

BAB III

METODE PENELITIAN

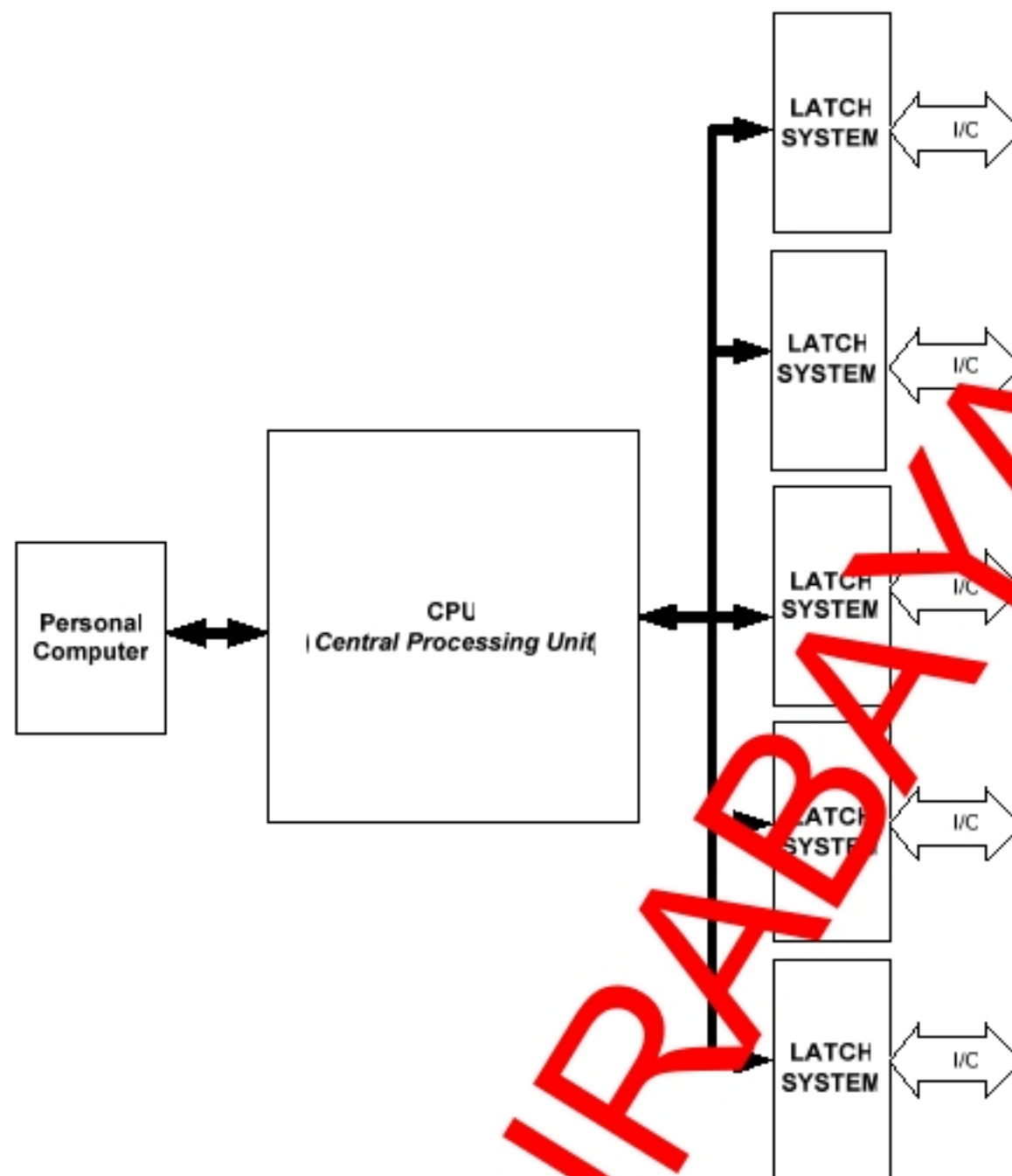
Metode penelitian yang digunakan pada pembuatan perangkat keras dan perangkat lunak, yaitu dengan studi kepustakaan. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku, bahan-bahan kuliah dan internet yang berkaitan dengan permasalahan.

Dari data-data untuk memecahkan permasalahan yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras. Setelah pembuatan perangkat keras, maka dilakukan pengujian terhadap perangkat keras tersebut, baik secara teori maupun dengan cara membuat program kecil pada tiap bagiannya. Terakhir adalah pembuatan perangkat lunak yang dapat berinteraksi dengan perangkat keras yang dibuat.

Perencanaan dan pembuatan yang dilakukan dalam tugas akhir ini bersifat praktis, yaitu perencanaan dan pembahasan yang mengarah pada pembuatan perangkat keras maupun perangkat lunak secara nyata dan aplikatif.

3.1. Perancangan Perangkat Keras (*Hardware*)

Dalam perancangan dan pembuatan Programmable Controller ini dibuat diagram blok seperti di bawah ini.



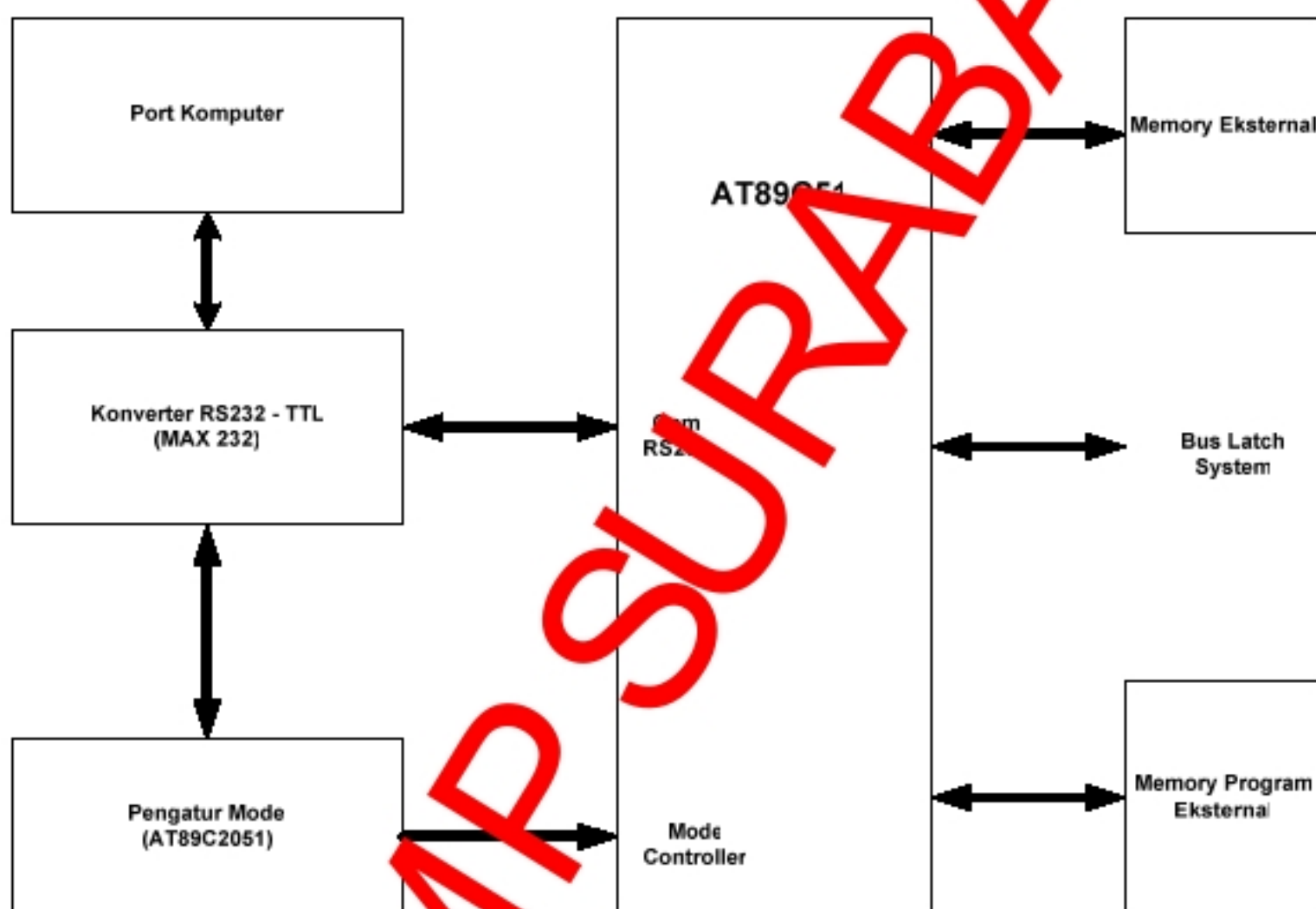
Gambar 3.1. Diagram Blok Programmable Controller.

Pada diagram blok di atas terlihat bahwa perancangan perangkat keras terdapat dua jenis modul, di antaranya: modul CPU dan Latch System yang berjumlah 5 buah.

3.1.1. CPU (*Central Processing Unit*)

Central Processing Unit yang selanjutnya disebut CPU adalah pusat pemrosesan yang berhubungan dengan Programmable Controller. Basis perancangan CPU menggunakan microcontroller MCS-51, selain itu CPU juga membutuhkan

komponen yang lain untuk mendukung kerjanya. Komponen yang dimaksud di antaranya adalah dekoder menggunakan IC 74LS138, multiplexer menggunakan IC 74LS157, Latch Address menggunakan IC 74HCT573. Seperti terlihat pada gambar diagram blok, CPU memiliki beberapa modul di antaranya adalah AT89C51, Memory Data Eksternal, Memory Program Eksternal, Konverter RS232-TTL, dan Pengatur Mode. Pada gambar diagram blok terdapat Bus Latch System yang berfungsi sebagai *bus* komunikasi antara Latch System dengan CPU.



