

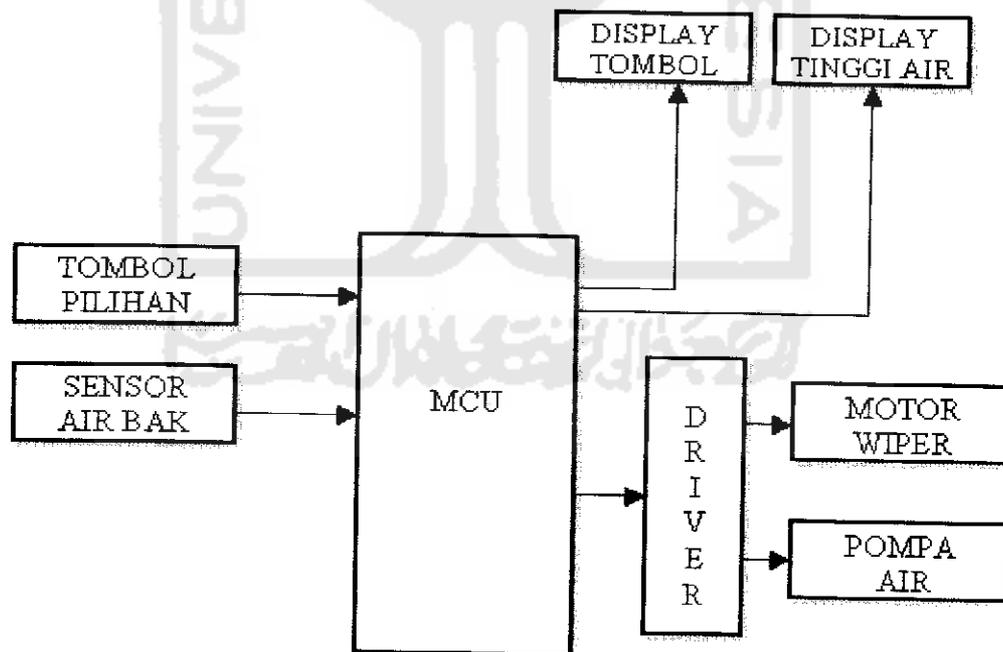
BAB III PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Hardware yang dibangun adalah sebuah sistem pengendali dan *monitoring* pada sebuah sirkulasi air. Sistem ini menggunakan mikrokontroler MC68HC908KX8 sebagai pusat pengendali.

Sistem pengendali sirkulasi air ini terdiri dari beberapa rangkaian pengendali yang diantaranya rangkaian *keypad*, rangkaian sensor air, rangkaian mikrokontroler, rangkaian *display*, dan rangkaian *driver* untuk motor dan pompa.

Secara umum cara kerja sistem dapat dijelaskan sebagai berikut:



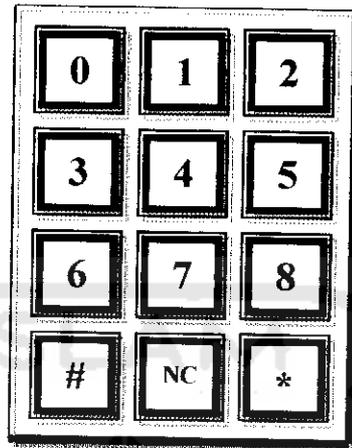
Gambar 3.1. Diagram Blok Sistem

Gambar 3.1 merupakan diagram blok dari keseluruhan sistem. Proses dimulai dari pembacaan sensor air yang ditempatkan pada bak air, dimana sensor ini berupa sebuah probe-probe yang dipasang secara berurutan dari bawah sampai atas bak dengan tingkatan *level* 0 sampai dengan 8. Keluaran dari sensor air ini diumpankan pada mikrokontroler yang pada akhirnya akan ditampilkan di *display* tinggi air. Selain membaca *level* dari sensor air, mikrokontroler juga melihat permintaan *level* air yang diinginkan sesuai dengan pilihan *level* dari tombol pilihan yang ditampilkan pada *display* tombol. Selama proses pengendalian mikrokontroler akan selalu melihat *display* tampilan, dimana angka pilihan tersebut akan digunakan sebagai acuan bagi mikrokontroler untuk mempertahankan kondisi *level* air sesuai dengan permintaan.

