

3. PERANCANGAN SISTEM

Temperatur dan kelembaban merupakan 2 faktor yang paling signifikan dalam perancangan dan pengelolaan mesin tetas semimodern. Oleh karena itu untuk perancangan kontroler temperatur dan kelembaban yang memiliki standar industri perlu memperhatikan beberapa faktor seperti reliabilitas, fleksibilitas dan kemudahan penggunaan alat tersebut.

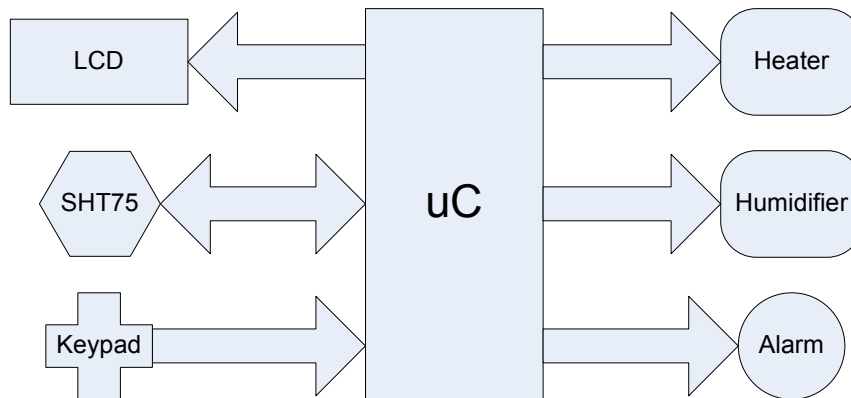
Pemilihan sensor dan *microcontroller* dalam perancangan alat ini memberi dampak yang besar terhadap kualitas alat ini. Sensor kelembaban & temperatur SHTxx memiliki banyak kelebihan yang membuatnya menjadi pilihan tepat untuk aplikasi ini. Pemilihan *microcontroller* yang menjadi otak dari kontroler ini jatuh pada Atmel ATmega8 yang memiliki performa dan fleksibilitas yang lebih tinggi dibandingkan keluarga MCS-51.

Pada awal perancangan alat ini memiliki 2 input yaitu sensor dan *keypad*, 3 output yaitu LCD, *heater*, *humidifier*. Untuk *heater* digunakan 8 buah lampu pijar 40 Watt dan untuk *humidifier* digunakan *ultrasonic water mister*. *Ultrasonic water mister* merupakan alat yang mampu mengubah air menjadi kabut dengan bergetar pada frekuensi yang sangat tinggi.



Gambar 3.1. Ultrasonic Water Mister

Dalam perancangan lebih lanjut ditambahkan fungsi *alarm* yang dapat berbunyi apabila *present value* temperatur 1°C lebih tinggi dari *set value* temperatur. Berikut ini adalah blok diagram sistem.



Gambar 3.2. Blok Diagram Sistem

Pada realisasinya nanti akan dibuat sebuah mesin penetas telur dengan fitur-fitur sebagai berikut:

- Kapasitas maksimal mesin tetas 576 butir telur.
- Dilengkapi LCD untuk menampilkan *present value* dan *set value* temperatur dan kelembaban dengan ketelitian 1 angka di belakang koma untuk *set value* dan 2 angka dibelakang koma untuk *present value*.
- Proses *input set value* yang mudah dengan *keypad* didukung dengan menu pengaturan *set value* yang mudah dimengerti pengguna. *Set value* yang telah dimasukkan akan disimpan pada internal RAM AVR.
- Desain kontroler yang kompak dan mudah dihubungkan (*wiring*) dengan perangkat lainnya.
- Mampu mempertahankan kondisi temperatur dan kelembaban sesuai dengan *set value* yang diinginkan. Untuk *setting* temperatur menggunakan histerisis 0,2°C (agar *heater* tidak menyala dan mati terlalu cepat).
- Dilengkapi *alarm* yang akan menyala apabila temperatur lebih tinggi 1°C dari *set value* temperatur (untuk mencegah matinya embrio telur akibat temperatur yang terlalu panas).
- Untuk mencegah embrio lengket pada sisi telur, mesin tetas ini dilengkapi mekanisme pemutaran telur dilakukan serempak pada semua rak *setter* (hanya dengan memutar tuas pemutar searah/berlawanan arah jarum jam). Mekanisme ini lebih efektif daripada memutar setiap butir telur secara manual.