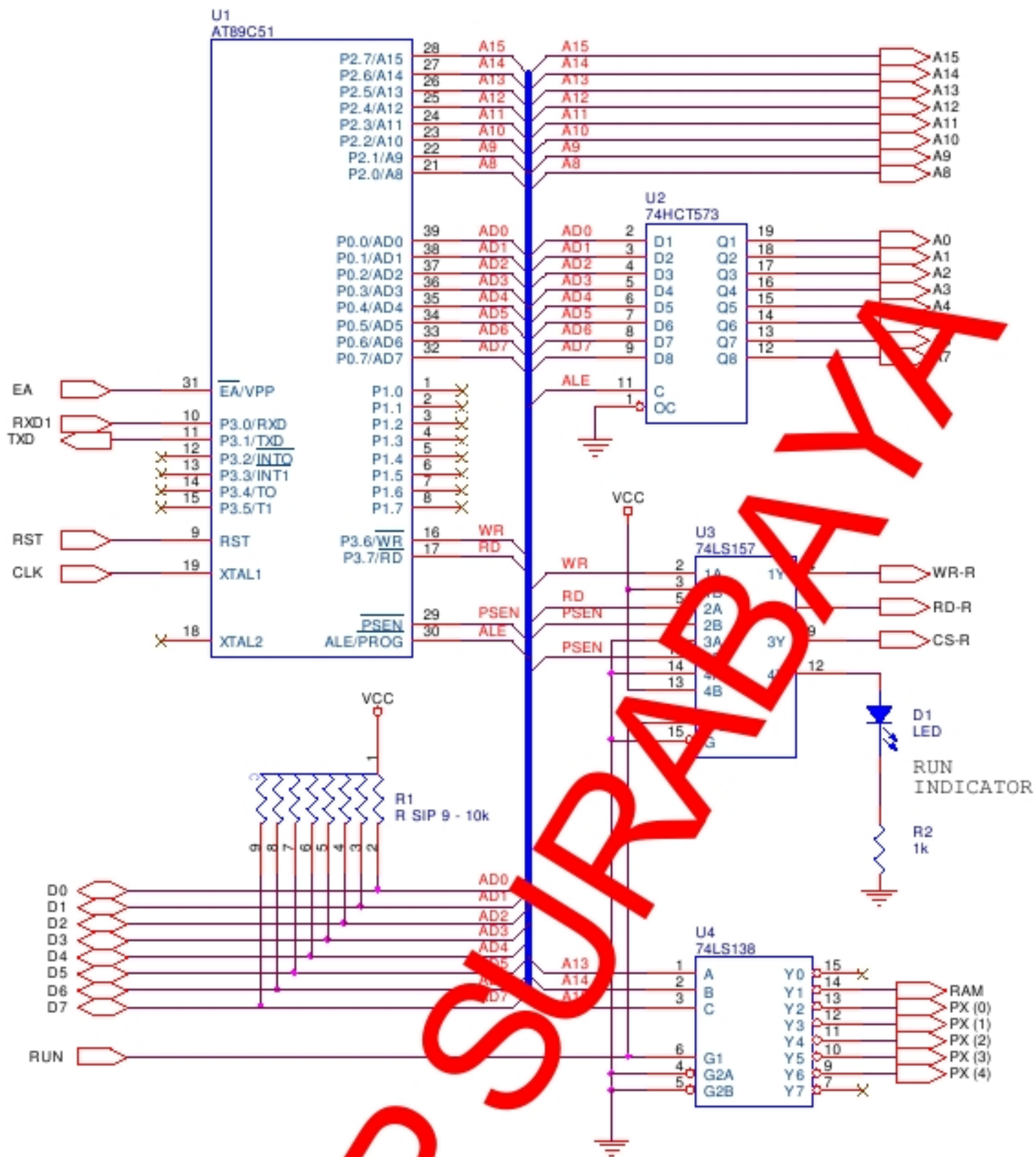
Gambar 3.2. Diagram blok CPU.

Port 0 pada AT89C51 digunakan sebagai 8 bit bus data dan 8 bit alamat rendah dengan sistem *multiplexing addressing* (pengalamatan bergantian). Port 0 harus diberi resistor *pull-up* agar dapat menerima data dari Memory Data Eksternal, Memory Program Eksternal, dan Latch System Bus. Port 2 digunakan sebagai 8 bit

bus alamat tinggi. Port 3.0 (pin RXD) sebagai port serial input, dan port 3.1 (pin TXD) sebagai serial output yang digunakan untuk komunikasi dengan komputer, semua pin tersebut disambungkan ke *line* RXD dan *line* TXD. Port 3.6 (pin \overline{WR}) digunakan untuk menulis data ke Memory Data Eksternal. Port 3.7 (pin \overline{RD}) digunakan untuk membaca data dari Memory Data Eksternal.

Pin ALE atau *Address Latch Enable* berfungsi sebagai penunjuk keadaan bahwa port 0 sedang dalam kondisi bus data atau sebagai bus alamat. Pin ini digunakan ENABLE dari IC 74HCT573 (pin 11 / C) yang berfungsi sebagai buffer penampung alamat rendah dari bus alamat. Pin RESET berfungsi untuk mengembalikan ke keadaan awal proses dari AT89C51, pin ini disambungkan ke *line* RST. Pin X1 dihubungkan ke rangkaian clock oscillator sebagai penentu kecepatan proses dari AT89C51. Pin \overline{PSEN} (*Program Store Enable*) berfungsi sebagai sinyal pembacaan program dari Memory Data Eksternal. Pin \overline{EA} (*External Access Enabled*) berfungsi untuk menentukan Memory Program yang dipakai, eksternal atau internal Memory Program.

STIKOMP SURABAYA



Gambar 3.3 Rangkaian elektronik minimum sistem CPU.

CPU dirancang untuk dapat bekerja pada mode standby dan mode running.

Pada mode standby CPU hanya dapat menulis dan membaca Memory Program Eksternal. Pada mode running CPU hanya dapat membaca dan menulis Memory Data Eksternal dan mengakses Bus Latch System.

3.1.1.1. Mode Standby

Pada kondisi standby komputer hanya dapat meng-akses Memory Program Eksternal, dan komputer tidak dapat meng-akses Memory Data Eksternal beserta Bus Latch System.

Kondisi standby *line* EA harus diberi sinyal *high*, sehingga pin \overline{EA} pada AT89C51 mendapat sinyal *high* pula, pada kondisi ini AT89C51 menjalankan program yang ada pada Memory Program Internal.

Selain itu kondisi standby *line* RUN harus diberi sinyal *low*. Sehingga pin 1 pada IC 74LS157 yang berfungsi sebagai selektor mendapatkan sinyal *low* sehingga sinyal yang masuk pada port input A ke untuk diteruskan ke outputnya. Semua output yang ada pada IC 74LS157 berhubungan langsung dengan bus kontrol yang ada pada Memory Program Eksternal dan indikator RUN. Port input 1A menerima sinyal \overline{WR} dari AT89C51 untuk diteruskan ke port output 1 yang disambungkan ke *line* WR-R. Port input 2A menerima sinyal \overline{RD} dari AT89C51 untuk diteruskan ke port output 2 yang disambungkan ke *line* RD-R. Port input 3A disambungkan ke ground secara langsung agar CS-R yang disambungkan ke output 3 menerima sinyal *low* secara terus-menerus. Dan port input 4A disambungkan ke ground pula yang difungsikan untuk mematikan lampu D1 sebagai indikator status running.

Pin G1 dari IC 74LS138 mendapat sinyal *low* dari *line* RUN, yang berarti tidak mengaktifkan IC 74LS138 sebagai dekoder dan juga mematikan akses ke Memory Data Eksternal dan Bus Latch System.

3.1.1.2. Mode Running

Kondisi running CPU menjalankan Memory Program Eksternal yang telah di-*download* dari komputer, setelah menjalankan mode running komputer tidak dapat mengubah isi Memory Program Eksternal, komputer hanya dapat meng-akses Memory Data Eksternal dan Bus Latch System.

Kondisi running *line* EA harus diberi sinyal *low*, sehingga port \overline{EA} pada AT89C51 mendapat sinyal *low*. Pada kondisi seperti ini AT89C51 menjalankan program yang ada pada Memory Program Eksternal.

Pada kondisi ini *line* RUN harus diberi sinyal *high*, sehingga pin 1 pada IC 75LS157 yang berfungsi sebagai selektor mendapatkan sinyal *high*, yang mengkondisikan IC 74LS157 akan meneruskan input ke outputnya. Semua output IC 74LS157 berhubungan langsung dengan bus kontrol yang ada pada Memory Program Eksternal. Port input 1B disambungkan langsung ke VCC yang berarti akan selalu memberi sinyal *high* pada output 1, sehingga *line* WR-R selalu mendapatkan sinyal *high*. Port input 2B disambungkan dengan port \overline{PSEN} pada AT89C51 yang kemudian sinyal tersebut diteruskan ke output 2 yang disambungkan ke *line* RD-R. Port input 3B juga disambungkan dengan port \overline{PSEN} pada AT89C51 pula, dan sinyal yang keluar dari port output diteruskan ke *line* CS-R. Dan port input 4B dihubungkan secara langsung dengan VCC yang berfungsi untuk menyalakan lampu D1 sebagai indikator status running.

Kondisi running dimana *line* RUN mendapatkan sinyal *high*, maka port G1 pada IC 74LS138 mendapatkan sinyal *high* pula. Dari sinyal yang dibawa oleh *line* RUN, IC 74LS138 statusnya menjadi aktif, sehingga Memory Data Eksternal dan Bus Latch System dapat di-akses oleh AT89C51. Dengan aktifnya IC 74LS138, port

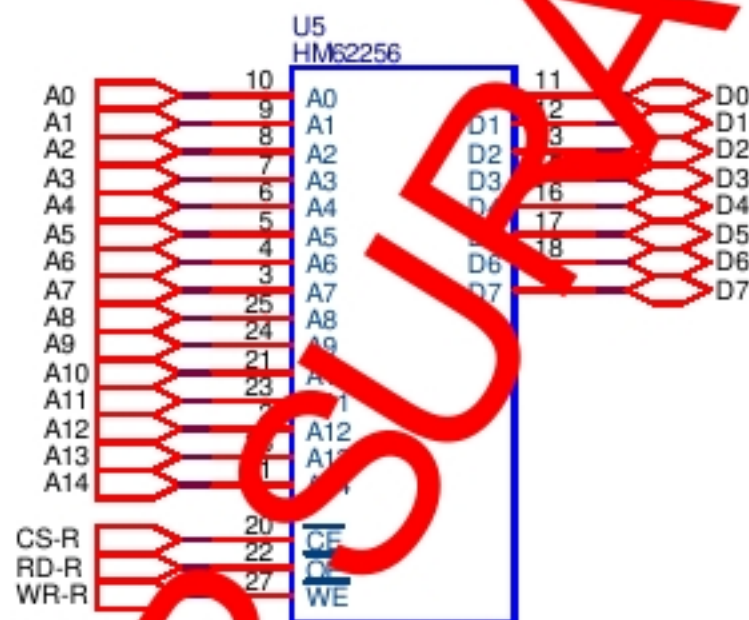
outputnya akan mengkondisikan sesuai dengan kondisi port input yang disambungkan ke alamat A13,A14,A15 pada AT89C51. Pada saat IC 74LS138 aktif, outputnya akan meng-aktifkan RAM, P0, P1, P2, dan P4 dengan alamat sebagai berikut:

- RAM: 02000h
- P0 (Port 0): 04000h
- P1 (Port 1): 06000h
- P2 (Port 2): 08000h
- P3 (Port 3): 0A000h
- P4 (Port 4): 0C000h