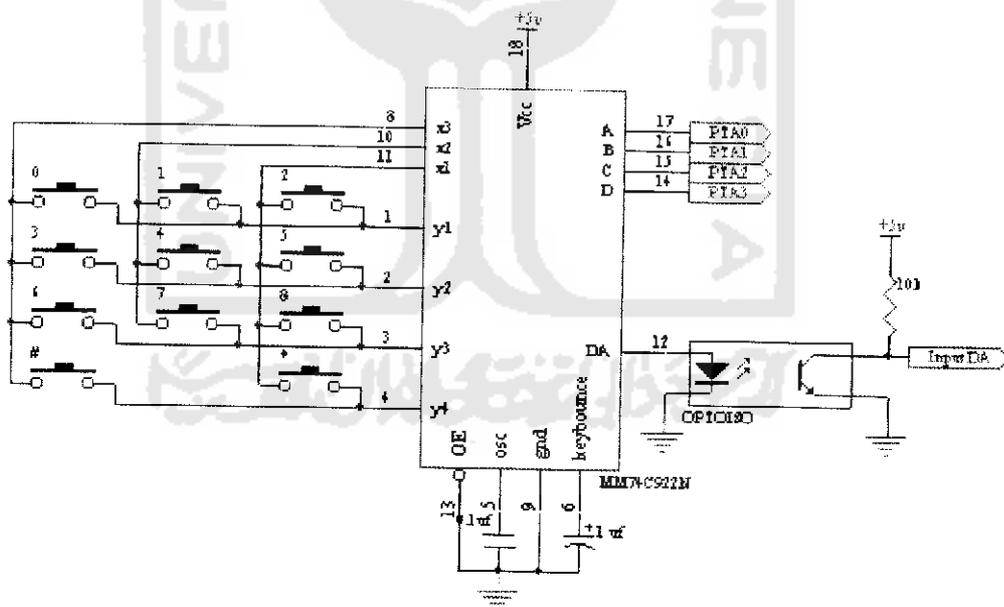
3.2. Perancangan Perangkat Keras

3.2.1. Perancangan Tombol Pilihan

Tombol pilihan dibuat dengan sebuah IC *keypad* dengan nomor seri MM74C922N. IC ini merupakan IC *keypad* dengan 4 bit keluaran berupa kode-kode biner sesuai dengan tombol yang ditekan. Dalam pembuatan, IC ini dikonfigurasi untuk 12 tombol yang ditata dalam bentuk matrik 3x4 dan diberi keterangan 0,1,2,...,8, serta 2 buah simbol, yaitu : # dan *. Susunan tombol *keypad* dapat terlihat pada gambar 3.2.



