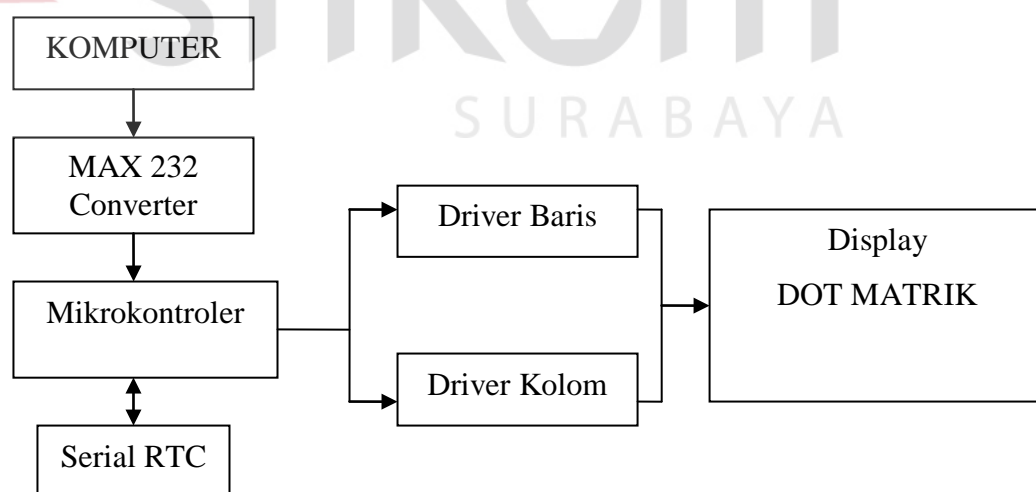


### BAB III

#### METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian tugas akhir ini adalah studi kepustakaan dan melakukan percobaan. Dengan ini penulis berusaha untuk mengumpulkan data, informasi serta materi–materi dasar yang bersifat teoritis yang sesuai dengan permasalahan. Hal tersebut diperoleh dari buku, materi kuliah, literatur melalui *browsing* di internet dan melakukan berbagai percobaan. Dari data-data yang diperoleh penulis berusaha menerapkannya untuk menyelesaikan permasalahan yang ada dalam penelitian ini.

Pada sub bab ini akan membahas tentang perancangan sistem secara keseluruhan dari penelitian ini, yaitu tentang perancangan perangkat keras dan perangkat lunak. Keseluruhan sistem pada penelitian ini sesuai dengan blok diagram pada Gambar 3.1.



Gambar 3.1 Blok Diagram rangkaian keseluruhan.

### 3.1 Perancangan Perangkat Keras

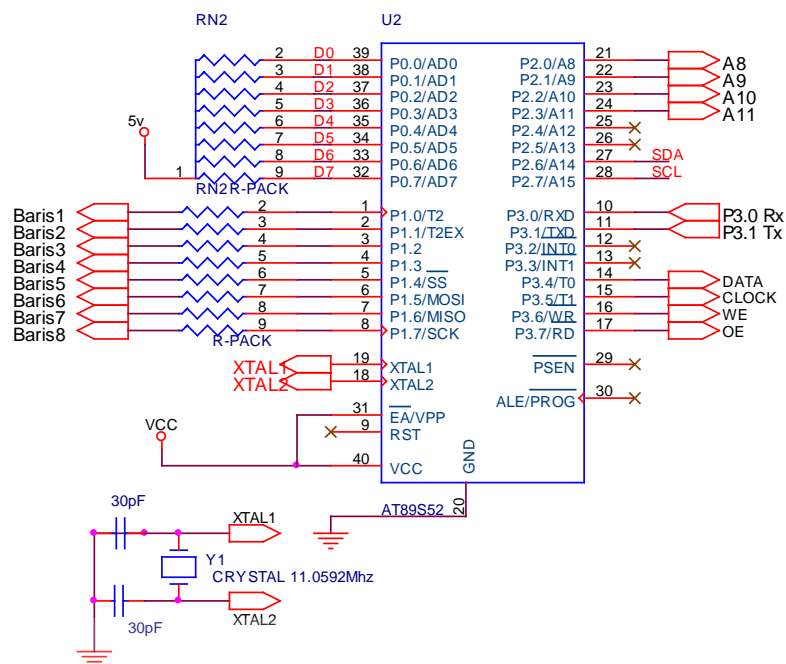
Langkah selanjutnya dalam perancangan perangkat keras pada sistem kalender digital menggunakan *dot matrix* ini adalah merealisasikan rangkaian pada diagram diatas. Rangkaian-rangkaian yang akan direalisasikan adalah:

1. Rangkaian Minimum Sistem AT89S52.
2. Rangkaian Komunikasi Serial RS232.
3. Rangkaian Serial RTC DS1307.
4. Rangkaian *Driver Baris* (TIP42).
5. Rangkaian *Shift Register* 74LS164.
6. Rangkaian *Display Dot Matrix*.

Dalam perancangan perangkat lunak terdapat proses-proses sebagai berikut: program utama, program *interrupt* serial, program konversi kalender .

#### 3.1.1 Rangkaian Minimum Sistem AT89S52

Rangkaian mikrokontroler berfungsi sebagai pusat pengontrol dari rangkaian Kalender Digital ini. Pada Tugas Akhir ini digunakan mikrokontroler keluaran ATMEL yaitu Mikrokontroler AT89S52. Mikrokontroler ini mempunyai 40 pin dengan 4 jalur *port* yaitu *Port 0*, *Port 1*, *Port 2*, dan *Port 3*. Untuk mengetahui lebih lanjut konfigurasi mikrokontroler sebagai pengendali sistem, skematik rangkaian terlihat pada Gambar 3.2.