STIKOMP SURABAYA

3.1.2. Memory Program Eksternal

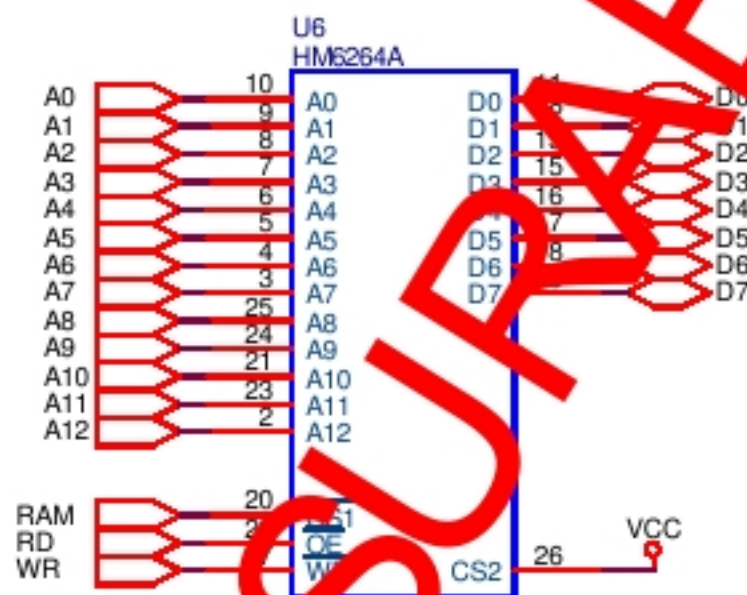
Memory Program Eksternal yang digunakan CPU adalah tipe IC RAM 62256 berkapasitas 32 Kbyte. Seperti yang telah dijelaskan pada kondisi standby dan kondisi running, Memory Program Eksternal hanya dapat di-akses pada kondisi standby dan dijalankan pada kondisi running. Alamat A0 – A14 dihubungkan pada bus alamat CPU, data D0 – D7 dihubungkan ke bus data CPU dan controller memory dihubungkan pada rangkaian IC 74LS157 pada line yang telah ditentukan, di antaranya port \overline{CE} disambungkan ke line CS-R, port \overline{OE} disambungkan ke line RD-R, port \overline{WE} disambungkan ke line WR-R. Penjelasan Memory Program Eksternal dapat dilihat seperti pada gambar 3.4.



Gambar 3.4. Konfigurasi rangkaian elektronik line dan bus pada Memory Program Eksternal.

3.1.3. Memory Data Eksternal

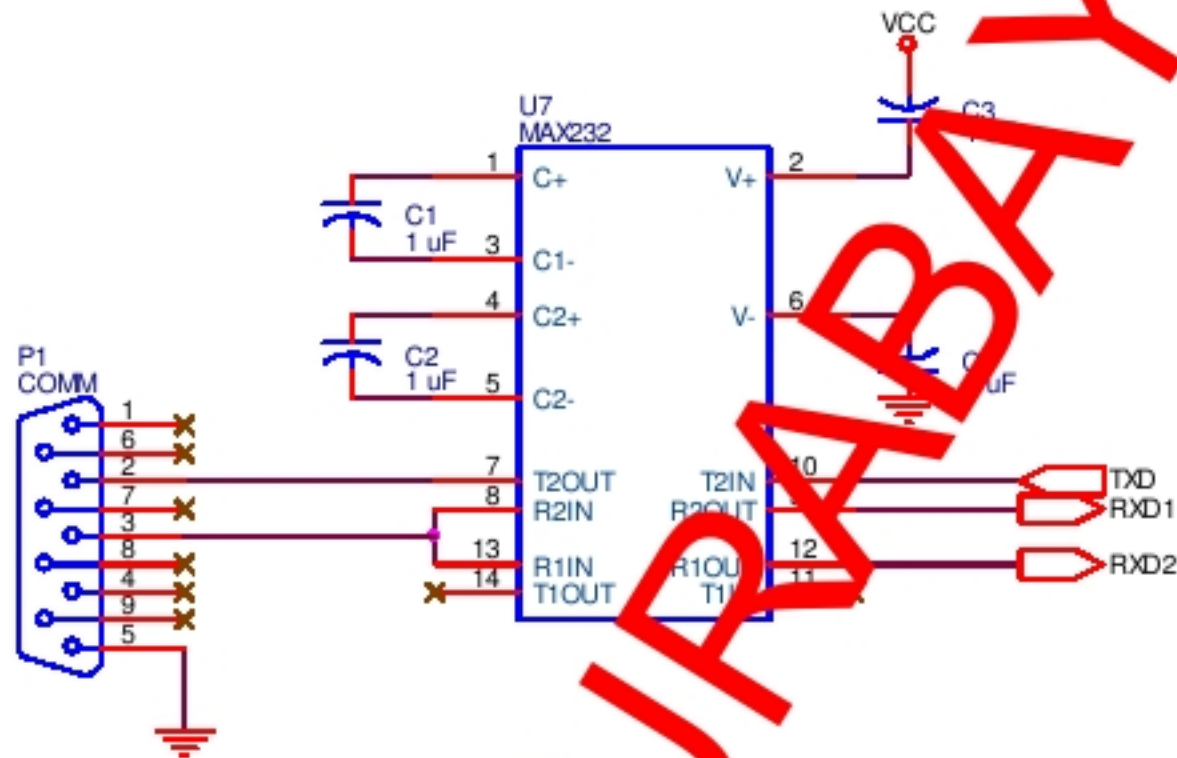
CPU dirancang dengan Memory Data Eksternal 8 Kbyte yang berfungsi untuk menyimpan data sementara pada saat program dijalankan. Data-data tersebut meliputi Timer, Counter, Flag, Register, dan data serial. Supaya Memory bekerja sesuai dengan kondisi yang diinginkan maka port alamat A0 – A12 disambungkan ke bus alamat CPU, port data D0 – D7 disambungkan ke bus data CPU, dan untuk kontroller memory disambungkan ke line-line yang sudah ditentukan di antaranya yaitu port $\overline{CS1}$ disambungkan ke line RAM, port CS2 disambungkan langsung ke VCC, port \overline{OE} disambungkan ke line RD, dan port \overline{WE} disambungkan ke line WR.



Gambar 3.5. Konfigurasi rangkaian elektronik line dan bus pada Memory Data Eksternal.

3.1.4. Konverter RS232 – TTL

CPU dapat berkomunikasi dengan komputer dengan standart RS232. Karena tegangan pada CPU menggunakan tegangan TTL, maka agar dapat berkomunikasi antara CPU dan komputer, tegangan RS232 dikonversikan menjadi tegangan TTL atau sebaliknya. Konverter RS232 – TTL pada CPU dirancang menggunakan IC MAX232.



Gambar 3.6. Konfigurasi rangkaian elektronik *line* dan *bus* pada Konverter RS232 - TTL.

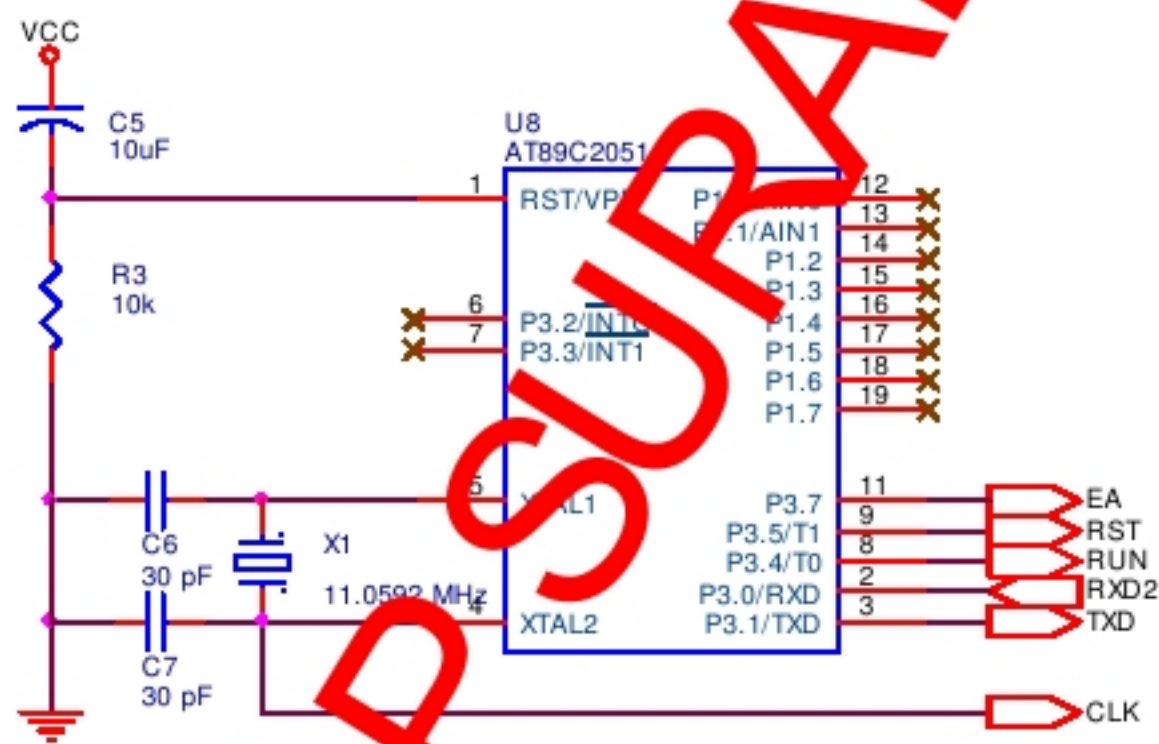
C1, C2, C3 dan C4 adalah kapasitor yang digunakan untuk menggetar generator yang ada dalam IC MAX232, generator pada MAX232 difungsikan sebagai pengganda tegangan dan pembalik tegangan, sesuai dengan tegangan komunikasi RS232 yang dibutuhkan.

Pin 8 MAX232 (Port R2IN) disambungkan ke RXD, yang berfungsi untuk menerima data dari komputer, data yang masuk dari Port R2IN dengan tegangan RS232 menghasilkan tegangan TTL yang dikeluarkan pada pin 9 (Port R2OUT),

kemudian di-broadcast ke *line* RXD yang nantinya data tersebut diolah oleh AT89C51 dan Mode Controller. Pin 10 (Port T2IN) menerima data dari Mode Controller dan AT89C51 melalui *line* TXD berupa tegangan TTL, dan menghasilkan data bertegangan RS232 melalui pin 7 (Port T2OUT) yang kemudian diterima oleh komputer.

3.1.5. Pengatur Mode

Mode Controller berfungsi untuk mengatur kerja CPU sehingga CPU dapat bekerja pada mode standby dan mode running. Untuk mengatur mode-mode tersebut digunakanlah *microcontroller* AT89C2051.



Gambar 3.7. Konfigurasi rangkaian elektronik *line* pada Mode Controller.

Kerja AT89C2051 mempengaruhi seluruh rangkaian kerja CPU, karena segala mode diatur disini. Jalur-jalur yang dapat diatur oleh AT89C2051 adalah *line* EA, RST, RUN dan CLK. Sedang *line* RXD dan TXD digunakan untuk menerima

perintah berupa data serial dari komputer, dan merespon perintah untuk dikirimkan ke komputer berupa data serial pula.

Mulai dari *line* EA yang berfungsi untuk mengatur mode AT89C51 agar dapat berjalan dalam dua mode, yaitu mode Memory Program Internal dan mode Memory Program Eksternal.

