBOL8,TBL9B
        MOV   A,#16
        CALL  SEND
TBL9B:   JMP   CEK_TOMBOL3

```

```

#####
#####
; INPUT PORT C PPI 4
#####
#####

```

```

CEK_TOMBOL3:
    CALL   ENABLE_PPI_4
    CALL   BACA_PORT_C
    CJNE  A,PPI4PC,SCAN3
    JMP   CEK_TOMBOL4
SCAN3:   MOV   PPI4PC,A
        MOV   20H,A
TBL1C:   JB   TOMBOL1,TBL2C
        MOV   A,#17
        CALL  SEND
        JMP   CEK_TOMBOL4
TBL2C:   JB   TOMBOL2,TBL3C
        MOV   A,#18
        CALL  SEND
        JMP   CEK_TOMBOL4
TBL3C:   JB   TOMBOL3,TBL4C
        MOV   A,#19
        CALL  SEND
        JMP   CEK_TOMBOL4
TBL4C:   JB   TOMBOL4,TBL5C
        MOV   A,#20
        CALL  SEND
        JMP   CEK_TOMBOL4
TBL5C:   JB   TOMBOL5,TBL6C
        MOV   A,#21
        CALL  SEND

```

```

        JMP      CEK_TOMBOLA
TBL6C:  JB      TOMBOL6,TBL7C
        MOV     A,#22
        CALL   SEND
        JMP     CEK_TOMBOLA
TBL7C:  JB      TOMBOL7,TBL8C
        MOV     A,#23
        CALL   SEND
        JMP     CEK_TOMBOLA
TBL8C:  JB      TOMBOL8,CEK_TOMBOLA4
        MOV     A,#24
        CALL   SEND

;#####
;#####
;      INPUT PORT A PPI 5
;#####
;#####
CEK_TOMBOLA4:
        CALL   ENABLE_PPI_5
        CALL   BACA_PORT_A
        CJNE  A,PPI5PA,SCAN15
        JMP     CEK_TOMBOL2B
SCAN15: MOV     PPI5PA,A
        MOV     20H,A
TBL15:  JB      TOMBOL1,TBL25
        MOV     A,#25
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL25:  JB      TOMBOL2,TBL35
        MOV     A,#26
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL35:  JB      TOMBOL3,TBL45
        MOV     A,#27
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL45:  JB      TOMBOL4,TBL55
        MOV     A,#28
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL55:  JB      TOMBOL5,TBL65
        MOV     A,#29
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL65:  JB      TOMBOL6,TBL75
        MOV     A,#30
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL75:  JB      TOMBOL7,TBL85
        MOV     A,#31
        CALL   SEND
        JMP     CEK_TOMBOL2B
TBL85:  JB      TOMBOL8,CEK_TOMBOL2B
        MOV     A,#32
        CALL   SEND

;#####
;#####
;      INPUT PORT B PPI 5
;#####
;#####
CEK_TOMBOL2B:
        CALL   ENABLE_PPI_5
        CALL   BACA_PORT_B
        CJNE  A,PPI5PB,SCAN25
        JMP     CEK_TOMBOL3C
SCAN25: MOV     PPI5PB,A
        MOV     20H,A
TBL1B5: JB      TOMBOL1,TBL2B5
        MOV     A,#33
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL2B5: JB      TOMBOL2,TBL3B5
        MOV     A,#34
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL3B5: JB      TOMBOL3,TBL4B5
        MOV     A,#35
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL4B5: JB      TOMBOL4,TBL5B5
        MOV     A,#36
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL5B5: JB      TOMBOL5,TBL6B5
        MOV     A,#37
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL6B5: JB      TOMBOL6,TBL7B5
        MOV     A,#38
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL7B5: JB      TOMBOL7,TBL8B5
        MOV     A,#39
        CALL   SEND
        JMP     CEK_TOMBOL3C
TBL8B5: JB      TOMBOL8,CEK_TOMBOL3C
        MOV     A,#40
        CALL   SEND

;#####
;#####
;      INPUT PORT C PPI 5
;#####
;#####
CEK_TOMBOL3C:
        CALL   ENABLE_PPI_5
        CALL   BACA_PORT_C
        CJNE  A,PPI5PC,SCAN35
        JMP     CEK_TOMBOL5_PPI6
SCAN35: MOV     PPI5PC,A
        MOV     20H,A

```