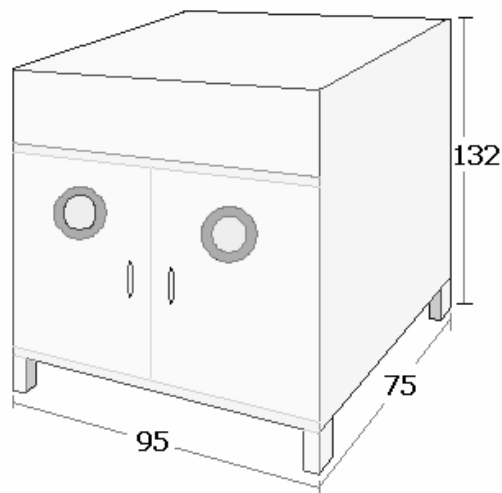
- Rak *setter* yang dapat dikeluarkan sehingga memudahkan pembersihan, perawatan dan perbaikan ruang inkubator.

3.1. Perancangan Mesin Penetas Telur

3.1.1. Ruang Inkubator

Ruang ini didesain berbentuk mirip almari dengan 2 pintu yang masing masing dilengkapi dengan jendela berbentuk lingkaran, sehingga memungkinkan untuk mengontrol keadaan di dalam ruangan inkubator. Kotak juga ini dilengkapi dengan beberapa lubang kecil pada bagian atasnya untuk sirkulasi udara.

Bahan yang digunakan untuk membuat ruangan ini adalah triplek dengan ketebalan 1 cm dan untuk kaki-kakinya menggunakan balok kayu. Ruangan ini memiliki dimensi panjang 95 cm, lebar 75 cm dan tinggi 132 cm.

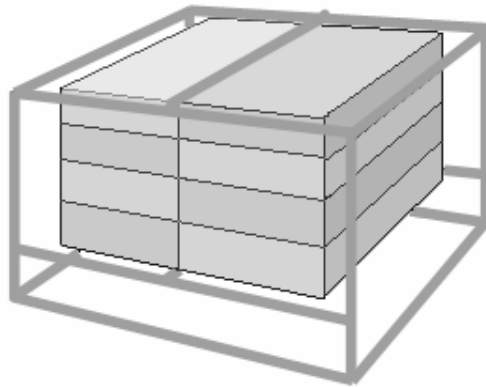


Gambar 3.3. Desain Ruang Inkubator

3.1.2. Kerangka Tempat Telur

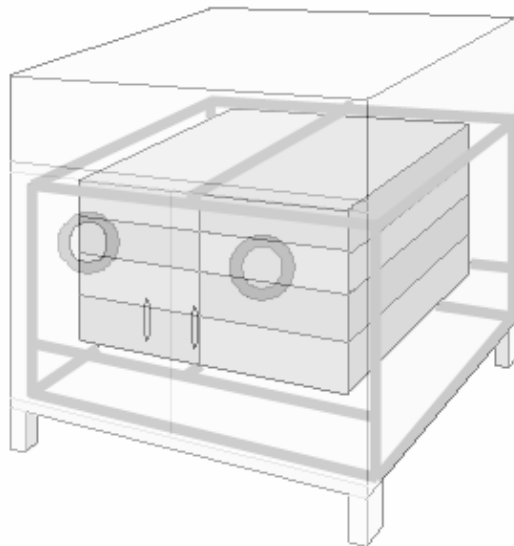
Kerangka ini didesain dengan bentuk yang menyerupai rak dengan tinggi 4 baris dan mempunyai 2 buah kolom. Tiap rak dapat diisi 2 buah *egg tray* dengan kapasitas 1 *egg tray* adalah 36 butir telur sehingga kapasitas total dari rak ini adalah $4 \times 2 \times 2 \times 36$ butir = 576 butir.

Kerangka ini dilengkapi dengan mekanisme pemutaran telur secara manual dengan cara mengubah sudut kemiringan dari 0° sampai 45° (CW/CCW) dari tiap rak secara bersamaan. Hal ini membantu untuk memudahkan proses pemutaran daripada harus memutar telur sebutir demi sebutir.