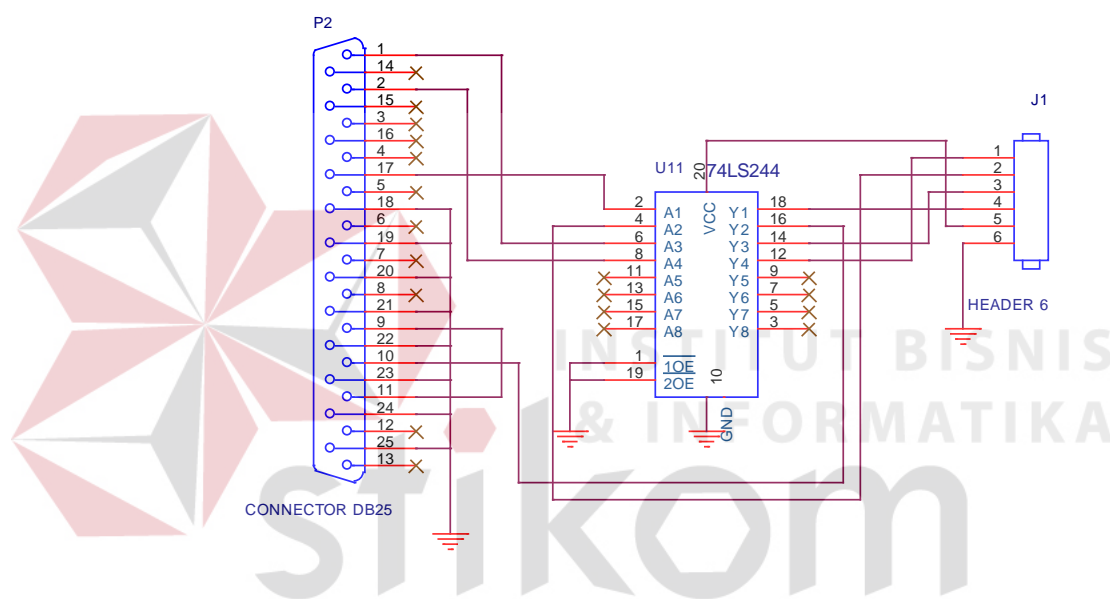
Gambar 3.2 Minimum sistem Mikrokontroler AT89S52

Pada skematik yang tercantum pada gambar 3.2 terdapat beberapa *port* yang berhubungan dengan komponen yang lainnya. *Port 0* dihubungkan dengan komponen 74HC573 yang berfungsi sebagai penyangga data yang diterima dari mikrokontroler. *Port 1* terhubung dengan rangkaian *driver* baris. Pada *port 1* ini digunakan untuk melakukan *scanning* 8 baris. *Port 3* (P3.4 dan P3.5) masing-masing berfungsi sebagai *input* data dan *input clock* pada *shift register*. *Pin* Rx/D dan Tx/D berfungsi sebagai penerima dan pengirim data serial ke komputer, terhubung melalui MAX 232 sebagai konverter.

Pada rangkaian mikrokontroler ini, digunakan komponen XTAL 11,0592 Mhz yang terhubung pada *pin* XTAL1 dan XTAL2.

#### A. Rangkaian Programmer

Penulis menggunakan rangkaian programmer yang terdiri dari sebuah IC 74LS244 yang berguna sebagai *buffer* dan kabel *downloader* dengan *interface* DB25 yang terhubung pada *port* LPT1 pada komputer dalam melakukan proses *download* program dalam format .HEX dari komputer ke mikrokontroler. Sedangkan software yang digunakan adalah Atmel *Microcontroller* ISP *Software*. Skematik kabel *downloader* yang digunakan untuk mendownload program ke mikrokontroler seperti pada Gambar 3.3.

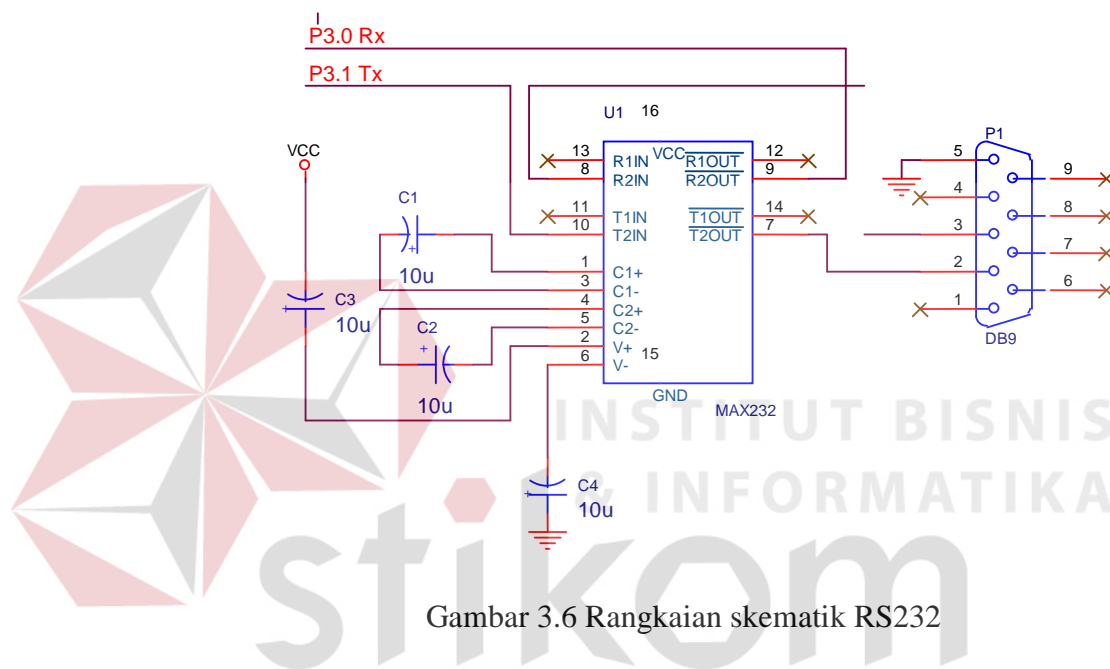


Gambar 3.3 Rangkaian kabel *downloader* pada *port* LPT

Setelah kabel *downloader* terhubung ke *Port paralel* pada PC melalui DB25 dan terhubung ke mikrokontroler melalui konektor 6 *pin*. Tahap selanjutnya adalah melakukan *download* program ke mikro. Penulis menggunakan *software* Atmel *Microcontroller*. Antarmuka *software* dapat dilihat pada Gambar 3.4.



untuk kondisi *low*. MAX232 akan mengubah level tegangan RS-232 menjadi *level* tegangan TTL agar dapat diolah oleh mikrokontroler. Demikian pula sebaliknya, data yang dikirim mikrokontroler akan diubah ke *level* tegangan RS-232 agar dapat diolah oleh komputer. Pengiriman data dari program visual di *PC* ke mikrokontroler menggunakan komunikasi serial RS232. Diagram skematik dari rangkaian serial terlihat pada Gambar 3.6.