```

TBL1C5:          CALL SEND
                JMP    CEK_TOMBOL2B6
                JB     TOMBOL1,TBL2C5
                MOV    A,#41
                CALL   SEND
TBL2C5:          JMP    CEK_TOMBOL5_PPI6
                JB     TOMBOL2,TBL3C5
                MOV    A,#42
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL3C5:          JB     TOMBOL3,TBL4C5
                MOV    A,#43
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL4C5:          JB     TOMBOL4,TBL5C5
                MOV    A,#44
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL5C5:          JB     TOMBOL5,TBL6C5
                MOV    A,#45
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL6C5:          JB     TOMBOL6,TBL7C5
                MOV    A,#46
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL7C5:          JB     TOMBOL7,TBL8C5
                MOV    A,#47
                CALL   SEND
                JMP    CEK_TOMBOL5_PPI6
TBL8C5:          JB     TOMBOL8,CEK_TOMBOL5_PPI6
                MOV    A,#48
                CALL   SEND

;#####
;#####
;      INPUT PORT A PPI 6
;#####
;#####
CEK_TOMBOL5_PPI6:
    CALL  ENABLE_PPI_6
    CALL  BACA_PORT_A
    CJNE A,PPI6PA,SCAN16
    JMP   CEK_TOMBOL2B6
SCAN16:
    MOV   PPI6PA,A
    MOV   20H,A
TBL16:   JB     TOMBOL1,TBL26
    MOV   A,#49
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL26:   JB     TOMBOL2,TBL36
    MOV   A,#50
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL36:   JB     TOMBOL3,TBL46
    MOV   A,#51
                CALL  SEND
                JMP   CEK_TOMBOL2B6
TBL46:   JB     TOMBOL4,TBL56
    MOV   A,#52
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL56:   JB     TOMBOL5,TBL66
    MOV   A,#53
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL66:   JB     TOMBOL6,TBL76
    MOV   A,#54
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL76:   JB     TOMBOL7,TBL86
    MOV   A,#55
    CALL  SEND
    JMP   CEK_TOMBOL2B6
TBL86:   JB     TOMBOL8,CEK_TOMBOL2B6
    MOV   A,#56
    CALL  SEND

;#####
;#####
;      INPUT PORT B PPI 6
;#####
;#####
CEK_TOMBOL2B6:
    CALL  ENABLE_PPI_6
    CALL  BACA_PORT_B
    CJNE A,PPI6PB,SCAN26
    JMP   CEK_TOMBOL3C6
SCAN26:
    MOV   PPI6PB,A
    MOV   20H,A
TBL1B6:   JB     TOMBOL1,TBL2B6
    MOV   A,#57
    CALL  SEND
    JMP   CEK_TOMBOL3C6
TBL2B6:   JB     TOMBOL2,TBL3B6
    MOV   A,#58
    CALL  SEND
    JMP   CEK_TOMBOL3C6
TBL3B6:   JB     TOMBOL3,TBL4B6
    MOV   A,#59
    CALL  SEND
    JMP   CEK_TOMBOL3C6
TBL4B6:   JB     TOMBOL4,TBL5B6
    MOV   A,#60
    CALL  SEND
    JMP   CEK_TOMBOL3C6
TBL5B6:   JB     TOMBOL5,TBL6B6
    MOV   A,#61
    CALL  SEND
    JMP   CEK_TOMBOL3C6
TBL6B6:

```

```

        JB      TOMBOL6,TBL7B6
        MOV     A,#62
        CALL   SEND
        JMP     CEK_TOMBOL3C6

TBL7B6:
        JB      TOMBOL7,TBL8B6
        MOV     A,#63
        CALL   SEND
        JMP     CEK_TOMBOL3C6

TBL8B6:
        JB      TOMBOL8,CEK_TOMBOL3C6
        MOV     A,#64
        CALL   SEND

;#####
;#####
;      INPUT PORT C PPI 6
;#####
;#####
CEK_TOMBOL3C6:
        CALL   ENABLE_PPI_6
        CALL   BACA_PORT_C
        CJNE   A,PPI6PC,SCAN36
        JMP     CEK_TOMBOL