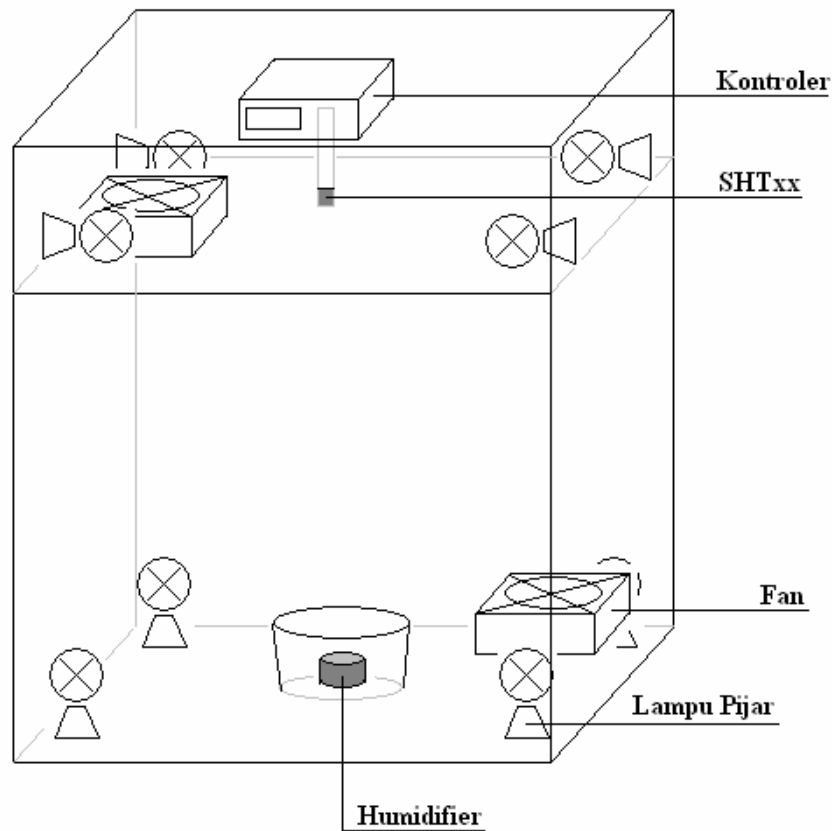
Gambar 3.4. Desain Kerangka Tempat Telur

Untuk memudahkan pembersihan dan perbaikan bagian dalam ruang inkubator, maka kerangka ini dibuat agar bisa dengan mudah keluar-masuk ruang inkubator. Sketsa posisi kerangka dalam ruang inkubator dapat dilihat pada gambar di bawah ini.



Gambar 3.5. Sketsa Posisi Kerangka dalam Ruang Inkubator

3.1.3. Perencanaan Tata Letak

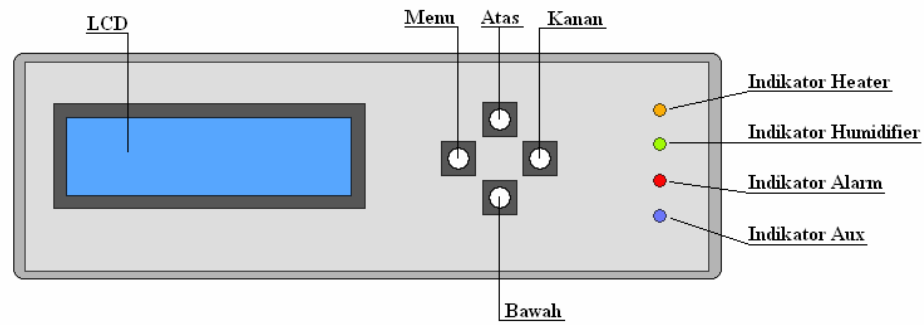


Gambar 3.6. Perencanaan Tata Letak

3.1.4. Layout Kontroler

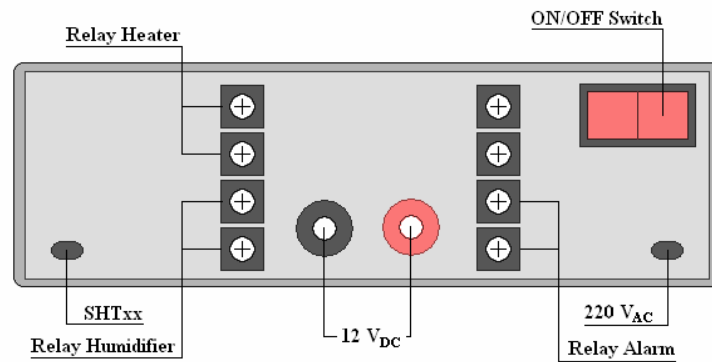
Rangkaian elektronik (*microcontroller*) dikemas dengan bahan kotak PVC dengan desain kompak, rapi dan mudah dihubungkan ke perangkat lain.

Pada bagian depan kotak akan terdapat LCD, *keypad*, lampu indikator untuk masing-masing perangkat yang terhubung.