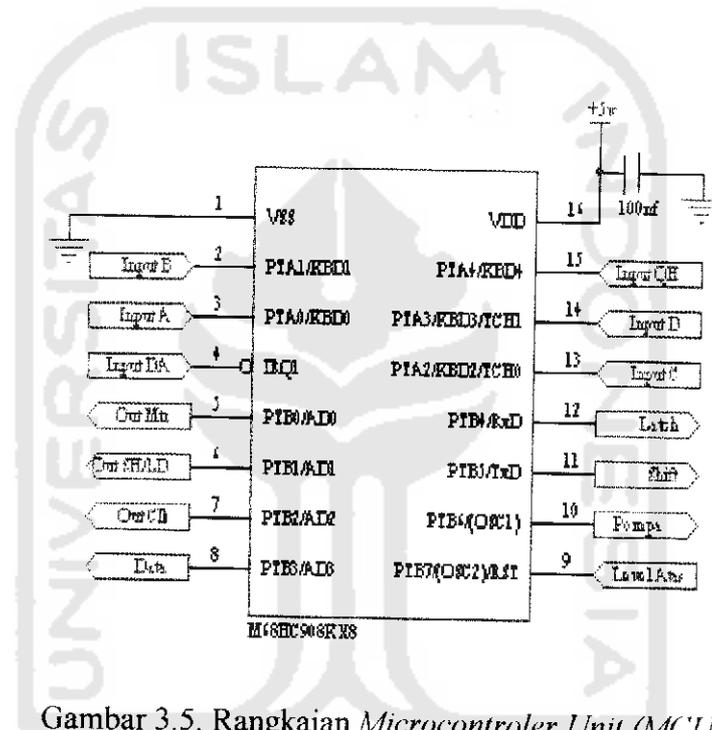
Gambar 3.2. Susunan Tombol Keypad



Gambar 3.3. Rangkaian tombol Keypad

3.2.3. Perancangan *Microcontroller Unit (MCU)*.

Sistem pengendali yang dibuat menggunakan mikrokontroler keluaran motorola dengan nomor seri MC68HC908KX8. Gambar 3.5 merupakan gambar ke-16 pin dari MC68HC908KX8 yang masing-masing pinnya sudah difungsikan untuk pengendalian seluruh sistem.



Gambar 3.5. Rangkaian *Microcontroller Unit (MCU)*

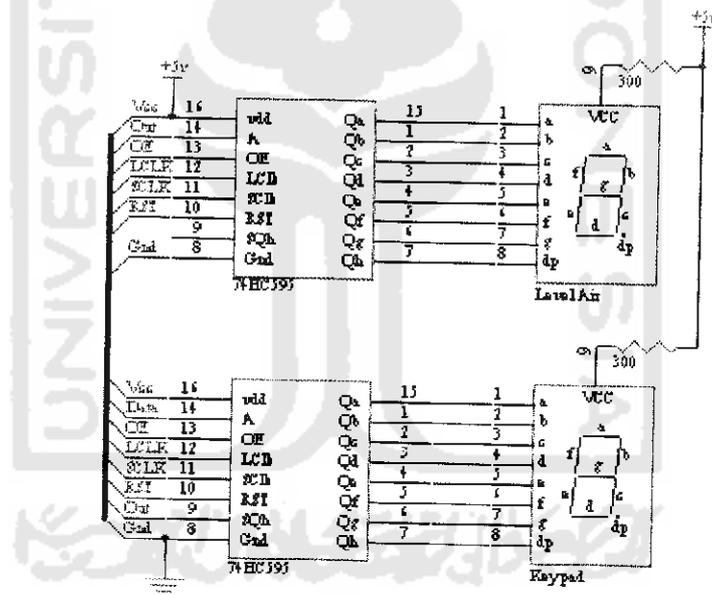
Secara keseluruhan semua pin dari MC68HC908KX8 tersebut difungsikan sebagai *port Input Output (I/O)*, dimana *PortA0,1,2,dan 3* sebagai masukan untuk menerima data 4 bit dari keluaran IC *keypad MM74C922N*, sedangkan *PortA4* difungsikan sebagai input untuk menerima data serial dari sensor air. Pin *IRQ* difungsikan sebagai tanda bagi mikrokontroler untuk mengambil data 4 bit dari IC *MM74C922N* ketika ada tombol ditekan.

PortB3,4,5 sebagai data keluaran untuk tampilan *display* pada *shift register 74HC595*. *PortB1,2* difungsikan untuk mengakses data masukan dari

sensor air. *PortB7* digunakan untuk mendeteksi air pada level teratas, dan *PortB6* digunakan untuk menghidup matikan pompa air.

3.2.4. Perancangan *Display*

Dalam pembuatan *display* digunakan 2 buah *7-segment common anoda* sebagai tampilan *level* air dan tampilan tombol pilihan *level*. Untuk pengiriman data digunakan IC *shift register 74HC595*, dimana IC ini berfungsi mengubah data serial menjadi data paralel.



