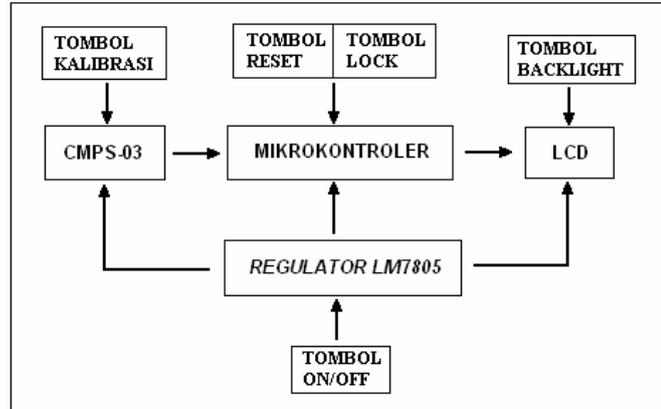


3. DESAIN SISTEM

Pada alat ini memiliki beberapa bagian, yaitu mikrokontroler dengan menggunakan *single chip* AT89S53, rangkaian *power supply* yang menurunkan tegangan DC dari 9 V menjadi 5 V, sensor magnetik bumi CMPS-03, LCD grafik LPH 7779 sebagai layar tampilan *output*-nya, dan memiliki lima buah tombol, yaitu:

- *On/Off*, yaitu tombol untuk memberikan tegangan *supply* pada sistem.
- *Reset*, yaitu tombol untuk me-*reset* mikrokontroler AT89S53.
- Kalibrasi, yaitu tombol untuk mengkalibrasi modul kompas CMPS-03 sehingga penunjukan arah mata angin-nya sesuai dengan kompas konvensional lainnya.
- *Backlight*, yaitu tombol untuk menyalakan enam buah lampu LED yang terletak di belakang LCD grafik.
- *Lock*, yaitu tombol untuk mengunci sudut.



Gambar 3.1. Blok Diagram Sistem

Gambar 3.1 menunjukkan gambar blok diagram dari sistem kompas digital ini. Sensor CMPS-03 akan menghasilkan *output* PWM yang berubah-ubah sesuai dengan besarnya sudut, dengan tolok ukur utara magnet bumi sebagai titik nol derajat dan bergerak searah dengan jarum jam. Lebar pulsa positif dari CMPS-03 ini bervariasi dari 1 mS (0°) sampai 36,99 mS ($359,9^\circ$) atau dengan kata lain, besar lebar pulsa positifnya bertambah secara linear 100 μ S tiap derajat dengan *offset* +1 mS.

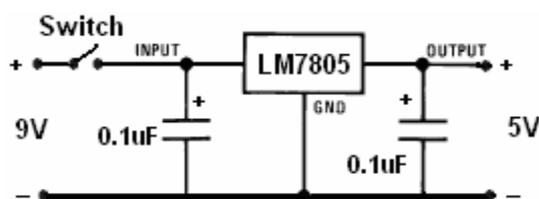
Kemudian dari sinyal PWM ini akan dibaca oleh mikrokontroler AT89S53. Selanjutnya, mikrokontroler akan melakukan pembacaan sudut melalui program konversi sudut yang dibuat dan kemudian menampilkan hasilnya melalui *display* LCD grafik berupa tampilan angka dan gambar seperti contoh yang terlihat pada gambar 3.2. Penjelasan lebih lengkap tentang proses pembacaan sudut dan penjelasan tentang *display* gambar akan dijelaskan pada sub bab 3.2.



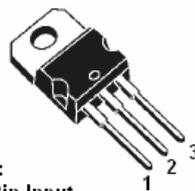
Gambar 3.2. *Display* Kompas Digital.

3.1. Perangkat Keras (*Hardware*)

3.1.1. Rangkaian untuk *Regulator* LM 7805C



Gambar 3.3. Rangkaian *Regulator* LM 7805C.



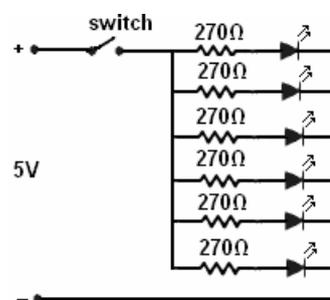
Ket:
 1. Pin Input
 2. Pin Ground
 3. Pin Output

Gambar 3.4. IC *Regulator* LM 7805C.

Untuk rangkaian penghasil tegangan 5 V yang digunakan pada rangkaian kompas digital ini tampak seperti pada gambar 3.3. Pada kaki nomor 1, yaitu *pin input* dihubungkan dengan sumber tegangan DC 9 V karena tegangan *input*-nya harus lebih besar daripada tegangan *output* yang dihasilkan. Sebagai sumber tegangan DC 9 V ini digunakan sebuah baterai 9 V yang mudah didapat dipasaran. Maka tegangan *output* yang dihasilkan dari LM 7805C ini *fix* sebesar 5 V dengan toleransi kesalahan $\pm 2\%$ dan mampu melewati arus sampai batas maksimum 1A.

Pemberian kapasitor $0.1\mu\text{F}$ dari V_{IN} ke *ground* sebenarnya tidak terlalu diperlukan pada rangkaian kompas digital ini, hanya saja diperlukan bila letak *regulator* jauh dari *power supply*. Sedangkan pemasangan kapasitor $0.1\mu\text{F}$ dari V_{OUT} ke *ground* (diperlukan untuk mengatasi *noise* yang kadang-kadang dihasilkan *power supply* yang digunakan), sehingga tegangan *output* 5 V yang dihasilkan dari LM 7805C ini lebih halus (*noise* berkurang).

3.1.2. Rangkaian *Backlight*



Gambar 3.5. Rangkaian *Backlight*

Untuk rangkaian *backlight* tampak seperti gambar 3.5. Rangkaian *backlight* terdiri dari enam buah lampu LED. Tiap LED tersebut diseri dengan sebuah resistor dengan besar 270Ω dan kemudian dirangkai secara paralel (supaya bila salah satu lampu LED mati, maka lampu LED yang lain masih tetap hidup) satu sama lain serta dihubungkan dengan sebuah *switch* yang terhubung dengan *supply* 5 V. Pemilihan resistor sebesar 270Ω ini dikarenakan oleh alasan sebagai berikut:

- Pada awalnya, lampu LED ingin dinyalakan dengan konsumsi arus tiap LED sebesar 10 mA. Dari perhitungan diperoleh $R = 200 \Omega$. Karena terdapat resistor sebesar 270Ω , maka dicoba untuk menggunakannya pada rangkaian *backlight*. Dari perhitungan diperoleh $I_{LED} = 7,15 \text{ mA}$, dan nyala LED dirasa masih cukup sebagai *backlight*. Selain itu, hal ini memberi keuntungan lebih, yaitu dapat me-minimal-kan konsumsi daya sistem pada alat *display* kompas digital ini. Sehingga dipilih resistor sebesar 270Ω untuk digunakan pada rangkaian *backlight*.