Gambar 3.7. Desain Layout Depan

Pada bagian belakang kotak akan terdapat terminal *output* untuk *relay* yang mengatur masing-masing perangkat (seperti *heater*, *humidifier*, *alarm*), *output* 12 V_{DC}, lubang untuk *supply* listrik lubang untuk sensor, *switch* ON/OFF.



Gambar 3.8. Desain Layout Belakang

3.2. Perancangan Hardware

3.2.1. Sensor

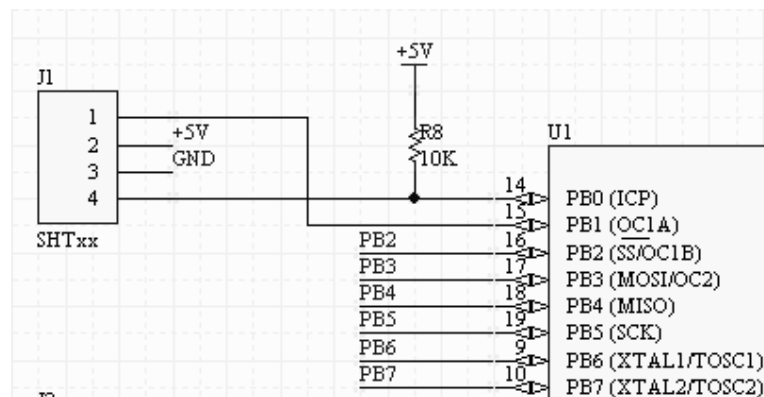
Sensor yang digunakan adalah SHT75 buatan SENSIRION yang menggunakan komunikasi *2-wire serial interface*. Sensor ini memiliki 4 pin yang terdiri dari SCK, VDD, GND, DATA seperti pada tabel di bawah.

Tabel 3.1. Konfigurasi Pin SHT75

Pin	Name	Comment
1	SCK	Serial clock input
2	VDD	Supply 2.4 – 5.5V
3	GND	Ground
4	DATA	Serial data bidirectional

Sumber: Sensirion. *SHT1x/SHT7x Humidity & Temperature Sensor*. 12 May 2006. <<http://www.sensirion.com/en/download/humiditysensor/SHT11.htm>>, p. 5

Untuk menjaga agar sinyal DATA dari sensor tidak *drop*, perlu dipasang *pull-up* resistor sebesar 10K. Untuk lebih jelas dapat dilihat pada gambar di bawah ini.

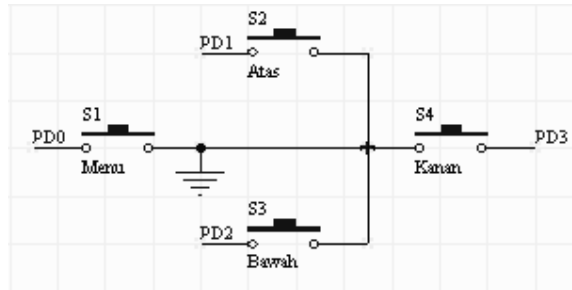


Gambar 3.9. Pemasangan Pull-up Resistor

3.2.2. Keypad

Untuk sarana *input* ke *microcontroller* digunakan *keypad* yang terdiri dari 4 buah *push button* yang masing masing berguna untuk menu, atas, bawah, kanan. Masing-masing tombol akan memberi *logic low* (langsung terhubung dengan GND), jika ditekan.

Dalam perancangan *keypad* pada AVR tidak diperlukan eksternal *pull-up* resistor untuk memberi *logic high* pada saat tombol tidak ditekan, sebab tiap *pin* AVR memiliki internal *pull-up* resistor yang dapat diaktifkan dengan mengatur DDR (*Data Direction Register*).



Gambar 3.10. Rangkaian Skematik Tombol

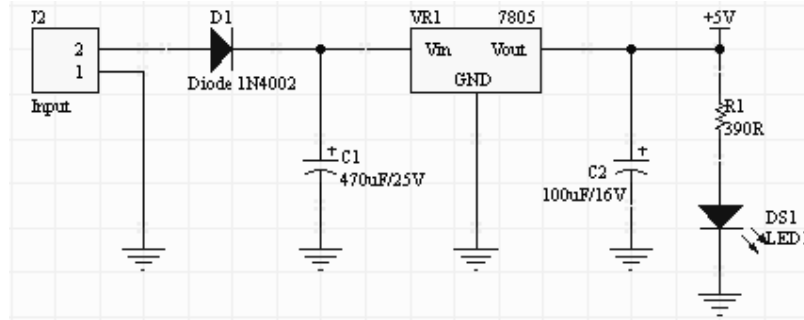
Fungsi masing-masing tombol adalah sebagai berikut:

- Tombol menu: digunakan untuk menampilkan menu pengaturan *set value* temperatur dan kelembaban.
- Tombol atas: digunakan untuk menambah 1 digit (dimana kursor berada) dari *set value*. Tombol ini hanya berfungsi di dalam menu pengaturan *set value*.
- Tombol bawah: digunakan untuk mengurangi 1 digit (dimana kursor berada) dari *set value*. Tombol ini hanya berfungsi di dalam menu pengaturan *set value*.
- Tombol kanan: digunakan untuk menggerakkan kursor ke kanan, untuk memilih satuan/puluhan/sepuluhan yang akan diubah nilainya. Tombol ini hanya berfungsi di dalam menu pengaturan *set value*.

3.2.3. Power Supply

Sumber tegangan rangkaian *microcontroller* diperoleh dari transformer 500mA dengan *input* 220V_{AC} dan *output* 12V_{AC} yang kemudian dihubungkan ke rangkaian *bridge rectifier* sehingga *output* menjadi 12V_{DC}.

Sebelum masuk ke rangkaian mikro tegangan 12V akan diturunkan menjadi 5V dengan menggunakan IC *voltage regulator* LM7805. Untuk menghindari kerusakan akibat polaritas terbalik, rangkaian ini dilengkapi diode.

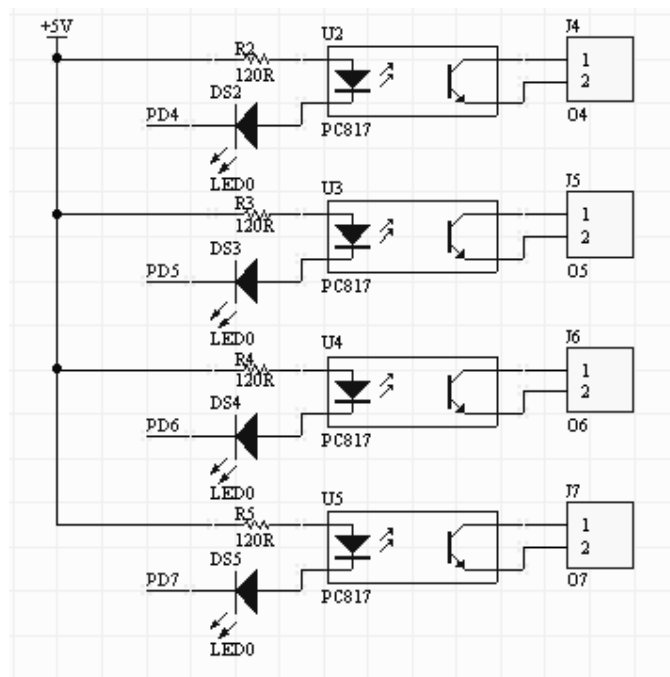


Gambar 3.11. Rangkaian Skematik Power Supply

3.2.4. Rangkaian Optocoupler

Untuk mengamankan rangkaian *microcontroller* dari arus balik yang ditimbulkan *relay*, maka pada setiap *output port* (yang nantinya digunakan untuk mengaktifkan *relay*) dipasang *optocoupler*.

Rangkaian pengaman ini juga dilengkapi dengan LED (*Light Emitting Dioda*) indikator yang digunakan untuk memastikan rangkaian *optocoupler* berjalan, sekaligus mengetahui status dari alat yang terpasang pada *output* kontroler (*heater, humidifier, alarm*).



Gambar 3.12. Rangkaian Skematik Optocoupler

Perhitungan resistor yang dipasang adalah sebagai berikut:

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

$$V_{LED} = 2,5 \text{ V}$$

$$V_{LED \text{ Opto}} = 1 \text{ V}$$

$$I_{LED} = 10 \text{ mA}$$

$$V_{CC} = V_{resistor} + V_{LED \text{ Opto}} + V_{LED}$$

$$V_{resistor} = V_{CC} - V_{LED \text{ Opto}} - V_{LED}$$

$$= 5\text{V} - 1\text{V} - 2,5\text{V}$$

$$= 1,5\text{V}$$

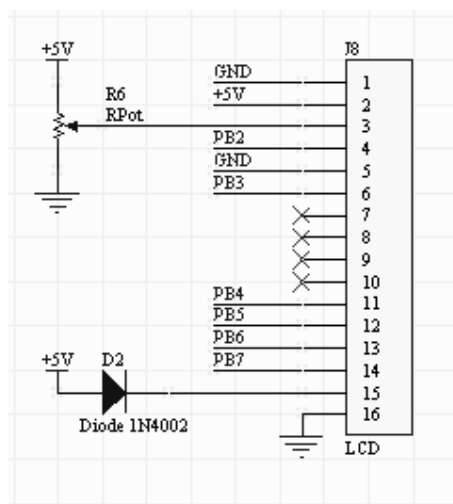
$$R_{resistor} = V_{resistor} / I_{LED}$$

