

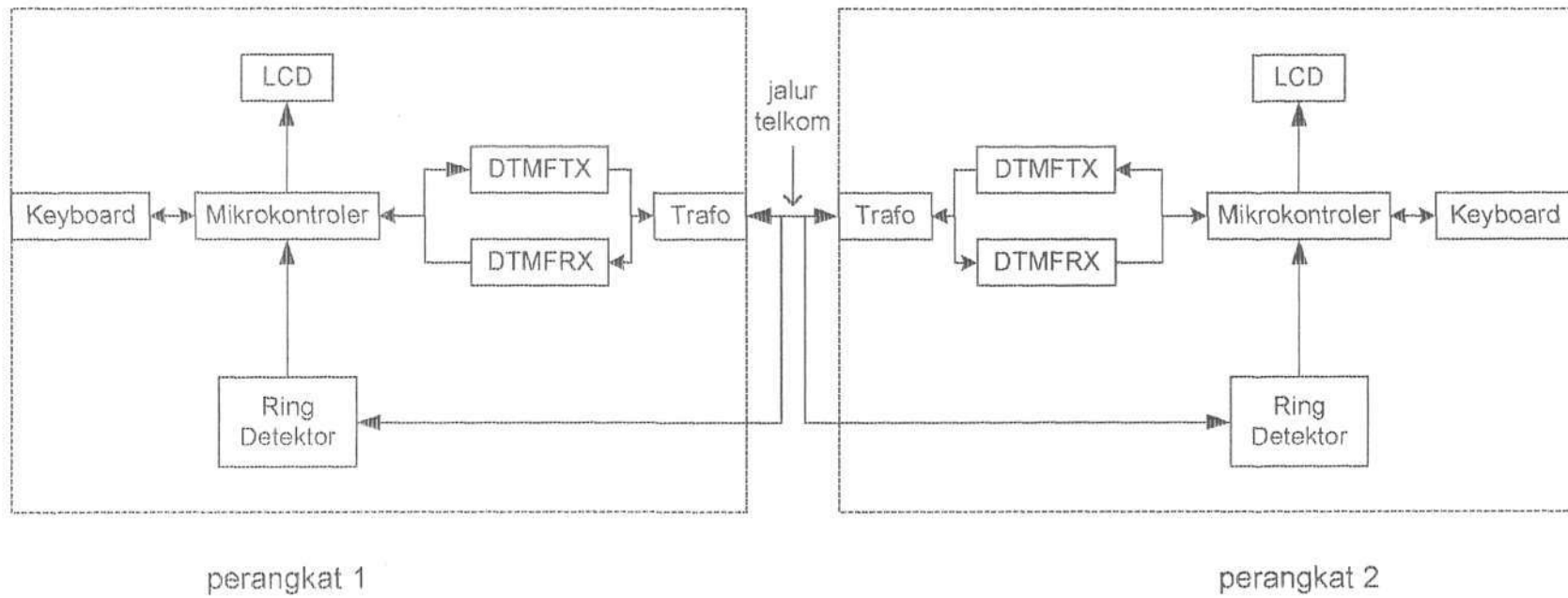
III. PERANCANGAN HARDWARE DAN SOFTWARE

1. CARA KERJA HARDWARE

Perancangan *hardware* dalam pembuatan alat pengirim dan penerima pesan melalui jalur telepon kabel ini, digunakan komponen-komponen piranti keras sebagai berikut :

- Minimum sistem
- LCD display
- Keyboard
- Penerima dan pemancar DTMF
- Ring detektor
- Trafo

Secara umum, hubungan antara relasi dari masing-masing komponen penyusun piranti pengirim pesan ini akan dijelaskan dalam blok diagram dalam gambar 3.1 berikut..



Gambar 3.1
Blok Diagram

Data pesan berupa karakter-karakter yang akan dikirim, di-*input*-kan melalui *keyboard* kemudian mikrokontroler menampilkannya ke dalam LCD dan sebelum dilakukan proses pengiriman mikrokontroler akan menerjemahkan data karakter dalam bentuk ASCII ke dalam kode DTMF. Sebelum proses pengiriman pesan berlangsung sistem terlebih dahulu men-*dial* nomor tujuan dan menunggu penerima siap menerima pesan.

1.1 Hubungan Mikrokontroler Dengan Keyboard



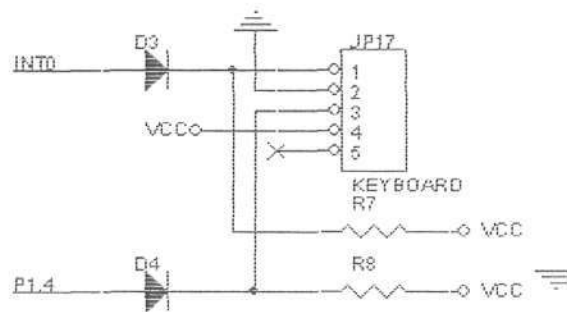
Keyboard PC (Personal Computer) digunakan sebagai *input* dalam penulisan pesan, dipilihnya *keyboard* sebagai *input* karena *keyboard* mempunyai keunggulan yaitu I/O yang dibutuhkan lebih sedikit dibandingkan jika menggunakan *keypad*.

