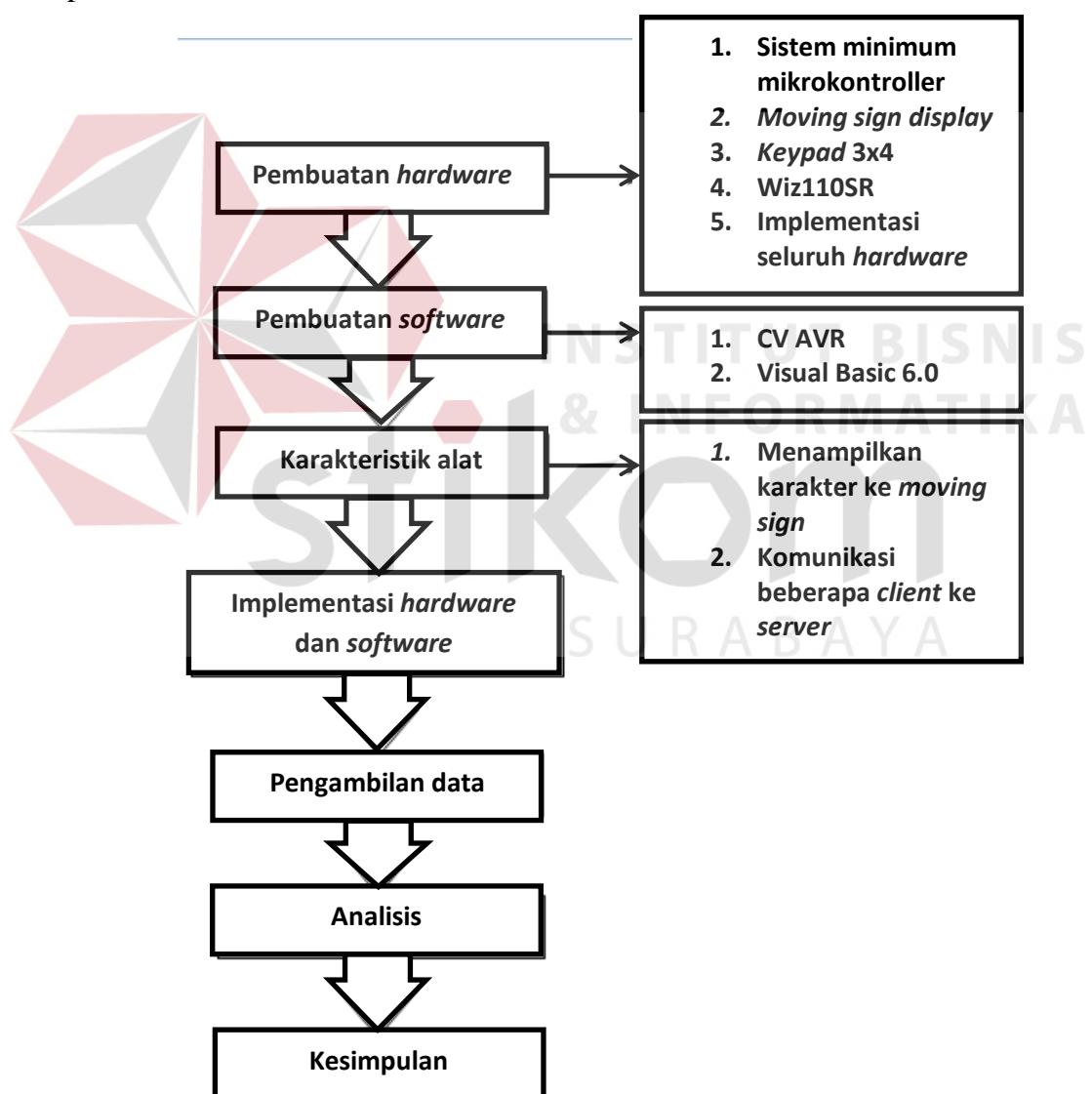


BAB III

METODE PENELITIAN

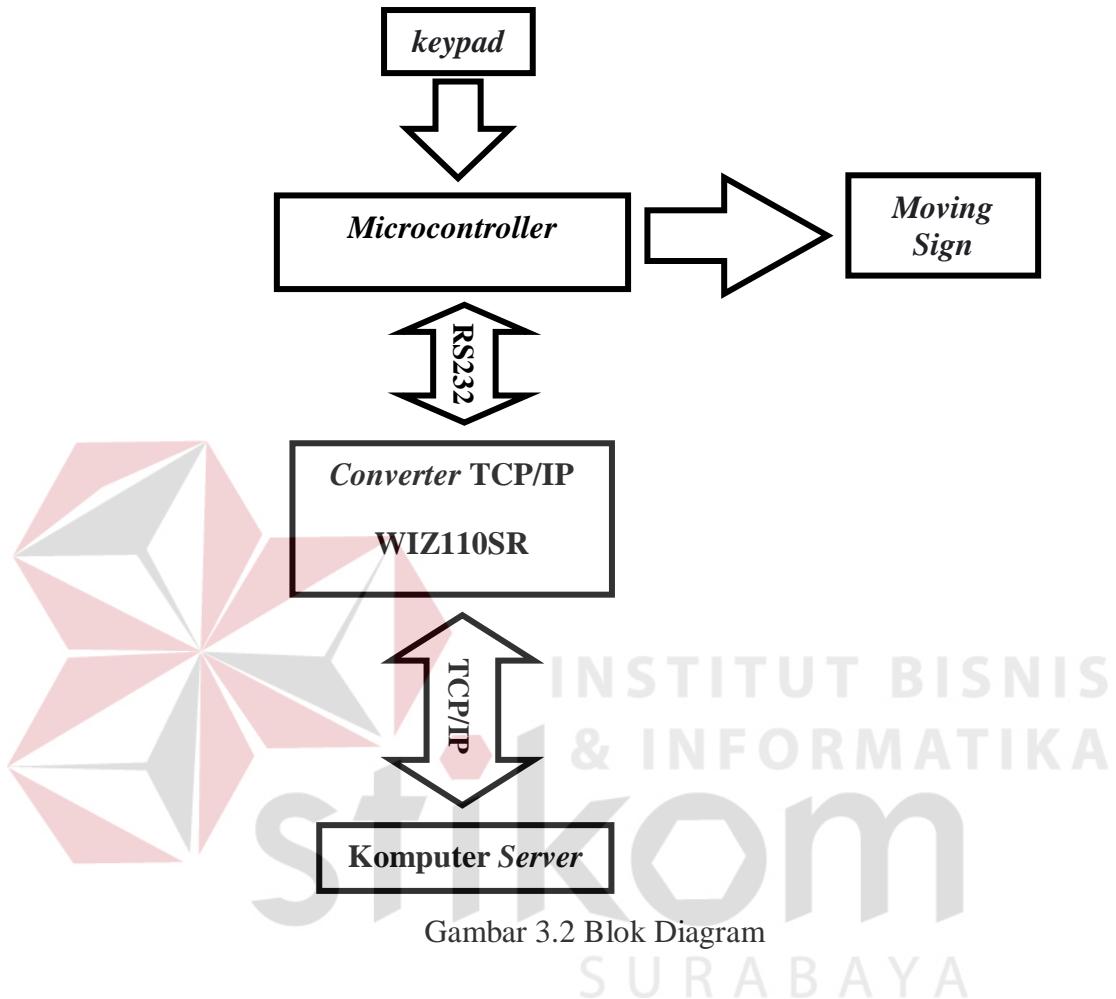
3.1 Metode Penelitian

Penelitian dilakukan berdasarkan tahapan yang telah disusun. Tahapan umum penelitian dibuat dalam sebuah skema kerja yang menggambarkan alur penggerjaan penelitian. Tahapan penelitian secara umum dijelaskan dalam skema pada Gambar 3.1.