$$= 1,5\text{V} / 10\text{mA}$$

$$= 150 \text{ Ohm}$$

3.2.5. Display

Untuk menampilkan hasil pembacaan temperatur dan kelembaban dari sensor digunakan LCD (*Liquid Crystal Display*) dengan ukuran 2x16. Selain menampilkan *present value*, *set value* temperatur dan kelembaban, LCD juga digunakan untuk menampilkan menu pengaturan *set value* temperatur dan kelembaban yang diinginkan. Koneksi LCD ke *port microcontroller* dapat dilihat pada gambar di bawah ini.



Gambar 3.13. Rangkaian Skematik LCD

3.3. Perancangan Software

Pada perancangan *software* untuk AVR digunakan *compiler* BASCOM-AVR versi 1.11.8.1 buatan MCS Electronics. BASOM-AVR merupakan *compiler* AVR yang menggunakan *high level language* yang mudah dimengerti.

Secara garis besar *software* dibagi menjadi beberapa bagian yaitu: bagian deklarasi awal, bagian *display*, akses menu dan akses sensor. Untuk memahami alur program lebih jelas dapat dilihat pada gambar *flowchart* program.

Bagian deklarasi awal mencakup beberapa bagian antara lain:

- Konfigurasi pin LCD

```
...
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 ,
Db6 = Portb.6 , Db7 = Portb.7 , E = Portb.3 , Rs = Portb.2
Config Lcd = 16 * 2
...
```

- Penentuan dimensi untuk masing-masing variabel

```
...
Dim Ctr As Byte
Dim Dataword As Word
Dim Command As Byte
Dim Dis As String * 20
...
```

- Pendeklarasian nilai konstanta

```
...
Const C1 = -4
Const C2 = 0.0405
Const C3 = -0.0000028
Const T1c = .01
Const T2 = .00008
Const T1f = .018
...
```

- Penamaan *port*

```
...
Sck Alias Portb.1
Dataout Alias Portb.0
Datain Alias Pinb.0
...
```

- Deklarasi *subroutine*

```
...
Declare Sub Getit()
Declare Sub Getsv()
...
```

- Konfigurasi *port* sebagai *input/output*

```
...
Ddrb = &B11111111
Config Pinb.0 = Output
Config Pinb.1 = Output
...
```

Bagian *display* berfungsi untuk menampilkan *present value* dan *set value* temperatur dan kelembaban pada LCD dengan format tertentu. Dalam perancangan *software* ini nilai temperatur dan kelembaban ditampilkan pada LCD dengan ketelitian 2 angka di belakang koma.

```

Dis = Fusing(tempc , "##.##")
Locate 1 , 1
Lcd Dis ; " C"

Dis = Fusing(rhlintemp , "##.##")
Locate 1 , 9
Lcd Dis ; " RH"

Dis = Fusing(svtemp , "##.##")
Locate 2 , 1
Lcd Dis ; " C"

Dis = Fusing(svhumi , "##.##")
Locate 2 , 9
Lcd Dis ; " RH"

```

Sebelum temperatur dan kelembaban ditampilkan, data yang diperoleh dari sensor terlebih dahulu diolah terlebih dahulu dari data digital ke nilai fisik. Untuk temperatur nilai fisik yang diperoleh (dari rumus) merupakan nilai yang sudah linear, namun untuk kelembaban perlu dilakukan kompensasi linearitas untuk memperoleh nilai fisik yang linier dan kompensasi pengaruh temperatur terhadap kelembaban untuk memperoleh nilai RH yang sebenarnya. Data temperatur dan kelembaban yang ditampilkan pada LCD akan diperbarui tiap detik.

```

Tempc = T1c * Dataword
Tempc = Tempc - 40

Calc = C2 * Dataword
Calc2 = Dataword * Dataword
Calc2 = C3 * Calc2
Calc = Calc + C1
Rhlinear = Calc + Calc2

Calc = T2 * Dataword
Calc = Calc + T1c
Calc2 = Tempc - 25
Calc = Calc2 * Calc
Rhlintemp = Calc + Rhlinear