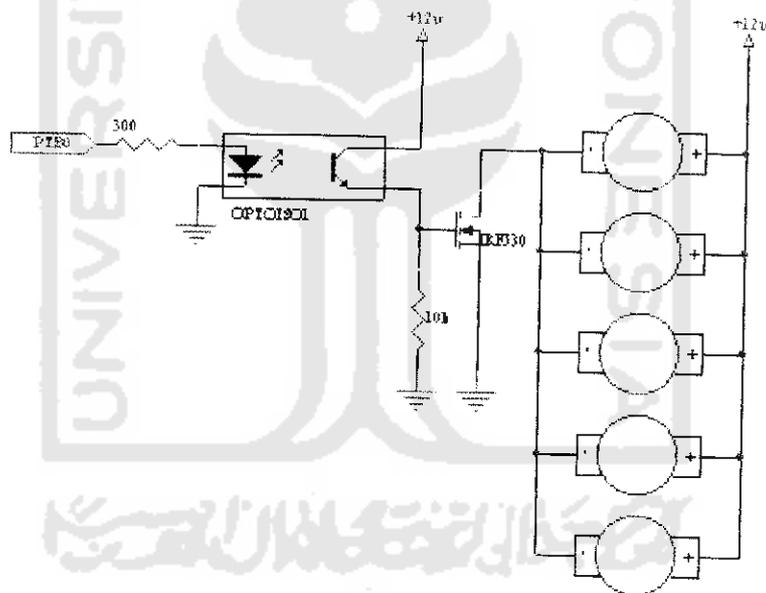
Gambar 3.6. Rangkaian *Display*

3.2.5. Perancangan Rangkaian *Driver*

Rangkaian *driver* terdiri dari 2 bagian. Bagian pertama digunakan untuk *driver* motor *washer* dan bagian kedua digunakan untuk *driver* pompa air

3.2.5.1. Rangkaian *Driver* Motor *Washer*

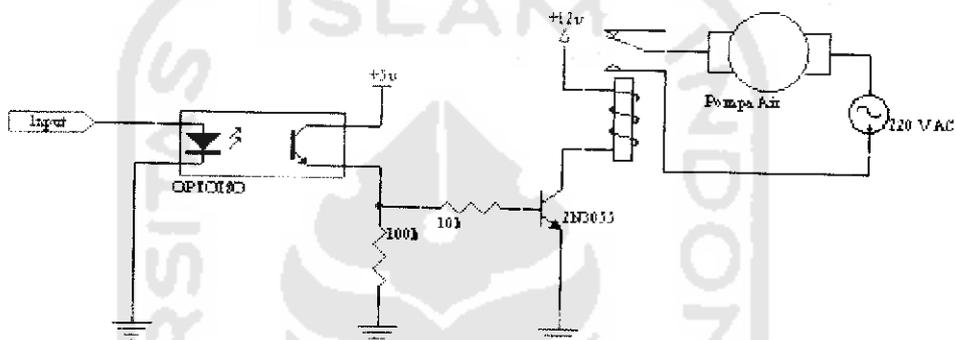
Dalam perencanaan digunakan 5 buah motor *washer* yang terhubung paralel dan untuk mengendalikannya digunakan sebuah MOSFET IRF 530 yang mempunyai rating arus keluaran sampai 14 Ampere dan sebuah *optocoupler* yang berfungsi untuk isolator antara mikrokontroler dengan MOSFET.



Gambar 3.7. Rangkaian *Driver* Motor *Washer*

3.2.5.2. Rangkaian *Driver* Pompa

Rangkaian *driver* pompa air menggunakan relay untuk memutus dan menyambungkan tegangan 220 Volt AC ke pompa. Relay diaktifkan oleh sebuah transistor yang difungsikan sebagai *switch* transistor.



