

## BAB III METODE PENELITIAN

Pembuatan simulasi otomatisasi untuk peralatan analisa aktivasi neutron cepat ini terbagi menjadi dua bagian yaitu:

1. Perangkat lunak
2. Perangkat keras

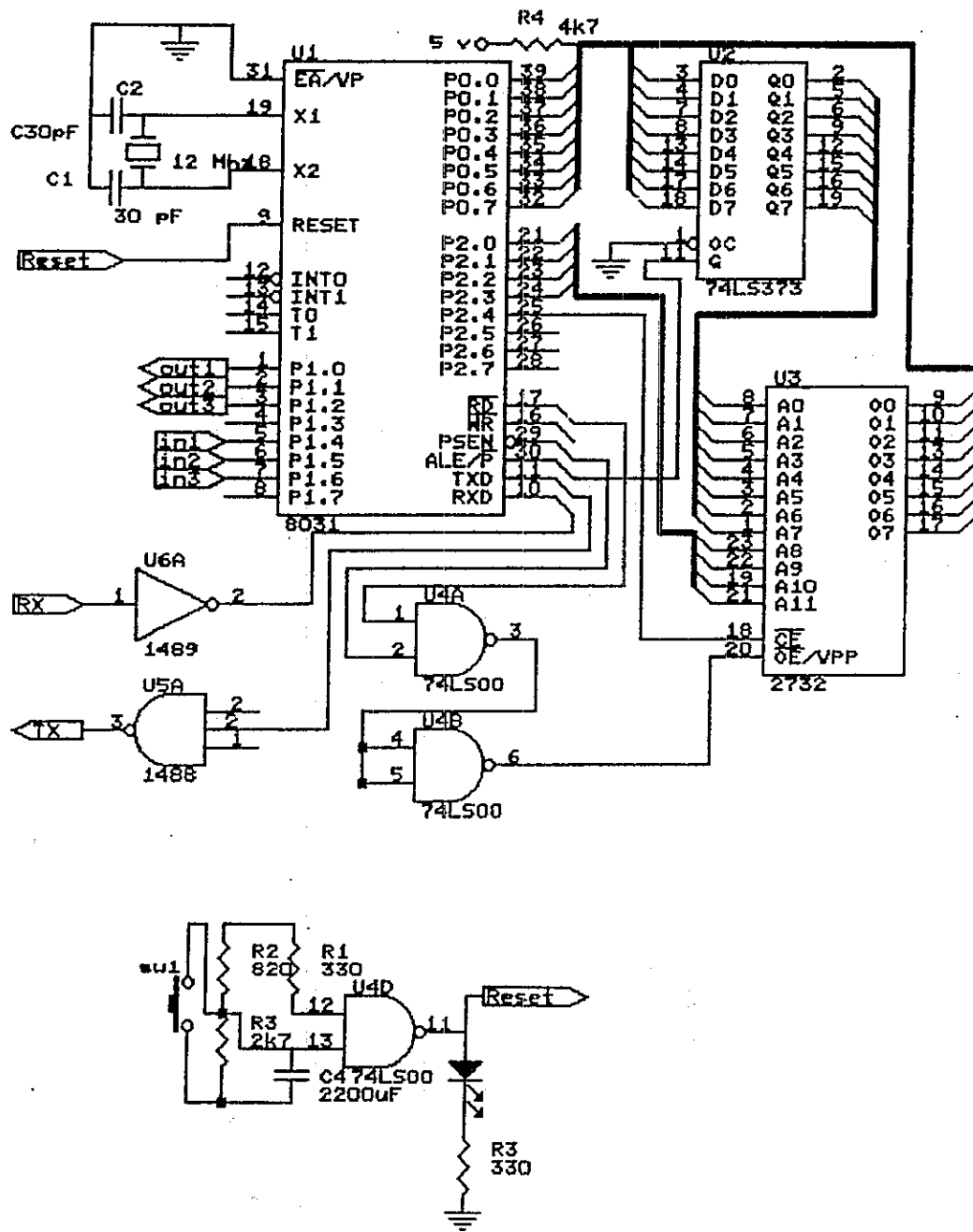
Masing-masing akan dijelaskan pada subbab berikut ini.