Gambar 3.1 Skema umum penelitian

Pada penelitian terdapat tahapan akuisisi data dan analisis data yang dijelaskan oleh blok diagram pada Gambar 3.2.



Pada bagian akuisisi data, terdapat *keypad*, *moving sign display*, dan *microcontroller*. *Microcontroller* yang digunakan pada tugas akhir ini adalah ATmega16 yang mempunyai fungsi sebagai pengolah data yang diperoleh dari inputan *keypad* yang kemudian akan ditampilkan ke *moving sign*. Selanjutnya, data-data yang diterima *microcontroller* dikirim ke komputer *server* melalui konverter TCP/IP. Konverter TCP/IP yang digunakan adalah WIZ110SR. Proses ini dilakukan setiap 5 detik sekali agar tidak ada menu yang belum terdaftar.

Bagian analisis data merupakan bagian untuk mengelola data yang sudah ada untuk mendapatkan kesimpulan. Dengan tujuan untuk mempermudah proses analisis data, dibuatlah aplikasi analisis data dengan Visual Basic 6.0. Aplikasi ini bertujuan untuk membaca semua data yang tersimpan pada PC *server*. Semua data yang berjumlah ratusan tersebut akan diseleksi sesuai dengan jenisnya, yaitu data makanan, data minuman dan data harga. Pada tahap terakhir, dibuatlah *display* yang memudahkan koki melihat tampilan *server*.

3.2. Perancangan *Hardware*

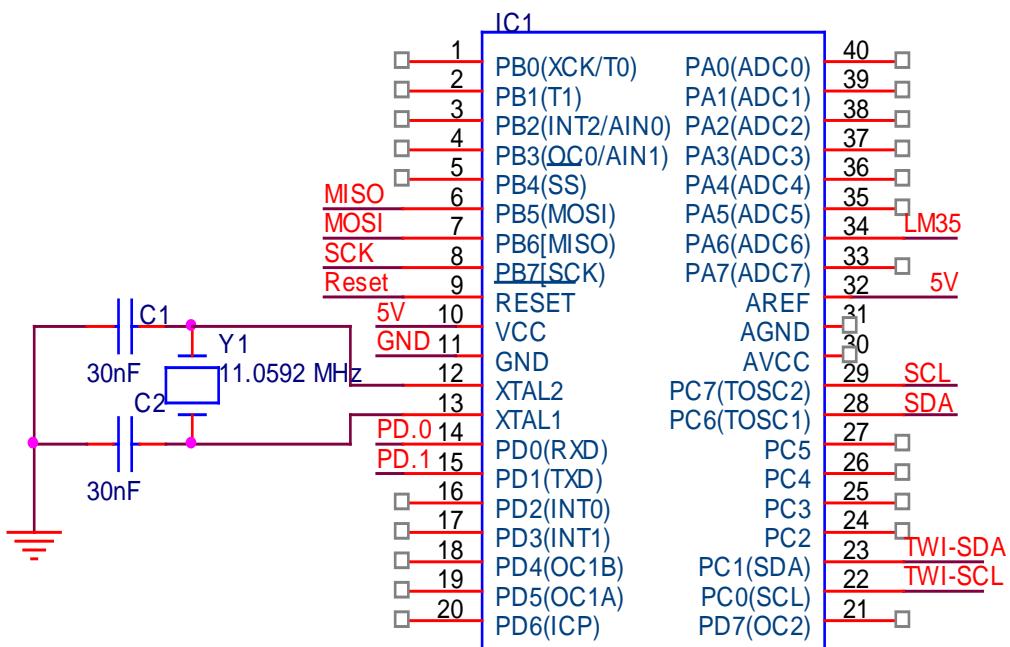
Hardware merupakan perangkat keras dalam sistem komputer yang secara fisik terlihat dan dapat disentuh. Perangkat keras yang dibutuhkan guna mendukung kelancaran program antara lain terdiri dari *keypad* 3x4, *moving sign display*, modul WIZ110SR serta penggunaan rangkaian sistem minimum.

3.2.1. Rangkaian *Microcontroller*

Pada proyek tugas akhir ini dibuat piranti pengendali menggunakan *microcontroller* keluaran AVR, yaitu ATMega16. Pengendalian program pada *microcontroller* ini diperlukan rangkaian sistem minimum. Rangkaian sistem minimum adalah rangkaian elektronika yang terdiri dari komponen-komponen dasar yang dibutuhkan oleh suatu *microcontroller* dapat berfungsi dengan baik.

Rangkaian sistem minimum *microcontroller*

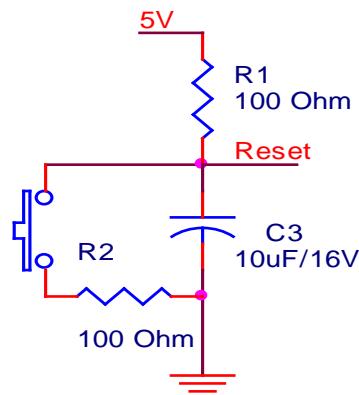
Rangkaian sistem minimum terbagi menjadi 3 rangkaian utama yaitu rangkaian IC, rangkaian *reset* dan rangkaian *crystal*. Rangkaian IC sistem minimum dapat lihat pada Gambar 3.3.



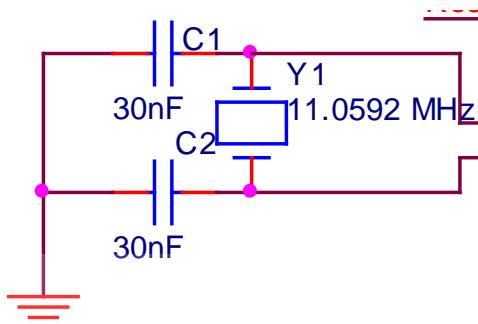
Gambar 3.3. Rangkaian Sistem Minimum ATMega 16

Pada Gambar 3.3 menjelaskan bahwa rangkaian pada *microcontroller* tipe ATMega 16 terdapat 40 pin. (Atmel, 2009). IC ini memiliki 4 *Port*. *Port A* digunakan untuk *IC 74HCT573* dan kolom untuk *keypad*. *Port B* digunakan untuk *output* data yang telah dikonfersikan. *Port C* digunakan untuk *IC 74HCT595* dan baris untuk *keypad*. *Port D* digunakan untuk pengiriman secara serial. Tegangan masukan DC 5 Volt diparalel dengan kapasitor 100 uF sebagai filter supaya tidak ada kekacauan data. Sedangkan rangkaian *reset* dapat dilihat pada

Gambar 3.4.