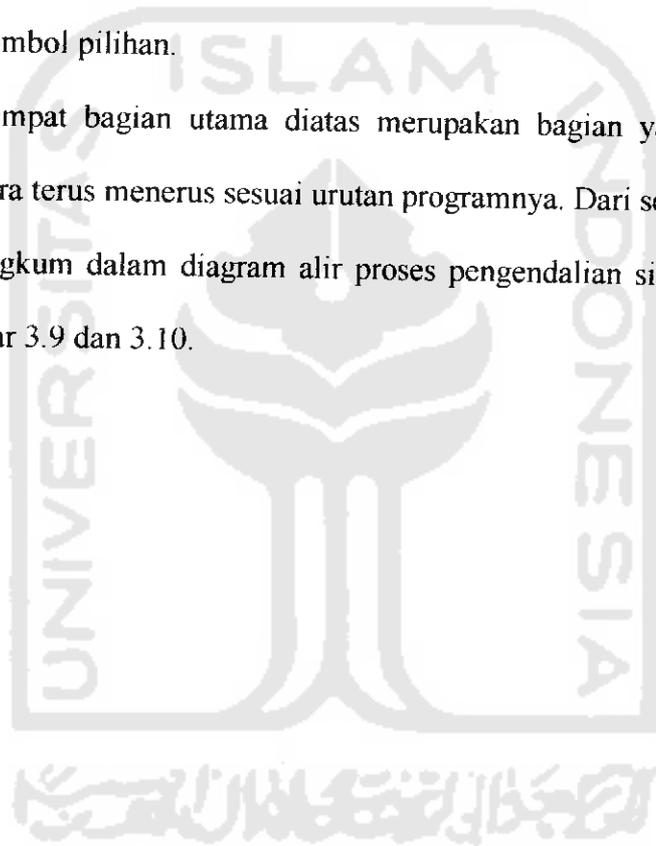
Gambar 3.8. Rangkaian *driver* pompa air