Line RST digunakan untuk men-set program AT89C51 agar berjalan pada awal program, baik program yang dijalankan melalui Memory Program Internal maupun Memory Program Eksternal.

Line RUN digunakan untuk men-set supaya komputer dapat meng-akses Memory Program Eksternal melalui AT89C51, selain itu digunakan untuk mematikan IC dekoder 74LS138, agar Memory Program Eksternal dan Bus Latch System tidak dapat diakses oleh AT89C51.

Line CLK digunakan untuk sebagai Clock Osilator *external* pada AT89C51. Clock Osilator diambil dari hasil output AT89C2051 pada pin XTAL2.

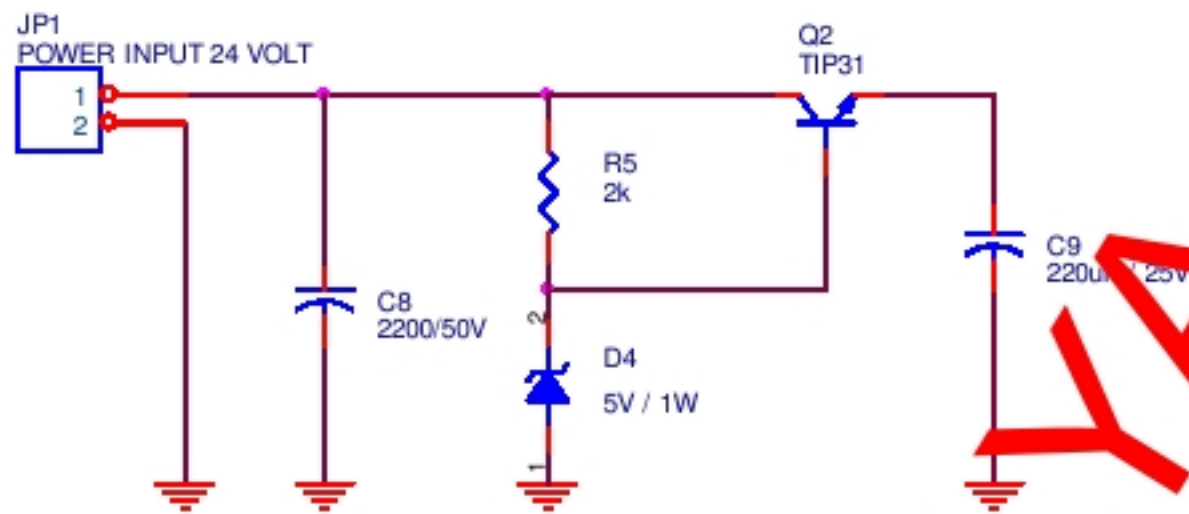
Rangkaian C5 dan R3 berfungsi untuk me-reset AT89C2051 secara otomatis pada saat tegangan VCC pertama kali diberikan. C5 berukuran 10 μ F dan R3 berukuran 10k Ω .

C6 dan C7 berfungsi untuk mendukung sistem kerja kristal osilator, dan kedua pin kristal tersebut disambungkan ke pin XTAL1 dan pin XTAL2 pada *microcontroller* AT89C2051.

3.1.6. Power Supply CPU

CPU tidak dapat bekerja bila tidak ada Power Supply, karena power supply berfungsi untuk memenuhi daya yang dibutuhkan oleh cpu dan perangkat modul

yang lain. Power supply yang dibuat di sini dapat diberi input tegangan sebesar 10 volt sampai dengan 36 volt dengan keluaran tegangan sebesar 5 volt.



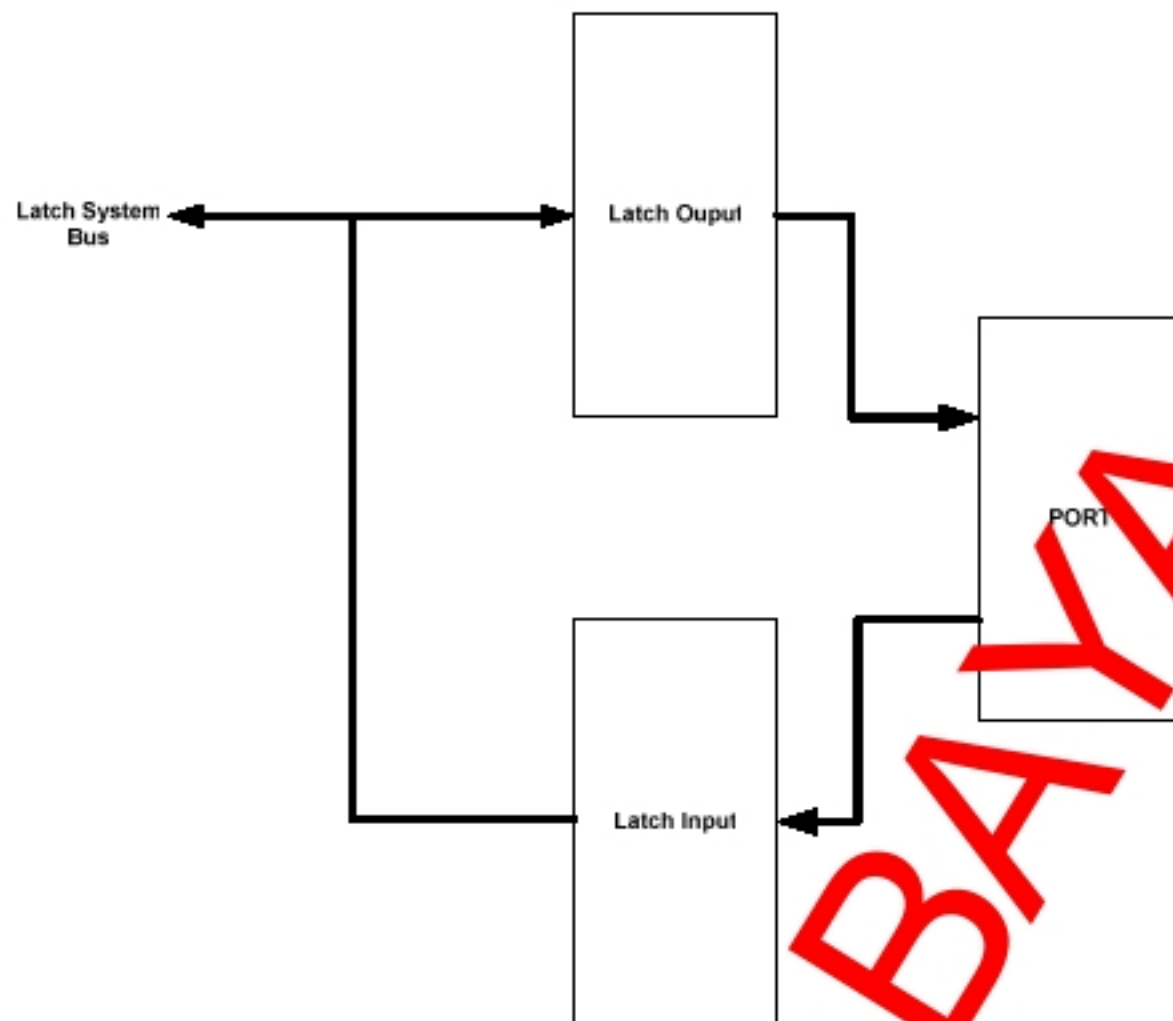
Gambar 3.8. Rangkaian elektronik power supply CPU.

Power supply menggunakan transistor switching transistor TIP-31. Agar output *emitter* pada transistor dapat mengeluarkan tegangan 5 volt secara konstan, pin *basis* pada transistor diberi tegangan konstan sebesar 5 volt pula. Tegangan konstan 5 volt didapat dari rangkaian pemertama tegangan yang dibangun oleh resistor dan diode zener.

3.1.7. Latch System

Latch System adalah suatu sistem yang digunakan untuk menahan data yang akan dikeluarkan sebagai keluaran (*output*) ke port PC, dan berfungsi juga untuk membaca data masukan (*input*) dari port PC. Latch System menerima semua data dari CPU, dengan menggunakan jalur-jalur telah dirancang pada CPU, jalur tersebut adalah *bus data*, *line WP*, *line RD*, dan *line PORT*.

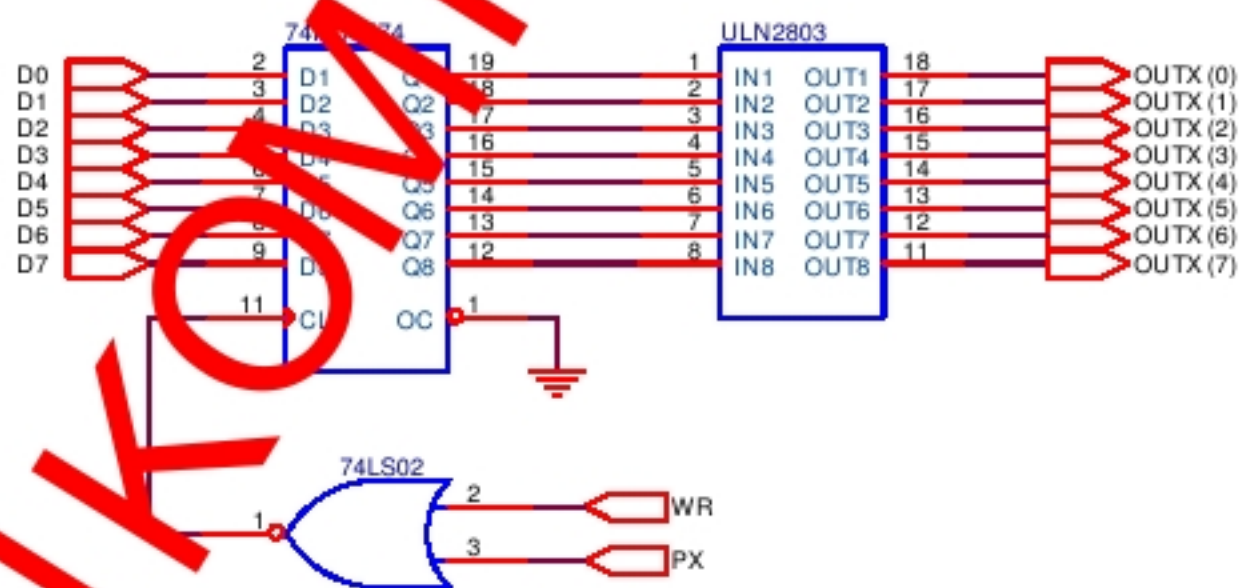
Seperti yang tampak pada diagram blok, Latch System dibagi menjadi 3 modul diantaranya modul Latch Output, modul Latch Input, dan yang terakhir modul Port.



Gambar 3.9. Diagram blok Latch System.

3.1.7.1. Latch Output

Latch Output bertugas menerima data dari CPU dan kemudian diteruskan data tersebut ke PORT. Sistem ini dirancang menggunakan perangkat IC 74HCT574 sebagai penahan data, IC ULN2803 difungsikan untuk sebagai pembalik tegangan, dan IC 74LS02 difungsikan sebagai gerbang NOR.