### 3.1. Perangkat Keras

Pada perancangan perangkat keras ini hubungan antara serpih serta fungsinya masing-masing telah dijelaskan pada bab terdahulu. Setelah mengetahui fungsi dari masing-masing serpih, serta hubungan serpih yang satu dengan serpih yang lainnya, maka dapat dirancang suatu sistem kontrol. Perancangan perangkat keras ini dikerjakan dalam beberapa tahap yaitu:

1. Membuat diagram perancangan dari mikrokontroler atau membuat skema dari rangkaian seperti yang terdapat pada gambar 3.1. Pembuatan gambar rangkaian ini ditulis dengan menggunakan ORCAD yaitu perangkat lunak yang biasa digunakan untuk menggambar rangkaian atau skema elektronik.
2. Membuat alur untuk PCB

Pembuatan alur atau layout PCB ini dibuat berdasarkan



Gambar 3.1. Skema lengkap board pengontrol (Malik dan Anistardi, 1997)

pada skema yang telah dibuat dari diagram perancangan mikrokontroler. Pada pembuatan layout PCB ini menggunakan Smartwork yaitu perangkat lunak yang biasa digunakan untuk membuat layout PCB.

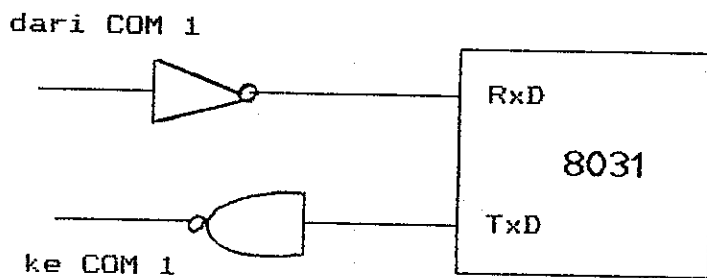
3. Membuat klise dari layout PCB
4. Memindahkan layout PCB dari klise ke pertinak
5. Melarutkan pertinaks yang telah ada gambar layout PCBnya.
6. Mengebor pertinaks pada tempat-tempat untuk memasukan komponen.
7. Memasang komponen.
8. Menguji kerja perangkat keras.

### 3.2. Perangkat Lunak

#### 3.2.1. Kontrol untuk input output data

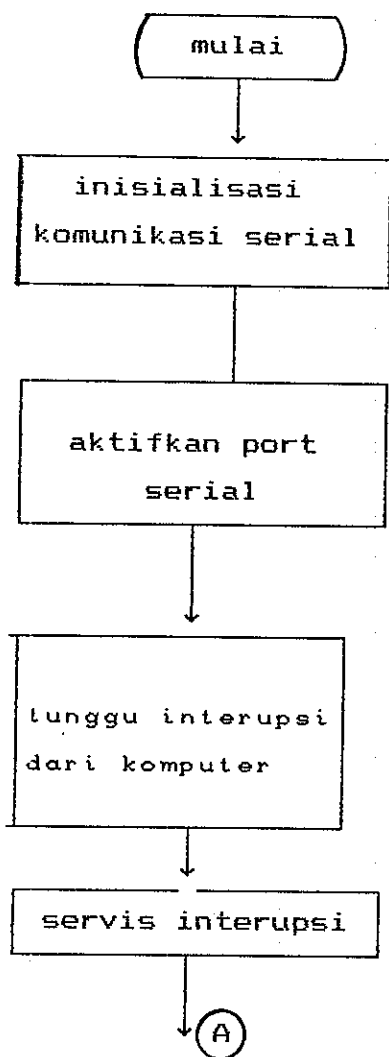
Seperti telah diuraikan didepan bahwa untuk dapat mengaktifkan kontrol harus dengan program yang diisikan pada eprom. Untuk itu terlebih dahulu harus menyusun program perangkat lunak untuk tujuan tertentu. Program ditulis dengan perangkat instruksi assembler Intel 8031.

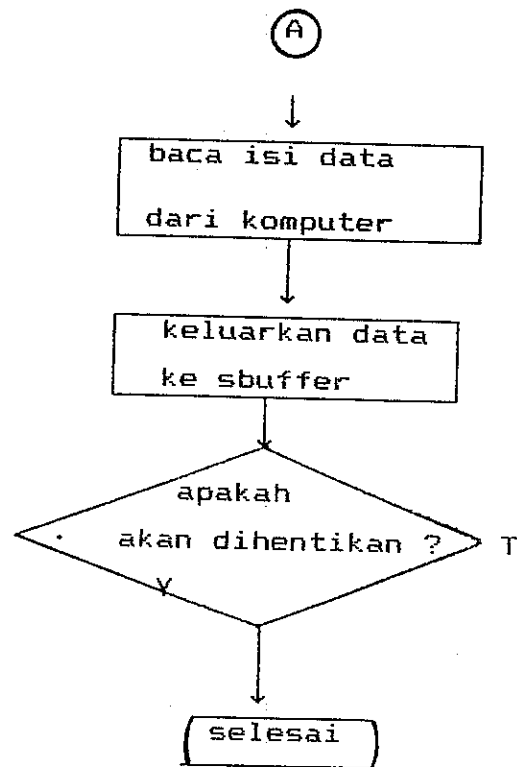
Untuk input-output data secara serial, hubungan serpih-serpihnya dtunjukkan pada gambar 3.2.