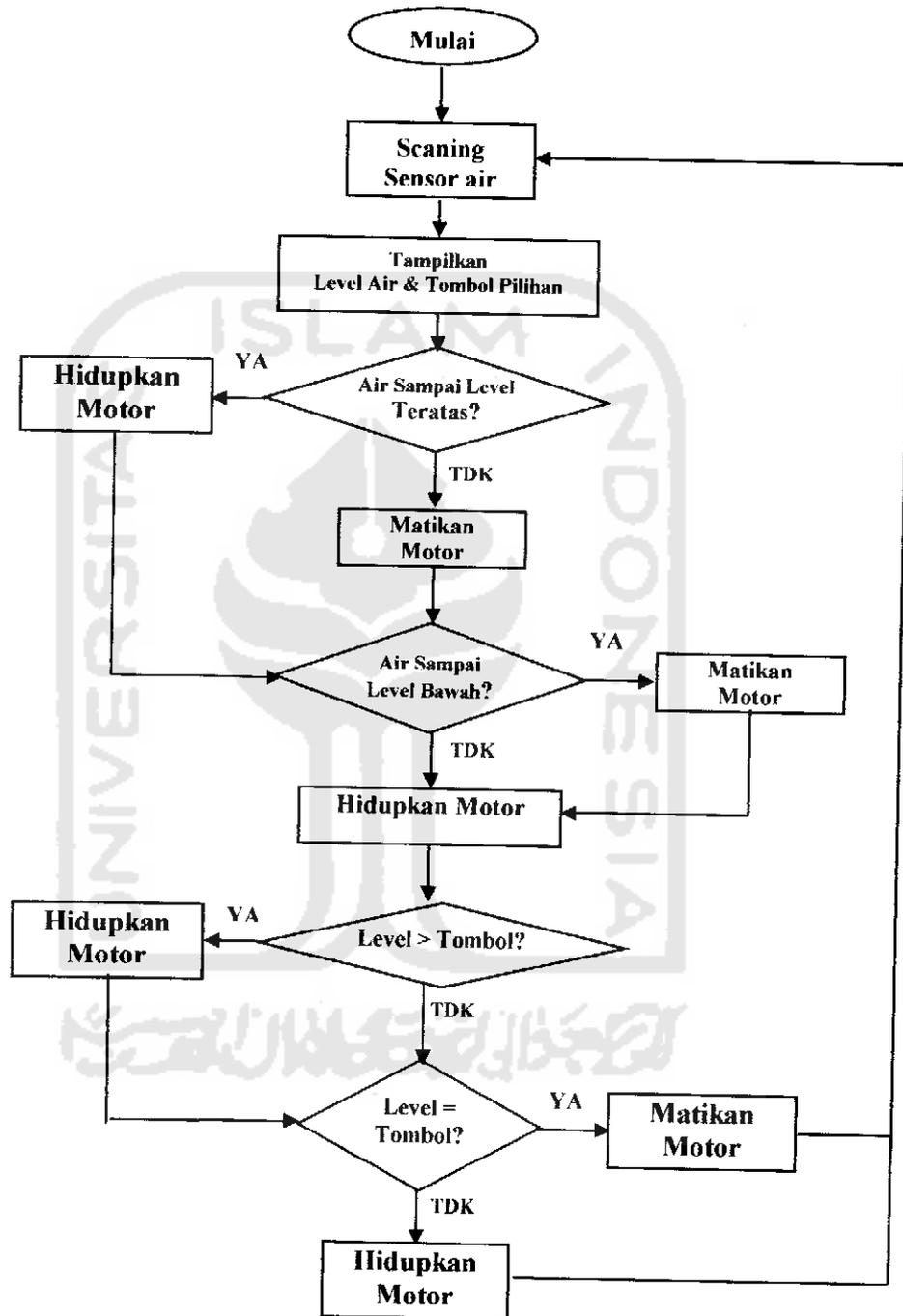
3.3. Perangkat Lunak

3.3.1. Diagram Alir Proses Pengendalian Sistem

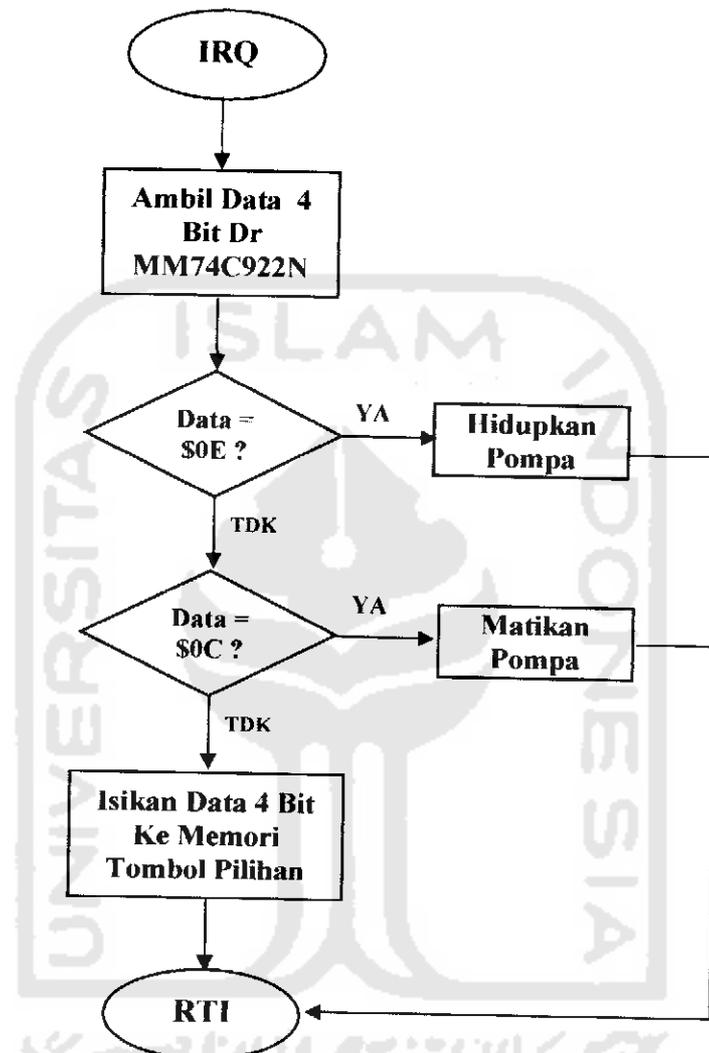
Proses pengendalian sistem yang dikerjakan mikrokontroler dapat dibagi menjadi empat bagian utama, yaitu: Proses scanning sensor air, menampilkan tombol pilihan serta level air, proses membuat level air konstan, dan proses scanning tombol pilihan.

Keempat bagian utama diatas merupakan bagian yang dikerjakan oleh sistem secara terus menerus sesuai urutan programnya. Dari semua kegiatan diatas dapat dirangkum dalam diagram alir proses pengendalian sistem seperti tampak pada gambar 3.9 dan 3.10.