Gambar 3.6 Rangkaian skematik RS232

Penulis menggunakan komunikasi *serial mode 1* dengan *baudrate* sebesar 9600 bps. Sehingga pengaturan register SCON dan register PCON adalah seperti pada tabel 3.1 dan tabel 3.2.

Tabel 3.1 Susunan *bit* dalam register SCON

| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
|-----|-----|-----|-----|-----|-----|----|----|
| 0   | 1   | 0   | 1   | 0   | 0   | 0  | 0  |

Dari tabel 3.1 maka SCON bernilai 0x50 dimana SM0 = 0 dan SM1 = 1 berarti menggunakan mode 1, sedangkan REN = 1 berarti mengaktifkan *port* serial untuk menerima data.

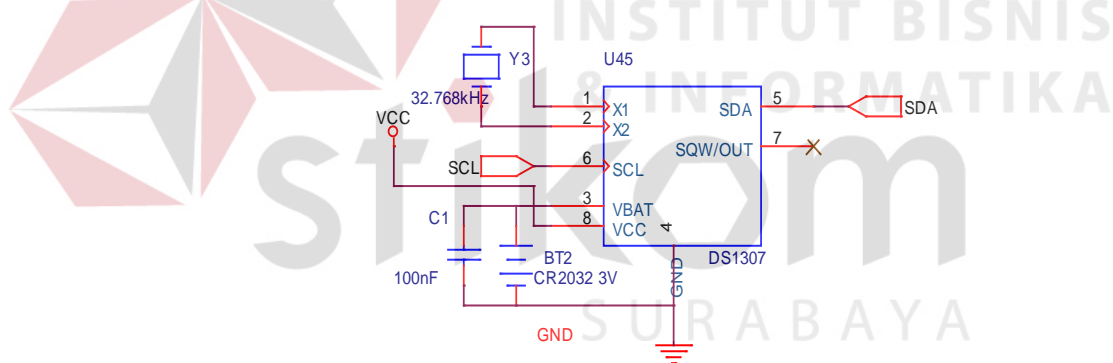
Tabel 3.2 Susunan *bit* dalam register PCON

| SMOD | - | - | - | GF1 | GF0 | PD | IDL |
|------|---|---|---|-----|-----|----|-----|
| 0    | 0 | 0 | 0 | 0   | 0   | 0  | 0   |

Dari tabel 3.2 maka PCON bernilai 0 x 00,

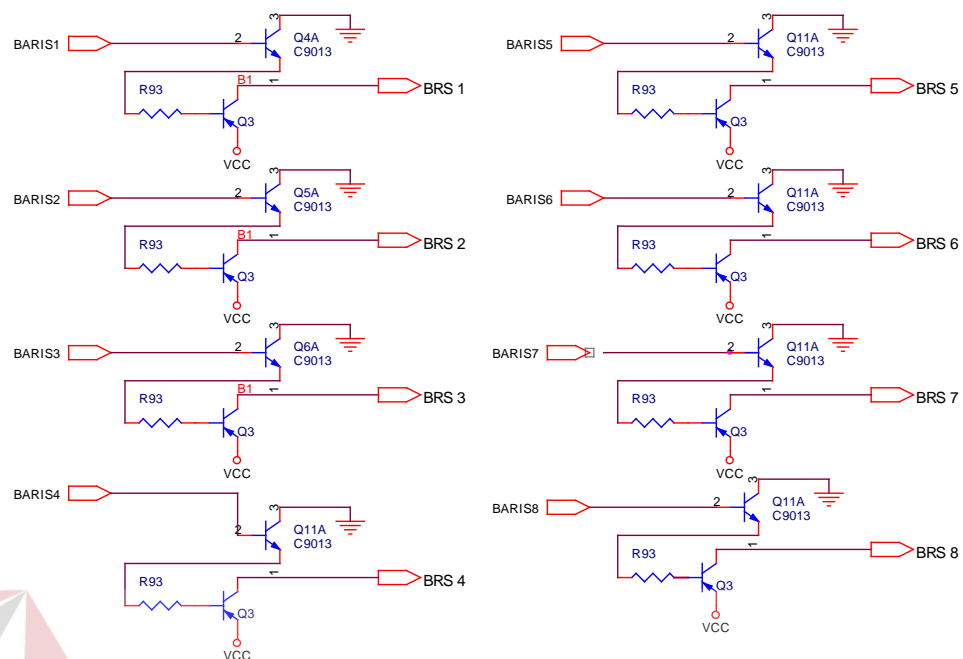
### 3.1.3 Rangkaian Serial RTC DS1307

*Real Time Clock* DS1307 digunakan untuk merancang jam digital. RTC ini berkomunikasi secara serial dengan mikrokontroler melalui kaki SDA (*serial data*) dan SCL (*serial clock*). Pada rangkaian ini DS1307 beroperasi sebagai *slave* dengan mengirimkan data waktu ke mikrokontroler yang berfungsi sebagai *master*. Konfigurasi dari pin RTC DS1307 yang digunakan dalam Tugas Akhir ini dapat dilihat pada Gambar 3.7.

Gambar 3.7 Rangkaian *Real Time Clock* DS1307

### 3.1.4 Rangkaian *Driver* Baris (TIP42)

Pada rangkaian kontroler memiliki 8 pasang transistor yang berfungsi sebagai *driver* baris. Dimana setiap pasang terdiri dari transistor 9013 dan TIP42. Rangkaian *driver* baris terhubung ke *Port* 1 mikrokontroler dan ke baris *dot matrix*. Agar lebih jelas mengenai uraian diatas, dapat dilihat skematik *driver* baris seperti Gambar 3.8.



Gambar 3.8 Rangkaian *driver* baris transistor TIP42

*Output* dari mikrokontroler tidak cukup kuat untuk menyalakan satu baris *led dot matrix* yang terdiri atas 288 *led*. Dibutuhkan transistor yang berdaya besar untuk memperkuat arus dari mikrokontroler agar dapat menyalakan atau mematikan tiap baris *led dot matrix*.