Berikut ini adalah perhitungan untuk memperoleh $R = 200 \Omega$ dengan diinginkan arus yang dikonsumsi tiap LED sebesar 10 mA.

$$V_{CC} = 5 \text{ V}$$

$$V_{LED} = 3 \text{ V (Tegangan tiap LED yang diperoleh dari pengukuran).}$$

$$V_{CC} = V_{LED} + I_{LED} \cdot R \dots\dots\dots(3.1)$$

Dari data dan persamaan (3.1) di atas diperoleh hasil:

$$5 \text{ V} = 3 \text{ V} + 10 \cdot 10^{-3} \text{ A} \cdot R$$

$$R = (5 \text{ V} - 3 \text{ V}) / 10 \cdot 10^{-3} \text{ A}$$

$$= 2 \text{ V} / 10 \cdot 10^{-3} \text{ A}$$

$$= 0,2 \cdot 10^3 \Omega$$

$$= 200 \Omega.$$

Berikut adalah perhitungan untuk mencari arus yang dikonsumsi tiap LED dengan resistor sebesar 270Ω .

$$V_{CC} = 5 \text{ V}$$

$$R = 270 \Omega$$

$V_{LED} = 3,07 \text{ V}$ (Tegangan tiap LED yang diperoleh dari pengukuran pada rangkaian dari gambar 3.5)

$$V_{CC} = V_{LED} + I_{LED} \cdot R \dots\dots\dots(3.2)$$

Dari data dan persamaan (3.1) di atas diperoleh hasil:

$$5 \text{ V} = 3,07 \text{ V} + I_{LED} \cdot 270 \Omega$$

$$I_{LED} = (5 \text{ V} - 3,07 \text{ V}) / 270 \Omega$$

$$= 1,93 \text{ V} / 270 \Omega$$

$$= 0,00715 \text{ A}$$

$$= 7,15 \text{ mA}.$$

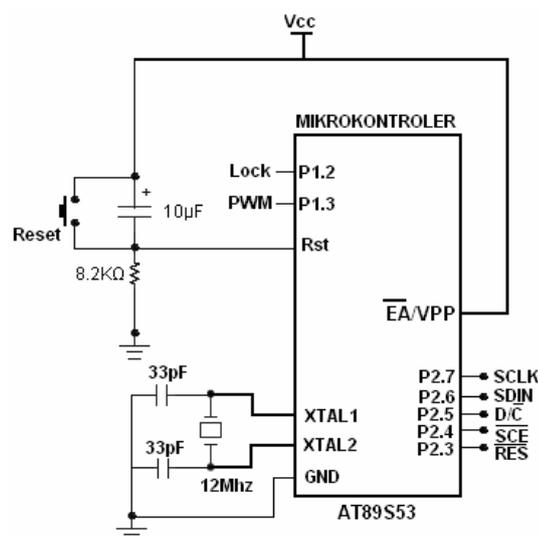
Jadi, arus yang dikonsumsi oleh masing-masing LED adalah sebesar 7,15 mA.

3.1.3. Rangkaian untuk Modul Kompas CMPS-03

Untuk modul kompas CMPS-03, rangkaian koneksinya seperti terlihat pada bab 2, halaman 17 (gambar 2.11), yaitu *pin 1*, *pin 2*, dan *pin 3* dihubungkan dengan *supply* positif 5V. *Pin 4* yang merupakan *output* dari sinyal PWM yang dihasilkan oleh modul tersebut dihubungkan dengan *port 1.3* pada mikrokontroler AT89S53. Untuk *pin 5*, *pin 7*, dan *pin 8*, tidak dihubungkan karena *pin* tersebut *no-connect* (dapat dilihat pada *datasheet CMPS-03-Robot Compass Module* yang terletak pada bagian lampiran). *Pin 6* yang berfungsi sebagai *pin* kalibrasi dihubungkan dengan menggunakan *push-button switch* ke *ground* untuk dapat melakukan proses kalibrasi.

3.1.4. Rangkaian untuk Mikrokontroler AT89S53

Rangkaian mikrokontroler AT89S53 ini terdiri dari mikrokontroler AT89S53, kristal 12 Mhz, beberapa resistor, beberapa kapasitor baik kapasitor *polar* maupun *non-polar*, dan beberapa komponen tambahan lainnya. Untuk lebih jelasnya dapat dilihat pada gambar 3.6, yaitu gambar rangkaian untuk mikrokontroler AT89S53.



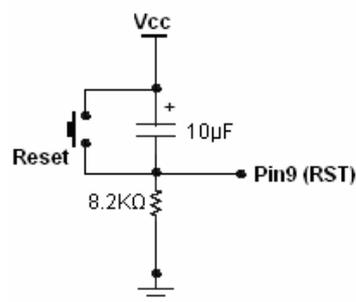
Gambar 3.6. Rangkaian untuk Mikrokontroler AT89S53.

Rangkaian ini berfungsi sebagai pembaca dan pengolah *output* PWM dari modul kompas CMPS-03. Penerimaan data tersebut dilakukan melalui *port* 1.3, yang berfungsi sebagai pengontrol untuk mengaktifkan dan me-*non*-aktifkan *timer* pada mikrokontroler. Kemudian, setelah data diolah hasilnya akan ditampilkan pada LCD grafik LPH 7779.

3.1.4.1. Rangkaian *Reset* dan Kristal pada Mikrokontroler AT89S53

Rangkaian *reset* berguna untuk mengembalikan semua nilai *register* seperti keadaan semula dan mikrokontroler akan menuju ke alamat 0000_H. Proses *reset* ini akan terjadi apabila ada logika ‘1’ (*high*) selama minimal 2 siklus pada kaki *pin* RST.

Untuk rangkaian *reset*-nya, digunakan sebuah resistor 8.2 k Ω dan sebuah kapasitor elektrolit (*polar*) dengan kapasitansi 10 μ F/16V, juga sebuah *switch push-button* sebagai tombol *reset*-nya. Antara kaki resistor dan kaki negatif kapasitor elektrolit dihubungkan dengan *pin reset* pada kaki mikrokontroler AT89S53 (*pin* 9).

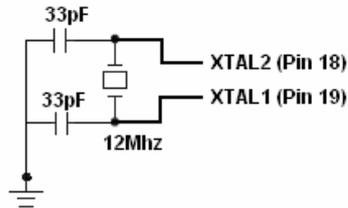


Gambar 3.7. Rangkaian *Reset* pada Mikrokontroler AT89S53

Sedangkan untuk rangkaian kristalnya, digunakan sebuah kristal 12Mhz dengan dua buah kapasitor keramik (*non-polar*) dengan kapasitansi 33pF.

- Frekuensi eksekusi = $1/12 \times \text{kapasitas kristal} \dots \dots \dots (3.3)$
 $= 1/12 \times 12\text{Mhz}$
 $= 1 \text{ Mhz.}$
- Kecepatan eksekusi = $1/F \dots \dots \dots (3.4)$
 $= 1/1\text{Mhz}$
 $= 1 \mu\text{S.}$

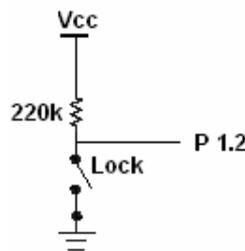
Untuk koneksi antara kristal, kapasitor *non-polar*, dan juga *pin* XTAL1 dan XTAL2 (pada mikrokontroler) dapat dilihat pada gambar berikut ini.



Gambar 3.8. Rangkaian Osilator pada Mikrokontroler AT89S53

3.1.5. Rangkaian untuk Tombol *Lock*

Untuk tombol pengunci sudutnya (*lock*), digunakan sebuah resistor sebagai *pull-up* sebesar 220 k Ω dan sebuah *push-button switch*. Pertemuan antara salah satu kaki resistor dengan salah satu kaki *push-button switch* tersebut dihubungkan dengan *port* 1.2 pada mikrokontroler. Gambar rangkaiannya dapat dilihat pada gambar 3.9 berikut ini.