Gambar 3.2. Kontroler untuk I/O serial (Anonim, 1991)

Pertama-tama harus dibuat dahulu bagan alirnya.





Gambar 3.3. Bagan alir I/O serial.

Pada program I/O serial proses penginisialisasiannya melibatkan *Tcon*, *Scan*, *Pcon*, *Tmod* dan *TH1*, yang semuanya merupakan bagian dari *Special Function Register*.

a. Tabel 3.1 *Tmod* (Timer/Counter Mode Register)

| Gate  | C/T   | M1    | M0    | Gate  | C/T   | M1    | M0    |
|-------|-------|-------|-------|-------|-------|-------|-------|
| Tmod7 | Tmod6 | Tmod5 | Tmod4 | Tmod3 | Tmod2 | Tmod1 | Tmod0 |

1). *Gate*

Jika *high* (1), *timer/counter* aktif, hanya saat "*int*" *high* (1) dan "*Tr*" juga *High* (1).

Jika *low* (0), *timer/counter* aktif jika "*Tr*" *high*

(1).

2). C/T

High (1), *counter* aktif

Low (0), *timer* aktif

3). Tabel 3.2 M1 dan M0- Mode pengoperasian

| M1 | M0 | Operasi  |
|----|----|--|
| 0  | 0  | Timer/counter 8 bit dengan pewaktu   |
| 0  | 1  | Timer/counter 16 bit   |
| 1  | 0  | Timer/counter autoreload 8 bit   |
| 1  | 1  | Counter 0 - Timer/counter 8 bit (TLO)<br>Counter 1 - Timer/counter berhenti. |

b. Tabel 3.3 Tcon (Timer/Counter Control Register)

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| TF1   | TR1   | TF0   | TR0   | IE1   | IT1   | IE0   | IT0   |
| Tcon7 | Tcon6 | Tcon5 | Tcon4 | Tcon3 | Tcon2 | Tcon1 | Tcon0 |

1). TF (1/0) *Timer overflow flag*

set/clear oleh perangkat keras

2). TR (1/0) *Timer Run control bit*

Set/clear oleh perangkat lunak

Set (1) *Timer/counter On*

Set (0) *Timer/counter off*

3). IE (1/0) *Interupt (1/0) Edge Flag*

Set/clear oleh perangkat keras

Set (1) saat ada interupsi dari luar

Clear (0) saat tidak ada interupsi atau sedang memproses interupsi

4).IT (1/0) *Interrupt (1/0) type control bit*

Set/clear oleh perangkat lunak

Set (1) INT (0/1) aktif dan IE

berfungsi sebagai flag.

clear (0) saat INT (0/1) aktif, tapi IE

tidak berpengaruh.

c. Tabel 3.4 Scon (Serial Control Register)

| SM0   | SM1   | SM2   | REN   | TB8   | RB8   | TI    | RI    |
|-------|-------|-------|-------|-------|-------|-------|-------|
| Scon7 | scon6 | scon5 | scon4 | scon3 | scon2 | scon1 | scon0 |

1) SM (0/1) *Serial port operation mode*

set/clear oleh perangkat lunak

Tabel 3.5. Pemilihan mode port serial

| SM0 | SM1 | Mode | Operasi   |
|-----|-----|------|---|
| 0   | 0   | 0    | I/O sinkron menggunakan register geser TTL atau CMOS  |
| 0   | 1   | 1    | UART (Universal Asinkronous Receiver/Transmitter) interface dengan bentuk 8 bit dan laju transmisi variabel |
| 1   | 0   | 2    | UART interface dengan bentuk 9 bit dan laju transmisi tetap   |
| 1   | 1   | 3    | UART interface dengan bentuk 9 bit dan laju transmisi variabel  |

Set/clear oleh perangkat lunak

Set (1) semua bentuk bit dapat diterima

Clear (0) Untuk menerima, kemudian mengirim (transmit)