Gambar 3.4. Rangkaian *Reset*

Pada Gambar 3.4 menjelaskan bahwa rangkaian *reset* pada tombol *reset* (SW1) digunakan untuk mengurangi *noise* serta memiliki fungsi terpenting yaitu untuk melakukan *reset* saat pertama kali catu daya dinyalakan. *Reset* untuk pertama kali merupakan hal yang terpenting sehingga dapat memastikan bahwa program telah berada pada posisi awal. (Wahyuni, 2013)



Gambar 3.5. Rangkaian *Crystal*

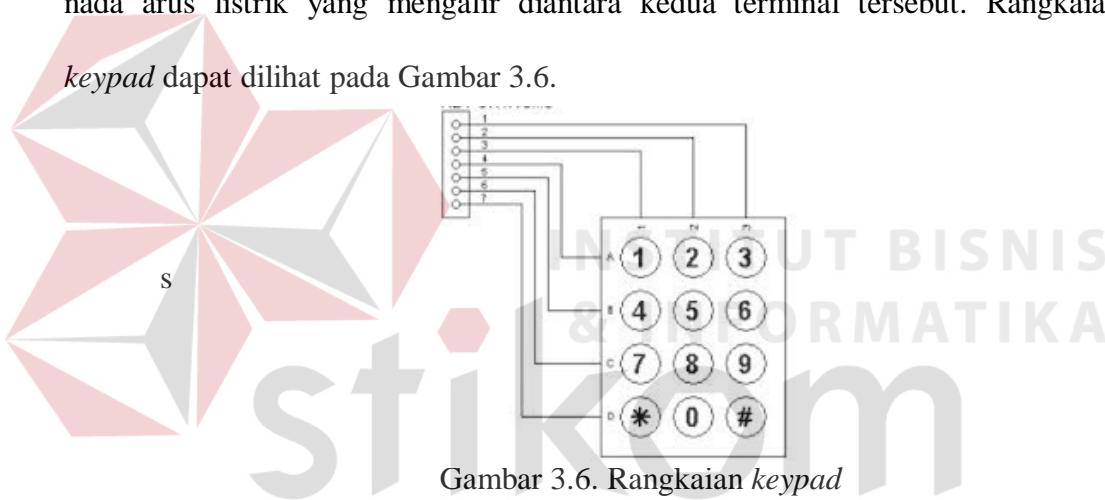
Pada Gambar 3.5 menjelaskan bahwa rangkaian *crystal* yang digunakan sebesar 11,0592 MHz karena sistem minimum dapat berjalan normal diatas *crystal* 14 MHz. Penggunaan *crystal* 11,0592 MHz dapat diperoleh dengan perhitungan UBRR karena mempunyai nilai *error* komunikasi serial 0 persen.

3.2.2. Rangkaian *Keypad 3x4*

Pada umumnya *keypad* ini digolongkan berdasarkan jumlah tombol. Dan disebut sebagai *keypad* matrix mxn. *Keypad* matriks 3 x 3, berarti bahwa *keypad* tersebut memiliki 12 tombol. Jenis *keypad* yang umum dijual di pasaran antara lain *keypad* matriks 3x3, 3x4, dan 4x4. Pada proyek tugas akhir ini *keypad* yang digunakan adalah *keypad* 3x4.

Pada dasarnya *keypad* berisi saklar-saklar biasa, semisal saklar push button. Dimana cara kerjanya hanyalah berdasarkan proses konduktansi pada saat tombol

saklar ditekan. Tombol yang tertekan oleh jari, akan mendorong lempeng konduktor yang melekat pada lapisan pegas menyentuh lapisan konduktor yang terdapat di bagian dasar (lapisan atas dari PCB). Selanjutnya hal ini akan mengakibatkan terminal-terminal konduktor yang ada di dasar saling berhubungan. Jika saja pada kedua terminal tersebut diberi beda potensial maka pada saat tombol dilepas, tidak ada arus listrik yang mengalir diantara kedua terminal. Sebaliknya, saat tombol ditekan, maka konduktor yang terdapat pada bagian dasar pegas akan menyentuh kedua terminal tersebut. Dengan kata lain ada arus listrik yang mengalir diantara kedua terminal tersebut. Rangkaian *keypad* dapat dilihat pada Gambar 3.6.



Gambar 3.6. Rangkaian *keypad*

Konfigurasi *keypad* pada CV AVR dapat dilihat pada Gambar 3.7 berikut ini.

Port A	Port B	Port C	Port D	Port A	Port B	Port C	Port D
Data Direction		Pullup/Output Value		Data Direction		Pullup/Output Value	
Bit 0	Out	0	Bit 0	Bit 0	Out	0	Bit 0
Bit 1	Out	0	Bit 1	Bit 1	Out	0	Bit 1
Bit 2	Out	0	Bit 2	Bit 2	Out	0	Bit 2
Bit 3	Out	0	Bit 3	Bit 3	Out	0	Bit 3
Bit 4	Out	0	Bit 4	Bit 4	In	P	Bit 4
Bit 5	Out	0	Bit 5	Bit 5	In	P	Bit 5
Bit 6	Out	0	Bit 6	Bit 6	In	P	Bit 6
Bit 7	Out	0	Bit 7	Bit 7	In	P	Bit 7

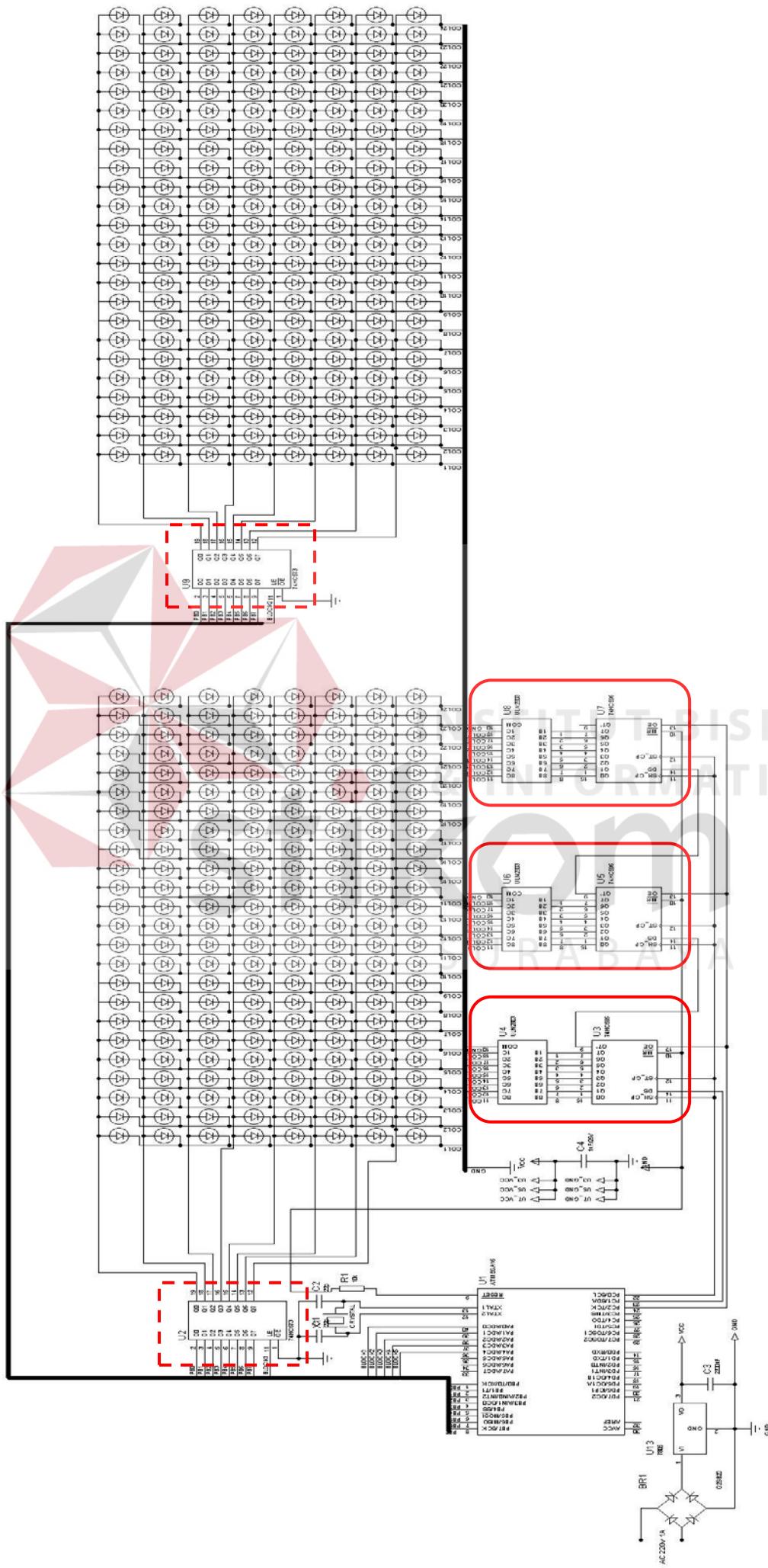
Gambar 3.7 Konfigurasi *keypad* pada CV AVR