Gambar 3.9. Rangkaian untuk Tombol *Lock*

Dari rangkaian tersebut, diperoleh bahwa arus yang masuk ke *port* 1.2 AT89S53 tidak melebihi batas maksimum I_{IL} (*logical 0 input current*) pada *port* 1, yaitu sebesar 50 μA , sehingga dapat diterima oleh mikrokontroler AT89S53.

$$V_{CC} = 5 \text{ V}$$

$$R = 220 \text{ k}\Omega$$

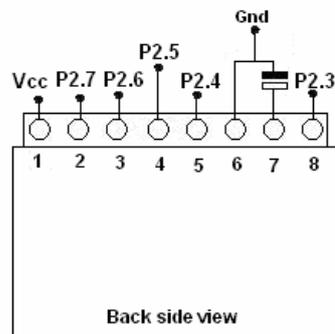
$$V_{CC} = I \cdot R \dots\dots\dots(3.5)$$

$$5 \text{ V} = I \cdot 220 \cdot 10^3 \Omega$$

$$I = 5 \text{ V} / 220 \cdot 10^3 \Omega$$

$$= 0,02273 \text{ mA} = 22,73 \mu\text{A}.$$

3.1.6. Rangkaian untuk LCD LPH 7779



Gambar 3.10. Rangkaian untuk LCD LPH 7779

Untuk rangkaian pada LCD LPH 7779 seperti terlihat pada gambar 3.10. *Pin* 1 (V_{CC}) dari LCD tersebut dihubungkan dengan tegangan 5 V sebagai tegangan *supply*-nya. Kemudian berturut-turut kaki nomor 2 (SCLK), 3 (SDIN), 4 (D/\overline{C}), dan 5 (\overline{SCE}) dihubungkan dengan *port* 2.7, *port* 2.6, *port* 2.5, dan *port* 2.4 pada mikrokontroler AT89S53. Sedangkan kaki nomor 6 (*ground*) dari LCD dihubungkan dengan kaki negatif dari sebuah kapasitor sebesar $4.7\mu\text{F}/16\text{V}$ dan kaki nomor 7 (*Output*) dari LCD dihubungkan dengan kaki positif dari kapasitor tersebut. Penambahan kapasitor ini bertujuan untuk mengurangi *ripple* (Dapat dilihat pada *Datasheet PCD 8544, 48x 84 pixels matrix LCD controller/driver*, p.23.) sehingga LCD dapat bekerja dengan stabil. Kaki nomor 8 (*reset*) dari LCD dihubungkan dengan *port* 2.3 pada mikrokontroler.

3.2. Perangkat Lunak (Software)

3.2.1. Program Insialisasi LCD LPH 7779

Pertama-tama, semua *pin* LCD yang terhubung dengan *port* 2 pada AT89S53 diberi logika '0' melalui mikrokontroler tersebut. Kemudian diberikan *delay* sebesar 50 mS, sebelum akhirnya *pin reset* pada LCD diaktifkan selama 250 mS. Setelah di-*reset* selama 250 mS kemudian *pin reset* LCD dimatikan. Sinyal *reset* harus diberikan pada LCD ini. Karena begitu tegangan *supply* dinyalakan, maka isi dari semua *register* internal dan RAM dari PCD 8544 dalam keadaan '*undefined*'. Sinyal *reset* ini harus diberikan dengan ketentuan maksimum 100 mS setelah V_{DD} *high* dengan besar pulsa *reset* minimal 100 nS. Penjelasan tentang *reset timing* ini dapat dilihat pada bab 2, halaman 28 (gambar 2.25).

Bagian program yang digunakan untuk meng-inisialisasi LCD LPH 7779 adalah sebagai berikut.

```

;-----INISIALISASI_LCD-----
INIT:
    MOV P2,#00H
DELAY_50ms:MOV    R1,#1Ah
L0006:  MOV    R2,#04h
L0007:  MOV    R3,#9Fh
L0008:  NOP
        DJNZ   R3,L0008
        DJNZ   R2,L0007
        DJNZ   R1,L0006

        CLR RES

DELAY_250ms:MOV   R1,#0A7h
L0009:  MOV    R2,#02h
L000A:  MOV    R3,#0F8h
L000B:  NOP
        DJNZ   R3,L000B
        DJNZ   R2,L000A
        DJNZ   R1,L0009

        SETB RES

SETB SCE
    MOV DPTR,#BYTE_INIT
    MOV R4,#6
KIRIM_INS: MOV A,#00H
            MOVC A,@A+DPTR
            LCALL WVR_A
            INC DPTR
            DJNZ R4,KIRIM_INS
;-----CLEAR_DDRAM-----
NLCD_CLEAR:
    CLR SCE
    SETB DC
    MOV R7,#6 ;6 PAGE
LROW:    MOV R6,#84 ;84 KOLOM
LCOLUMN: MOV R5,#8
LCLOCK:  CLR SCL ;CLOCK UNTUK 8 BIT DARI TIAP PAGE
    CLR SDA
    SETB SCL
    DJNZ R5,LCLOCK
    DJNZ R6,LCOLUMN
    DJNZ R7,LROW
    SETB SCE

```

Gambar 3.11. Bagian Program untuk Inisialisasi LCD LPH 7779.

Setelah sinyal *reset* ini selesai diberikan, kemudian *pin* \overline{SCE} pada LCD diaktifkan dengan memberikan logika '0' untuk menginisialisasi *serial interface*. Baru kemudian dikirimkan *byte-byte* data sebagai *command* sesuai dengan intruksi *set* pada PCD 8544 untuk proses inisialisasi, yaitu:

- 21_H = Set PD=0,V=0, dan H=1 (*Extended instruction set*).
- $0E3_H$ = Set Vop LCD pada $0E3_H$ sehingga memberikan tegangan LCD sebesar 9 Volt.
- 06_H = Set pada koefisien suhu tegangan LCD sesuai kurva tipikal.
- 13_H = Set Mux Rate pada 1:48.
- 20_H = Set PD=0,V=0, dan H=0 (*Normal instruction set*).
- $0C_H$ = Set *display control* pada mode normal.

'CLEAR_DDRAM' pada bagian program tersebut diberikan supaya *display* LCD tidak dipenuhi oleh *pixel-pixel* yang muncul secara acak.

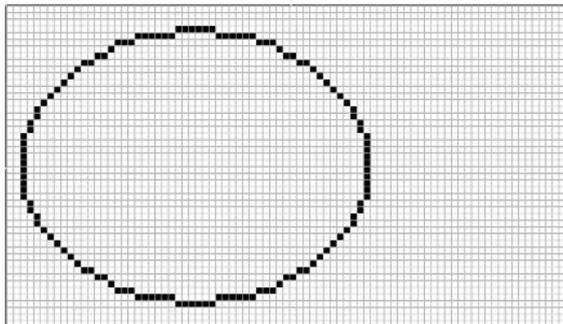
3.2.2. Program Pembuatan Lingkaran dan *Pointer* pada *Display* LCD.

Lingkaran pada *display* dibuat secara manual dengan terlebih dulu menggambar sebuah lingkaran dengan menggunakan program aplikasi Microsoft

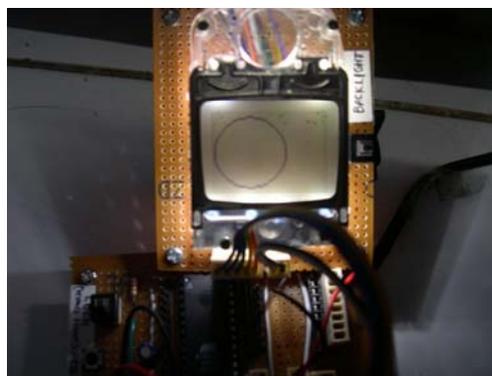
Paint. Dari gambar lingkaran yang telah dibuat pada Microsoft Paint ini kemudian diambil semua data yang ada yang membentuk gambar lingkaran tersebut kemudian mengirimkan data tersebut pada LCD.