SM2 ini tidak berpengaruh pada mode 0 dan 1

### 3). REN Receiver Enable

Set/clear oleh perangkat lunak

Set (1) penerima serial aktif

lear (0) penerima serial non aktif



- 4). TBB *Transmite data bit 8*  
 Set/clear oleh perangkat keras  
 Set (1) saat penerimaan bit data ke 9
- 5). RBB *Receiver data bit 8*  
 Set/clear oleh perangkat keras  
 Set (1) saat penerimaan bit data ke 9
- 6). TI *Transmission Complete Interupt Flag*  
 Set oleh perangkat keras saat selesai mengirim data  
 Clear oleh perangkat lunak
- 7). RI *Reception Complete Interupt Flag*  
 Set oleh perangkat keras saat menerima data  
 Clear oleh perangkat lunak

d. Tabel 3.6 Pcon (*Power Control Register*)

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| Smod  | -     | -     | -     | GF1   | GF0   | PD    | IDL   |
| Pcon7 | Pcon6 | Pcon5 | Pcon4 | Pcon3 | Pcon2 | Pcon1 | Pcon0 |

- 1). Smod *Serial Mode*  
 Set/clear oleh perangkat lunak  
 Set (1) timer 1 tidak dipakai untuk pembangkit laju bit  
 Clear (0) timer1 dipakai untuk pembangkit laju bit
- 2). - *Tidak dipakai*

- 3). - Tidak dipakai
- 4). - Tidak dipakai
- 5). GF1 *General Purpose Flag Bit 1*  
Set/clear oleh perangkat keras
- 6). GF0 *General purpose Flag Bit 0*  
Set/clear oleh perangkat keras
- 7). PD *Power Down Bit*  
Set/clear oleh perangkat lunak  
Set (1) untuk mengaktifkan power
- 8).IDL *Idle Mode Bit*  
Set/clear oleh perangkat lunak  
Set (1) untuk mengaktifkan operasi idle

e. TH1 (*Timer/Counter 1-High*)

Saat mode 1 atau 3, digunakan (Scon ), Laju transmisi dapat divariabel mulai dari 122 bit/detik sampai dengan 31250 bit/detik. Pada frekuensi osilator 12 Mhz, dengan cara memasukan data dari 00H sampai dengan FFH. Saat TH1 00H, maka (Malik dan Anistardi,1997):

$$\frac{12 \text{ Mhz}}{16 \cdot (256 - 00H) \cdot 12} = 244 \text{ bit/detik}$$

12 Mhz = Frekuensi osilator

32 = Banyaknya pulsa yang dibutuhkan untuk membangkitkan laju transmisi

12 = Banyaknya pulsa yang dibutuhkan untuk

memindahkan data dari/ke serial buffer

Dan saat TH1 = FFH maka :

$$\frac{12 \text{ Mhz}}{16 \cdot (256 - \text{FFH}) \cdot 12} = 62.500 \text{ bit/detik}$$

Sedangkan untuk mode 0, laju transmisi saat frekuensi osilator 12 Mhz adalah sebesar :

$$\frac{12 \text{ Mhz}}{12^*} = 1.000.000 \text{ bit/detik}$$

$12^*$  = banyaknya pulsa yang dibutuhkan untuk membangkitkan laju transmisi.

f. EA, ES, PS (*Enable All, Enable Serial Port, Priority Serial Port*)

- 1). EA     *Enable all*  
           Set/clear oleh perangkat lunak  
           Set (1) mengaktifkan semua bit pengontrol interupsi  
           Clear (0) non aktif
- 2). ES     *Enable Serial Port*  
           Set/clear oleh perangkat lunak  
           Set (1) mengaktifkan interupsi dari TI atau RI  
           Clear (0) non aktif
- 3). PS     *Priority Serial Port*

Set/clear oleh perangkat lunak

Set (1) mengaktifkan interupsi tinggi  
(utama) untuk serial port

Clear (0) prioritas rendah (terakhir)

Sedangkan program penerimaan dan pengiriman data ke/dari komputer mempergunakan bahasa Pascal. Programnya ada pada lampiran.