Program untuk *keypad 3x4* adalah sebagai berikut :

```
void Scan_KeyPad(void)
{
    unsigned char pad;
    PORTA=0xFF;
    PORTA.7=0; // kolom 0
    pad=PIND & 0xF0;
    if(pad==0x70)key=3;else
    if(pad==0xB0)key=6;else
    if(pad==0xD0)key=9;else
    if(pad==0xE0)key=11;else
    {
        PORTA=0xFF;
        PORTA.6=0; // kolom 1
        pad=PIND & 0xF0;
        if(pad==0x70)key=2;else
        if(pad==0xB0)key=5;else
        if(pad==0xD0)key=8;else
        if(pad==0xE0)key=0;else
        {
            PORTA=0xFF;
            PORTA.5=0; // kolom 2
            pad=PIND & 0xF0;
            if(pad==0x70)key=1;else
            if(pad==0xB0)key=4;else
            if(pad==0xD0)key=7;else
            if(pad==0xE0)key=10;else
            {
                key=12;
            }
        }
    }
}
```

3.2.3. *Moving Sign Display*

Moving Sign Display atau yang sering disebut dengan *Running Text* dapat diprogram melalui beberapa cara. Pada tugas akhir ini moving sign di program dengan metode scanning kolom. Pembuatan *Moving Sign Display* ini membutuhkan beberapa komponen, yaitu *IC 74HCT573*, *IC 74HCT595*, *ULN 2803* dan *384 LED*. *Moving Sign Display* yang digunakan pada tugas akhir ini berukuran 8 baris x 48 kolom. Gambar rangkaian *Moving Sign Display* dapat dilihat pada Gambar 3.8.



Gambar 3.8. Rangkaian Moving Sign Display

Program untuk *scanning moving sign display* adalah sebagai berikut :

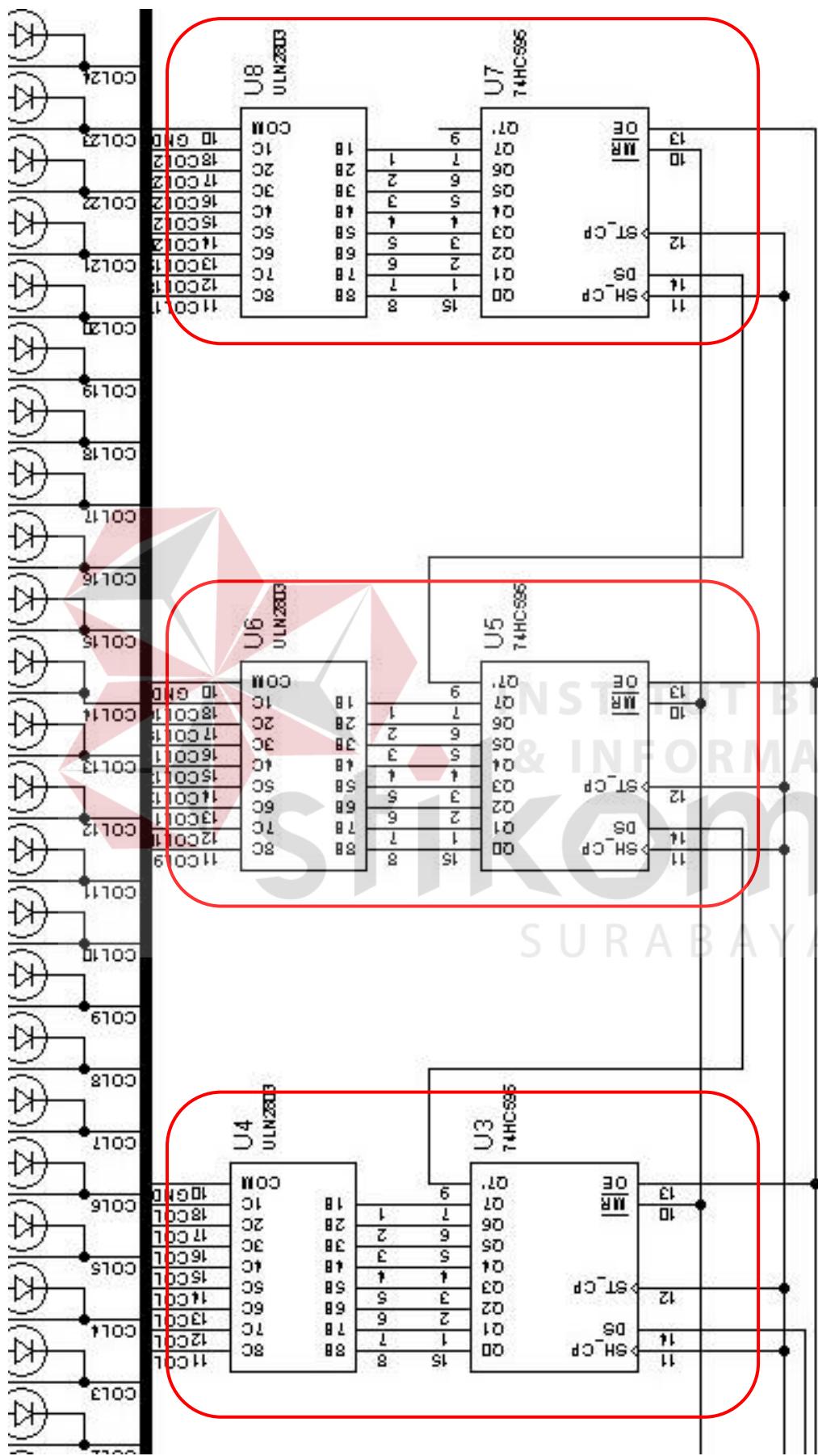
```
void scancodes(char n)
{
    if (n=='A') { kode[x]=0xFE;x++; kode[x]=0x11;x++;
                  kode[x]=0x11;x++; kode[x]=0xFE;x++;
                  kode[x]=0x00;x++; }
    else if (n=='B') { kode[x]=0xFF;x++; kode[x]=0x89;x++;
                        kode[x]=0x89;x++; kode[x]=0x76;x++;
                        kode[x]=0x00;x++; }
    else if (n=='C') { kode[x]=0x7E;x++; kode[x]=0x81;x++;
                        kode[x]=0x81;x++; kode[x]=0x42;x++;
                        kode[x]=0x00;x++; }
}
```