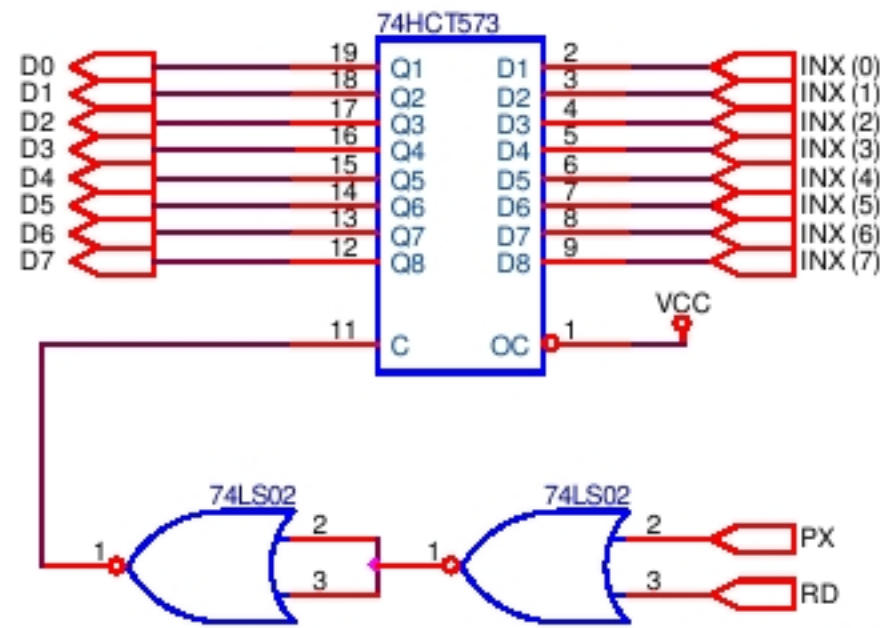
Gambar 3.10 Konfigurasi rangkaian elektronik Latch Output.

Pada kondisi normal atau *idle*, *bus* data berada pada logika *high*, *line* PORTX pada logika *high*, *line* WR berada pada logika *high*. Siklus penulisan sistem dilakukan berdasarkan urutan kerja dari CPU, dimana AT89C51 pada CPU mengeksekusi perintah "MOVX @DPTR,A". Mula-mula *line* PORTX mendapat sinyal *low* terlebih dahulu, kemudian data dikirimkan melalui *bus* data, dengan menunggu sinyal *clock* dari *line* WR, yaitu perubahan logika *low* ke logika *high*. Setelah mendapat sinyal *clock* dari CPU, maka data yang terdapat pada *bus* data tertahan oleh IC 74HCT574, dan dikeluarkan ke port outputnya. Dan setelah semua urutan siklus selesai, tiap kondisi *bus* dan *line* kembali pada posisi normal.

Tegangan PORT yang digunakan sekitar 24 Volt, maka output dari IC 74HCT574 tidak dapat langsung digunakan secara langsung, untuk merubah daya dan tegangan supaya dapat disesuaikan dengan tegangan 24 Volt, sinyal data pada IC 74HCT574 yang memiliki tegangan TTL dibalik dengan menggunakan transistor switch darlington. Transistor darlington yang dibutuhkan sebanyak pada tiap PORT sebanyak 8 buah, sesuai dengan kebutuhan karakter IC ULN2803 dapat digunakan dalam permasalahan ini.

3.1.7.2. Latch Input

Latch Input bertugas menerima data dari port, sesuai dengan permintaan yang diperintahkan dari CPU melalui Latch System Bus. Perangkat yang digunakan pada sistem ini menggunakan IC 74HCT573 sebagai pembaca data dari dari PORT, IC 74LS02 yang mempunyai fungsi sebagai gerbang NOR, dan kemudian dirancang sebagai pemilih untuk gerbang OR.

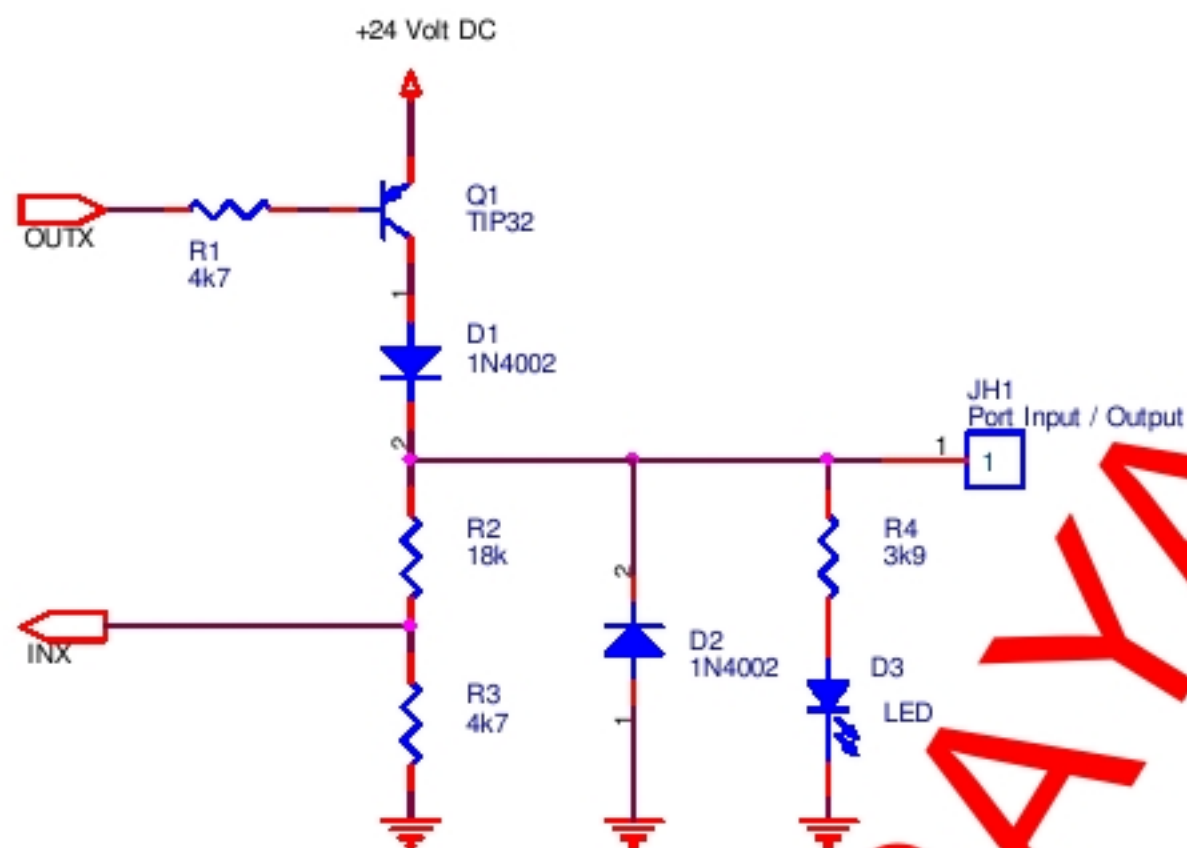


Gambar 3.11 Konfigurasi rangkaian elektronik Latch Input.

Pada sistem ini siklus pembacaannya juga berdasarkan permintaan dari CPU. Pada kondisi normal atau *idle*, *line* PORTX berada pada logika *high*, *line* RD berada pada logika *high*, dan *bus* data berada pada logika *high*. Urutan siklus pembacaan PORT adalah sebagai berikut: semua siklus yang dideskripsikan dianggap AT89C51 pada CPU meng-eksekusi perintah “MOVX A,@DPTR”, mula-mula *line* PORTX mendapat sinyal *low*, kemudian *line* RD juga mendapat sinyal *low* sesuai dengan perintah CPU, dengan kondisi seperti ini, data yang masuk pada port input IC 74HCT573 dapat dienkahkan ke *bus* data dan dibaca seketika oleh CPU, setelah CPU membaca data, CPU memerintahkan untuk memberikan sinyal *high* pada *line* RD. Setelah siklus pembacaan berakhir semua kondisi *bus* dan *line* kembali pada posisi logika normal.

3.1.7.3. PORT

PORT termasuk perangkat penting pada Programmable Controller, karena PORT berhubungan langsung dengan perangkat yang dikendalikan atau dibaca melalui sensor.



Gambar 3.12 Rangkaian elektronik PORT.

Siklus “output” rangkaian elektronik PORT pada saat mengirimkan data, yaitu pada saat *line* OUT mendapat sinyal *ground*, pin basis transistor terpicu untuk meneruskan arus yang berada pada pin emitter ke pin collector. Dengan begitu *jack-plug* JH1 mendapat tegangan *high*. Sedangkan jika *line* OUT mengambang, pin basis pada transistor tidak terpicu, sehingga *jack-plug* tidak mendapat tegangan *high*.

Pada siklus “input” rangkaian PORT menggunakan prinsip pembagi tegangan, yang dibentuk oleh R2 dan R3. Sebagai acuan tegangan yang diukur adalah tegangan 24 Volt pada *line* *high*, sedangkan sinyal *high* yang dapat dibaca oleh Latch Input berada pada tegangan 5 Volt.

3.2. Perancangan Perangkat Lunak (*Software*)

Selain *hardware* yang diperlukan pada perancangan dan pembuatan Programmable Controller, juga diperlukan *software* program pada *microcontroller* dan komputer agar dapat bekerja.

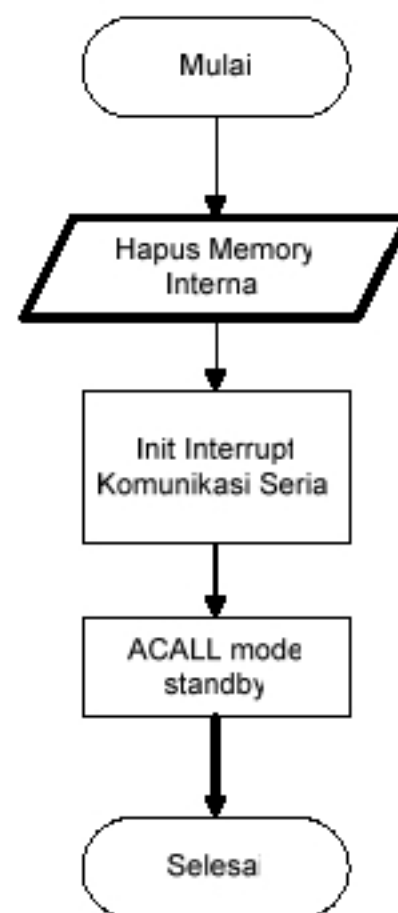
3.2.1. *Firmware* AT89C2051

AT89C2051 pada pembuatan Programmable Controller berfungsi untuk mengatur mode standby dan mode running. *Firmware* AT89C2051 menerima perintah dari komputer berupa kode ASCII melalui komunikasi serial, perintah-perintah tersebut adalah sebagai berikut:

- 'RST',0xD : mereset CPU.
- 'STB',0xD : meng-aktifkan mode standby.
- 'RUN',0xD : meng-aktifkan mode running.