Pada program pascal untuk pengiriman data secara serial lewat COM1 yang perlu diperhatikan adalah urutan program sebagai berikut (Busono,1991):

1. Inisialisasi LCR (*Line Control Register*)

Tabel 3.7.Fungsi LCR (*Line Control Register*)

| Bit | Fungsi   |
|-----|--|
| 0   | Jumlah data bit 00= 5 bit 01= 6 bit  |
| 1   | 10= 7 bit 11= 8 bit  |
| 2   | 0: satu stop bit, 1; bila data 5 bit stopbit = 1,<br>bila data 6, 7, 8 bit stop bit =2   |
| 3   | Parity :0= tidak ada parity,1= ada parity  |
| 4   | Parity genap, bila ada parity: 0= parity<br>ganjil,jumlah logika selalu ganjil<br>1 = parity genap, jumlah logika selalu genap |
| 5   | 1 : bila ada parity  |
| 6   | 1 : serial output selalu berlogika 0   |
| 7   | DLAB ( Divisor Latch Access Bit)   |

2. Mengeset laju transmisi (*baud rate*) pada (*Divisor Latch LSB*) dan DLM (*Divisor Latch MSB*)

Laju transmisi ini dapat divariasikan sedemikian rupa mengikuti rumus sebagai berikut :

$$\text{Laju transmisi} = \frac{1,843 \text{ Mhz}}{16 \cdot (\text{DLAB})}$$

$$\begin{aligned} \text{Untuk laju transmisi 4800 Bps, maka DLAB} &= \frac{1,84 \text{ Mhz}}{16 \cdot 4800 \text{ Bps}} \\ &= 24 \text{ atau } \$18 \end{aligned}$$

Alasan dipilihnya laju transmisi 4800 Bps, adalah karena laju transmisi tersebut yang amat cocok dengan laju transmisi dari board kontrol dan secara teori maupun praktek diketahui bahwa perbedaan dari laju transmisi komputer dan kontroler yang cukup besar, akan menyebabkan terjadinya kesalahan pada saat pengiriman maupun penerimaan data.

3. DLAB harus 0 (nol) untuk pengiriman maupun penerimaan data, caranya adalah dengan membuat bit ke-7 dari LCR ( $\$3FC$ ) = 0.

`Port[$3FB] = Port[$3FB] and $7F;`

7F H = 0111 1111 B, maka bit ke-7 dari alamat \$3FB pasti = 0, karena perintah *and* hanya 1 jika keduanya 1.

4. Mengisikan Out2 = 1 di MCR (*Modem Control Register*)

Tabel 3.8. Fungsi MCR pada alamat \$3FC

| Bit | Fungsi                              |
|-----|-------------------------------------|
| 0   | DTR (Data Terminal Ready)= 1        |
| 1   | RTS(Request To Send) = 1            |
| 2   | Out1                                |
| 3   | Out2 =1 untuk menyalurkan interupsi |
| 4   | Loop : 1 uji sendiri                |
| 5-7 | Selalu 0 (nol)                      |

5. Mengisikan data ke penyangga transmisi TX Buffer) di alamat \$3FB port[\$3f8]= data;

Sedangkan untuk program penerimaan interupsi, ada beberapa hal yang perlu diperhatikan. Urutan programnya adalah sebagai berikut:

1. Menyiapkan prosedur interupsi dan mengganti vektor interupsi dengan prosedur interupsi tersebut.

```
Getintvec(%c,vektor); %C = nomer interupsi untuk IRQ4
                        (Interrupt Request 4) yaitu pada
                        Com1 komputer.
```

```
Vektor = pointer; untuk menyimpan
                        vektor interupsi yang lama.
```

```
Setintvec(%C,@cetak); Cetak = prosedur interupsi yang
                        baru.
```

2. Mempersiapkan prioritas interupsi pada IER (*Interrupt Enable Register*)

Tabel 3.9. Fungsi IER \$3F9

| Bit | Fungsi             |
|-----|--------------------|
| 0   | 1 = prioritas ke-2 |
| 1   | 1 = prioritas ke 1 |
| 2   | 1 = prioritas ke-3 |
| 3   | 1 = prioritas ke-0 |
| 4-7 | selalu 0 (nol)     |

3. Sama seperti langkah 4 pada program pengiriman data, yaitu mengisikan `Out2 = 1` di MCR.
4. Mendefinisikan nomor IRQ di IMR (*Interrupt Memory Register*) `port[$21] = port[$21] and $EF;`
5. Sama seperti langkah 1 sampai dengan 3 pada program pengiriman data.
6. Memberikan sinyal EOI (*End Of Interupsi*) ke OCW (*Operation Control Word*).  
`Port[$20 = $20;`
7. Setelah selesai, vektor interupsi yang lama harus dikembalikan.  
`Setintvec($C,vektor);`

Cara pengujiannya adalah sebagai berikut:

1. Hubungkan I/O serial kontroler dengan Com 1 pada komputer.
2. Jalankan mikrokontroler dan komputer.
3. Lakukan pemasukan data dan amati perubahan pada komputer dan mikrokontroler.