Bila menggunakan *keypad* sebagai *input* maka akan membutuhkan I/O yang lebih banyak, untuk *keypad* dengan konfigurasi 4 X 3 dibutuhkan 7 I/O dan untuk konfigurasi 4 X 4 dibutuhkan 8 I/O. *Keypad* juga mempunyai keterbatasan dalam penyediaan *switch* yang mana jauh lebih sedikit dibandingkan dengan *keyboard*.

Keyboard hanya memerlukan dua buah pin dari mikrokontroler sehingga masih tersedia *pin-pin* untuk kebutuhan yang lain. Transmisi data dari *keyboard* ke MCS51 dilakukan melalui dua jalur yaitu *kbd data* (P1.4)

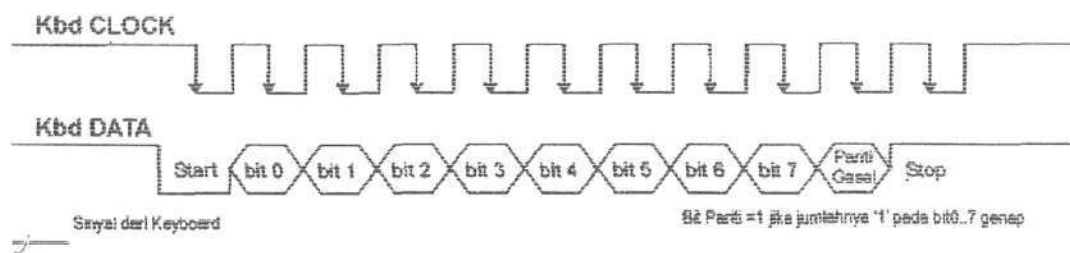
sebagai pengirim data dan *kbd clock* sebagai pengirim sinyal *clock* (int 0).

Rangkaian dari *keyboard* ini dapat dilihat pada gambar di bawah.



Gambar 3.2

Rangkaian Keyboard



Gambar 3.3¹

Kbd Data Dan Clock Dari Keyboard

¹ Philips Semiconductors Microcontroller Products, hal. 2

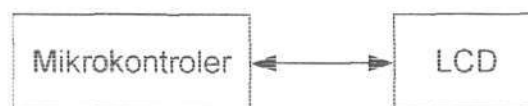
Dibawah ini adalah program untuk mengambil 1 byte (8 bit) data dari *keyboard*.

```

Ambil_Keyboard:
Clr  IE.0
Push 00H
CLR      A                ;Tunggu ada Interrupt
                        ;keyboard
JNB      INT0,*          ;Level high
Mov  R0,#08H
Ambil_Keyboard2:        ;Ambil data dari P1.4
                        ;setiap
JB      INT0,*          ;kali interrupt keyboard
                        ;transisi
MOV     C,P1.4          ;low
RR      A               ;Geser Acc A
MOV     A.7,C           ;Simpan data di bit 7AccA
JNB     INT0,*          ;Tunggu Int0 high
Djnz   R0,Ambil_Keyboard2 ;Lakukan 8x
Jb     INT0,*           ;Buang Stop Bit
Jnb    INT0,*           ;
Jb     INT0,*           ;
Jnb    INT0,*           ;
Pop    00H
Setb  IE.0
Ret

```

1.2 Hubungan Mikrokontroler Dengan LCD



Sebagai tampilan penulisan maupun pembacaan dari pesan, digunakan *display* LCD Hitachi HD44780 yang mempunyai fasilitas *backlight*. LCD Hitachi ini terdiri dari dua baris dengan 16 karakter tiap

baris. Untuk *backlight* dapat dinyalakan dengan memberikan sumber tegangan sebesar 4.5 volt. Akses LCD disini hanya membutuhkan sedikit *pin* dari mikrokontroler, yaitu dengan 4 bit *interface* dengan system *bus*.

Di bawah ini adalah *sub rutin* program dari LCD.

```

        .DATA
Alamat_LCD      Ds      1
RS      Bit      P2.1
        .CODE
        Org      *
;-----
;Geser Display ke kanan

GeserDisplay_Kanan:
        Mov      A,#1FH
        Acall     Kirim_Perintah
        Acall     Delay_LCD
        Ret

;-----
;Geser Display ke kiri

GeserDisplay_Kiri:
        Mov      A,#18H
        Acall     Kirim_Perintah
        Acall     Delay_LCD
        Ret

;-----
;Kembali ke posisi awal

PosisiAwal_LCD:
        Mov      A,#02H
        Acall     Kirim_Perintah
        Acall     Delay_LCD
        Ret

;-----
;Geser Cursor ke kiri

GeserCursor_Kiri:
        Mov      A,#10H