SCAN36:
        MOV     PPI6PC,A
        MOV     20H,A

TBL1C6:
        JB      TOMBOL1,TBL2C6
        MOV     A,65
        CALL   SEND
        JMP     WESO

TBL2C6:
        JB      TOMBOL2,TBL3C6
        MOV     A,#66
        CALL   SEND
        JMP     WESO

TBL3C6:
        JB      TOMBOL3,TBL4C6
        MOV     A,#67
        CALL   SEND
        JMP     WESO

TBL4C6:
        JB      TOMBOL4,TBL5C6
        MOV     A,#68
        CALL   SEND
        JMP     WESO

TBL5C6:
        JB      TOMBOL5,TBL6C6
        MOV     A,#69
        CALL   SEND
        JMP     WESO

TBL6C6:
        JB      TOMBOL6,TBL7C6
        MOV     A,#70
        CALL   SEND
        JMP     WESO

TBL7C6:
        JB      TOMBOL7,TBL8C6
        MOV     A,#71
        CALL   SEND
        JMP     WESO

TBL8C6:
        JB      TOMBOL8,WESO
        MOV     A,#72
        CALL   SEND

WESO:  LJMPL   CEK_TOMBOL

SEND:
        CLR     ES
        MOV     SBUF,A
        JNB    TI,$
        CLR    TI
        SETB   ES

DELAY: DJNZ    R7,$
        DJNZ    R7,$
        DJNZ    R6,DELAY
        RET

ENABLE_PPI_1:
        CLR     ADDR0
        CLR     ADDR1
        CLR     ADDR2
        RET

ENABLE_PPI_2:
        SETB   ADDR0
        CLR     ADDR1
        CLR     ADDR2
        RET

ENABLE_PPI_3:
        CLR     ADDR0
        SETB   ADDR1
        CLR     ADDR2
        RET

ENABLE_PPI_4:
        SETB   ADDR0
        SETB   ADDR1
        CLR     ADDR2
        RET

ENABLE_PPI_5:
        CLR     ADDR0
        CLR     ADDR1
        SETB   ADDR2
        RET

ENABLE_PPI_6:
        SETB   ADDR0
        CLR     ADDR1
        SETB   ADDR2
        RET

TULIS_PORT_A:
        MOV     R0,#0
        MOVX   @R0,A
        RET

TULIS_PORT_B:
        MOV     R0,#1
        MOVX   @R0,A
        RET

TULIS_PORT_C:
        MOV     R0,#2
        MOVX   @R0,A
        RET

```

```

BACA_PORT_A:
MOV R0,#0
MOVX A,@R0
RET

BACA_PORT_B:
MOV R0,#1
MOVX A,@R0
RET

BACA_PORT_C:
MOV R0,#2
MOVX A,@R0
RET

SERIAL:
CLR ES
PUSH ACC
JNB RI,$
MOV A,SBUF
CLR RI
CLR LAMPU
MOV DATA_LAMPU,A
POP ACC
SETB ES
RETI

;-----
; AKSES PPI UNTUK NYALAKAN LAMPU
; AKSES PPI 1
; AKSES PORT A PPI 1

AKSES: CJNE A,#0,LP2
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT0
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP2: CJNE A,#1,LP3
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT1
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP3: CJNE A,#2,LP4
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT2
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP4: CJNE A,#3,LP5
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT3
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP5: CJNE A,#4,LP6
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT4
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP6: CJNE A,#5,LP7

CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT5
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP7: CJNE A,#6,LP8
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT6
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES
LP8: CJNE A,#7,LP9
CALL ENABLE_PPI_1
MOV A,PPI1PA
CLR BIT7
MOV PPI1PA,A
CALL TULIS_PORT_A
JMP WES

; KSES PORT B PPI 1
LP9: CJNE A,#8,LP10
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT0
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP10: CJNE A,#9,LP11
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT1
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP11: CJNE A,#0AH,LP12
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT2
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP12: CJNE A,#0BH,LP13
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT3
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP13: CJNE A,#0CH,LP14
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT4
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP14: CJNE A,#0DH,LP15
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT5
MOV PPI1PB,A
CALL TULIS_PORT_B
JMP WES
LP15: CJNE A,#0EH,LP16
CALL ENABLE_PPI_1
MOV A,PPI1PB
CLR BIT6
MOV PPI1PB,A
CALL TULIS_PORT_B

```

```

LP16:  JMP      WES
        CJNE   A,#0FH,LP17
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PB
        CLR   BIT7
        MOV   PPI1PB,A
        CALL  TULIS_PORT_B
        JMP   WES