Pada kenyataannya, ukuran dari tiap *pixel* LCD LPH 7779 yang digunakan sebagai *display* tersebut tidak sama sisi. Sehingga untuk membuat sebuah lingkaran dengan jari-jari yang cukup panjang pada *display* tersebut, maka gambar yang dibentuk pada Microsoft Paint tersebut harus berbentuk *ellips*. Jika gambar yang dibentuk pada Microsoft Paint tersebut adalah lingkaran, maka pada *display* LCD LPH 7779 akan diperoleh gambar *ellips*. Sehingga, untuk memperoleh sebuah gambar lingkaran yang baik (jari-jari sama) pada *display*, dibutuhkan ketelitian dalam membuat *ellips* tersebut pada Microsoft Paint.

Gambar 3.12 berikut ini menunjukkan gambar *ellips* yang dibuat pada Microsoft Paint yang akan merepresentasikan sebuah lingkaran pada *display* LCD grafik LPH 7779.



Gambar 3.12. Gambar Lingkaran Pada Microsoft Paint



Gambar 3.13. Lingkaran Pada LCD LPH 7779 Yang dihasilkan Dari Gambar Lingkaran Pada Microsoft Paint

Ukuran *workspace* pada Microsoft Paint tersebut disesuaikan dengan ukuran resolusi pada LCD yaitu 48 x 84 *pixel*. Jadi, panjang dan lebar dari gambar 3.12 adalah 48 x 84 *pixel*. Data yang diperoleh dari gambar lingkaran tersebut kemudian dimasukkan ke dalam tabel *array* dalam format *assembler*. Berikut adalah data dari gambar lingkaran tersebut.

```
TABEL_1: db 80h,80h,40h,20h,20h,20h,10h,10h,10h,10h,10h,10h,08h,08h,08h,08h,08h
          db 10h,10h,10h,10h,10h,10h,20h,20h,20h,40h,80h,80h
TABEL_2: db 80h,40h,20h,10h,8,4,2,1,1
TABEL_3: db 1,1,2,4,8,10h,20h,40h,80h
TABEL_4: db 0f8h,06h,01h
TABEL_5: db 1,6,0f8h
TABEL_6: db 1fh,60h,80h
TABEL_7: db 80h,60h,1fh
TABEL_8: db 1,2,4,8,10h,20H,40H,80H,80H
TABEL_9: db 80h,80h,40h,20h,10h,8,4,2,1
TABEL_10: db 1,1,2,4,4,4,8,8,8,8,8,10H,10H,10H,10H,10H,10H,8,8,8,8,8,4,4,2,1,1
```

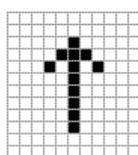
Gambar 3.14. Tabel *Array* dari Gambar Lingkaran.

Dari data-data tersebut kemudian dikirimkan sesuai dengan *page* dan kolom dari data tersebut pada LCD grafik. Berikut ini adalah bagian dari program untuk membuat lingkaran pada *display*.

```
LINGKARAN:
CHAR1: LCALL PAGE0
      ACALL KOLOM13
      MOV DPTR,#TABEL_1
      ACALL H1
CHAR2: LCALL PAGE1
      ACALL KOLOM4
      MOV DPTR,#TABEL_2
      ACALL H2
CHAR3: ACALL KOLOM43
      MOV DPTR,#TABEL_3
      ACALL H2
CHAR4: LCALL PAGE2
      ACALL KOLOM2
      MOV DPTR,#TABEL_4
      ACALL H3
CHAR5: ACALL KOLOM51
      MOV DPTR,#TABEL_5
      ACALL H3
CHAR6: LCALL PAGE3
      ACALL KOLOM2
      MOV DPTR,#TABEL_6
      ACALL H3
CHAR7: ACALL KOLOM51
      MOV DPTR,#TABEL_7
      ACALL H3
CHAR8: LCALL PAGE4
      ACALL KOLOM4
      MOV DPTR,#TABEL_8
      ACALL H2
CHAR9: ACALL KOLOM43
      MOV DPTR,#TABEL_9
      ACALL H2
CHAR10: LCALL PAGE5
      ACALL KOLOM13
      MOV DPTR,#TABEL_10
      ACALL H1
      RET
H1:   MOV R4,#30
      LCALL KIRIM_DATA
      RET
H2:   MOV R4,#9
      LCALL KIRIM_DATA
      RET
H3:   MOV R4,#3
      LCALL KIRIM_DATA
      RET
```

Gambar 3.15. Bagian Program Untuk Membuat Dan Menampilkan Lingkaran Pada *Display*.

Untuk membuat *pointer* (Pada kompas digital ini berfungsi sebagai jarum penunjuk arah mata angin modul CMPS-03), digunakan cara yang sama seperti dalam membuat lingkaran. Di bagian tengah dari gambar lingkaran pada Microsoft Paint tersebut dibuat sebuah gambar tanda panah. Kemudian diambil data-data *byte* berdasarkan gambar tanda panah tersebut dan selanjutnya data tersebut dikirimkan pada *display* sesuai dengan *page* dan kolom. Berikut adalah gambar *pointer*, *byte-byte* data yang dihasilkan, dan bagian program untuk membuat *pointer* tersebut pada *display*.



Gambar 3.16. Gambar *Pointer* pada Microsoft Paint.

```
NEEDLE: db 4,2,0FFH,2,4
```

Gambar 3.17. *Byte-Byte* Data dari Gambar *Pointer*.

```
NEEDLE1: LCALL PAGE2
          LCALL KOLOM26      ;pointer
          MOV DPTR,#NEEDLE
          LCALL ROT
```

Gambar 3.18. Bagian Program Membuat *Pointer* Pada *Display*.

3.2.3. Program Menampilkan Kata “Compass”, “CMPS03”, dan Karakter Derajat Pada *Display*

LCD grafik LPH 7779 tidak memiliki *on-board character-set*, sehingga untuk menampilkan berbagai karakter pada LCD tersebut, harus didesain terlebih dahulu bagaimana bentuk dan ukuran dari karakter-karakter yang hendak ditampilkan tersebut. Setelah bentuk dan ukuran dari karakter-karakter tersebut dibuat, kemudian diambil *byte-byte* data dari karakter dan kemudian ditampilkan pada LCD.

Berikut ini adalah desain karakter yang dibuat pada Microsoft Paint dan *byte-byte* data yang dihasilkan dari karakter-karakter tersebut.



Gambar 3.19. Bentuk Kata “Compass”, “CMPS03”, dan Karakter Derajat (°).

DATA_10:	db 06H,09H,09H,06H	;0(DERAJAT)
DATA_11:	db 38H,44H,44H,44H,20H	;c
	db 00H,38H,44H,44H,44H,38H	;o
	db 00H,7cH,04H,18H,04H,78H	;m
	db 00H,7cH,14H,14H,14H,08H	;p
	db 00h,20h,54h,54h,54h,78h	;a
	db 00H,48H,54H,54H,54H,20H	;s
	db 00H,48H,54H,54H,54H,20H	;s
DATA_12:	db 00h,38H,44H,44H,44H,20H	;c
	db 00H,7cH,04H,18H,04H,78H	;m
	db 00H,7cH,14H,14H,14H,08H	;p
	db 00H,48H,54H,54H,54H,20H	;s
	db 00H,38H,64H,54H,4cH,38H	;0
	db 00H,44H,54H,54H,7cH	;3

Gambar 3.20. *Byte-Byte* Data dari Kata “Compass”, “CMPS03”, dan Karakter Derajat (°).