Gambar 3.9. Diagram alir program utama



Gambar 3.10. Diagram alir proses scanning tombol pilihan

Setelah sistem dijalankan maka akan dilakukan proses pengambilan data paralel sensor dari ic 74HC165 dan hasil scanning dimasukkan ke memori 1 byte di *level* air. Setelah melakukan scanning data paralel mikrokontroller akan menampilkan data paralel tersebut ke *display* sebagai tampilan *level* air dan dilanjutkan menampilkan tombol pilihan. Setelah kedua proses tersebut dilaksanakan mikrokontroller akan mengecek kondisi ketinggian air, jika sudah sampai ke *level* teratas maka motor akan dihidupkan dan air akan dikurangi sampai ke *level* terendah dan ketika nyampai ke *level* terendah motor akan kembali dimatikan.

Jika ada permintaan ketinggian air dari tombol pilihan, maka sistem akan melihat apakah ketinggian air melebihi permintaan jika ya motor dihidupkan dan jika ketinggian air sama dengan permintaan motor akan dimatikan.

3.3.2. Pewaktuan

Agar dapat melaksanakan semua kegiatannya, maka mikrokontroller memerlukan referensi waktu yang baik dan tepat. Dengan referensi waktu ini maka mikrokontroller dapat membagi waktu kapan melakukan suatu kegiatan dapat melakukan berbagai macam kegiatan sekaligus.

Mikrokontroller menghasilkan sendiri pulsa yang berfungsi untuk menjalankan perintah dan proses pewaktuan. Pulsa ini dihasilkan dari bagian *Internal Clock Generator*. Frekuensi dari pulsa ini dapat diatur melalui program, yaitu dengan memberikan nilai pada register ICGCR, dan ICGMR sebagai perkaliannya.

Register ICGCR sudah bernilai “\$08” saat reset dan ini sesuai dengan yang dikehendaki yaitu mengaktifkan *internal clock* dan pulsa berasal dari dalam mikrokontroler.

Register ICGMR diberi nilai “100” sehingga *internal clock* menghasilkan pulsa dengan frekuensi 30,72 MHz. Nilai frekuensi ini didapat dari nilai $100 \times 307,2$ KHz (frekuensi dasar dari *internal clock*). Konfigurasi *internal clock* dapat dilihat di potongan program berikut:

```
;Inisialisasi Internal clock generator
MOV      #$08,ICGCR
MOV      #!100,ICGMR
```

3.3.3. Bagian Proses Scaning Sensor Air

Proses scanning dilakukan terus menerus dan diambil dengan cara mengeksekusi ic *shift register* 74HC165 pada portB1, portB2, dan portA4. Untuk batas level atas digunakan portB7.

Agar seluruh port dapat digunakan sebagai *input/output* (I/O) maka port-port tersebut harus diatur terlebih dahulu. Pertama adalah mengatur port-port tersebut sebagai I/O, yaitu dengan memasukkan nilai “\$00” ke dalam *register* DDRA (*Data Direction Register A*) dan nilai “\$7F” ke dalam *register* DDRB (*Data Direction Register B*).

Potongan program yang mengatur port tersebut sebagai I/O:

```
;Setup Port I/O
MOV      #%00000000,DDRA
MOV      #%01111111,DDRB
```

Setelah portB1, dan portB2 sebagai *output* serta portA4 sebagai *input* maka dilakukan proses pengambilan data dari ic 74HC165, ini dilakukan dalam potongan program :

```

;Rutin Mengambil Data 74HC165
SPI_Paralel  CLR          Data_Paralel
              BITCLR      Shift_Paral
              BITCLR      Clk_Paral
              BITSET      Clk_Paral
              BITSET      Shift_Paral

              BRASET      Data_Paral,Hi_bit
              BITCLR      Data_Bit
              LSL          Data_Paralel

Detak1
Paral2      MOV          #!6,Jml_Data
              BITCLR      Clk_Paral
              BITSET      Clk_Paral
              BRASET      Data_Paral,Hi_bit2
              BITCLR      Data_Bit
              LSL          Data_Paralel

Detak2      DEC          Jml_Data
              BNE          Paral2
              BITCLR      Clk_Paral
              BITSET      Clk_Paral
              BRASET      Data_Paral,Hi_bit3
              BITCLR      Data_Bit

Out         BRA          Tampil_Level

Hi_Bit      BITSET      Data_Bit
              LSL          Data_Paralel
              BRA          Detak1

Hi_Bit2     BITSET      Data_Bit
              LSL          Data_Paralel
              BRA          Detak2

Hi_Bit3     BITSET      Data_Bit
              BRA          Out