```

```

    Acall    Kirim_Perintah
    Acall    Delay_LCD
    Ret

;-----
;Geser Cursor ke kanan

GeserCursor_Kanan:
    Mov  A,#14H
    Acall    Kirim_Perintah
    Acall    Delay_LCD
    Ret

KirimPesan_LCD:
Send_Loop:
    Mov  A,#00H
    Movc A,@A+Dptr
    Cjne A,#0FH,Send
    Ret

Send:
    Acall    Kirim_Karakter
    Inc  Dptr
    Ajmp Send_Loop

Init_LCD:
    Setb RS

    Mov  A,#30H                ;Kirim 30H
    Acall    Kirim_Perintah    ;
    Acall    Delay_LCD

    Mov  A,#30H                ;Kirim 30H
    Acall    Kirim_Perintah    ;
    Acall    Delay_LCD

    Mov  A,#30H                ;Kirim 30H
    Acall    Kirim_Perintah    ;
    Acall    Delay_LCD

    Mov  A,#20H                ;Send Init
    Acall    Kirim_Perintah    ;
    Acall    Delay_LCD

    Mov  A,#2FH                ;8x5 2lines
    Acall    Kirim_Perintah    ;

```

```

    Acall      Delay_LCD

    Mov  A,#08H      ;
    Acall     Kirim_Perintah ;
    Acall     Delay_LCD

    Mov  A,#01H      ;
    Acall     Kirim_Perintah ;
    Acall     Delay_LCD

    Mov  A,#07H      ;
    Acall     Kirim_Perintah ;
    Acall     Delay_LCD ;

    Mov  A,#0EH      ;Display ON
    Acall     Kirim_Perintah ;
    Acall     Delay_LCD ;

    Mov  A,#06H      ;Mode Increment
                                ;Address

    Acall     Kirim_Perintah ;
    Acall     Delay_LCD ;
    Ret

Kirim_Perintah:
    Mov  P2,#00H
    Acall     Kirim_DataLCD
    Acall     Kirim_DataLCD
    Nop
    Ret

Kirim_DataLCD
    Swap A
    Mov  P0,A
    Nop
    Acall     PulseE_Clock
    ret

PulseE_Clock:
    Push A
    Mov  A,#0A0H
    Orl  A,P2
    Mov  P2,A
    Anl  A,#0FH
    Mov  P2,A
    Pop  A
    Ret

```

Kirim_Karakter:

```
Mov P2,#02H
Acall Kirim_DataLCD
Acall Kirim_DataLCD
Acall Delay_LCD
Ret
```

Baris2_LCD:

```
Mov A,#0C0H
Acall Kirim_Perintah
Ret
```

Baris1_LCD:

```
Mov A,#02H
Acall Kirim_Perintah
Acall Delay_LCD2
Ret
```

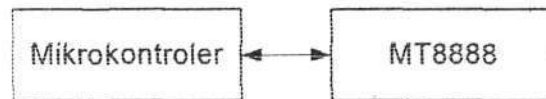
Delay_LCD:

```
Push B
Mov B,#06H
Delay_LCD_Loop:
Push B
Acall Delay_LCD2
Pop B
Djnz B,Delay_LCD_Loop
Pop B
Ret
```

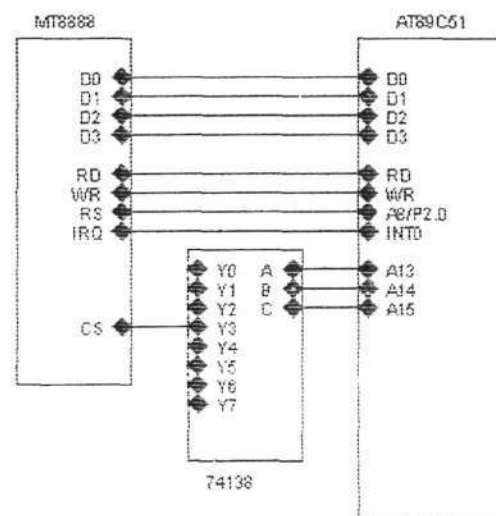
Delay_LCD2:

```
Mov B,#0FFH
Djnz B,*
Ret
```

1.3 Hubungan Mikrokontroler Dengan DTMF Decoder / Encoder



Pada rangkaian ini, CS dihubungkan ke sebuah decoder 74138 sehingga ke-aktifan MT8888 dapat diatur. MT8888 ataupun komponen-komponen lain yang berhubungan dengan *data bus*, RD maupun WR ditempatkan di alamat yang berbeda-beda. Pada saat AT89C51 mengakses komponen lain, maka MT8888 akan non aktif sehingga tidak bertabrakan dengan komponen lain tersebut dan demikian pula sebaliknya bila AT89C51 mengakses MT8888 maka komponen lain non aktif pula.