Kemudian dari data yang diperoleh tersebut dikirimkan pada LCD dengan program sebagai berikut:

```

CHAR11: LCALL PAGE0
        LCALL KOLOM43      ;KATA "COMPASS"
        MOV R4,#41
        MOV DPTR,#DATA_11
        LCALL KIRIM_DATA
CHAR13: LCALL PAGE3
        LCALL KOLOM77      ;SIMBOL DERAJAT (°)
        MOV R4,#4
        MOV DPTR,#DATA_10
        LCALL KIRIM_DATA
CHAR12: LCALL PAGE5
        LCALL KOLOM43      ;KATA "CMPS03"
        MOV R4,#35
        MOV DPTR,#DATA_12
        LCALL KIRIM_DATA

```

Gambar 3.21. Bagian Program untuk Menampilkan Kata “Compass”, “CMPS03”, dan Karakter Derajat (°) pada LCD.

3.2.4. Program Pembacaan *Output* PWM Modul Kompas CMPS-03

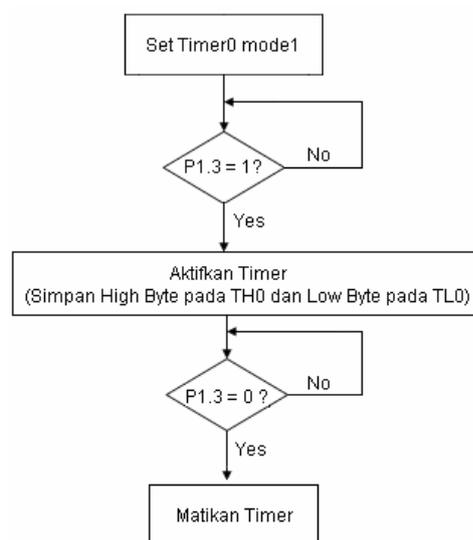
Seperti yang sudah diketahui sebelumnya bahwa modul CMPS-03 ini menghasilkan sinyal PWM, dengan lebar pulsa positif yang berubah-ubah sesuai

dengan besarnya sudut. Untuk membaca lebar pulsa positifnya digunakan *timer 0 mode 1* pada mikrokontroler AT89S53.

Untuk pembacaan PWM dari CMPS-03 digunakan pengaturan *timer* menggunakan *software*, yaitu *bit* TR0 sebagai penentu. *Bit* TR0 ini merupakan *bit addressable* (dapat diakses secara *bit*).

Jadi, untuk menjadikan *bit* TR0 menjadi kontrol, maka *bit* Gate harus berkondisi *low*, sehingga *bit* INTx tidak terpengaruh pada penentuan *timer*. *Bit* Gate ini terletak pada alamat 89_H, pada alamat 89_H ini tidak dapat dialamati secara *bit* (*not bit addressable*). Pada alamat 89_H ini, TMOD (*Timer Mode Register*) merupakan *register 8 bit* dengan fungsi sebagai pengatur mode *timer* atau mode *counter* yang akan digunakan.

Di sini akan digunakan *timer 0* dan mode 1, yaitu *timer* yang berfungsi sebagai 16 *bit timer* yang akan menghitung naik mulai dari 0000_H hingga FFFF_H. Hasil dari perhitungan disimpan pada *register* TL0 untuk *low byte* dan TH0 untuk *high byte*. Untuk lebih jelasnya dapat dilihat pada *flowchart* dari prosedur-prosedur pembacaan sudut modul CMPS-03 berikut ini.



Gambar 3.22. *Flowchart* Pembacaan *Output* PWM Modul CMPS-03.

Untuk dapat menggunakan *timer 0* mode 1 ini, harus dilakukan *setting* terlebih dahulu pada *register* TMOD yang memiliki alamat 89_H. Setelah itu *output* PWM pada modul CMPS-03 akan dideteksi oleh mikrokontroler AT89S53 pada

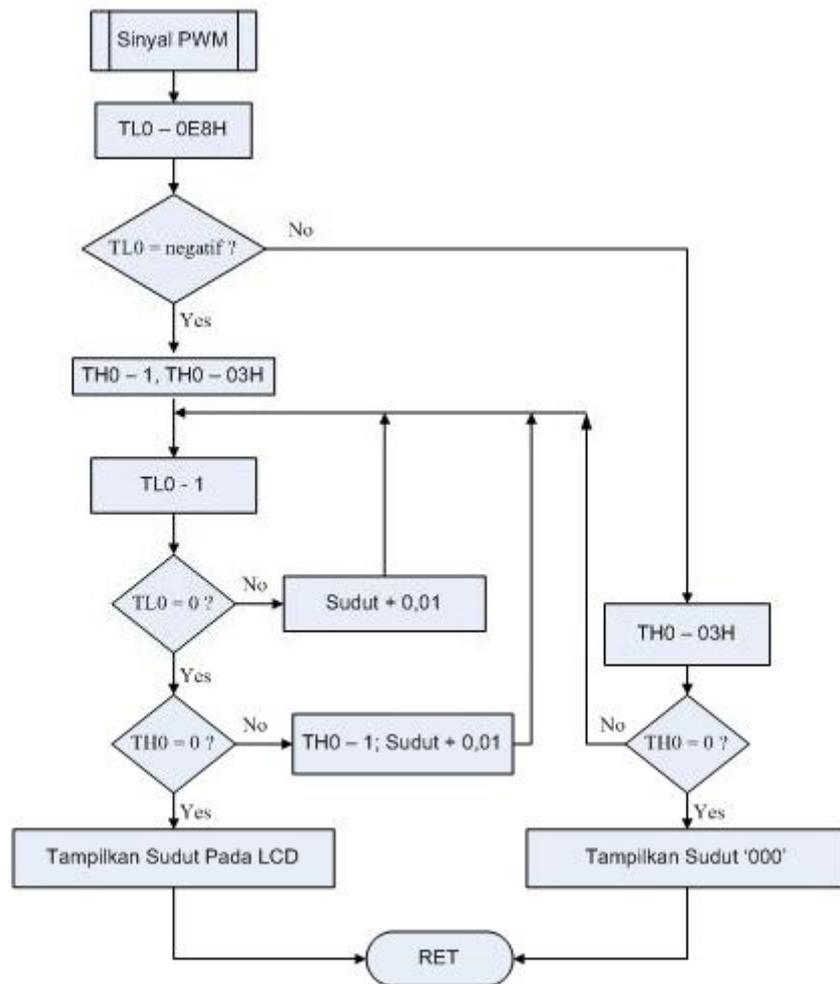
port 1.3, mikrokontroler akan menunggu sampai *port* 1.3 berkondisi *high*, pada saat *port* 1.3 berkondisi *high*, maka mikrokontroler akan mengaktifkan *timer* 0 dan akan menunggu sampai *port* 1.3 berkondisi *low*. Pada saat *port* 1.3 berkondisi *low*, maka mikrokontroler AT89S53 akan me-*non*-aktifkan *timer* 0, dan kemudian hasil perhitungan *timer*-nya disimpan pada *register* TH0 (untuk *high byte*-nya) dan pada TL0 (untuk *low byte*-nya). Berikut ini adalah bagian program untuk membaca *output* PWM dari modul CMPS-03.