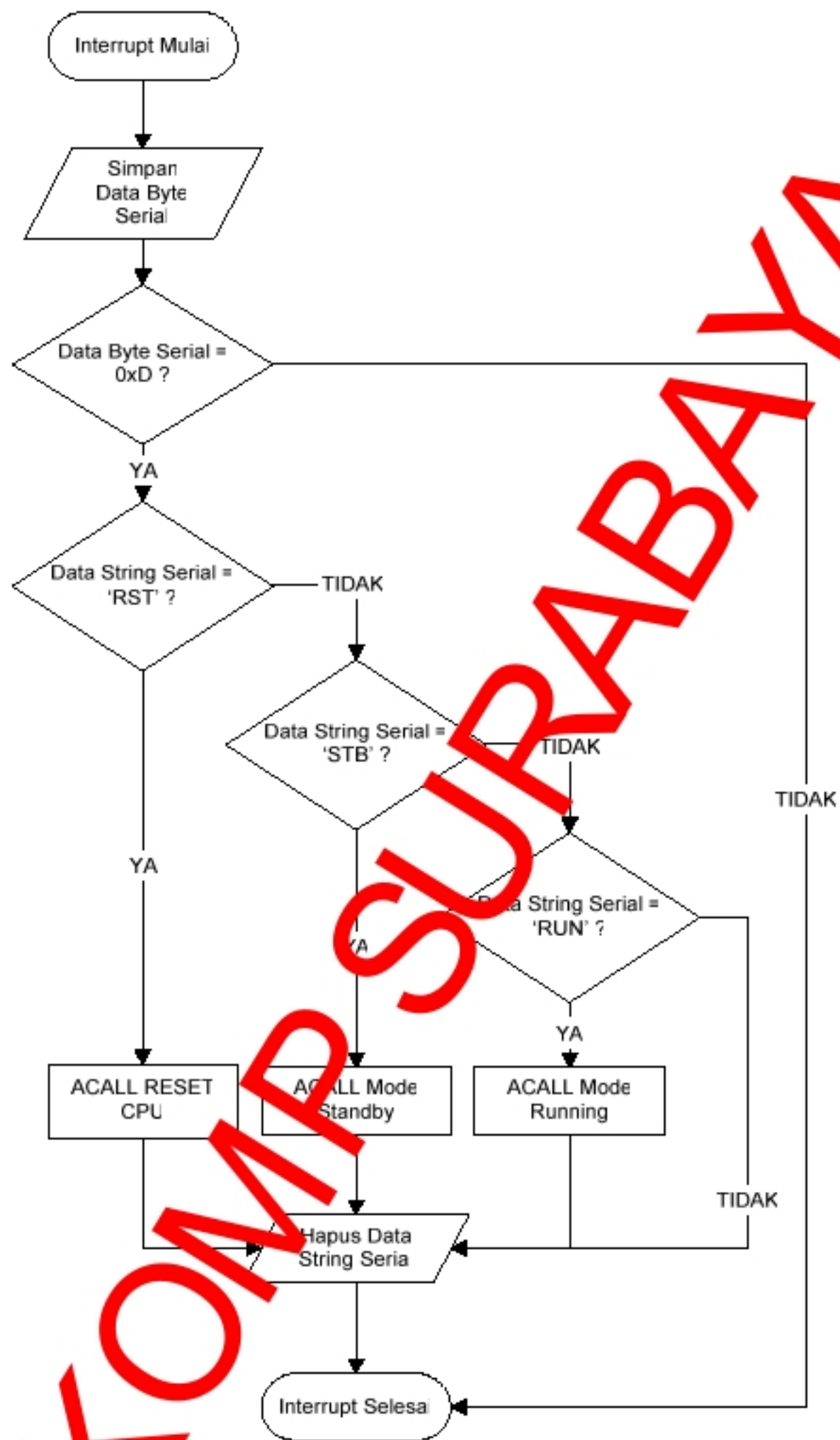
Listing program *firmware* AT89C2051 menggunakan bahasa Assembly MCS-51 dapat dilihat pada Lampiran 2.

3.2.1.1. Diagram Alur Program Utama *Firmware* AT89C2051



Gambar 3.13 Diagram Alur Program Utama *Firmware* AT89C2051.

3.2.1.2. Diagram Alur Program Interrupt komunikasi serial *Firmware*
AT89C2051



Gambar 3.14. Diagram Alur Program Interrupt Komunikasi Serial *Firmware*
AT89C2051.

3.2.2. Firmware AT89C51

Firmware AT89C51 pada CPU digunakan untuk meng-akses Memory Program Eksternal melalui komputer, dengan standart kode ASCII, pada mode standby. Perintah yang dapat dilakukan di antaranya adalah:

- “WP” + [alamat] + [jumlah] + [d1,d2,...,dn] + 0xD

OK

Perintah di atas berguna untuk menuliskan data ke Memory Program Eksternal, baris pertama adalah perintah yang dikirimkan oleh komputer, sedangkan baris kedua adalah respon dari CPU.

- ✓ alamat adalah alamat Memory Program Eksternal, dengan format 4 karakter bilangan hexa desimal.
- ✓ jumlah adalah jumlah data yang akan dikirimkan ke Memory Program Eksternal, dengan format 2 karakter bilangan hexa desimal.
- ✓ d1 adalah data ke-1, d2 adalah data ke-2, dan seterusnya.
- ✓ 0xD digunakan untuk mengakhiri perintah, menggunakan bilangan hexa desimal.

- “RP” + [alamat] + [jumlah] + 0xD

RP + [alamat] + [jumlah] + [d1,d2,...,dn] + 0xD

OK

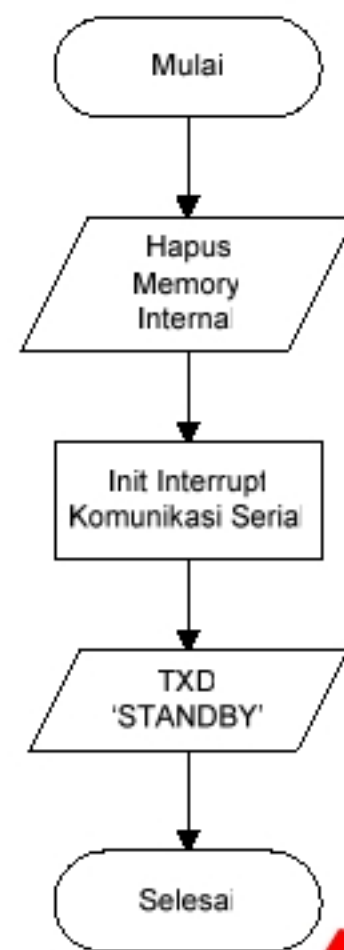
Perintah di atas digunakan untuk membaca data yang ada pada Memory Program Eksternal. Baris pertama adalah perintah yang dikirim oleh komputer, baris kedua dan ketiga adalah respon dari CPU.

- ✓ alamat adalah alamat Memory Program Eksternal, dengan format 4 karakter bilangan hexa desimal.
- ✓ jumlah adalah jumlah data yang akan dikirimkan ke Memory Program Eksternal, dengan format 2 karakter bilangan hexa desimal.
- ✓ d1 adalah data ke-1, d2 adalah data ke-2, dan seterusnya.
- ✓ 0xD digunakan untuk mengakhiri perintah, menggunakan bilangan hexa desimal.

Pemrograman yang digunakan pada *firmware* tersebut menggunakan bahasa Assembly MCS-51, untuk lebih jelasnya dapat dilihat pada Lampiran 3.

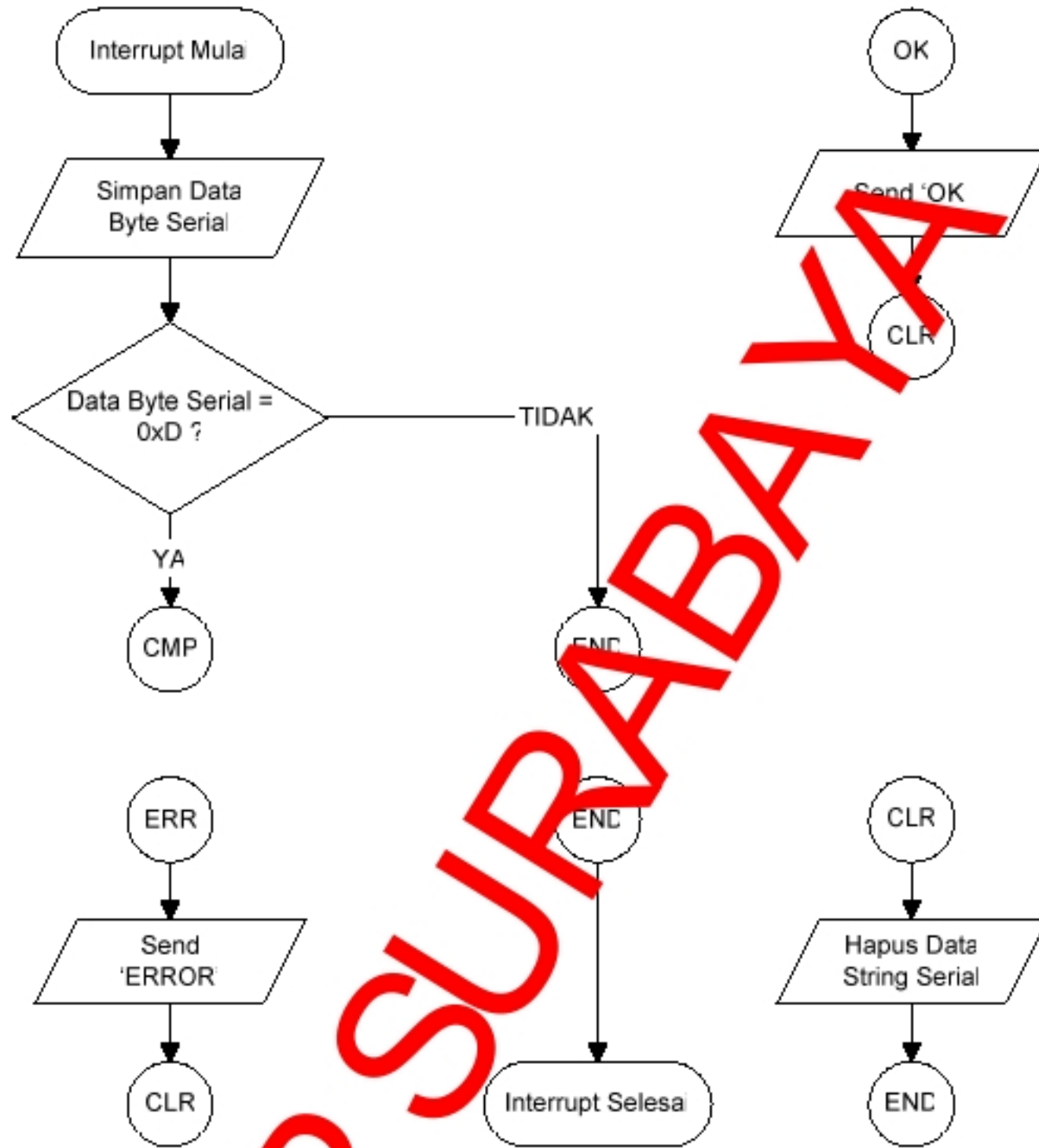
STIKOMP SURABAYA

3.2.2.1. Diagram Alur Program Utama *Firmware* AT89C51



Gambar 3.15 Diagram Alur Program Utama *Firmware* AT89C51.