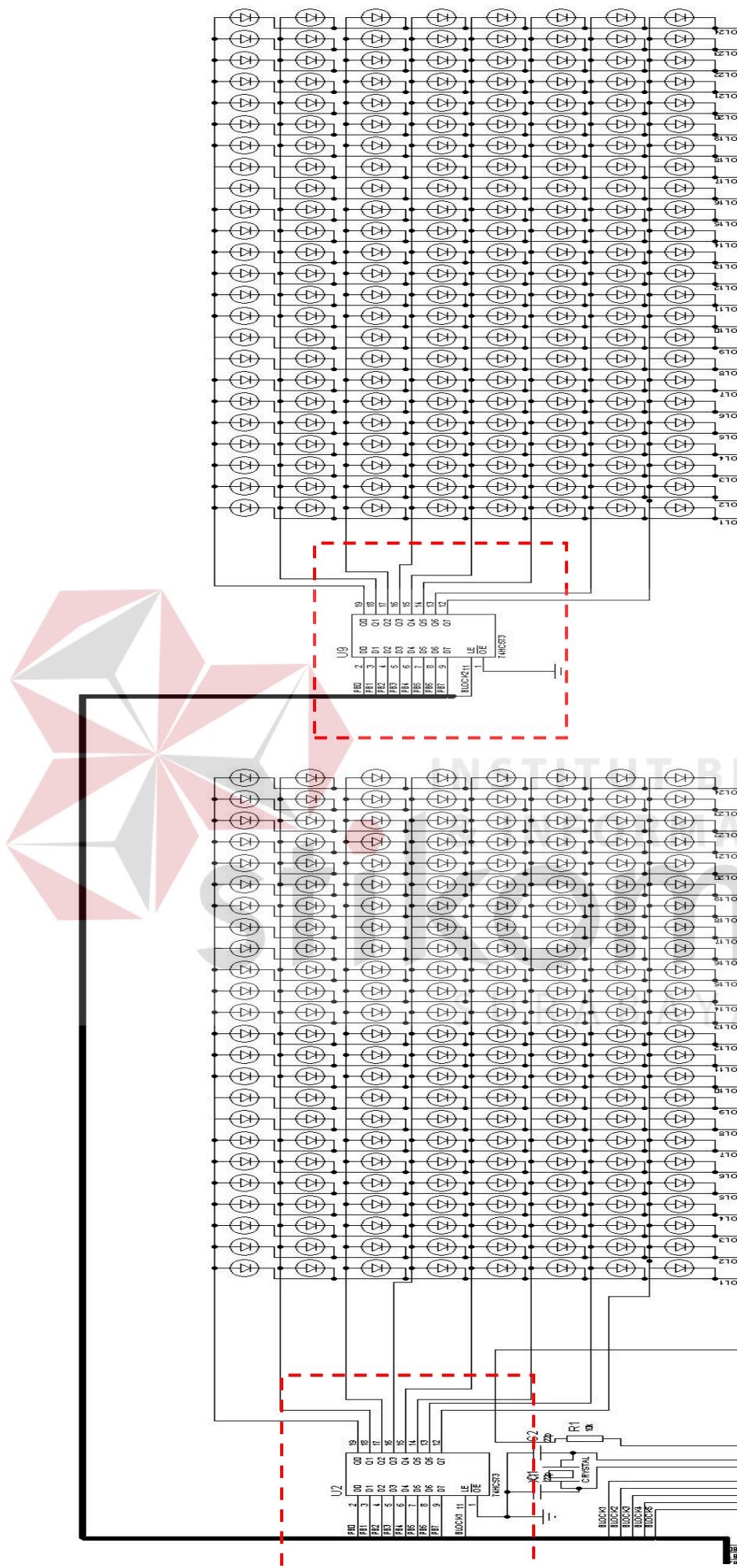
Pada Gambar 3.8 menjelaskan bahwa rangkaian pada *moving sign display* terdiri dari 2 blok, masing-masing blok terdiri dari 8 baris dan 24 kolom LED. Ada 2 buah *IC 74HCT573* dan 3 buah IC *IC 74HCT595* dan *ULN2803*. Gambar rangkaian *IC 74HCT595* dapat dilihat pada Gambar 3.9.

Pada Gambar 3.9 menjelaskan bahwa ada 3 buah *IC 74HCT595* dan 3 buah *ULN2803* yang saling terhubung. *ULN2803* merupakan transistor NPN, pada tugas akhir ini digunakan sebagai *driver* dari LED yang kemudian diteruskan pada *IC 74HCT595* dan *output* di masukkan pada minimum sistem.

Gambar Rangkaian *74HCT735* dapat dilihat pada Gambar 3.10, yang menjelaskan bahwa ada 2 buah *IC 74HCT573*. *IC 74HCT573* berfungsi sebagai perpindahan setiap blok *moving sign display* dan meneruskan *output* dari PORT.B dari minimum sistem saat LE (*Latch Enable*) bernilai 1.

Gambar 3.9. Rangkai IC 74HC595

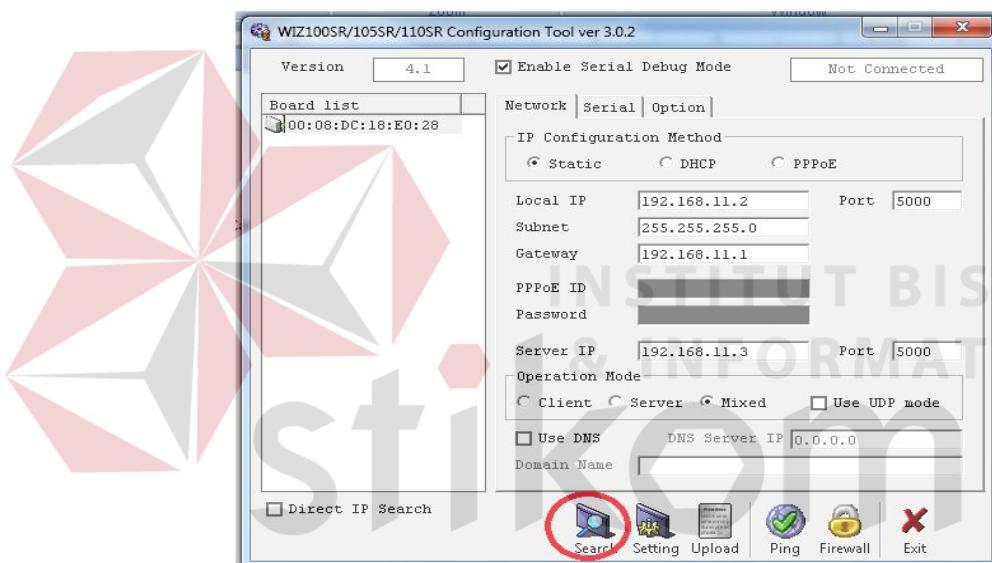




Gambar 3.9. Rangkaia IC 74HCT573

3.2.4. Konfigurasi WIZ110SR

Microcontroller dapat berkomunikasi melalui jaringan berbasis *internet protocol* menggunakan modul WIZ110SR, untuk itu diperlukan beberapa pengaturan pada modul WIZ110SR. Pengaturan tersebut dapat dilakukan melalui WIZ110SR *Configuration Tool*. Konfigurasi ini digunakan untuk merubah komunikasi serial ke komunikasi TCP/IP supaya *microcontroller* dapat berkomunikasi dengan PC *server*. Tampilan jendela pengaturan modul WIZ110SR dapat dilihat pada Gambar 3.11.



Gambar 3.11. Konfigurasi WIZ110SR

Pada Gambar 3.11 dijelaskan bahwa pertama kali pada konfigurasi WIZ110SR yaitu kita pilih tombol *search* supaya kotak *box* pada konfigurasi keluar IP *default* dari modul beserta informasi lainnya seperti versi *firmware* dari modul dan *mac address*.