	MOV TMOD,#01H	;Gunakan TIMER_0_MODE_1
	MOV TH0,#00H	;Reset TH0
	MOV TL0,#00H	;Reset TL0
LOCK2:	JB P1.3,LOCK2	;MENORMALKAN_PULSA
LOCK3:	JNB P1.3,LOCK3	;MENORMALKAN_PULSA
	SETB TR0	;AKTIFKAN_TIMER
LOCK4:	JB P1.3,LOCK4	
	CLR TR0	;MATIKAN_TIMER

Gambar 3.23. Bagian Program Pembacaan *Output* PWM Modul CMPS-03 Menggunakan Timer 0 Mode 1 AT89S53

3.2.5. Program Konversi Nilai TH0 dan TL0 Menjadi Besar Sudut

Setelah didapatkan nilai pada *register* TH0 dan TL0, kemudian nilai pada *register* ini dikonversikan menjadi sudut melalui beberapa proses perhitungan dan pengolahan data. Berikut adalah *flowchart* dari proses konversi nilai yang disimpan pada TH0 dan TL0 menjadi besar sudut.



Gambar 3.24. Flowchart Konversi Nilai TH0 dan TL0 Menjadi Sudut.

Untuk konversi *register* TH0 dan TL0 menjadi besar sudut kompas, pertama-tama nilai pada TL0 dikurangi dengan 0E8_H. Hal ini dikarenakan pulsa positif yang dihasilkan *output* PWM modul CMPS-03 harus dikurangi dengan 1000_D (03E8_H) untuk merepresentasikan besar sudut sebenarnya. Kemudian dicek apakah nilai pada TL0 negatif atau positif.

Jika negatif, maka TH0 di-*decrement* satu kali dan kemudian dikurangi dengan 03_H. Setelah itu dilakukan siklus pengurangan pada nilai TL0. Tiap satu kali pengurangan terhadap nilai TL0, maka besar sudut ditambah dengan 0.01. Apabila nilai *register* TL0 sudah bernilai 0, maka proses *decrement* dilakukan pada TH0. Akan tetapi tiap satu kali siklus *decrement* terhadap nilai TH0, maka dilakukan proses yang sebelumnya yaitu *decrement* terhadap nilai TL0 dan proses

pertambahan 0.01 derajat terhadap nilai sudut. Bila nilai pada *register* TH0 sudah nol maka hasil dari pengolahan data tersebut ditampilkan pada LCD.

Sedangkan, jika setelah proses pengurangan terhadap *register* TL0 dengan nilai 0E8_H menghasilkan nilai positif, maka *register* TH0 dikurangi dengan nilai 03_H. Dari hasil pengurangan tersebut kemudian di-cek apakah pada *register* TH0 tersebut bernilai 0. Bila bernilai 0, maka dapat dipastikan bahwa *output* PWM dari modul CMPS-03 tersebut merepresentasikan sudut 0°. Sehingga pada *display* LCD ditampilkan sudut '000°'. Sedangkan jika nilai pada *register* TH0 tidak nol, maka dilakukan proses yang sama seperti proses sebelumnya di atas yaitu pengurangan terhadap *register* TL0. Dan tiap satu kali pengurangan terhadap nilai TL0 maka besar sudut ditambah dengan 0.01. Jika nilai *register* TL0 sudah bernilai 0, maka proses *decrement* dilakukan pada TH0. Akan tetapi tiap satu kali siklus *decrement* terhadap nilai TH0, maka dilakukan proses yang sebelumnya yaitu *decrement* terhadap nilai TL0 dan proses pertambahan 0.01 derajat terhadap nilai sudut. Bila nilai pada *register* TH0 sudah nol, maka hasil dari pengolahan data tersebut ditampilkan pada LCD.

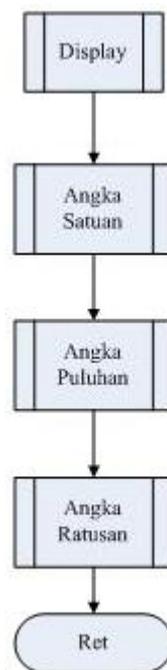
3.2.6. Program Menampilkan Besar Sudut Pada *Display* LCD Grafik LPH 7779

Untuk dapat menampilkan angka-angka pada *display* LCD LPH 7779, maka pertama-tama harus dibuat sebuah kumpulan *byte-byte* data sebagai pembentuk karakter-karakter angka mulai dari 0 sampai 9. Untuk membentuk karakter angka ini digunakan program aplikasi Microsoft Paint seperti halnya pada sub bab 3.2.2 dan 3.2.3. Berikut ini adalah *byte-byte* data dari karakter angka 0 sampai angka 9.

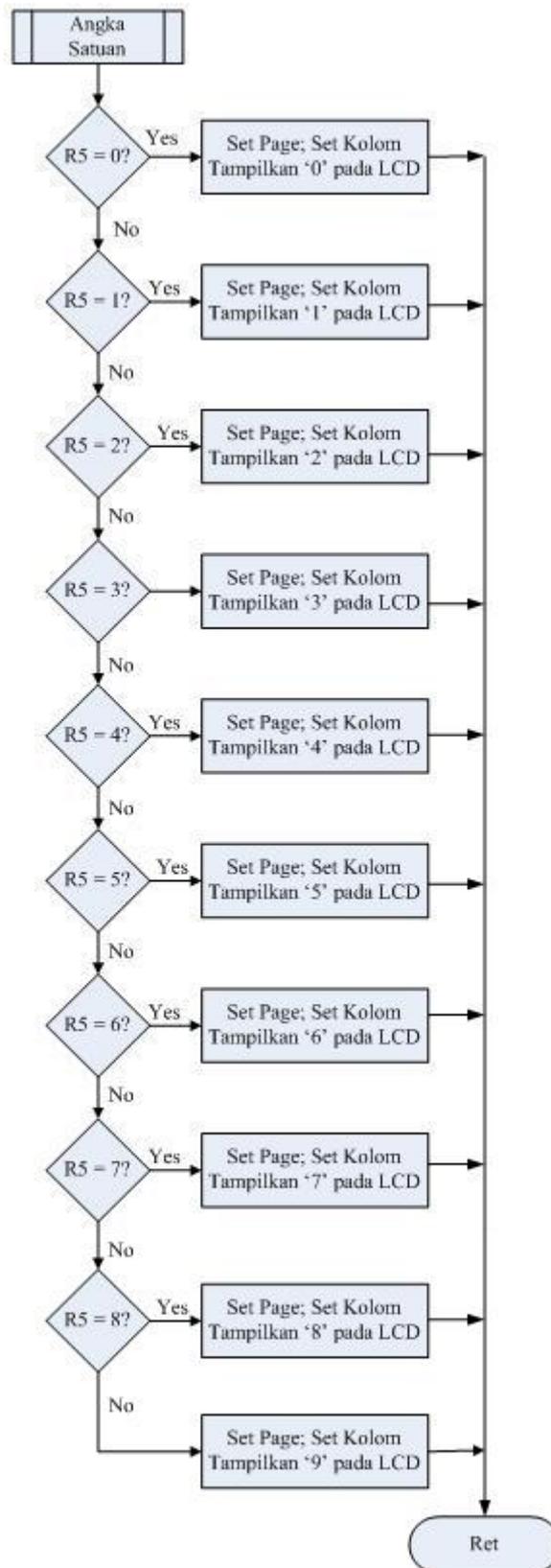
DATA_0:	db	3eH,51H,49H,45H,3eH	;0
DATA_1:	db	00H,42H,7fH,40H,00H	;1
DATA_2:	db	42H,61H,51H,49H,46H	;2
DATA_3:	db	21H,41H,45H,4bH,31H	;3
DATA_4:	db	18H,14H,12H,7fH,10H	;4
DATA_5:	db	27H,45H,45H,45H,39H	;5
DATA_6:	db	3cH,4aH,49H,49H,30H	;6
DATA_7:	db	01H,71H,09H,05H,03H	;7
DATA_8:	db	36H,49H,49H,49H,36H	;8
DATA_9:	db	06H,49H,49H,29H,1eH	;9

Gambar 3.25. *Byte-Byte* Data Pembentuk Karakter Angka.