### 3.2.2. Kontroler Sebagai Counter

Mikrokontroler 8031 memiliki dua buah timer/couter yang dapat diatur melalui perangkat lunak. Apabila timer/counter diaktifkan pada frekuensi kerja mikrokontroler 12 Mhz, timer/counter akan melakukan perhitungan waktu sekali setiap satu detik secara independen tidak tergantung pada pelaksanaan suatu instruksi. Satu siklus pencacahan waktu berpadanan dengan satu siklus pelaksanaan instruksi, sedangkan satu siklus diselenggarakan dalam waktu satu mikrodetik.

Pada perangkat lunaknya, program untuk pewaktu cara menghitungnya adalah sebagai berikut.

- Pada instruksi `Mov Rn,#data` mikrokontroler membutuhkan 1 cycle pulsa osilator untuk mengerjakannya, yang berarti 12 periode osilator.
- Sedangkan untuk instruksi `DJNZ Rn,Rel` dibutuhkan 2 cycle = 24 periode osilator.

Karena osilator mikrkontroler 8031 memiliki frekuensi 12 Mhz maka :

$$1 \text{ periode} = \frac{1}{12 \cdot 10^6} = 8,333 \cdot 10^{-8} \text{ detik}$$

Sehingga untuk instruksi `Mov Rn,#data` dibutuhkan waktu sebesar :  $12 \times 8,333 \cdot 10^{-8} \text{ detik} = 1 \cdot 10^{-6} \text{ detik}$  dan untuk instruksi `DJNZ Rn,rel` dibutuhkan waktu sebesar  $24 \times 8,333 \cdot 10^{-8} \text{ detik} = 2 \cdot 10^{-6} \text{ detik}$ .

Contoh program pewaktu 1 detik setelah diurutkan

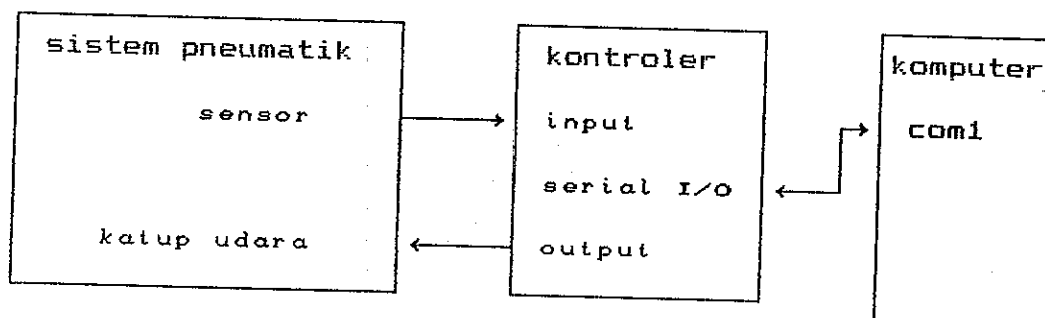


adalah sebagai berikut :

$$\begin{aligned}
 \text{mov R5, \#08H} &: 1 \times 1.10^{-6} \text{ detik} &= 1.10^{-6} \text{ detik} \\
 \text{lima mov R4, \#F5H} &: 1 \times 8 \times 1.10^{-6} \text{ detik} &= 8.10^{-6} \text{ detik} \\
 \text{empat mov R3, \#FEH} &: 1 \times 8 \times 245 \times 1.10^{-6} \text{ detik} &= 1960.10^{-6} \text{ detik} \\
 \text{tiga djnz R3, tiga} &: 2 \times 8 \times 245 \times 254 \times 1.10^{-6} \text{ detik} &= 995680.10^{-6} \text{ dt} \\
 \text{djnz R4, empat} &: 2 \times 8 \times 245 \times 10^{-6} \text{ detik} &= 3920.10^{-6} \text{ detik} \\
 \text{djnz R5, lima} &: 2 \times 8 \times 10^{-6} \text{ detik} &= 16.10^{-6} \text{ detik} \\
 & & \text{-----} + \\
 & & = 1001585.10^{-6} \\
 & & = 1,001585 \text{ detik} \\
 & & \sim 1 \text{ detik}
 \end{aligned}$$

### 3.3. Kontroler Untuk Simulasi Otomatisasi Pada Peralatan Analisa Aktivasi Neutron Cepat.

Pada penggunaan kontroler untuk simulasi otomatisasi pada peralatan analisa aktivasi neutron cepat, perlu diketahui terlebih dahulu hubungan sistem otomatis dengan peralatan analisa aktivasi neutron cepat. Peralatan yang dipergunakan digambarkan pada gambar 3.4.