LP17:  ;      AKSES PORT C PPI 1
        CJNE   A,#10H,LP18
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT0
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP18:  CJNE   A,#11H,LP19
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT1
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP19:  CJNE   A,#12H,LP20
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT2
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP20:  CJNE   A,#13H,LP21
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT3
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP21:  CJNE   A,#14H,LP22
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT4
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP22:  CJNE   A,#15H,LP23
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT5
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP23:  CJNE   A,#16H,LP24
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT6
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP24:  CJNE   A,#17H,LP25
        CALL  ENABLE_PPI_1
        MOV   A,PPI1PC
        CLR   BIT7
        MOV   PPI1PC,A
        CALL  TULIS_PORT_C
        JMP   WES

;-----AKSES PPI 2-----
LP25:  ;      AKSES PORT A PPI 2
        CJNE   A,#18H,LP26

        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT0
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP26:  CJNE   A,#19H,LP27
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT1
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP27:  CJNE   A,#1AH,LP28
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT2
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP28:  CJNE   A,#1BH,LP29
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT3
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP29:  CJNE   A,#1CH,LP30
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT4
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP30:  CJNE   A,#1DH,LP31
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT5
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP31:  CJNE   A,#1EH,LP32
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT6
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP32:  CJNE   A,#1FH,LP33
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PA
        CLR   BIT7
        MOV   PPI2PA,A
        CALL  TULIS_PORT_A
        JMP   WES

;      AKSES PORT B PPI 2
LP33:  CJNE   A,#20H,LP34
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT0
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP34:  CJNE   A,#21H,LP35
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT1
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B

```

```

LP35:  JMP      WES
        CJNE   A,#22H,LP36
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT2
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP36:  CJNE   A,#23H,LP37
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT3
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP37:  CJNE   A,#24H,LP38
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT4
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP38:  CJNE   A,#25H,LP39
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT5
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP39:  CJNE   A,#26H,LP40
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT6
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP40:  CJNE   A,#27H,LP41
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        CLR   BIT7
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP41:  ;        AKSES PORT C PPI 2
        CJNE   A,#28H,LP42
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT0
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP42:  CJNE   A,#29H,LP43
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT1
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP43:  CJNE   A,#2AH,LP44
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT2
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP44:  CJNE   A,#2BH,LP45
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT3
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP45:  CJNE   A,#2CH,LP46
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT4
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP46:  CJNE   A,#2DH,LP47
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT5
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP47:  CJNE   A,#2EH,LP48
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT6
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
LP48:  CJNE   A,#2FH,LP49
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        CLR   BIT7
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES
;-----AKSES PPI 3 -----
LP49:  ;        AKSES PORT A PPI 3
        CJNE   A,#30H,LP50
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        CLR   BIT0
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP50:  CJNE   A,#31H,LP51
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        CLR   BIT1
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP51:  CJNE   A,#32H,LP52
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        CLR   BIT2
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP52:  CJNE   A,#33H,LP53
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        CLR   BIT3
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP53:  CJNE   A,#34H,LP54
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        CLR   BIT4
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP54:  CJNE   A,#35H,LP55

```

```

CALL    ENABLE_PPI_3
MOV     A,PPI3PA
CLR     BIT5
MOV     PPI3PA,A
CALL    TULIS_PORT_A
JMP     WES
LP55:   CJNE   A,#36H,LP56
CALL    ENABLE_PPI_3
MOV     A,PPI3PA
CLR     BIT6
MOV     PPI3PA,A
CALL    TULIS_PORT_A
JMP     WES
LP56:   CJNE   A,#37H,LP57
CALL    ENABLE_PPI_3
MOV     A,PPI3PA
CLR     BIT7
MOV     PPI3PA,A
CALL    TULIS_PORT_A
JMP     WES

;       AKSES PORT B PPI 3
LP57:   CJNE   A,#38H,LP58
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT0
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP58:   CJNE   A,#39H,LP59
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT1
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP59:   CJNE   A,#3AH,LP60
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT2
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP60:   CJNE   A,#3BH,LP61
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT3
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP61:   CJNE   A,#3CH,LP62
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT4
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP62:   CJNE   A,#3DH,LP63
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT5
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES
LP63:   CJNE   A,#3EH,LP64
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT6
MOV     PPI3PB,A
CALL    TULIS_PORT_B

LP64:   JMP     WES
        CJNE   A,#3FH,LP65
CALL    ENABLE_PPI_3
MOV     A,PPI3PB
CLR     BIT7
MOV     PPI3PB,A
CALL    TULIS_PORT_B
JMP     WES