Penulis menggunakan 2 buah transistor PNP tipe TIP42 dan 9013 yang dirangkai seperti pada gambar 3.8. Transistor berfungsi sebagai saklar untuk menyalakan atau mematikan tiap baris dari *led dot matrix*. *Display dot matrix* terdiri dari 8 baris *led* sehingga digunakan 8 pasang rangkaian dengan setiap pasang transistor terhubung ke *Port P1.0* sampai *Port 1.7*.

*Pin basis* pada TIP42 terhubung ke mikrokontroler, *pin collector* sebagai *output* yang terhubung ke *pin baris* pada *led dot matrix*, sedangkan *pin emitter* terhubung pada tegangan 5V. Rangkaian *driver* ini mempunyai karakteristik akan aktif jika mendapat *input low*. Saat *output* dari mikrokontroler *high*, maka

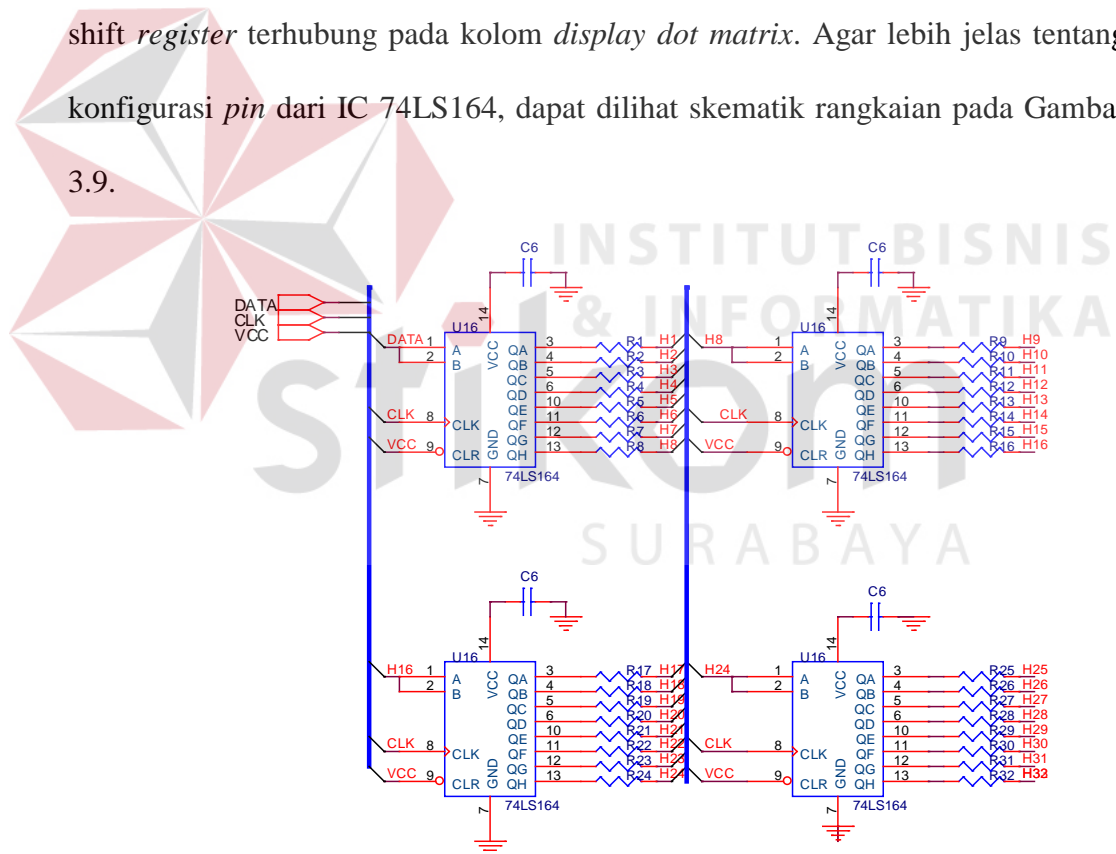


transistor 9013 akan *ON*, tegangan di kolektor akan menjadi 0 V dan transistor TIP42 akan *OFF*, sehingga baris *led* akan mati. Sebaliknya jika *output* mikrokontroler *low*, maka transistor 9013 akan *OFF*, tegangan di kolektor 9013 akan menjadi 12 V dan transistor TIP42 akan *ON* sehingga baris *led* akan hidup.

### 3.1.5 Rangkaian *Shift Register* 74LS164

Rangkaian *shift register* digunakan sebagai *driver* kolom pada *display dot matrix*. *Input* pada IC *shift register* berupa data, *clock* dan *clear* dimana masing-masing terhubung ke *Port* P3.4, *Port* P3.5 dan *VCC* dari mikrokontroler. *Output* masing-masing terhubung ke kolom *display dot matrix*. Agar lebih jelas tentang konfigurasi *pin* dari IC 74LS164, dapat dilihat skematik rangkaian pada Gambar

3.9.



Gambar 3.9 Rangkaian *Shift Register* 74LS164

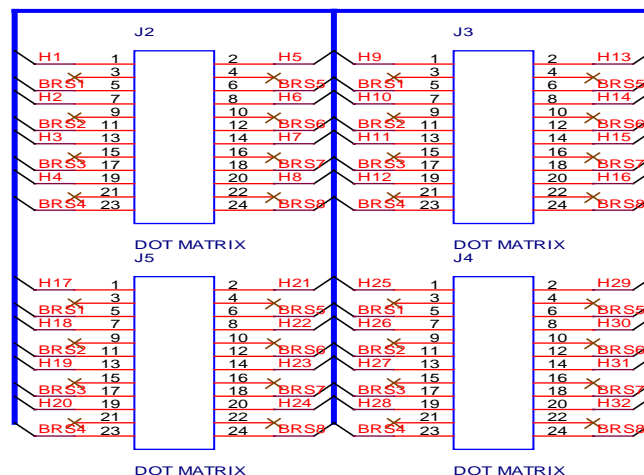
Pada rangkaian *display dot matrix* terdiri dari 288 kolom sehingga masing-masing kolom tidak dapat terhubung langsung ke *port* mikrokontroler. *Shift*

*Register* digunakan untuk mengatasi masalah ini, dimana cukup dipakai 3 output dari mikrokontroler untuk mengatur seluruh 288 kolom *led*.