Gambar 3.4

AT89C51 Dan MT8888 Melalui Decoder 74138

Listing Program MT8888

```
MT8888 EQU 8000H
DTMF_Int bit INT1
```

```
Inisialisasi MT8888
```

```
Init_MT8888:
    INC DPH ;baca status register
    MOVX A,@DPTR ;
    CLR A ;kirim 0 ke kontrol
    ;register 2 kali
    MOVX @DPTR,A ;
    MOVX @DPTR,A ;
    MOV A,#008h ;kirim 8 ke kontrol
    ;register
    MOVX @DPTR,A ;
    CLR A ;kirim 0 ke kontrol
    ;register
    MOVX @DPTR,A ;
    MOVX A,@DPTR ;baca status register
    MOV A,#00Dh ;kirim 0d ke kontro
    ;register
    MOVX @DPTR,A ;
    MOV A,#000h ;kirim 0 ke kontrol
    ;pregister
    MOVX @DPTR,A ;
    Ret
```

```
Pengambilan nada DTMF
```

```
Ambil_DTMF:
    Mov DPTR,#MT8888 ;baca data register
    Movx A,@DPTR ;
    Inc DPH ;baca status register
    Push A ;
    Movx A,@DPTR ;
    Pop A ;
    Jnb DTMF_Int,* ;tunggu interrupt DTMF
    ;high
    Ret
```

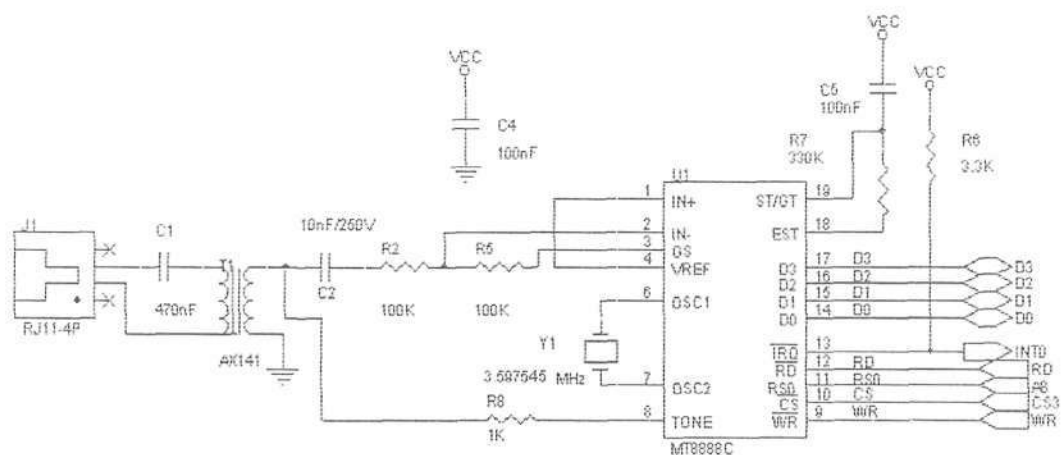
```
Pengiriman nada DTMF
```

```
Kirim_DTMF:
    Anl A,#0FH ;Bila terjadi angka 0
    ;maka
    Cjne A,#00H,BukanNol_DTMF ;ubah menjadi 0AH
    ;atau 10 desimal
    Mov A,#0AH ;
```

```

MOVX @DPTR, A      ; Kirim_DTMF
INC DPH             ;
MOVX A, @DPTR      ; Baca Status Register
RET

```



Gambar 3.5

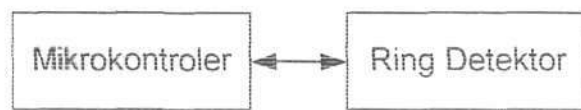
Rangkaian DTMF Transceiver

AT89C51 adalah mikrokontroler dengan *Timing Bus Intel* sehingga penggunaan MT8888 sebagai DTMF *Transceiver* lebih sesuai. Pada rangkaian diatas digunakan *transformer* AX141 yaitu sebuah transformer dengan perbandingan 1:1 dan impedansi 600 ohm yang berfungsi sebagai *matching impedance* antara jalur telpon dan *Interface DTMF Transceiver*.

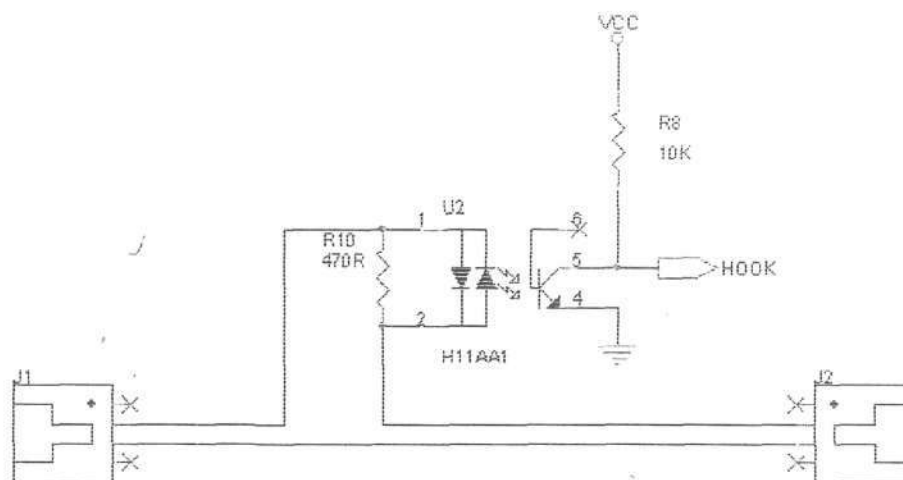
Kapasitor C1 berfungsi untuk melakukan *blocking* tegangan DC dari jalur telepon sehingga rangkaian ini tidak membebani jalur telepon. Sinyal-sinyal DTMF baik dari jalur telepon menuju ke IC MT8888

ataupun sebaliknya dari IC MT8888 ke jalur telepon yang merupakan sinyal AC saja yang dapat melalui kapasitor ini.