Langkah-langkah keseluruhan dari pengaturan modul *ethernet* WIZ110SR adalah sebagai berikut :

1. Modul WIZ110SR dikoneksikan dengan komputer yang akan digunakan untuk proses konfigurasi melalui *network switch*.
2. Konfigurasi modul dilakukan dengan menggunakan WIZ110SR *configuration tool* seperti pada Gambar 3.11.
3. Memulai proses konfigurasi tekan tombol *search* pada *tool* untuk menampilkan daftar modul yang terkoneksi ke jaringan. Daftar modul akan tampil di sebelah kiri (*Board List*) pada Gambar 3.11.
4. Pilih salah satu *board* yang akan dikonfigurasi. Ketika dipilih, pada bagian kanan akan muncul konfigurasi yang telah disimpan ke dalam modul sebelumnya.
5. Pada *tool* ini terdapat 2 tab yang wajib dikonfigurasi. Masing-masing tab tersebut memiliki fungsi sebagai berikut :

a. **Network**

Mengkonfigurasi modul WIZ110SR terkait dengan bagaimana modul tersebut dapat berkomunikasi melalui jaringan, seperti *IP Address*, *Subnet Mask*, *Gateway*, dan *Port*. Pada tab ini, beberapa hal yang dapat dikonfigurasi adalah sebagai berikut:

1. *IP Configuration Method*, digunakan untuk menentukan pengaturan alamat IP. Pengaturan alamat IP yang digunakan yaitu menggunakan *static IP*.
2. *Operation Mode*, digunakan untuk menentukan *mode* operasi dari modul WIZ110SR. *Mode* yang digunakan adalah *mode client*.

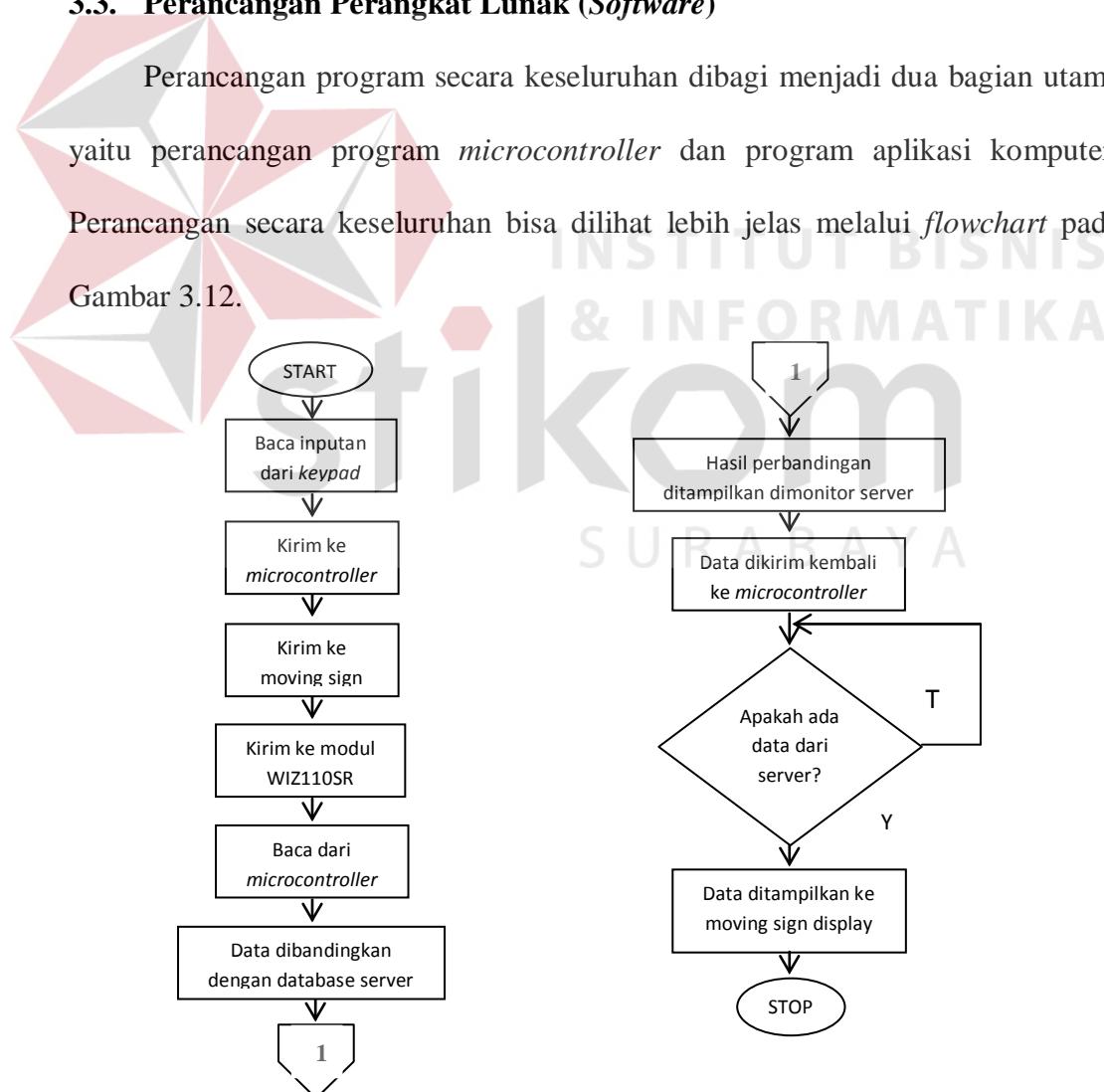
b. Serial

Mengkonfigurasi modul terkait dengan bagaimana modul dapat berkomunikasi dengan *microcontroller* melalui *Universal Asynchronous Receiver Transmitter* (UART) seperti *Baud Rate (Speed)*, Jumlah bit data setiap paket (*DataBit*), *Parity*, *Stop Bit*, dan *Flow Control*. Setelah semua terkonfigurasi sesuai (*Network & Serial*) tekan tombol pengaturan untuk mengirimkan konfigurasi ke modul WIZ110SR.

3.3. Perancangan Perangkat Lunak (*Software*)

Perancangan program secara keseluruhan dibagi menjadi dua bagian utama yaitu perancangan program *microcontroller* dan program aplikasi komputer.

Perancangan secara keseluruhan bisa dilihat lebih jelas melalui *flowchart* pada Gambar 3.12.

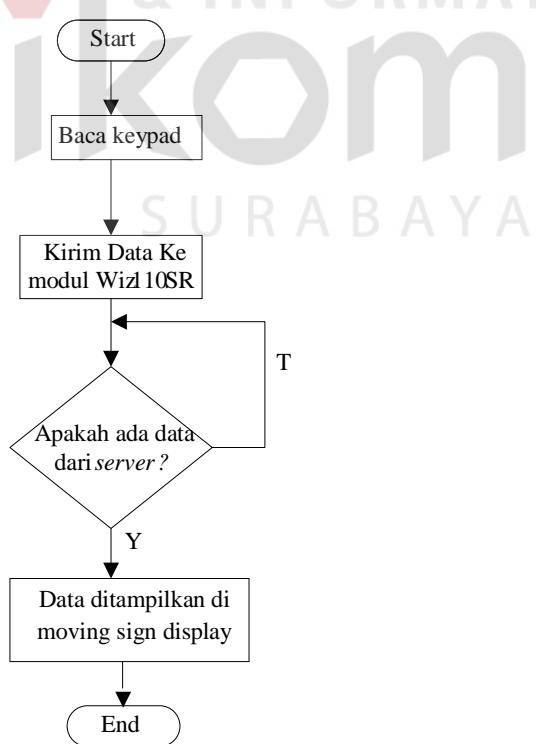


Gambar 3.12. *Flowchart* Keseluruhan Sistem

Pada Gambar 3.12 dijelaskan bahwa pada awalnya *microcontroller* menerima *input* dari *keypad* yang kemudian ditampilkan pada *moving sign display*. Setelah ditampilkan maka *microcontroller* mengirimkan data ke PC *server* melalui modul WIZ110SR. Setelah data diterima oleh *server* maka data pesanan menu makanan ditampilkan pada monitor PC *server*. Data yang telah diterima oleh *server* dikirim kembali ke *microcontroller* dan ditampilkan kembali ke *moving sign display*.