*Shift Register* mempunyai 2 input A dan B yang terhubung oleh gerbang 'and', kedua *input* ini dihubungkan jadi satu dan dihubungkan ke *Port* P3.5 dari mikrokontroler. *Output* dan *Shift Register* hanya ada 8 (QA-QH) jadi dipakai 36 buah *Shift Register* untuk mengatur 288 kolom LED. Output terakhir dari *Shift Register* (QH) dihubungkan ke input *Shift Register* yang berikutnya agar semua data dapat digeser oleh *Shift Register*. Semua kaki *Clock* dari *Shift Register* terhubung ke *Port* P3.4 dan semua kaki *Clear* terhubung ke VCC agar semua *Shift Register* berjalan secara sinkron. Rangkaian ini menggunakan sistem SIPO (*Serial Input Parallel Output*).

### 3.1.6 Rangkaian *Display Dot Matrix*

Dalam perancangan kalender digital dengan *dot matrix* ini, ukuran *display* yang digunakan 48x48, dimana mikrokontroler mempunyai *display* ukuran 8x288. Pola *display* tidak memanjang tetapi berbentuk persegi, karena *dot matrix* yang disusun ke bawah secara rapat sehingga membentuk suatu *display dot matrix* ukuran 48x48. Maksud dari rangkaian *display dot matrix* diperjelas melalui skematik seperti pada gambar 3.10.



Gambar 3.10. Rangkaian *display dot matrix*

### 3.1 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak dibagi atas 2 jenis : perancangan perangkat lunak pada komputer dan perancangan perangkat lunak pada mikrokontroler.

#### 3.2.1 Perangkat Lunak pada Komputer

Perancangan perangkat lunak pada komputer berfungsi sebagai *interface* dan digunakan dalam proses konversi kalender yang selanjutnya digunakan untuk mengirimkan data pada mikrokontroler. Proses-proses utama pada perangkat lunak komputer antara lain proses konversi kalender dan proses menentukan hari dari kalender yang di-*input*-kan.

Pada perancangan perangkat lunak, penulis menggunakan *software Borland Delphi 5.0*. *Software* ini berfungsi untuk melakukan konversi sistem penanggalan yang di-*input*-kan. Kalender Masehi yang akan ditampilkan ke

*display dot matrix* dilakukan proses konversi terlebih dahulu menjadi beberapa macam kalender serta menampilkan hari.

### A. Proses Penentuan Hari

Proses penentuan hari bertujuan untuk mengetahui hari dari kalender yang telah di-*input*-kan. Hal ini dikarenakan terkadang kita tidak mengetahui hari pada kalender di masa lalu ataupun di masa mendatang. Kalender Masehi yang dijadikan *input* memiliki rentang waktu dari tahun 2000 sampai tahun 2099.

Berikut ini algoritma untuk mencari hari dalam kalender :

$$\begin{aligned} a &= \frac{14 - \text{month}}{12} \\ y &= \text{year} - a \\ m &= \text{month} + 12a - 2 \end{aligned} \quad (3.2)$$

Untuk kalender Masehi:

$$d = \left( \text{day} + y + \frac{y}{4} - \frac{y}{100} + \frac{y}{400} + \frac{31m}{12} \right) \bmod 7$$

Dari perhitungan  $d$  akan didapatkan nilai sisa pembagian yang memiliki arti :

0 = Hari Minggu

1 = Hari Senin

2 = Hari Selasa.

3 = Hari Rabu

4 = Hari Kamis

5 = Hari Jumat

6 = Hari Sabtu

*Listing* program proses penentuan hari seperti berikut:

```

begin
  Dum1 := (14 - Month) DIV 12;
  Dum2 := Year - Dum1;
  Dum3 := Month + (12 * Dum1) - 2;
  Day := (Date + Dum2 + (Dum2 DIV 4) - (Dum2 DIV 100) + (Dum2 DIV 400) +
    (31*Dum3) DIV 12);
  Day := Day MOD 7;
  Case Day of
    0 : Label21.Caption := 'Senin';
    1 : Label21.Caption := 'Selasa';
    2 : Label21.Caption := 'Rabu';
    3 : Label21.Caption := 'Kamis';
    4 : Label21.Caption := 'Jumat';
    5 : Label21.Caption := 'Sabtu';
    6 : Label21.Caption := 'Minggu';
  end;
end;

```

## B. Proses Konversi Kalender

Proses konversi kalender ini meliputi konversi dari kalender Masehi ke kalender Hijriyah, kalender Cina, kalender Jawa. Proses konversi kalender tercantum dalam listing program berikut:

```

// Konversi kalender Masehi ke Hijriyah
Begin
  if ((y2>1582) OR((y2=1582) AND (m2>10))OR((y2=1582) AND (m2=10)
  AND (d2>14))) then
    jd := intPart((1461*(y2+4800+intPart((m2-14)/12))/4)+
      intPart((367*(m2-2-12*(intPart((m2-14)/12))))/12)-
      intPart((3*(intPart(
        (y2+4900+intPart((m2-14)/12))/100))/4)+d2-32075
    else
      jd := 367*y2-intPart((7*(y2+5001+intPart((m2-9)/7)))/4)+
        intPart((275*m2)/9)+d2+1729777;
    L := jd-1948440+10632;
    N := intPart((L-1)/10631);
    L := L-10631*n+354;
    J := (intPart((10985-L)/5316))*(intPart((50*L)/17719))+
      (intPart(L/5670))*(intPart((43*1)/15238));
    L := L-(intPart((30-j)/15))*(intPart((17719*j)/50))-
      (intPart(j/16))*(intPart((15238*j)/43))+29;
    Rm := intPart((24*L)/709);
    Rd := 1-intPart((709*Rm)/24);
    Ry := 30*n+j-30;

// Konversi kalender Masehi ke Cina
Function TForm1.DecodeGregToCNDate (dtGreg: TDateTime): TCNDate;