1.4 Hubungan Mikrokontroler Dengan Ring Detector



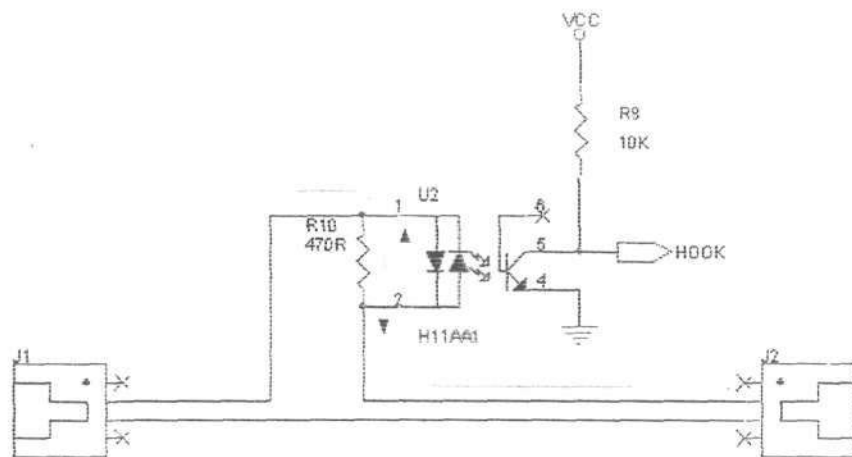
Dalam mendeteksi sinyal dering menggunakan *counter* dari MCS51 dengan memberi harga awal dari *counter* D8_H, *counter* ini akan bertambah setiap kali ada pulsa dari sinyal dering. Setelah *counter* penuh maka akan meng-*interrupt* sistem yang menunjukkan bahwa satu sinyal *ring* terdeteksi.



Gambar 3.6

Rangkaian Hook Dan Ring Detektor

Pada saat gagang pesawat telepon diangkat (*off hook*) maka impedansi pesawat telepon akan turun menjadi 600 hingga 1 K ohm sehingga arus akan mengalir dari jalur telkom menuju pesawat telepon melalui *opto coupler* H11AA1 atau sebaliknya dari pesawat telepon ke H11AA1. Hal ini tergantung dari posisi TIP dan RING yang terhubung dari jalur telkom.

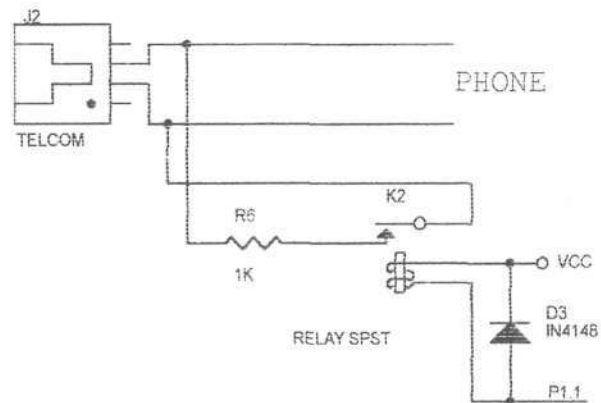


Gambar 3.7

Aliran arus pada Hook Detektor

Aliran arus ini akan menyebabkan *opto coupler* menjadi saturasi dan tegangan di titik *hook* akan turun menjadi 0 volt yang menandakan bahwa posisi gagang telepon pada kondisi *off hook*. Pada saat gagang telepon pada kondisi *on hook* maka impedansi pesawat telepon akan kembali tinggi sehingga arus tidak akan mengalir melalui