3.3.1. Perancangan Program *Microcontroller*

Microcontroller digunakan untuk mengolah data dari *keypad* lalu dikirimkan ke modul WIZ110SR dan ditampilkan ke *moving sign display* melalui komunikasi serial. Perancangan program *microcontroller* secara keseluruhan bisa dilihat lebih jelas melalui *flowchart* pada Gambar 3.13.



Gambar 3.13. *Flowchart* Program *Microcontroller*

Dari Gambar 3.12 dijelaskan bahwa keypad mengirimkan *inputan* ke *microcontroller*. *Inputan* tersebut dikirimkan ke *microcontroller* kemudian dilanjutkan ke PC melalui WIZ110SR. Data diolah di PC *server* untuk dibandingkan sesuai dengan database. Jika data tersebut belum diterima maka *microcontroller* menunggu. Jika data tersebut sudah diterima sesuai dengan pesanan maka data tersebut dikirimkan ke *microcontroller* dan data ditampilkan di *moving sign display*.

3.4. Perancangan Aplikasi Pemesanan Menu Makanan

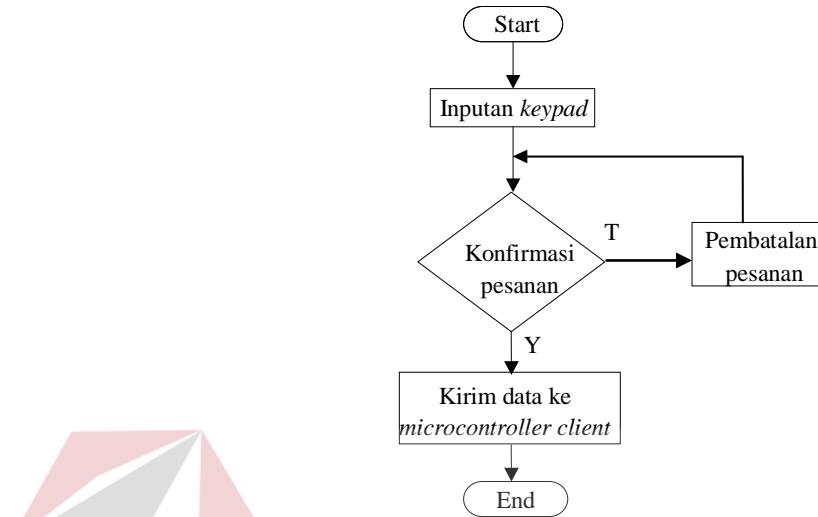
Aplikasi data pemesanan menu makanan dibuat menggunakan Visual Basic 6.0 sebagai antarmuka *software* dengan alat. Aturan penulisan program Visual Basic 6.0 didasari dengan aturan penulisan bahasa basic. Aplikasi komputer ini digunakan untuk mengolah semua data yang dikirim dari *microcontroller*. Data dikirim secara serial ke komputer. Aplikasi ini terdiri dari 2 bagian yaitu Aplikasi *server* dan aplikasi simulasi. Aplikasi *server* ini digunakan untuk menerima dan menampilkan data dari *microcontroller*.

3.4.1. Aplikasi Simulasi

Aplikasi simulasi pemesanan menu makanan dibuat menggunakan Visual Basic 6.0 sebagai antarmuka *software* dengan alat. Aturan penulisan program Visual Basic 6.0 didasari dengan aturan penulisan bahasa basic. Aplikasi simulasi digunakan untuk mensimulasikan alat pemesanan menu makanan dan *software* simulasi pada komputer, hal ini dikarenakan keterbatasan *hardware* yang dibuat hanya satu buah.

3.4.1.1 Simulasi Keypad

Simulasi *keypad* dipermudah dengan penjelasan *flowchart* pada Gambar 3.14.



Gambar 3.14. Flowchart Simulasi *keypad*

Penjelasan *flowchart* pada Gambar 3.14 bahwa program simulasi *keypad* ini berawal dari inputan *keypad* yang disimulasikan pada visual basic 6.0. Setelah pengunjung memilih menu makanan yg akan ditampilkan, maka pengunjung melakukan konfirmasi pemesanan. Jika pesanan telah dikonfirmasi maka data dikirim ke *microcontroller client*, dan apabila pesenan dibatalkan maka pengunjung bisa memulai memesan menu dari awal.

Program untuk simulasi inputan *keypad* adalah :

```

makanan = Text1.Text & Text5.Text
minuman = Text2.Text & Text6.Text
snack   = Text3.Text & Text7.Text
  
```

Program untuk konfirmasi pesanan adalah :

```

Private Sub Command5_Click()
    x = Int(10000)
    y = Int(8000)
    z = Int(6000)
    jumlah = (Text5.Text * x) + (Text6.Text * y) +
    (Text7.Text * z)
  
```

```

With List1
    .AddItem "nomer meja" & Text4.Text
    .AddItem "Pesanan Anda adalah :"
    .AddItem "makanan : " & Text1.Text & " " & Text5.Text
    & " porsi"
    .AddItem "minuman : " & Text2.Text & " " & Text6.Text
    & " gelas"
    .AddItem "snack : " & Text3.Text & " " & Text7.Text
    & " porsi"
    .AddItem "Total yang harus dibayar sebesar : " & "Rp
" & jumlah
End With
End Sub

```

Program untuk pembatalan pesanan adalah :

```

Private Sub Command2_Click() ' cancel pesanan
Timer1.Enabled = False
'MSComm1.PortOpen = False

Text1.Text = "silahkan isi kembali"
Text2.Text = "silahkan isi kembali"
Text3.Text = "silahkan isi kembali"
Text5.Text = "0"
Text6.Text = "0"
Text7.Text = "0"
List1.Clear
End Sub

```

Program untuk mengirim data ke *microcontroller client* adalah :

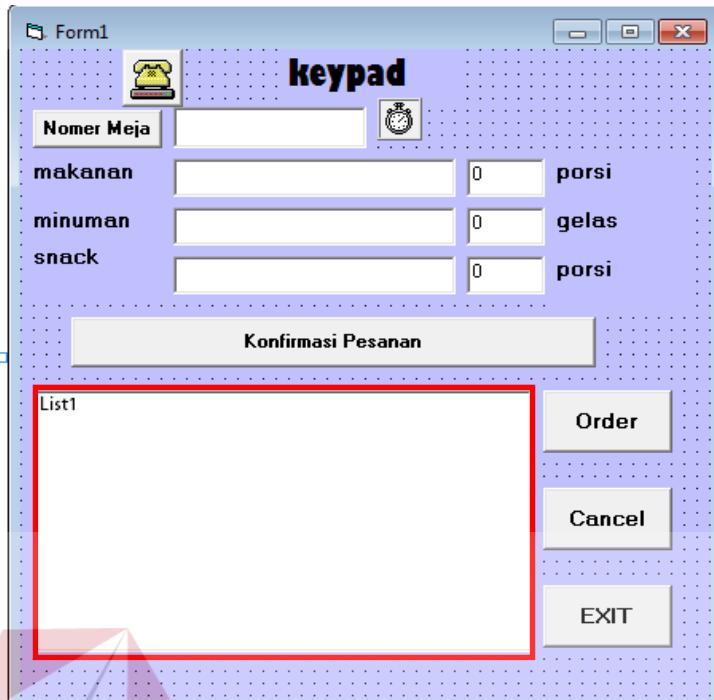
```

Private Sub Timer1_Timer()
makanan = Text1.Text & Text5.Text
minuman = Text2.Text & Text6.Text
snack = Text3.Text & Text7.Text

MSComm1.Output = "P" + makanan + " " + "porsi" + "M" +
minuman + " " + "gelas" + "I" + snack + " " + "porsi" +
"R"
Timer1.Enabled = False
End Sub

```

Desain dan kegunaan *form keypad* dapat dilihat pada Gambar 3.15.