```

```

Var
  IDayLeave: Integer;
  WYear, wMonth, wDay: WORD;
  I, j: integer;
  WBigSmallDist, wLeap, wCount, wLeapShift: WORD;
  Label OK;
Begin
  Result := 0;
  IDayLeave := Trunc (dtGreg) - cstDateOrg;
  DecodeDate (IncMonth (dtGreg, -1), wYear, wMonth, wDay);
  If (iDayLeave <0) or (iDayLeave > 22295) then Exit;
  For i:= Low (cstCNTTable) to High (cstCNTTable) do begin
    WBigSmallDist := cstCNTTable [i];
    WLeap := wBigSmallDist shr 12;
    If wLeap > 12 then begin
      WLeap := wLeap and 7;
      WLeapShift := 1;
    End else
      WLeapShift := 0;
    For j:= 1 to 12 do begin
      WCount := (wBigSmallDist and 1) + 29;
      If j = wLeap then wCount := wCount - wLeapShift;
      If iDayLeave <wCount then begin
        Result := (i shl 9) + (j shl 5) + iDayLeave + 1;
        Exit;
      End;
      IDayLeave := iDayLeave - wCount;
      If j = wLeap then begin
        WCount := 29 + wLeapShift;
        If iDayLeave <wCount then begin
          Result := (i shl 9) + (j shl 5) + iDayLeave + 1 + (1 shl 21);
          Exit;
        End;
        IDayLeave := iDayLeave - wCount;
      End;
      WBigSmallDist := wBigSmallDist shr 1;
    End;
  End;
End;
Function TForm1.isCNLeap(cnDate: TCNDate): boolean;
Begin
  Result := (cnDate and $200000) <> 0;
End;
Function GetGregDateFromCN (cnYear, cnMonth, cnDay: word; bLeap: Boolean =
  False): TDateTime;
Var
  I, j: integer;
  DayCount: integer;
  WBigSmallDist, wLeap, wLeapShift: WORD;

```

```

Begin
  DayCount := 0;
  If (cnYear <1990) or (cnYear> 2050) then begin
    Result := 0;
    Exit;
  End;
  For i := cstCYearOrg to cnYear-1 do begin
    WBigSmallDist := cstCNTabel [i];
    If (wBigSmallDist and $F000) <> 0 then DayCount := DayCount + 29;
    DayCount := DayCount + 12 * 29;
    For j := 1 to 12 do begin
      DayCount := DayCount + wBigSmallDist and 1;
      WBigSmallDist := wBigSmallDist shr 1;
    End;
  End;
  WBigSmallDist := cstCNTabel [cnYear];
  WLeap := wBigSmallDist shr 12;
  If wLeap > 12 then begin
    WLeap := wLeap and 7;
    WLeapShift := 1; // Tai, in Runru.
  End else
    WLeapShift := 0;
  For j := 1 to cnMonth-1 do begin
    DayCount := DayCount + (wBigSmallDist and 1) + 29;
    If j = wLeap then DayCount := DayCount + 29;
    WBigSmallDist := wBigSmallDist shr 1;
  End;
  If bLeap and (cnMonth = wLeap) then begin
    DayCount := DayCount + 30 - wLeapShift;
    Result := DayCount + cstDateOrg + cnDay - 1;
  End;
End;

```

### 3.2.2 Perangkat Lunak pada Mikrokontroler

Perancangan perangkat lunak pada mikrokontroler berfungsi sebagai pengendali sistem dan digunakan dalam proses pengujian *display dot matrix* yang selanjutnya digunakan untuk mengirimkan data. Perangkat lunak yang digunakan adalah bahasa *assembly* dengan *software* MIDE. Proses-proses utama pada perangkat lunak mikrokontroler antara lain proses *scanning* baris, proses *update* waktu dan proses serial *interrupt*.

### A. Proses *Scanning* Baris

Untuk menghasilkan tampilan *display dot matrix* yang tidak berkedip, maka frekuensi dari *scanning* baris harus melebihi frekuensi penglihatan mata manusia dalam keadaan normal (60 Hz).

Sehingga mikrokontroler harus dapat melakukan proses *scanning* delapan baris dengan frekuensi diatas 60 Hz. Perhitungannya adalah sebagai berikut:

$$T = \frac{1}{f} \quad (3.1)$$

$$= \frac{1}{60\text{Hz}}$$

$$\cong 0.0167 \text{ s}$$

$$= 16,7 \text{ ms}$$

Sehingga masing-masing baris membutuhkan waktu maksimal sebesar:

$$\frac{16,7}{8} = 2.09 \text{ ms}$$

Karena dalam satu baris terdapat 288 kolom titik *dot matrix*, maka untuk menyalakan masing-masing *led* dalam *dot matrix* diberikan waktu sebesar:

$$\frac{2.09\text{ms}}{288\text{led}} = 0.007 \text{ ms/led.}$$

Dalam perangkat tugas akhir ini penulis menentukan waktu *scanning* tiap baris sebesar 2.09ms. Dengan waktu tersebut mikrokontroler dapat menghasilkan frekuensi sebesar 61,27 Hz. perhitungannya adalah sebagai berikut:

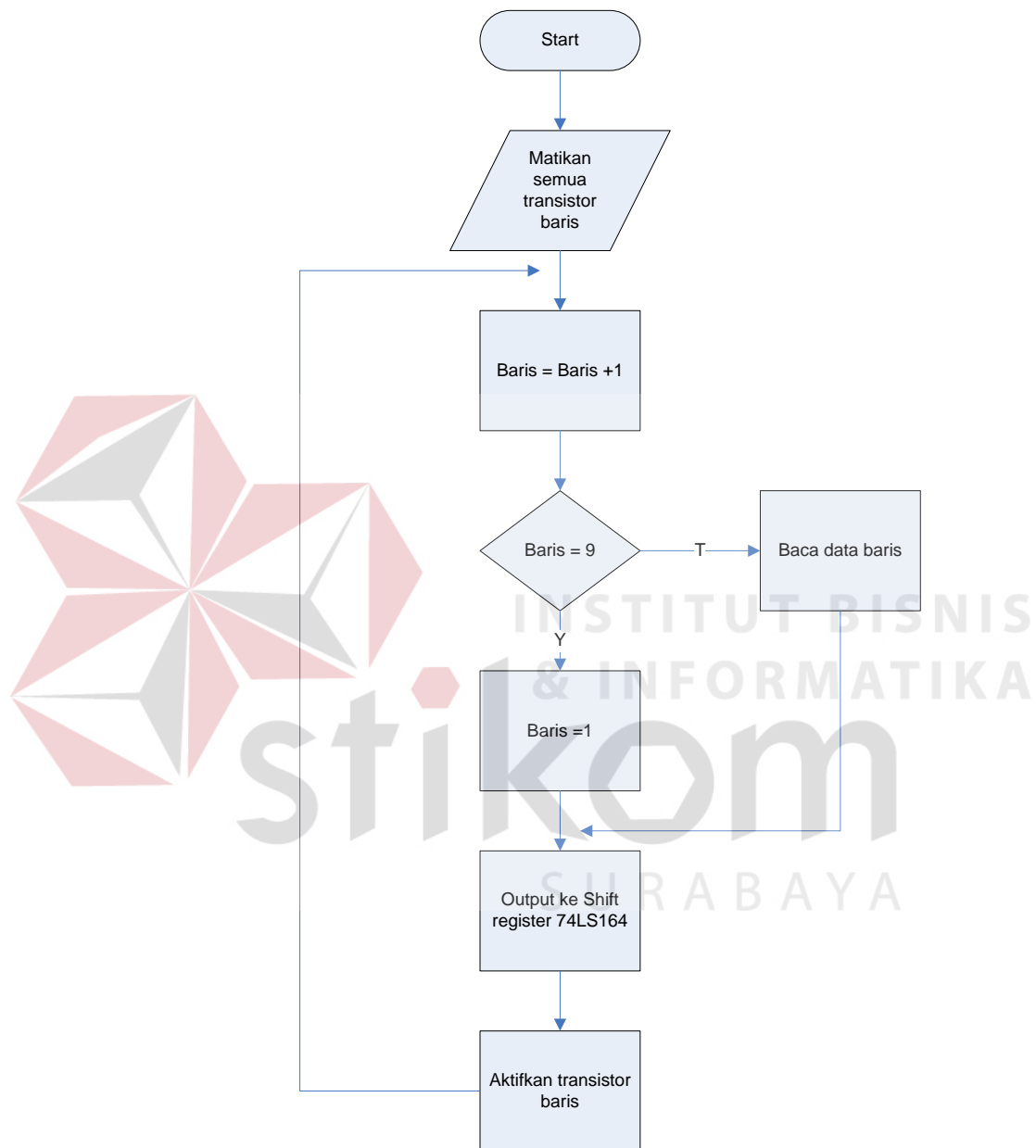
$$2.09 \times 8 \text{ baris} = 16.32 \text{ ms}$$

$$F = \frac{1}{16,32\text{ms}}$$

$$= 61.27 \text{ Hz}$$



Proses *scanning* baris pada modul mikrokontroler dapat ditunjukkan oleh diagram alir pada Gambar 3.11.



Gambar 3.11. Diagram alir proses *scanning* baris

Berikut listing program *scanning* pada baris :

```

Program      EQU      0000h
TH0Val_C    EQU      0F8h      ; nilai timer untuk scanning
TL0Val_C    EQU      000h;      : (65536 - TH0:TL0) * (12 MHz / 11.0592 MHz)

JCol_C      EQU      200
  
```

```

DispBuffAddr_C      EQU      0h

;-----
;          PORTS
;-----

Clk_P      BIT      p3.4
Data_P     BIT      p3.5

Row_P      EQU      p1

;-----
Timer_0
;-----

MOV        th0,#TH0Val_C
MOV        t10,#TL0Val_C
PUSH      a
PUSH      psw
PUSH      dph
PUSH      dpl
PUSH      7

; off all transistor

MOV        Row_P,#0
MOV        dptr,#DispBuffAddr_C
MOV        r7,#JCol_C
T0J3
MOVX      a,@dptr          ; 2 cycles (24 osc.periods)
CLR      Data_P          ; 1 cycle (12 osc.periods)
ANL      a,BitMask_M     ; 1 cycle (12 osc.periods)
JZ       T0J4            ; 2 cycles (24 osc.periods)
SETB     Data_P          ; 1 cycle (12 osc.periods)
T0J4
SETB     Clk_P           ; 1 cycle (12 osc.periods)
CLR      Clk_P           ; 1 cycle (12 osc.periods)
INC      dptr            ; 2 cycles (24 osc.periods)
DJNZ    r7,T0J3
MOV      Row_P,RowMask_M

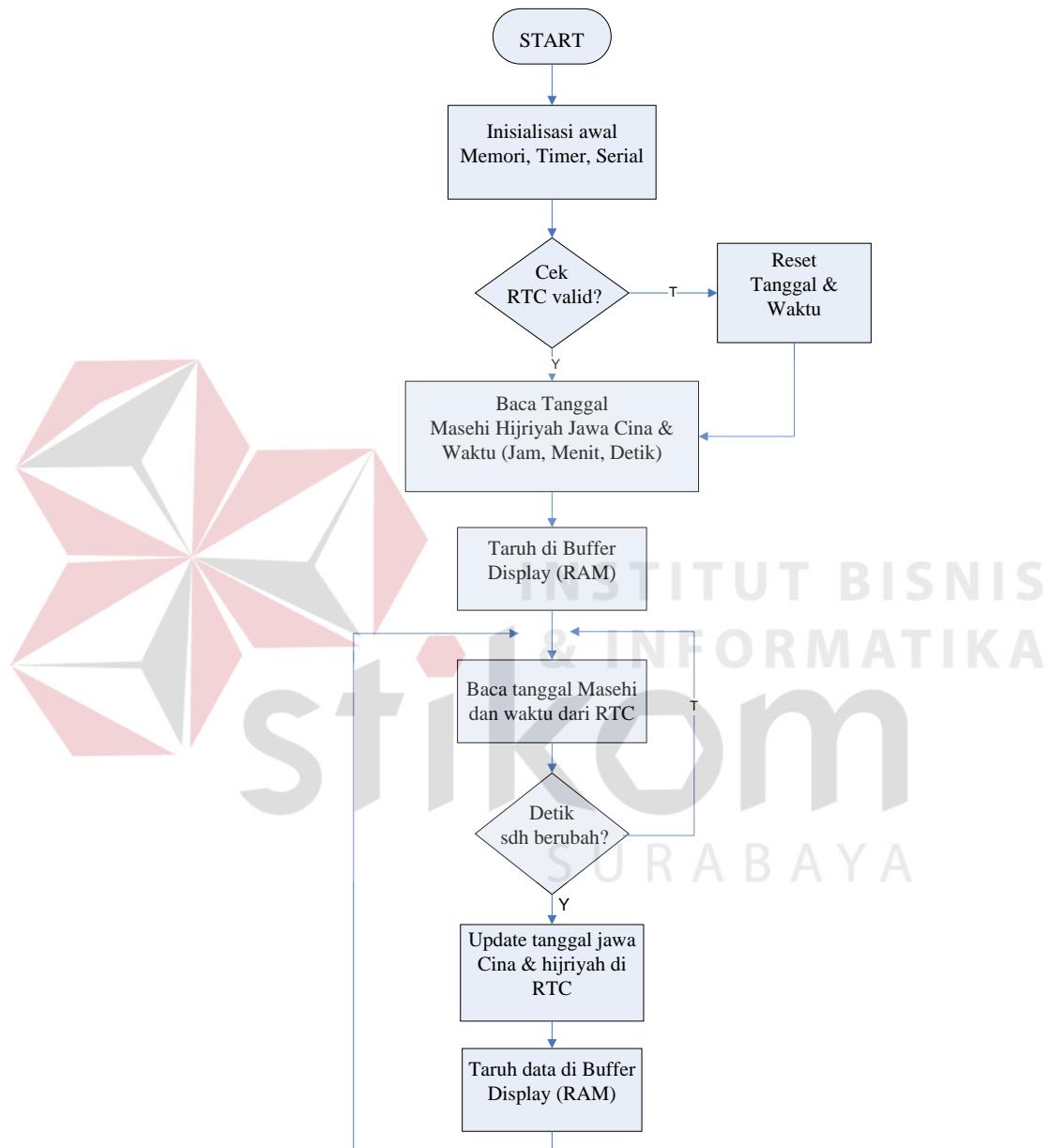
MOV      a,BitMask_M
RR      a
MOV      BitMask_M,a
MOV      a,RowMask_M
RL      a
MOV      RowMask_M,a
CLR      LastRow_F
INC      ScanCtr_M
MOV      a,ScanCtr_M
CJNE    a,#8,T0J1
SETB     LastRow_F
MOV      ScanCtr_M,#0
MOV      RowMask_M,#0000001b

T0J1
POP      7
POP      dpl
POP      dph
POP      psw
POP      a
RETI