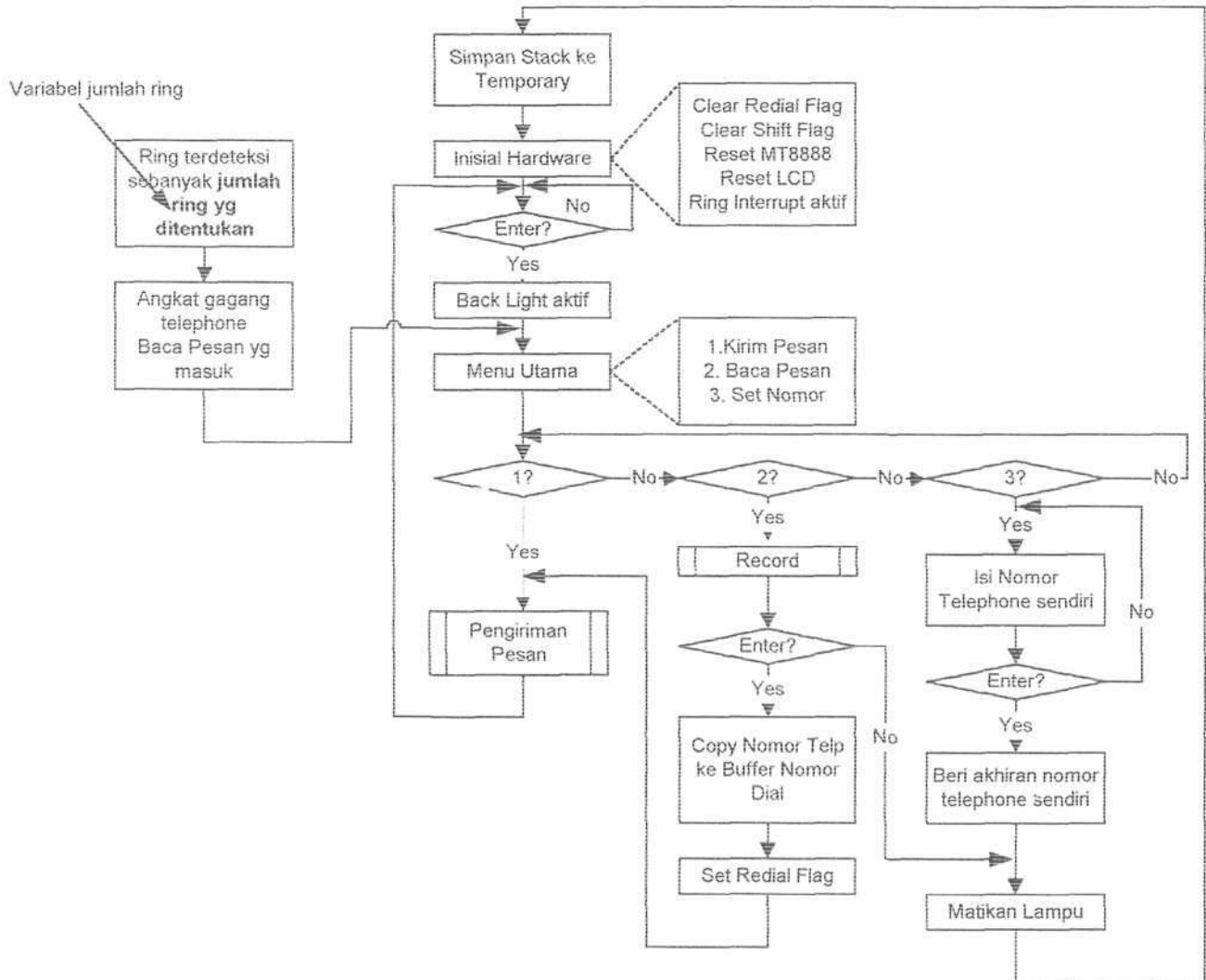
opto coupler. Hal ini menyebabkan *opto coupler* menjadi *cut off* dan tegangan pada titik *hook* berubah menjadi 5 volt.



Gambar 3.8

Rangkaian Simulasi Off Hook

2. FLOW-CHART



Gambar 3.9

Flow-chart Main Program

Seperti yang tampak pada *flowchart* di atas, maka pertama-tama sistem akan menyimpan nilai *stack* ke dalam *temporary*. Nilai ini akan dibutuhkan

pada saat setelah terjadi sinyal *ring* di mana sistem ter-*interrupt* dan program *counter* akan menuju ke bagian *Subroutine Ring Interrupt*.

Agar program dapat kembali hanya ke label pada *main* program maka nilai *stack* terlebih dahulu dikembalikan ke posisi asal, oleh karena itu nilai *stack* perlu disimpan pada saat program pertama kali berjalan.

Kemudian dilanjutkan dengan proses inisialisasi *hardware* di mana pada bagian ini, sistem melakukan *reset* pada LCD, MT8888, *redial flag* maupun *shift flag*.

Setelah itu sistem akan menyalakan *backlight* setelah tombol *enter* ditekan dan menu utama berupa:

1. Kirim Pesan
2. Baca Pesan
3. Set Nomor

Tampil dimana perintah Kirim Pesan adalah perintah untuk masuk ke bagian proses pengiriman pesan, perintah Baca Pesan adalah perintah untuk masuk ke bagian proses pembacaan pesan dan Perintah Set Nomor adalah perintah untuk mengisi nomor telepon sendiri yang akan dikirimkan setiap kali pengiriman pesan dilakukan.