;       AKSES PORT C PPI 3
LP65:   CJNE   A,#40H,LP66
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT0
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP66:   CJNE   A,#41H,LP67
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT1
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP67:   CJNE   A,#42H,LP68
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT2
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP68:   CJNE   A,#43H,LP69
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT3
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP69:   CJNE   A,#44H,LP70
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT4
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP70:   CJNE   A,#45H,LP71
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT5
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP71:   CJNE   A,#46H,LP72
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT6
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES
LP72:   CJNE   A,#47H,LP10
CALL    ENABLE_PPI_3
MOV     A,PPI3PC
CLR     BIT7
MOV     PPI3PC,A
CALL    TULIS_PORT_C
JMP     WES

; -----
;       AKSES PPI UNTUK MATIKAN LAMPU (
DATA : 48 S/D 8F)

```

```

; AKSES PPI 1
; AKSES PORT A PPI 1
LP10: CJNE A,#48H,LP20
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT0
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP20: CJNE A,#49H,LP30
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT1
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP30: CJNE A,#4AH,LP40
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT2
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP40: CJNE A,#4BH,LP50
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT3
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP50: CJNE A,#4CH,LP60
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT4
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP60: CJNE A,#4DH,LP70
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT5
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP70: CJNE A,#4EH,LP80
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT6
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP80: CJNE A,#4FH,LP90
      CALL ENABLE_PPI_1
      MOV A,PPI1PA
      SETB BIT7
      MOV PPI1PA,A
      CALL TULIS_PORT_A
      JMP WES
LP90: ; AKSES PORT B PPI 1
      CJNE A,#50H,LP100
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT0
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP100: CJNE A,#51H,LP110
      CALL

```

```

      MOV A,PPI1PB
      SETB BIT1
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP110: CJNE A,#52H,LP120
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT2
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP120: CJNE A,#53H,LP130
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT3
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP130: CJNE A,#54H,LP140
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT4
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP140: CJNE A,#55H,LP150
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT5
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP150: CJNE A,#56H,LP160
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT6
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP160: CJNE A,#57H,LP170
      CALL ENABLE_PPI_1
      MOV A,PPI1PB
      SETB BIT7
      MOV PPI1PB,A
      CALL TULIS_PORT_B
      JMP WES
LP170: ; AKSES PORT C PPI 1
      CJNE A,#58H,LP180
      CALL ENABLE_PPI_1
      MOV A,PPI1PC
      SETB BIT0
      MOV PPI1PC,A
      CALL TULIS_PORT_C
      JMP WES
LP180: CJNE A,#59H,LP190
      CALL ENABLE_PPI_1
      MOV A,PPI1PC
      SETB BIT1
      MOV PPI1PC,A
      CALL TULIS_PORT_C
      JMP WES
LP190: CJNE A,#5AH,LP200
      CALL ENABLE_PPI_1
      MOV A,PPI1PC
      SETB BIT2
      MOV PPI1PC,A
      CALL TULIS_PORT_C
      JMP WES

```

```

LP200: CJNE    A,#5BH,LP210
        CALL    ENABLE_PPI_1
        MOV    A,PPI1PC
        SETB   BIT3
        MOV    PPI1PC,A
        CALL   TULIS_PORT_C
        JMP    WES
LP210: CJNE    A,#5CH,LP220
        CALL    ENABLE_PPI_1
        MOV    A,PPI1PC
        SETB   BIT4
        MOV    PPI1PC,A
        CALL   TULIS_PORT_C
        JMP    WES
LP220: CJNE    A,#5DH,LP230
        CALL    ENABLE_PPI_1
        MOV    A,PPI1PC
        SETB   BIT5
        MOV    PPI1PC,A
        CALL   TULIS_PORT_C
        JMP    WES
LP230: CJNE    A,#5EH,LP240
        CALL    ENABLE_PPI_1
        MOV    A,PPI1PC
        SETB   BIT6
        MOV    PPI1PC,A
        CALL   TULIS_PORT_C
        JMP    WES
LP240: CJNE    A,#5FH,LP250
        CALL    ENABLE_PPI_1
        MOV    A,PPI1PC
        SETB   BIT7
        MOV    PPI1PC,A
        CALL   TULIS_PORT_C
        JMP    WES

;-----AKSES PPI 2-----
; AKSES PORT A PPI 2
LP250: CJNE    A,#60H,LP260
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT0
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP260: CJNE    A,#61H,LP270
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT1
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP270: CJNE    A,#62H,LP280
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT2
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP280: CJNE    A,#63H,LP290
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT3
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP290: CJNE    A,#64H,LP300
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT4
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP300: CJNE    A,#65H,LP310
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT5
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP310: CJNE    A,#66H,LP320
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT6
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES
LP320: CJNE    A,#67H,LP330
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PA
        SETB   BIT7
        MOV    PPI2PA,A
        CALL   TULIS_PORT_A
        JMP    WES