```

Untuk akses menu dilakukan dengan memeriksa apakah ada penekanan tombol menu, apabila ada penekanan maka program akan langsung meloncat ke label Menu1.

```

If Menu = 0 Then
Gosub Menu1
End If

```

Menu1 akan menampilkan data *set value* dari temperatur yang dapat diubah dengan menekan tombol atas/bawah untuk menambah/mengurangi nilai dari digit

dimana kursor berada, tombol kanan digunakan untuk menggeser posisi kursor ke kanan 1 digit. Data *set value* temperatur yang baru (vt) akan disimpan ke variabel svtemp dan nilai svtemp akan disimpan pada internal RAM AVR (svt), sehingga data *set value* tidak hilang (jika sumber listrik padam).

```
...
Dim Svt As Eram Single
...
Svtemp = vt
Svt = Svtemp
...
```

Apabila tombol menu ditekan lagi, maka akan berpindah ke Menu2.

```
If Menu = 0 Then
Goto Menu2
End If
```

Menu2 akan menampilkan data *set value* dari kelembaban yang dapat diubah dengan cara yang sama dengan cara mengatur *set value* temperatur. Data *set value* kelembaban yang baru (vt) akan disimpan ke variabel svhumi dan nilai svhumi akan disimpan juga pada internal RAM AVR (svh).

```
...
Dim SvH As Eram Single
...
Svhumi = vt
Svh = Svhumi
...
```

Penekanan tombol menu akan membuat program kembali ke tampilan awal, yaitu penampilan *present value* dan *set value* temperatur dan kelembaban.

```
...
If Menu = 0 Then
Return
End If
...
```

Akses sensor diawali dengan melakukan sekuen *Connection Reset* dengan cara membiarkan *pin DATA high* dan menjalankan *clock SCK* sebanyak 9 kali atau lebih diikuti dengan sekuen *Transmission Start*.

```
Set Dataout
For Ctr = 1 To 12
Set Sck
waitus 2
Reset Sck
waitus 2
Next Ctr
```

Untuk mengirimkan perintah pada SHTxx, nilai *binary* dari perintah tersebut dimasukkan pada variabel *command*, diikuti dengan pemanggilan *subroutine* Getit.

```
...
Command = &B00000011
Call Getit
...
```

Subroutine Getit diawali dengan sekuen *Transmission Start* selanjutnya dilakukan pengiriman variabel *command* secara *serial* dengan perintah *shiftout* diikuti dengan melakukan *clock SCK* sebanyak 1 kali sebagai *ACK bit*.

```
...
Set Sck
Reset Dataout
Reset Sck
Set Sck
Set Dataout
Reset Sck

Shiftout Dataout , Sck , Command , 1

Ddrb = &B11111110
Config Pinb.0 = Input

Set Sck
Reset Sck
Waitus 10
...
```