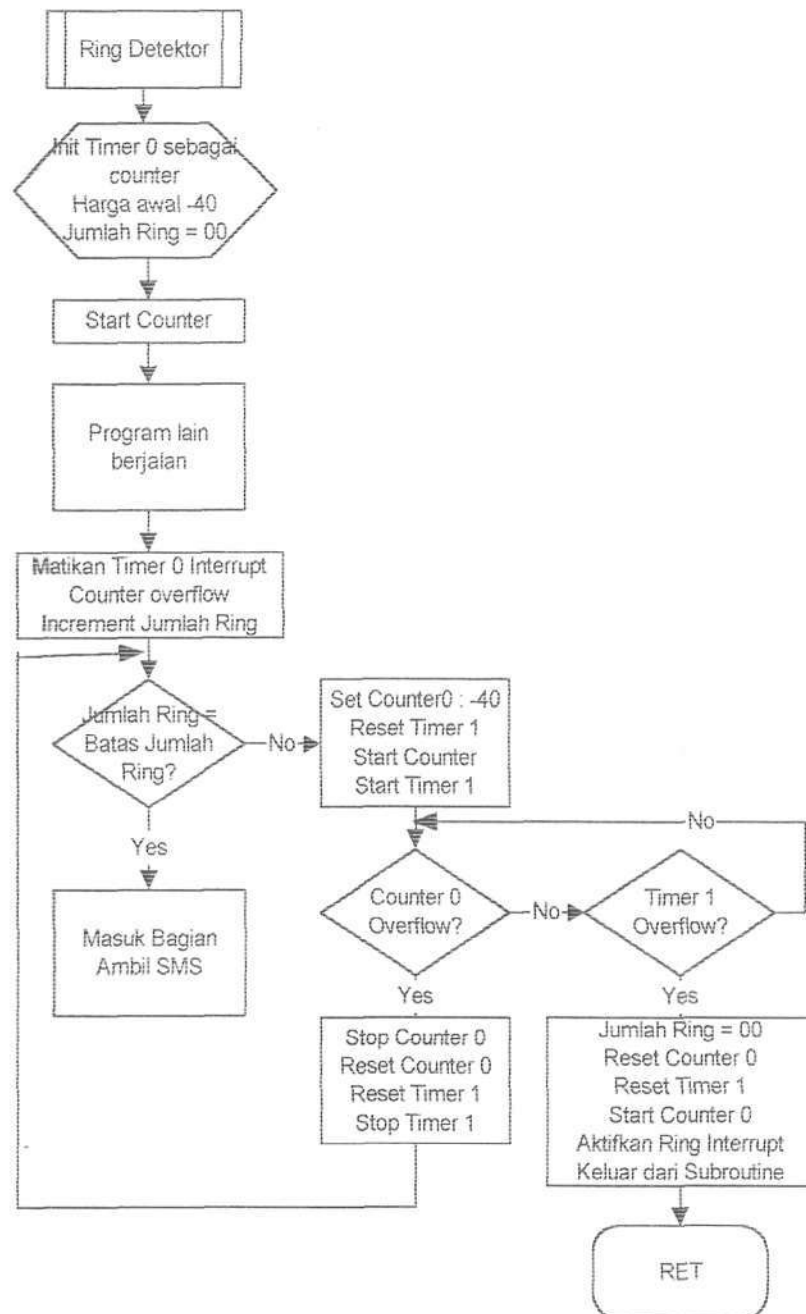
Pada proses pengiriman pesan maka program akan menuju ke menu pengiriman pesan dan setelah penulisan pesan selesai maka proses selanjutnya yaitu men-*dial* nomor tujuan dan setelah selesai proses pengiriman maka program akan kembali ke keadaan awal dengan mematikan LCD.

Pada proses pembacaan pesan maka program akan menuju ke menu pembacaan pesan dan menampilkan pesan yang tersimpan di dalam *record*. Setelah selesai membaca pesan terdapat dua pilihan yaitu membalas pesan atau tidak. Bila ingin membalas pesan maka program akan menuju ke menu pengiriman dan menjalankan proses seperti penekanan tombol satu. Tetapi bila pesan tidak dibalas maka program akan mematikan LCD dan kembali ke keadaan awal.

Pada proses *set* nomor telepon maka program akan menuju ke menu *set* nomor yaitu mengisi nomor telepon sendiri (nomor pengirim) setelah pengisian nomor selesai maka mikrokontroler akan memberikan tanda akhiran pada nomor dan selanjutnya mematikan LCD dan kembali ke keadaan awal.

Pada waktu penerimaan pesan maka sinyal *ring* akan meng-*interrupt* jalannya program dan sistem akan mendeteksi *ring* sebanyak jumlah yang ditentukan dan mengangkat gagang telepon dan menerima pesan yang masuk dan menampilkan tulisan ada pesan di LCD

Setelah menampilkan tulisan ada pesan di LCD dan terjadi penekanan tombol *enter* maka sistem terlebih dahulu mengembalikan isi *stack* ke posisi yang tercatat pada saat pertama kali program berjalan dan dilanjutkan dengan menampilkan menu utama di mana pengguna dapat memilih untuk membaca pesan atau mengirim pesan lain.