Gambar 3.4. Diagram blok sistem otomatis Untuk pengaturan input/output pada kontroler, yang

digunakan adalah port 1. Untuk lebih terinci diperlihatkan pada tabel berikut.

Tabel 3.10. Port 1 sebagai input serial.

| Pin   | Input           | Keadaan |        | S  |
|-------|-----------------|---------|--------|----|
|       |                 | high(1) | low(0) |    |
| pi.4  | sensor pengirim | ada     | tidak  | S1 |
| pi.5. | sensor aktivasi | ada     | tidak  | S2 |
| pi.6  | sensor analisa  | ada     | tidak  | S3 |

Tabel 3.11. Port 1 sebagai output serial

| Pin  | output           | keadaan |        | LED |
|------|------------------|---------|--------|-----|
|      |                  | high(1) | low(0) |     |
| pi.0 | katup pengirim   | buka    | tutup  | L1  |
| pi.1 | katup pengembali | buka    | tutup  | L2  |
| pi.2 | katup pemutar    | buka    | tutup  | L3  |
| pi.3 | katup penghenti  | buka    | tutup  | L4  |

### 3.3.1. Diagram Alir Sistem Otomatis

Seperti yang telah diuraikan didepan pada bab terdahulu, bahwa untuk mengoperasikan kontroler harus dilakukan dengan program perangkat lunaknya. Untuk itu terlebih dahulu dibuat diagram alir dari perangkat lunaknya yaitu diagram alir pada komputer dan diagram alir pada mikrokontroler. Diagram tersebut diperlihatkan pada

gambar lampiran.

Keterangan dari diagram alir dari sistem otomatis tersebut adalah :

- Generator neutron siap ?, sampel siap ?, pneumatik siap ? menanyakan apakah ketiga peralatan tersebut sudah siap atau belum. Pada penulisan ini program yang memuat hal tersebut belum ada. Jadi ketiga-tiganya dianggap sudah siap.
- Kirim ke mikrokontroler waktu aktivasi, siklik/tidak, berapa kali ?, ini meminta pemakai untuk memasukan lamanya waktu aktivasi yang diinginkan, operasi siklik atau tidak dan jika siklik berapakah sikliknya. Program yang memuat hal ini terdapat pada lampiran I dengan nama program kirim yang ditulis dalam bahasa pascal.
- Inisialisasi yaitu penginisialisasian mikrokontroler untuk komunikasi serial. Program yang mengatur hal ini adalah inout.asm yang ditulis dengan perangkat instruksi assembler intel 8031 yang terdapat pada lampiran II.
- Tunggu kiriman data, yaitu mikrokontroler menerima data waktu aktivasi, siklik/tidak serta jumlah siklik.
- kirim tanda terima, yaitu mikrokontroler mengirimkan tanda untuk memberitahukan ke komputer bahwa data waktu aktivasi, operasi siklik/tidak serta jumlah siklik yang dikirimkan komputer telah diterima oleh mikrokontroler.
- tunggu perintah kerjakan, yaitu mikrokontroler menunggu perintah dari komputer untuk mulai operasi. Program di

- komputer yang mengatur hal ini adalah program mulai mengerjakan proses yang terdapat pada lampiran III.
- buka katup penghenti, yaitu penghenti dibuka untuk memberi jalan sampel dapat lewat.
  - Buka katup pengirim, yaitu katup pengirim dibuka supaya udara dalam pipa mengirim sampel ke ruang aktivasi.
  - Hitung waktu transfer, yaitu mikrokontroler menghitung lamanya perjalanan sampel dari terminal pengirim ke terminal aktivasi. Data ini dikirimkan ke komputer dan di atur oleh program terima\_03 dan program kirim\_03\_terima\_waktu\_transfer yang terdapat pada lampiran IV dan V. Operasi siklik diatur oleh program kirim\_03 dan program kirim\_04\_terima\_waktu\_transfer yang terdapat pada lampiran VI dan VII.
  - Tutup katup pengirim, katup pengirim ditutup sehingga sampel berhenti di terminal aktivasi.
  - Buka katup pemutar, katup pemutar terbuka dan meniupkan udara ke sampel sehingga sampel berputar.
  - Aktivasi, yaitu sampel ditembaki dengan berkas neutron di terminal aktivasi yang lamanya sudah ditentukan oleh pemakai.
  - Buka katup pengembali, yaitu katup pengembali terbuka dan mengirim sampel ke terminal analisa. Lamanya waktu analisa sudah ditentukan oleh pemakai. Program yang mengatur hal ini adalah yang terdapat pada autosquence.
  - Tutup katup penghenti, katup ini berfungsi untuk menghentikan jalannya sampel di terminal analisa.