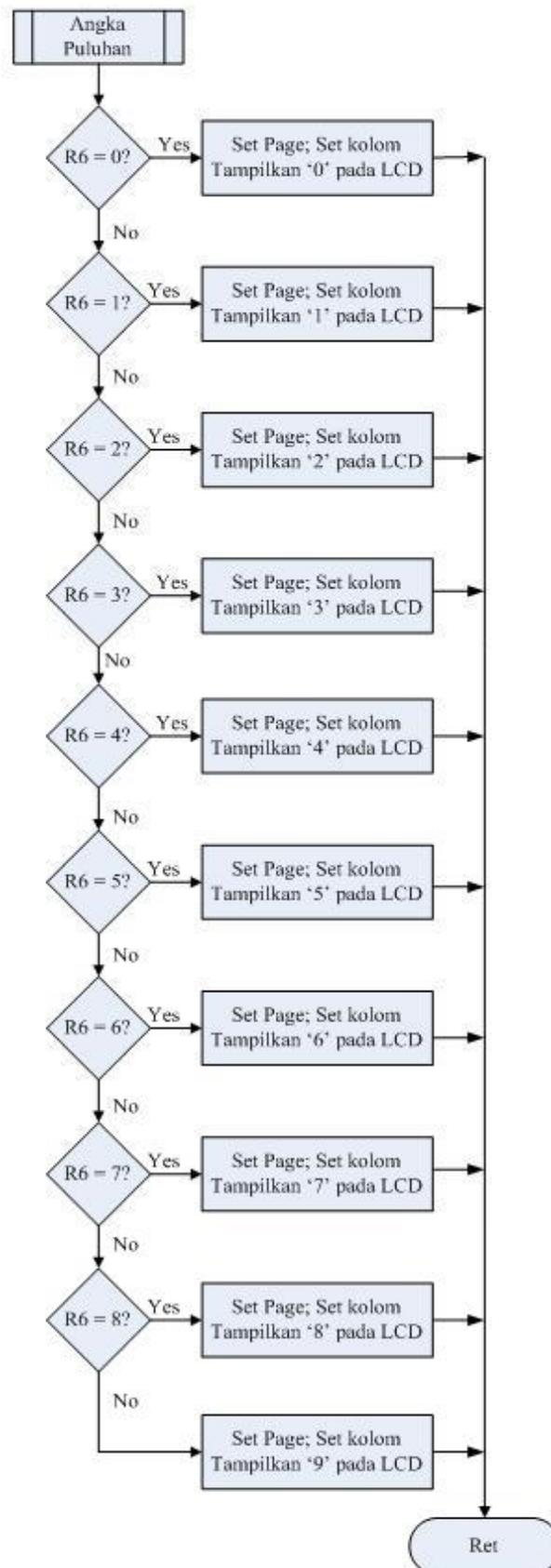
Kemudian dari data-data tersebut dipanggil dengan sub rutin tertentu dan selanjutnya dikirimkan melalui prosedur pengiriman data untuk ditampilkan pada LCD. Pada program kompas digital ini, digunakan *register-register* dari mikrokontroler AT89S53 untuk menyimpan data-data sebagai nilai besar sudut yang akan ditampilkan. *Register-register* tersebut adalah R7 untuk menampung data ratusan, R6 untuk menampung data puluhan, R5 untuk menampung data satuan, R4 untuk menampung data sepersepuluh, dan R3 untuk menampung data seperseratus. Selanjutnya, karena *display* angka pada kompas digital ini memiliki resolusi 1° , maka hanya nilai pada *register* R7, R6, dan R5 saja yang ditampilkan pada *display* LCD, yaitu ratusan, puluhan, dan satuan dengan proses membandingkan (*compare*) nilai pada *register-register* tersebut. Berikut ini adalah *flowchart* program untuk menampilkan nilai dari R7 (ratusan), R6 (puluhan) dan R5 (satuan) tersebut menjadi besar sudut pada *display*.



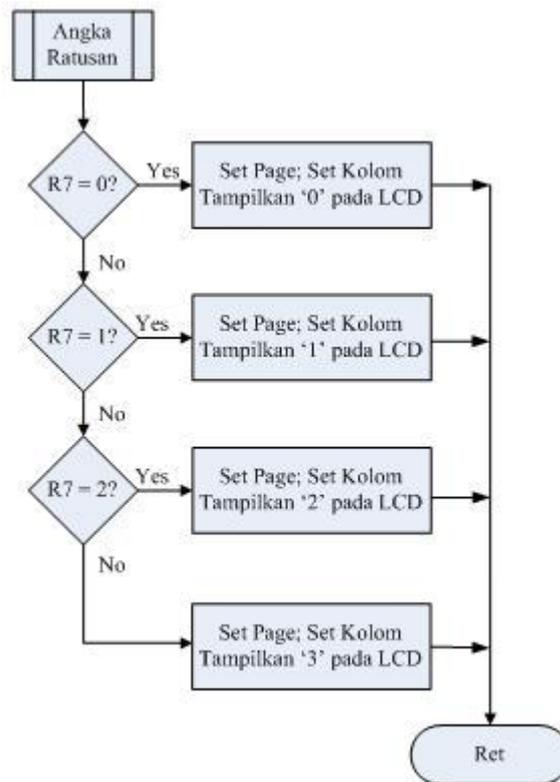
Gambar 3.26. *Flowchart* Program Menampilkan Angka Ratusan, Puluhan, dan Satuan Pada LCD Grafik



Gambar 3.27. Flowchart Program Menampilkan Angka Satuan.



Gambar 3.28. Flowchart Program Menampilkan Angka Puluhan.

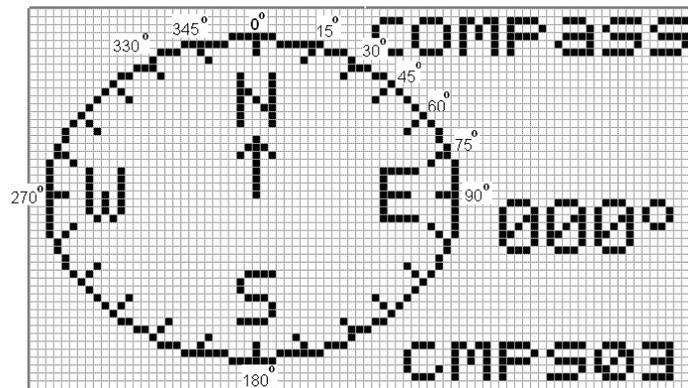


Gambar 3.29. Flowchart Program Menampilkan Angka Ratusan.

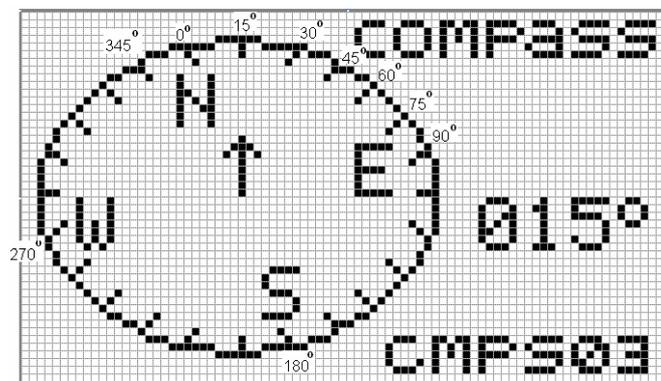
3.2.7. Program Menampilkan Huruf Penunjuk Arah Mata Angin.