```

## B. Proses *Update Waktu*

Proses set dan *update* waktu pada modul mikrokontroler dapat ditunjukkan oleh diagram alir pada Gambar 3.12.



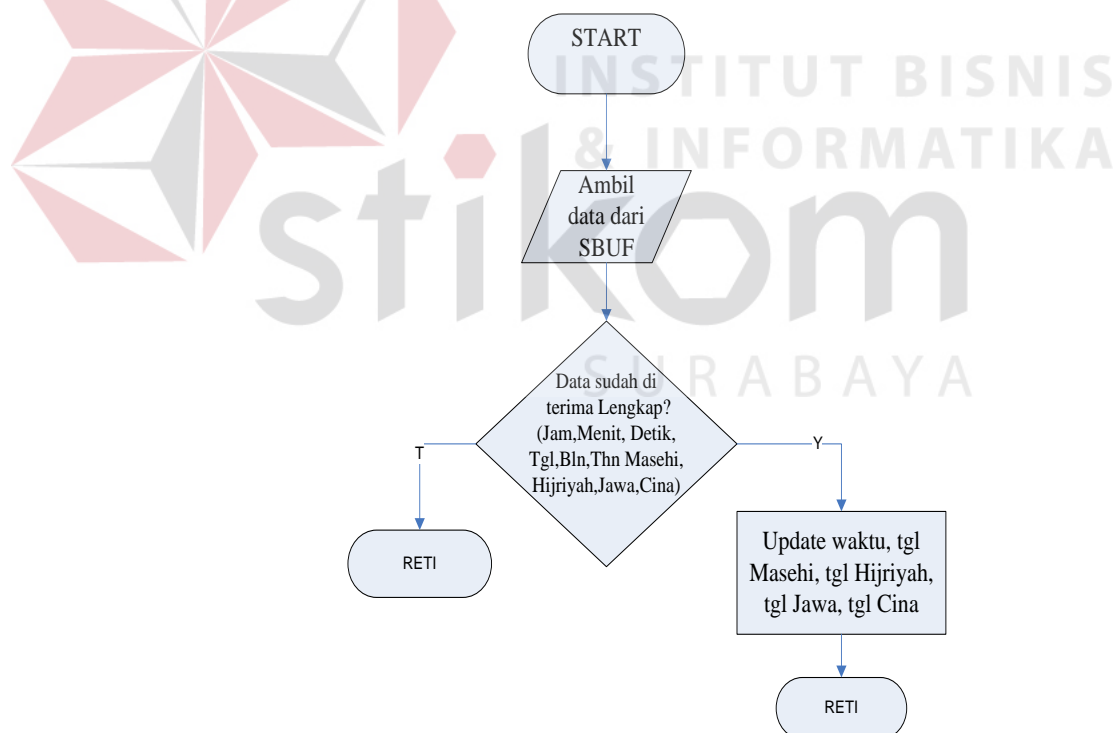
Gambar 3.12. Diagram alir program utama

Penjelasan dari diagram alir program utama pada Gambar 3.12 yaitu pada saat program pertama kali dijalankan, dilakukan inisialisasi terlebih dahulu yang

meliputi inisialisasi memori, *timer* dan serial. Dilakukan cek apakah *internal clock* serial RTC sudah sesuai. Jika *internal clock* tak sesuai dilakukan reset pada RTC. Sebaliknya jika sesuai dilakukan pembacaan tanggal dan waktu. Setelah di-set data *internal clock* diletakkan di *buffer* RAM. Setelah semua proses dilakukan, selanjutnya dilakukan *update internal clock* RTC. Proses ini dilakukan berulang-ulang dan setiap selesai *update* data disimpan pada *buffer display* pada RAM.

### C. Proses Serial *Interrupt*

Proses serial *interrupt* menangani apabila terdapat *interrupt* dari *user* untuk mengganti atau melakukan *update* pada kalender. Proses tersebut ditunjukkan oleh diagram alir pada Gambar 3.13.



Gambar 3.13. Diagram alir program serial *interrupt*

Penjelasan dari diagram alir program utama pada Gambar 3.13 adalah saat terjadi interupsi dari *user* dengan memberi *input* tanggal yang lain pada komputer dan melakukan *update* kalender. Program akan membaca data dari register SBUF. Kemudian program akan melakukan cek apakah data yang dikirimkan sudah diterima dengan lengkap meliputi jam, menit, detik, tanggal, bulan dan tahun Masehi, Hijriyah, Jawa, dan Cina. Seandainya data sudah lengkap akan dilakukan *update* kalender.