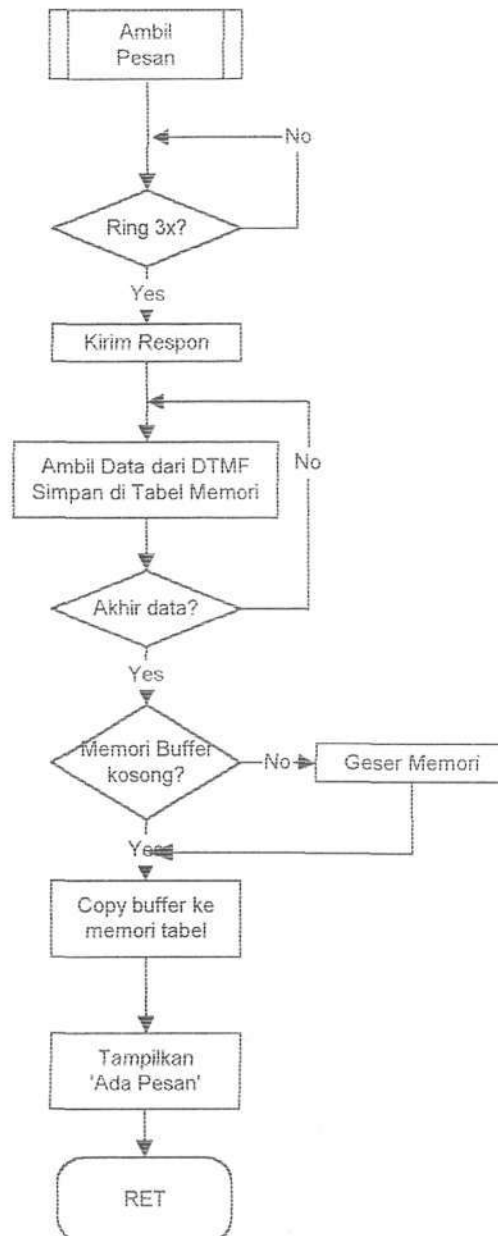


Gambar 3.10

Flow-chart Ring Detektor

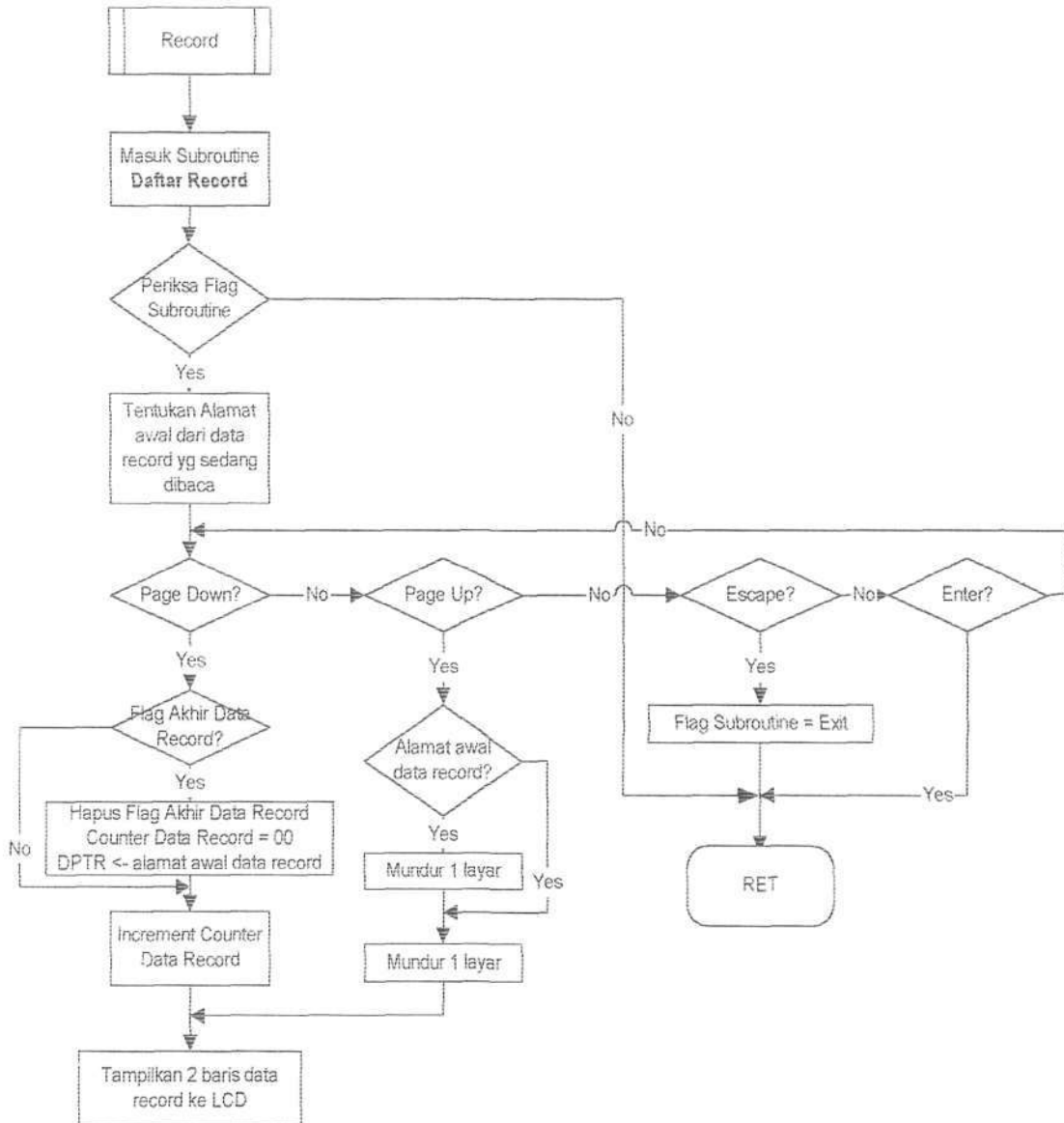


Gambar 3.11
Flow-chart Pengiriman Pesan



Gambar 3.12

Flow-chart Pengambilan Pesan



Gambar 3.13

Flow-chart Record