Sebagai penunjuk arah mata angin, digunakan huruf-huruf yang mewakili empat arah penjuror mata angin pada *display*, yaitu N (Utara), E (Timur), S (Selatan), dan W (Barat).

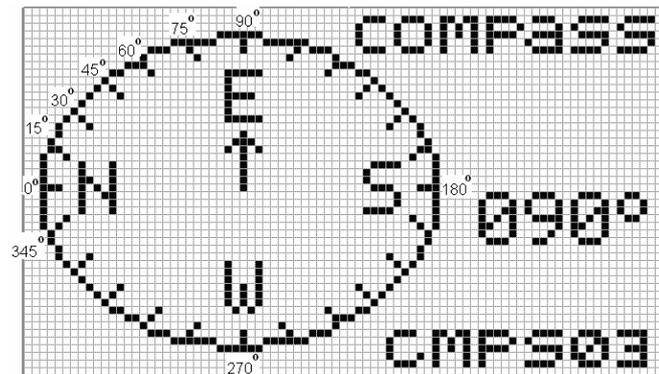
Huruf-huruf penunjuk arah mata angin yaitu N (Utara), E (Timur), S (Selatan), dan W (Barat) dibuat berdasarkan posisi empat arah mata angin tersebut tiap kenaikan sudut sebesar 15° (dimulai dari sudut 0°). Jadi, terdapat 24 gambar ($360^\circ/15^\circ = 24$) yang menunjukkan posisi empat arah penjuror mata angin tiap 15° . Untuk itu, terlebih dahulu perlu dibuat gambar bagaimana posisi empat arah penjuror mata angin tersebut pada tiap kenaikan sudut sebesar 15° . Sebagai media untuk membuat gambar ini digunakan program aplikasi Microsoft Paint. Berikut ini adalah contoh gambar yang dibuat berdasarkan posisi empat arah penjuror mata angin tersebut pada beberapa besar sudut.



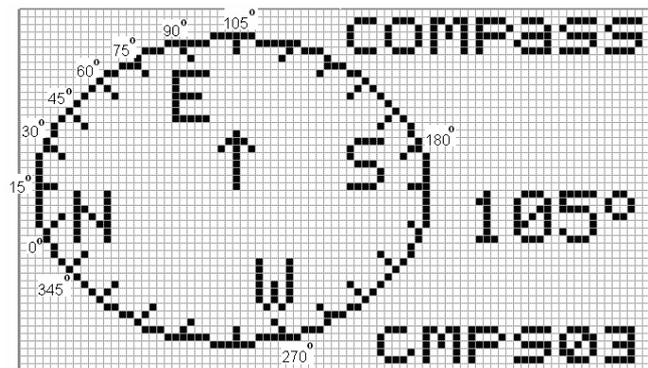
Gambar 3.30. Posisi Huruf Empat Arah Penjuru Mata Angin Pada Sudut 0°



Gambar 3.31. Posisi Huruf Empat Arah Penjuru Mata Angin Pada Sudut 15°



Gambar 3.32. Posisi Huruf Empat Arah Penjuru Mata Angin Pada Sudut 90°



Gambar 3.33. Posisi Huruf Empat Arah Penjuru Mata Angin Pada Sudut 105°

Dari gambar posisi huruf empat penjuru mata angin tersebut, yaitu posisi pada sudut-sudut 0° , 15° , 30° , 45° , 60° , 75° , 90° , 105° , 120° , 135° , 150° , 165° , 180° , 195° , 210° , 225° , 240° , 255° , 270° , 285° , 300° , 315° , 330° , dan 345° , selanjutnya diambil *byte-byte* data dan kemudian dimasukkan dalam suatu tabel *array* dalam format *assembler*. Dengan menggunakan proses *compare*, huruf-huruf tersebut kemudian ditampilkan pada LCD grafik LPH 7779.

Posisi huruf empat penjuru mata angin yang ditampilkan, berdasarkan *range* yang diambil dari nilai tengah tiap kenaikan sudut 15° yang telah dibuat. Contohnya, untuk sudut antara 0° dan 15° , nilai tengahnya berada antara sudut 7° dan 8° . Maka, untuk sudut $\leq 7^\circ$ (lebih mendekati sudut 0°), ditampilkan gambar posisi huruf empat penjuru mata angin pada sudut 0° . Sedangkan untuk sudut dengan *range* $8^\circ \geq \text{sudut} \leq 15^\circ$ (mendekati sudut 15°), ditampilkan gambar posisi huruf empat penjuru mata angin pada sudut 15° . Kemudian, untuk sudut antara 16° dan 30° , nilai tengahnya terdapat antara sudut 22° dan 23° . Maka, untuk sudut dengan *range* $16^\circ \leq \text{sudut} \leq 22^\circ$ (lebih mendekati 15°), ditampilkan gambar posisi huruf empat penjuru mata angin pada sudut 15° . Sedangkan untuk sudut dengan *range* $23^\circ \leq \text{sudut} \leq 30^\circ$ (lebih mendekati 30°), ditampilkan gambar posisi huruf empat penjuru mata angin pada sudut 30° . Begitu seterusnya dibuat *range* seperti demikian sampai sudut 359° . *Range* ini dibuat agar perubahan gambar posisi huruf empat penjuru mata angin yang ditampilkan menjadi lebih detail.

3.2.8. Program Penguncian Sudut (*Lock*)

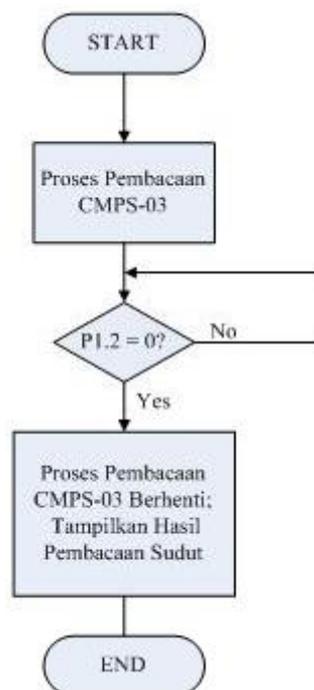
Selain terdapat tombol *reset*, kalibrasi, dan *backlight*, juga terdapat sebuah tombol untuk mengunci sudut (*lock*) yang ditampilkan pada *display*. Kunci

sudut ini berguna untuk kegiatan navigasi yaitu untuk menyesuaikan dengan arah perjalanan.

Program pada bagian penguncian sudut sebagai berikut, pertama-tama dilakukan proses pembacaan dan konversi sudut seperti yang sudah dijelaskan. Kemudian, setelah sudut ditampilkan pada LCD (apabila tidak ada penekanan pada tombol *lock*), maka program akan mengulangi proses pembacaan dan penampilan sudut ke *display* LCD grafik, dan demikian seterusnya.

Tetapi apabila ada penekanan terhadap tombol *lock*, maka *port* 1.2 akan berkondisi *low*, dan apabila *port* 1.2 berkondisi *low*, maka program pembacaan *output* PWM modul CMPS-03, proses konversi sudut, dan penampilan angka pada LCD akan dihentikan sementara pada nilai sudut terakhir yang ditampilkan.

Berikut ini adalah gambar *flowchart* program dari penguncian sudut.



Gambar 3.34. *Flowchart* Program Penguncian Sudut.