```

Pertama memori untuk penyimpanan data 1 byte data paralel sama dengan 0 kemudian memberi logika "1" pada serial/paralel load sehingga IC berfungsi sebagai paralel. Kemudian diberi clock dengan tujuan data paralel dari input

dimasukkan ke register di dalam IC. Setelah itu serial/paralel load diberi logika "0" sehingga IC berfungsi sebagai serial dan kemudian diberi clock sebanyak 8 kali sehingga data input 8 bit dikeluarkan secara serial lewat QH yang terhubung dengan portA4. Data serial ini diumpankan ke mikrokontroler dan disimpan dalam memori 1 byte yang dinamai Data_Paralel.



3.3.4. Bagian Penampil Level Air Dan Tombol Pilihan

Agar dapat memonitor seluruh proses pengendalian *level* air maupun tombol pilihan maka digunakan 2 *display* penampil dengan dilengkapi IC *shift register* sebagai pendorong data serial. Proses penampilan dilakukan dalam potongan program :

```

;Rutin untuk menampilkkan level air & Tombol
Show      LDA      Level
          JSR      SPI
          LDX      Tombol
          LDA      Tabel_Keypad,X
          JSR      SPI
          LDA      Tabel_Desimal,X
          STA      Keypad
          JSR      Delay

*****
*Sub Rutin Untuk SPI serial_in/paralel_out      *
*****
SPI       MOV      #!8,Data_Serial
Shift_Seri BITCLR   Sclk_SPIseri
          LSLA
          BCS      SPI_set
          BITCLR   Data_SPIseri
          BRA      SPI2
SPI_set   BITSET   Data_SPIseri

SPI2:    BITSET   Sclk_SPIseri
          BITCLR   Sclk_SPIseri
          DBNZ    Data_Serial,Shift_Seri
          BITSET   Lclk_SPIseri
          NOP
          BITCLR   Lclk_SPIseri
          RTS

```

3.3.5. Bagian Proses Membuat Air Konstan

Dalam proses selalu dideteksi *level* air agar sesuai dengan permintaan dan mempertahankan ketinggian airnya. Proses tersebut dilakukan dalam potongan program :

;Rutin untuk mengendalikan motor supaya level air konstan

	BRACLR	Batas_Atas, Terus
	BITSET	Motor
Terus	LDA	Keypad
	CBEQA	#\$09, Out_Scan
	CBEQ	Level_2, OFF_Motor
	SUB	Level_2
	BMI	ON_Motor
OFF_Motor	JSR	Delay
	JSR	Delay
	JSR	Delay
	BITCLR	Motor
Out_Scan	LDA	Level
	CMP	#\$C0
	BNE	Out2
	BITCLR	Motor
Out2	JMP	MAIN_LOOP

Setelah mengambil data level dan keypad kemudian dibandingkan apakah *level* lebih dari permintaan jika ya maka air akan dikurangi dan jika *level* dan *keypad* sama maka motor dimatikan.

3.3.6. Bagian *Scanning* Tombol Pilihan

Untuk pilihan *level* air diperlukan sebuah permintaan dari luar. Hal ini dilakukan dengan menekan tombol pilihan di luar. Proses pengenalan tombol pilihan dilakukan dengan potongan program seperti berikut:

```

*****
*Sub rutin utk External Interupt (Keypad)
*****
IRQ      BITSET      ACK1
          MOV         PortA,Data_Tombol
          BITCLR     Data_5
          BITCLR     Data_6
          BITCLR     Data_7
          BITCLR     Data_8
          LDA        Data_Tombol
Lanjut2  CMP         #$0E
          BNE       Lanjut
          BITSET     Pompa
          BRA        Out_IRQ

Lanjut   CMP         #$0C
          BNE       Lanjut1
          BITCLR     Pompa
          BRA        Out_IRQ

Lanjut1  MOV         Data_Tombol,Tombol
Out_IRQ  RTI

```