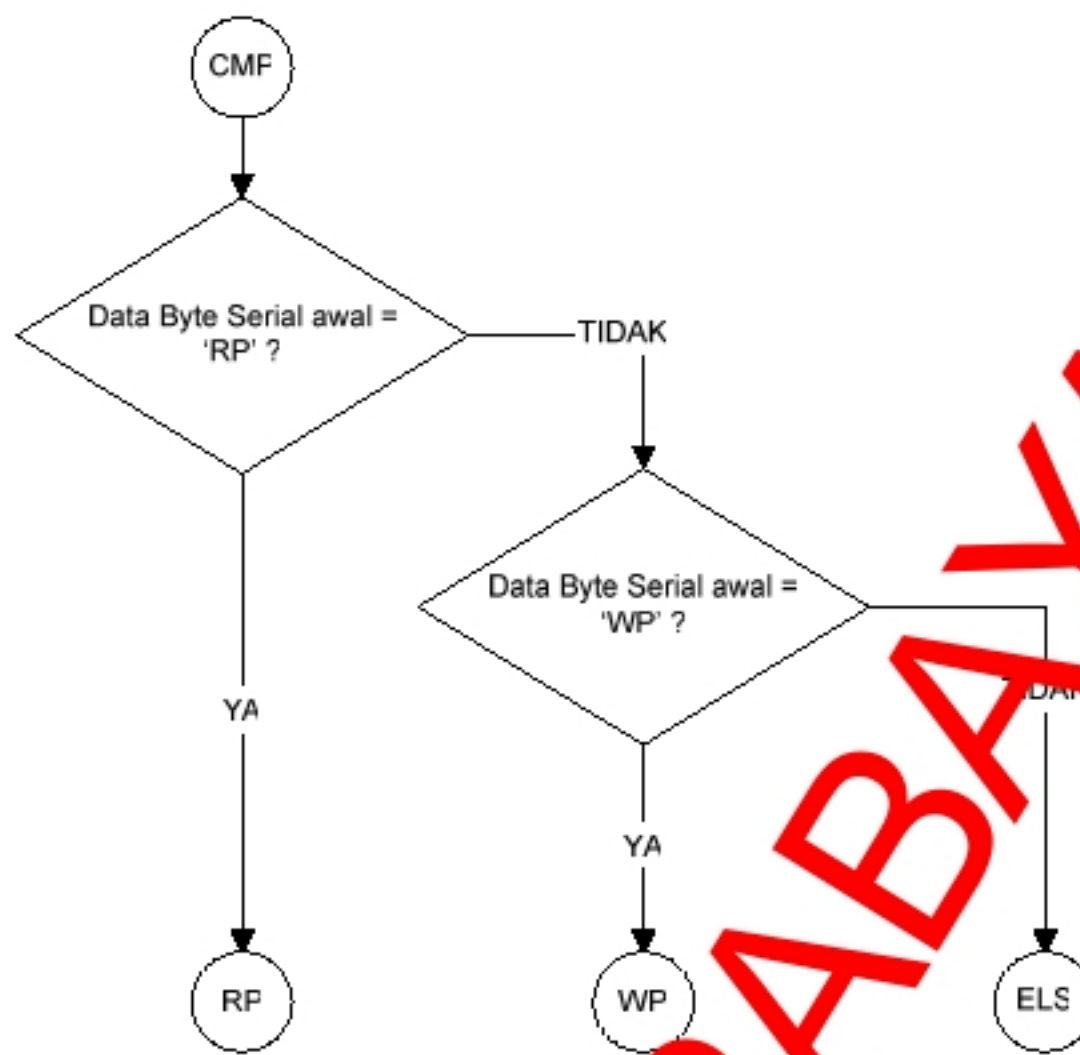
3.2.2.2. Diagram Alur Program Interrupt Komunikasi Serial *Firmware* AT89C51



Gambar 3.16 Diagram Alur Program Interrupt Komunikasi Serial

Firmware AT89C51.

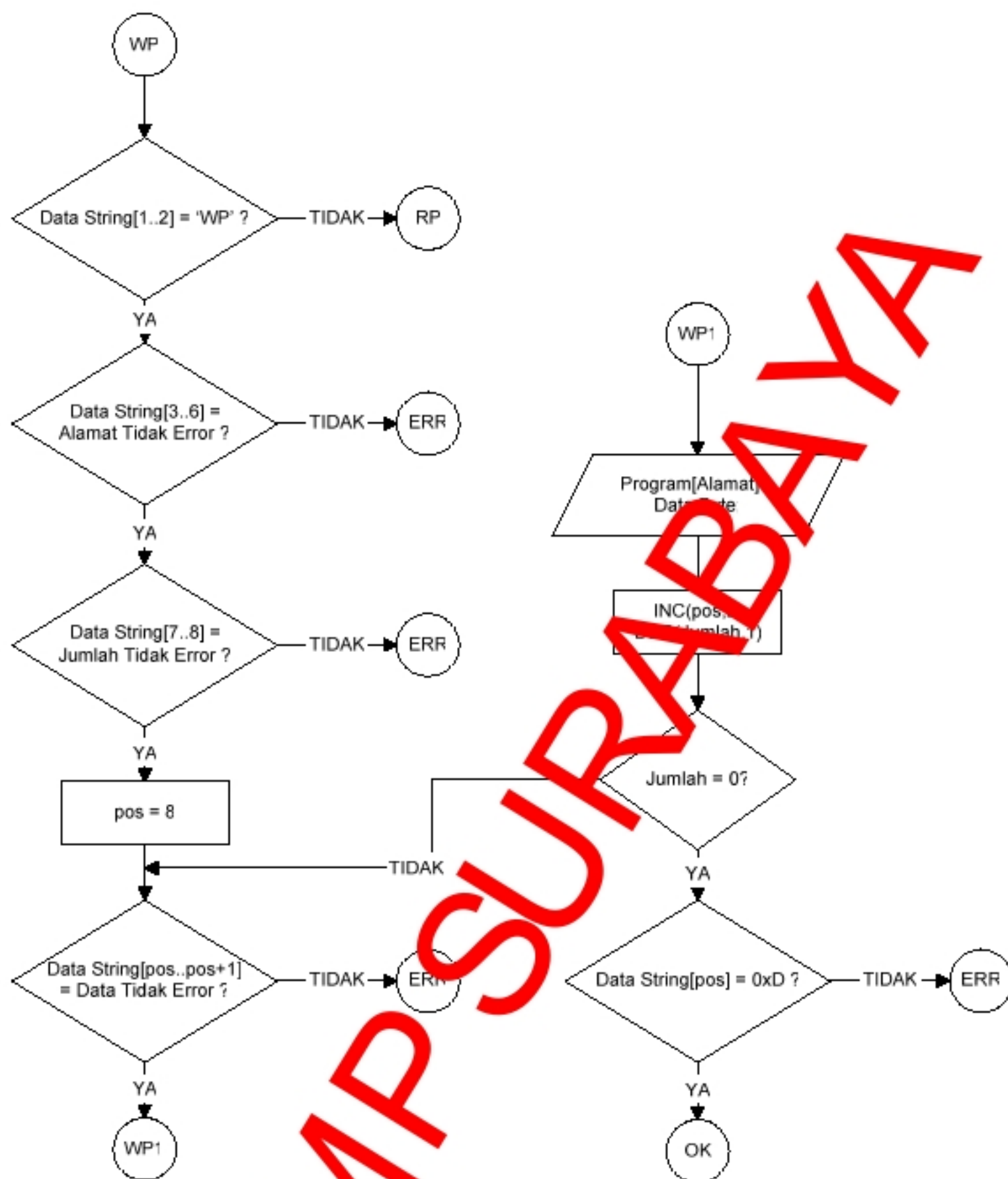
3.2.2.3. Diagram Alur Program Pembeding *Firmware* AT89C51