- Hitung waktu transfer, yaitu menghitung lamanya waktu yang diperlukan oleh sampel dari terminal aktivasi ke terminal analisa. Data ini dikirimkan ke komputer dan ditampilkan di layar monitor. Program yang mengatur hal ini terdapat pada lampiran IV, V, VI dan VII.
- Kirim ke komputer analisa, yaitu sampel yang sudah bersifat radioaktif dianalisa oleh komputer.
- simpan data, yaitu komputer menyimpan data yang diperoleh dari analisa sampel tadi.
- kirim tanda selesai, komputer memberitahukan mikrokontroler bahwa analisa sudah selesai.
- Kirim sampel ke pemuat, yaitu mengirim sampel ke tempat asal pengiriman atau ke terminal pengirim.
- Ingin mengaktivasi lagi ?, menanyakan kepada pemakai apakah akan mengaktivasi sampel lagi. Jika ingin mengaktivasi lagi maka harus dijawab "ya", jika tidak ingin mengaktivasi lagi maka dijawab "tidak" .

Dari diagram alir tersebut terlihat bahwa sistem otomatis tersebut terbagi menjadi tiga bagian yaitu :

1. Program yang mengatur parameter waktu. Yaitu waktu aktivasi, waktu analisa serta pemilihan operasi siklik atau tidak serta jumlah siklik.
2. Program yang mengatur katup-katup udara.
3. Program pengukur waktu transfer.

Pada penelitian ini untuk memudahkan pembuatan program, maka program dikerjakan secara bertahap. Program secara lengkap terdapat pada lampiran. Program yang telah dibuat lalu dikodekan dengan software x8051. Yang dimasukan ke dalam eprom adalah bilangan heksadesimal hasil dari pengkodean oleh x8051. Pemasukan ke eprom dengan menggunakan eprom programmer.

### 3.3.1. Autosequence

Untuk menjalankan program yang telah dibuat, maka dibuat suatu program untuk mengurutkan bagian-bagian program dari program di komputer maupu program di mikrokontroler. Program ini nanti juga yang akan menjalankan MCA, unit yang menganalisa sampel. Adapun programnya adalah sebagai berikut :

```
let a=" a:\kirim.exe
let b=" a:\start.exe
let c=" a:\terima03.exe
let d=" a:\krm03trm.exe
let e=" a:\kirim03.exe
let f=" a:\krm04trm.exe
:lagi
\util\run a
\acquire\erase
ask g,"MASUKAN WAKTU CACAH", "300","5"
\acquire\preset\live g
ask h,"siklik ya=1/tidak=0","0","5"
test\ne h,siklik
```

```
\util\run b
\util\run c
\acquire\on/off\on
\wait\adc
\move\data "coba"," ","b:data1"
\acquire\erase
\util\run d
ask i,"ingin mengukur lagi? ya=1 tidak=0","0","5"
test\ne i,lagi
goto selesai

:siklik
let j="0"
ask k,"berapa kali siklik ?","2"
\util\run b
:ulang
increment j
\util\run c
\acquire\on/off\on
\wait\adc
\move\data "coba"," ","b:data~j"
\acquire\erase
\util\run e
compare\ne j, k,ulang
\util\run f
ask l,"ingin mengukur lagi ? ya=1 tidak=0","0","5"
test\ne l,lagi

:selesai
\exit
halt
```

### 3.3.2. Pengujian Sistem Otomatis

Pengujian sistem otomatis yang telah dibuat dilaksanakan sebagai berikut:

1. Menghubungkan port output dengan LED, Yaitu pada p1.0, p1.1, p1.2 dan p1.3.
2. Menghubungkan port input dengan + 5 Volt dengan melalui sebuah saklar. Yaitu pada p1.4, p1.5, p1.6.
3. Menghubungkan RS 232 dengan Com1 pada komputer.
4. Menghidupkan mikrokontroler dan komputer.
5. Menjalankan autosequence.
6. Mengisi pesan yang ditampilkan pada layar monitor.
7. Mengamati perubahan pada layar monitor, pada mikrokontroler dan merubah-ubah posisi saklar.