; AKSES PORT B PPI 2
LP330: CJNE    A,#68H,LP340
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT0
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP340: CJNE    A,#69H,LP350
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT1
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP350: CJNE    A,#6AH,LP360
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT2
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP360: CJNE    A,#6BH,LP370
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT3
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP370: CJNE    A,#6CH,LP380
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT4
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP380: CJNE    A,#6DH,LP390
        CALL    ENABLE_PPI_2
        MOV    A,PPI2PB
        SETB   BIT5
        MOV    PPI2PB,A
        CALL   TULIS_PORT_B
        JMP    WES
LP390: CJNE    A,#6EH,LP400
        CALL    ENABLE_PPI_2

```



```

MOV     A,PPI2PB
SETB   BIT6
MOV     PPI2PB,A
CALL   TULIS_PORT_B
LP400: JMP     WES
        CJNE  A,#6FH,LP410
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PB
        SETB BIT7
        MOV   PPI2PB,A
        CALL  TULIS_PORT_B
        JMP   WES

;       AKSES PORT C PPI 2
LP410:  CJNE  A,#70H,LP420
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT0
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP420:  CJNE  A,#71H,LP430
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT1
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP430:  CJNE  A,#72H,LP440
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT2
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP440:  CJNE  A,#73H,LP450
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT3
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP450:  CJNE  A,#74H,LP460
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT4
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP460:  CJNE  A,#75H,LP470
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT5
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP470:  CJNE  A,#76H,LP480
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT6
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
LP480:  CJNE  A,#77H,LP490
        CALL  ENABLE_PPI_2
        MOV   A,PPI2PC
        SETB BIT7
        MOV   PPI2PC,A
        CALL  TULIS_PORT_C
        JMP   WES

;-----AKSES PPI 3 -----
;       AKSES PORT A PPI 3
LP490:  CJNE  A,#78H,LP500
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT0
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP500:  CJNE  A,#79H,LP510
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT1
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP510:  CJNE  A,#7AH,LP520
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT2
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP520:  CJNE  A,#7BH,LP530
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT3
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP530:  CJNE  A,#7CH,LP540
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT4
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP540:  CJNE  A,#7DH,LP550
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT5
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP550:  CJNE  A,#7EH,LP560
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT6
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES
LP560:  CJNE  A,#7FH,LP570
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PA
        SETB BIT7
        MOV   PPI3PA,A
        CALL  TULIS_PORT_A
        JMP   WES

;       AKSES PORT B PPI 3
LP570:  CJNE  A,#80H,LP580
        CALL  ENABLE_PPI_3
        MOV   A,PPI3PB
        SETB BIT0
        MOV   PPI3PB,A
        CALL  TULIS_PORT_B
        JMP   WES
LP580:  CJNE  A,#81H,LP590
        CALL  ENABLE_PPI_3

```

```

MOV     A,PPI3PB
SETB   BIT1
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP590: CJNE   A,#82H,LP600
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT2
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP600: CJNE   A,#83H,LP610
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT3
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP610: CJNE   A,#84H,LP620
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT4
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP620: CJNE   A,#85H,LP630
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT5
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP630: CJNE   A,#86H,LP640
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT6
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES
LP640: CJNE   A,#87H,LP650
CALL   ENABLE_PPI_3
MOV    A,PPI3PB
SETB   BIT7
MOV    PPI3PB,A
CALL   TULIS_PORT_B
JMP    WES

;      AKSES PORT C PPI 3
LP650: CJNE   A,#88H,LP660
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT0
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP660: CJNE   A,#89H,LP670
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT1
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP670: CJNE   A,#8AH,LP680
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT2
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES

LP680: CJNE   A,#8BH,LP690
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT3
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP690: CJNE   A,#8CH,LP700
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT4
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP700: CJNE   A,#8DH,LP710
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT5
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP710: CJNE   A,#8EH,LP720
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT6
MOV    PPI3PC,A
CALL   TULIS_PORT_C
JMP    WES
LP720: CJNE   A,#8FH,WES
CALL   ENABLE_PPI_3
MOV    A,PPI3PC
SETB   BIT7
MOV    PPI3PC,A
CALL   TULIS_PORT_C
WES:   RET
      END

```