Gambar 3.17 Diagram Alur Program Pembeding *Firmware* AT89C51

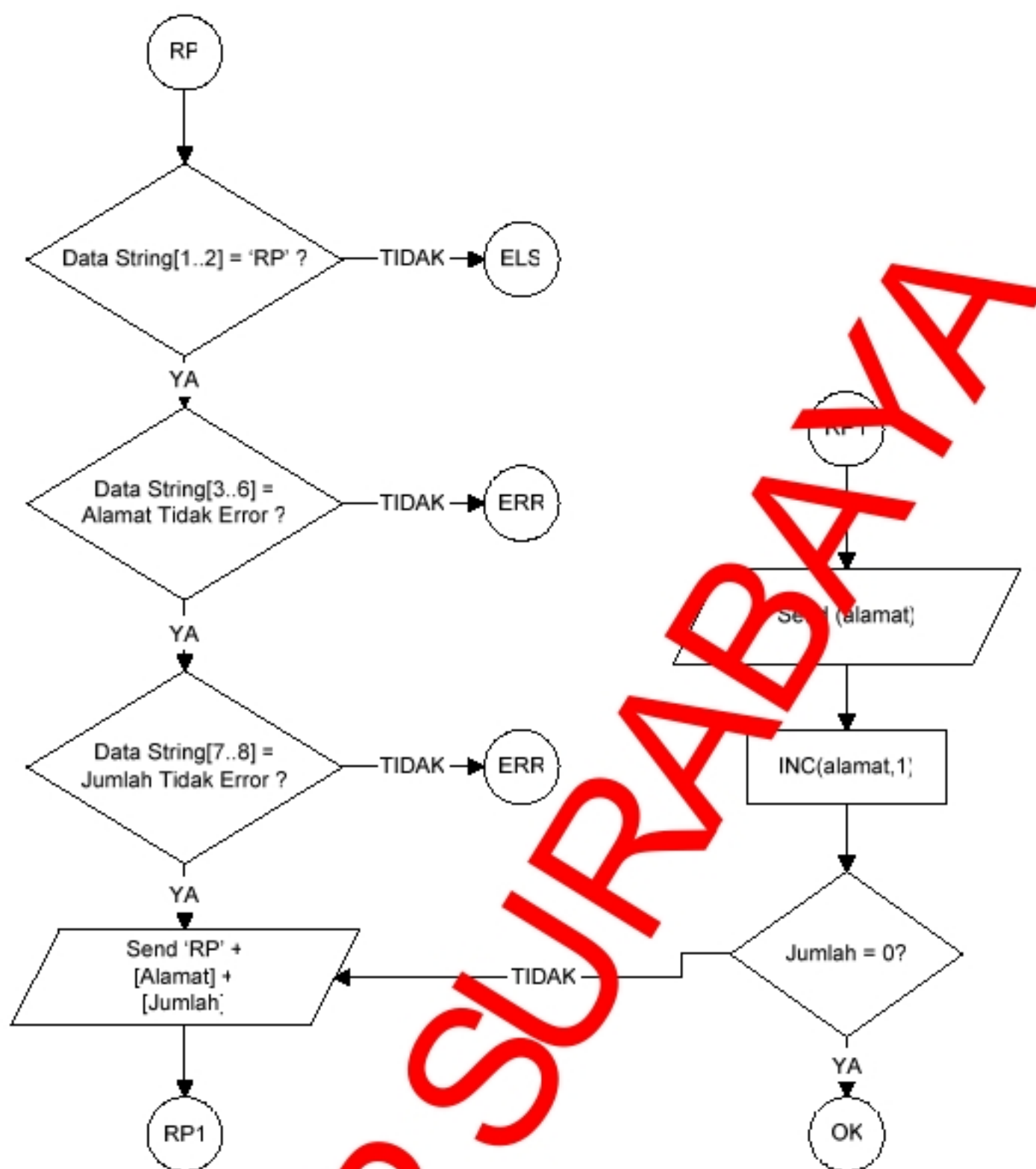
STIKOMMP SURABAYA

3.2.2.4. Diagram Alur Program Pemanding WP Firmware AT89C51

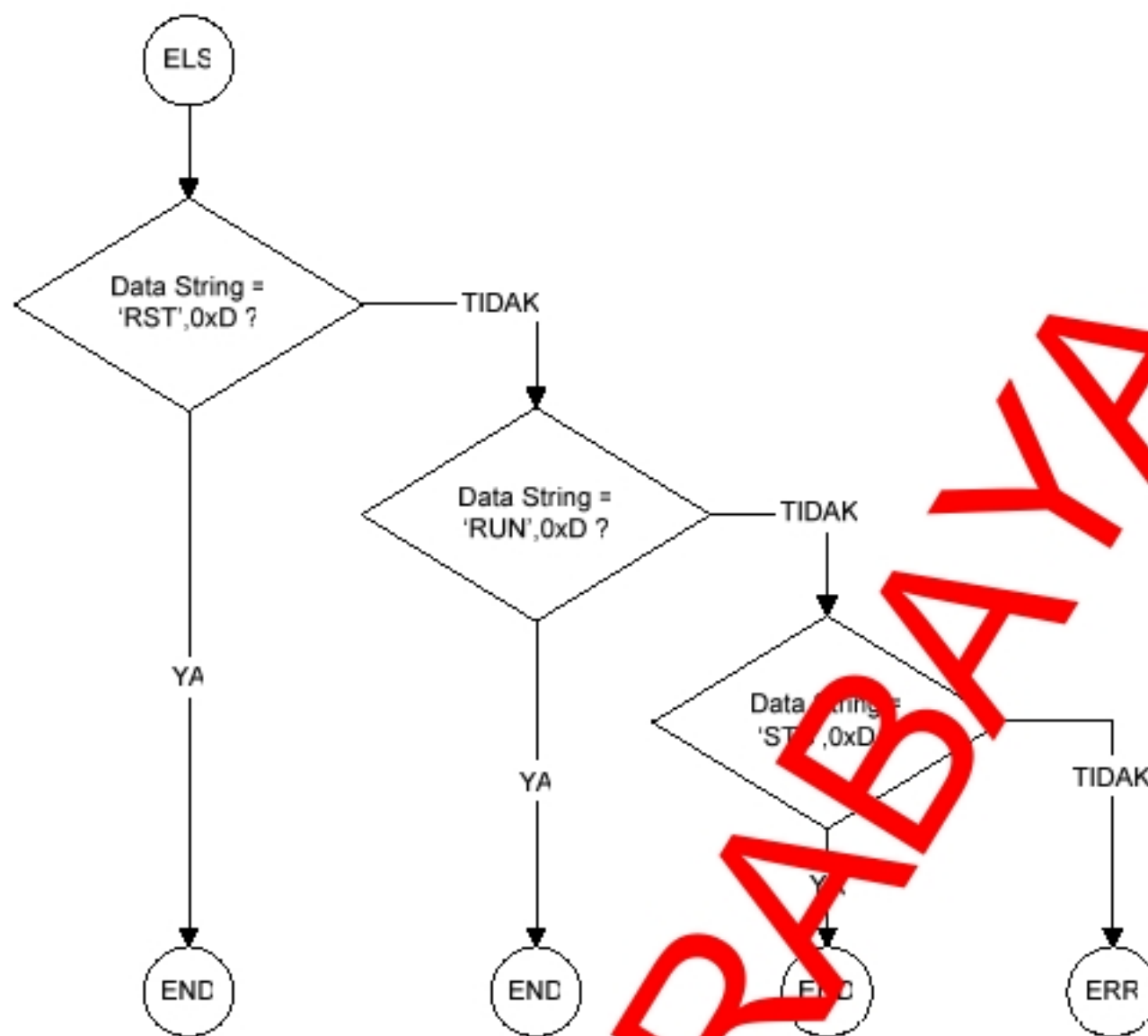


Gambar 3.18 Diagram Alur Program Pemanding WP Firmware AT89C51

3.2.2.5. Diagram Alur Program Pemanding RP *Firmware* AT89C51



Gambar 3.19 Diagram Alur Program Pemanding RP *Firmware* AT89C51

3.2.2.2.1. Diagram Alur Program Pembanding Lain *Firmware AT89C51*Gambar 3.20 Diagram Alur Program Pembanding Lain *Firmware AT89C51*.

3.2.3. Pemetaan Memory Programmable Controller

Agar penggunaan memory tidak kacau atau tercampur satu sama lain, memory yang sudah dirancang dipetakan posisinya untuk kegunaannya masing-masing. Memory yang dimaksud ada dua jenis pertama Memory Data Eksternal dan yang kedua Memory Internal.

Memory Internal digunakan untuk menyimpan data yang bersifat boolean dan byte yang kecil.

Tabel 3.1 Pemetaan Memory Internal

Alamat Memory Internal	Keterangan
08h	Posisi Serial
20h – 21h	Timer Status (16 bits)
22h – 25h	Counter Status (32 bits)
26h	Boolean System
30h – 4Fh	RAM Serial (32bytes)
60h – 7Fh	Stack Routine

Memory Data Eksternal berguna untuk menyimpan data yang dengan lebar 16 bit atau word, kecuali PORT hanya dapat di-akses secara 8 bit.

Tabel 3.2. Pemetaan Memory Data Eksternal / PORT.

Alamat Memory Data Eksternal / PORT	Keterangan
2000h – 201Fh	Timer Preselect
2020h – 203Fh	Timer Word
2100h – 213Fh	Counter Preselect
2140h – 217Fh	Counter Word
2200h – 221Fh	Flag
2300h – 237Fh	Register
2380h – 23FFh	Serial Communication Memory

Tabel 3.2. Pemetaan Memory Data Eksternal / PORT (lanjutan).

4000h	PORT 0
6000h	PORT 1
8000h	PORT 2
A000h	PORT 3
C000h	PORT 4

3.2.4. Pemrograman Timer CPU

Pemrograman Timer sangatlah penting untuk mendukung kerja Programmable Controller secara keseluruhan, karena timer yang digunakan lebih dari satu, maka pengaturan interrupt timer sangatlah diperlukan. Programmable Controller memiliki kepekaan timer sebesar $\frac{1}{100}$ detik, atau timer melakukan perhitungan dalam satu detik sebesar 100 Hz. Frekuensi clock osilator yang digunakan adalah 11.0592 Mhz atau sama dengan 11.059.200 Hz. Dengan menggunakan timer mode 1, maka didapat perhitungan sebagai berikut:

$$\text{Timer Clock} = \frac{\text{Osilator Clock}}{12} \quad (3.1)$$

$$\text{Timer Clock} = \frac{11059200}{12}$$

$$\text{Timer Clock} = 921600 \text{ Hz}$$

$$\text{Timer Counter} = 65536 - \left(\frac{\text{Timer Clock}}{\text{Interrupt Clock}} \right) \quad (3.2)$$

$$\text{Timer Counter} = 65536 - \left(\frac{921600}{100} \right) = 56320$$

Menurut perhitungan di atas, agar dapat menghasilkan interrupt setiap $\frac{1}{100}$ detik, maka timer counter harus diisi sebesar 56320 atau bila di-konversikan menjadi bilangan hexa desimal adalah 0xDC00.

Pada program interrupt timer, bila instruksi dilaksanakan secara maksimal akan diperoleh siklus intruksi sebesar 1058 siklus, semua perhitungan siklus instruksi dapat dilihat lebih detail pada Lampiran 4.

Perhitungan di bawah ini disimbolkan sebagai berikut:

- SSI = Sisa Siklus Instruksi
- SIIM = Siklus Instruksi Interrupt Maksimal
- APTC = Awal Posisi Timer Counter

$$SSI = 65536 - (APTC + SIIM) \quad (3.3)$$

$$SSI = 65536 - (56320 + 1058)$$

$$SSI = 65536 - (57436) = 8158$$

Jadi instruksi yang dapat dilaksanakan di antara *interrupt* adalah sebesar 8158 siklus instruksi, dengan sisa siklus instruksi sebesar 8158 siklus, maka program utama pada Programmable Controller masih dapat di-eksekusi.

3.2.5. Software PC Editor

Software PC Editor dibuat agar Programmable Controller dapat berkomunikasi dengan komputer. Software yang dibangun dengan Delphi 5.0 dapat melakukan transfer program dari komputer ke Programmable Controller.

Selain itu PC Editor juga berfungsi sebagai *editor* perintah-perintah program dari Programmable Controller. Perintah Program dari Programmable Controller yang

telah diketik dengan software ini dapat diketahui bahwa program tersebut mendapatkan perintah *error* atau tidak. Jika mendapatkan *error* maka program tidak dapat dikirimkan ke Programmable Controller. Sedangkan apabila tidak terjadi *error* berarti program tersebut dapat dikirimkan ke Programmable Controller. Sebelum pengiriman program ke Programmable Controller, program harus disimpan terlebih dahulu dengan ekstensi *.PCU, dan turunan program yang menyimpan daftar *allocation* disimpan secara software dengan ekstensi *.ALP.

Penambahan daftar *allocation* dapat dilakukan dengan menekan tombol “Option” pada menu, yang kemudian diteruskan dengan menekan tombol “Edit Allocation List” dan akan muncul *form* “Allocation List Editor”. Apabila penambahan tidak diperkenankan akan muncul pesan kesalahan (*error*).

Pada saat pengiriman yang perlu diperhatikan adalah pemilihan port komunikasi, port komunikasi yang disediakan: port COM1, COM2, COM3, dan COM4. Port komunikasi yang dipergunakan harus sesuai dengan port yang ditancapkan Programmable Controller.

Penjelasan penggunaan Program PE Editor dapat dilihat lebih *detail* pada Lampiran 13.