Setelah mengirim *ACK bit*, harus menunggu beberapa saat sampai pembacaan selesai dilakukan sensor. Selesaiannya perhitungan oleh sensor ditandai dengan nilai *logic* pin DATA menjadi *low*, selanjutnya hasil pembacaan dapat diminta secara *serial* dengan perintah *shiftin*. Karena data perhitungan memiliki besar 1 *word* (14 *bit* untuk temperatur, 12 *bit* untuk kelembaban) dan *right justified*, maka 2 *bit* pertama bernilai 0 untuk temperatur, 4 *bit* pertama untuk kelembaban bernilai 0. Data akan dikirim oleh sensor menjadi 2 bagian yaitu *high byte* diikuti *low byte* (MSB dikirim dahulu), setelah menerima *high byte* (disimpan pada variabel *Databyte*) harus dilakukan *clock SCK* sekali sebagai *ACK bit* dan dilanjutkan dengan penerimaan *low byte* (disimpan pada variabel *Databyte*).

```
...
Bitwait Pinb.0 , Reset
Shiftin Datain , Sck , Databyte , 1
Datavalue = Databyte

Ddrb = &B11111111
Config Pinb.0 = Output

Reset Dataout
Set Sck
Reset Sck

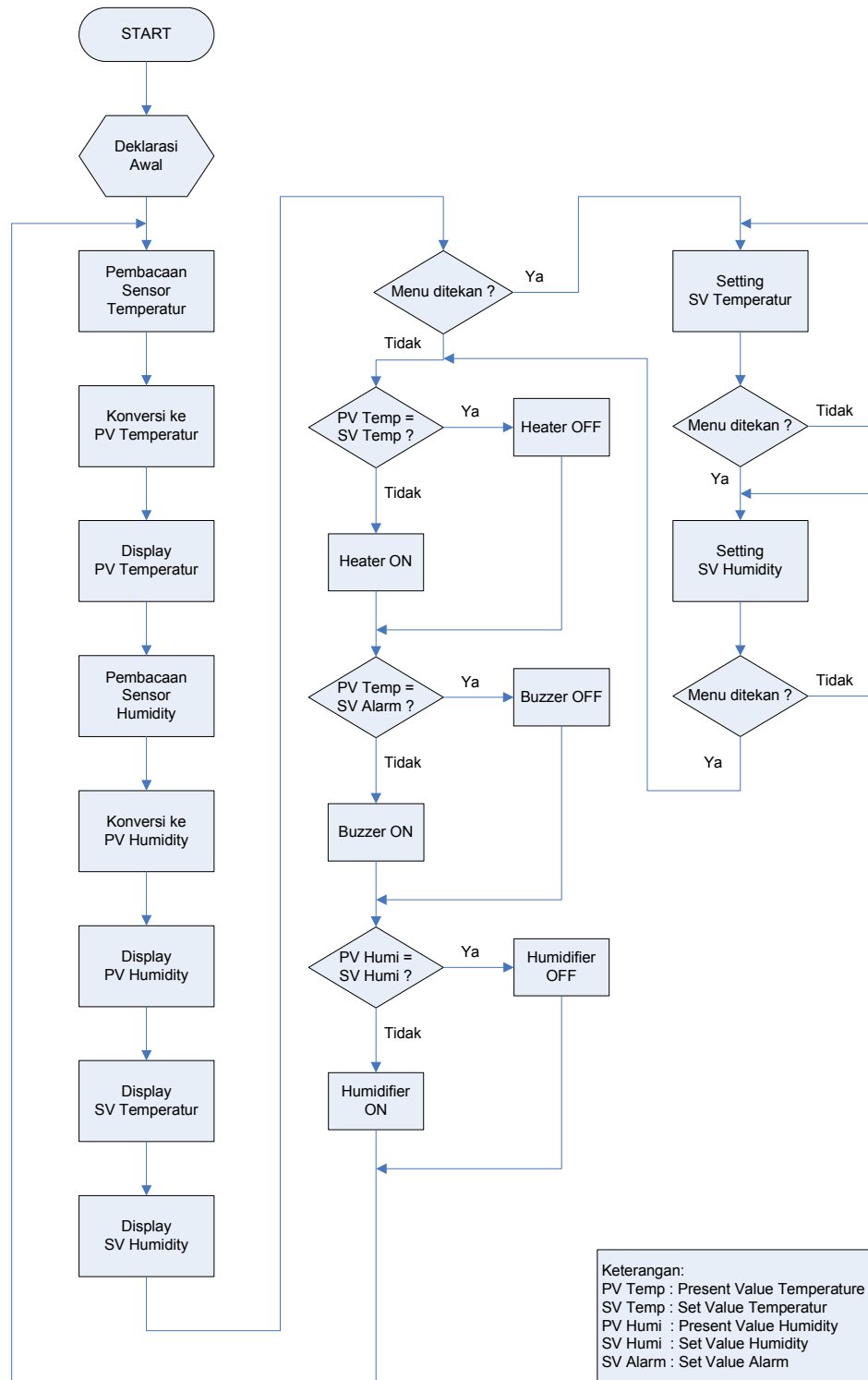
Ddrb = &B11111110
Config Pinb.0 = Input

Shiftin Datain , Sck , Databyte , 1
...
```

Penggabungan 2 *byte* data tersebut dilakukan dengan cara memindahkan *Databyte (high byte)* pada variabel *Datavalue* (berukuran 1 *word*), sehingga data pada *Databyte* akan menjadi *low byte* pada *Datavalue (high byte* pada *Datavalue* sekarang berisi '00000000'), menggeser data pada *Datavalue* ke kiri sebanyak 8 kali (*high byte* pada *Datavalue* sekarang berisi *Databyte*) dan melakukan operasi logika OR antara *Datavalue* dengan *Databyte (low byte)*.

```
...
Shift Datavalue , Left , 8
Datavalue = Datavalue Or Databyte
...
```

Pada *software* ini dilengkapi dengan pembacaan CRC-8 *Checksum* setelah pembacaan data temperatur/kelembaban, namun data CRC ini tidak akan digunakan lebih lanjut.



Gambar 3.14. Flowchart Program