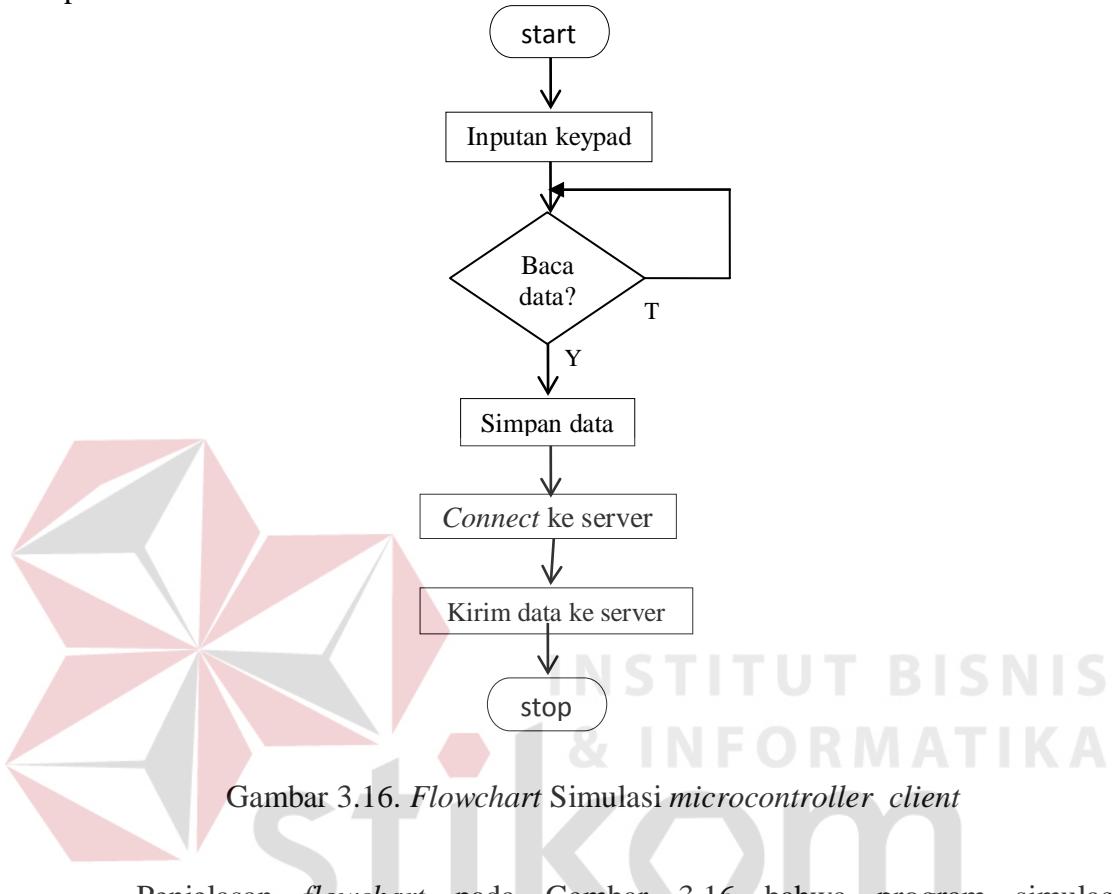
Gambar 3.15. *Form keypad*

Keterangan Gambar 3.15 :

- Tombol Nomer Meja: Mengkonfirmasi Nomer Meja yang telah di tempati.
- Text Nomer Meja: Nomer meja yang di tempati.
- Text makanan, minuman, snack: Menu makanan, minuman, dan snack yang telah dipilih/ dipesan.
- Text porsi dan gelas: Jumlah menu makanan, minuman, dan snack yang telah dipilih/ dipesan.
- Tombol konfirmasi pesanan: Menampilkan semua data yang sama dengan data *server*.
- Tombol Order: Mengirim pesanan pada *microcontroller client*.
- Tombol Cancel: Pembatalan pemesanan.
- Kotak Merah: Hasil menu yang telah dipesan.

3.4.1.2 Simulasi *Microcontroller Client*

Simulasi *microcontroller client* dipermudah dengan penjelasan *flowchart* pada Gambar 3.16 .



Penjelasan *flowchart* pada Gambar 3.16 bahwa program simulasi *microcontroller client* ini berawal dari inputan *keypad* yang kemudian dilakukan pembacaan oleh *client*. Setelah client berhasil membaca inputan maka client menampilkan data menu makanan yg dipesan. Data yang telah ditampilkan kemudian disimpan pada *microcontroller client*. Kemudian *client* connect ke *server*, setelah berhasil *connect* ke *server* data dikirim ke *server*.

Program pembacaan data dari inputan *keypad* adalah :

```

If MSComm1.CommEvent = comEvReceive Then
    datmasuk = MSComm1.Input
    If datmasuk = "P" Then
        b = ""
    ElseIf datmasuk = "X" Then
        makanan = b
  
```

```

p1 = b
b = ""
ElseIf datmasuk = "I" Then
    minuman = b
    p2 = b
    b = ""
ElseIf datmasuk = "R" Then
    snack = b
    p3 = b
    b = ""

```

Program untuk *connect server* adalah :

```

Private Sub Command1_Click() 'conect ke server
    If Winsock1.State = sckClosed Then
        Call Winsock1.Connect
    Else
        Call Winsock1.Close
        Call Winsock1.Connect
    End If
End Sub

```

Program untuk kirim data *server* adalah :

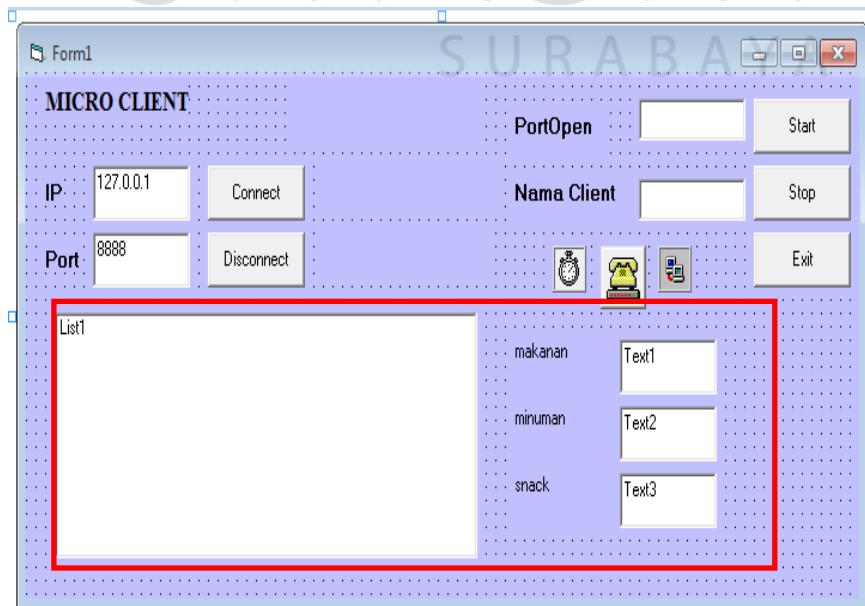
```

If Winsock1.State = sckConnected Then
    menu = "P" + p1 + "X" + p2 + "I" + p3 + "R" +
    namaclient + "D" + oke + "T"
    Call Winsock1.SendData(menu)
End If

```

Desain dan kegunaan *form microcontroller client* dapat dilihat pada Gambar

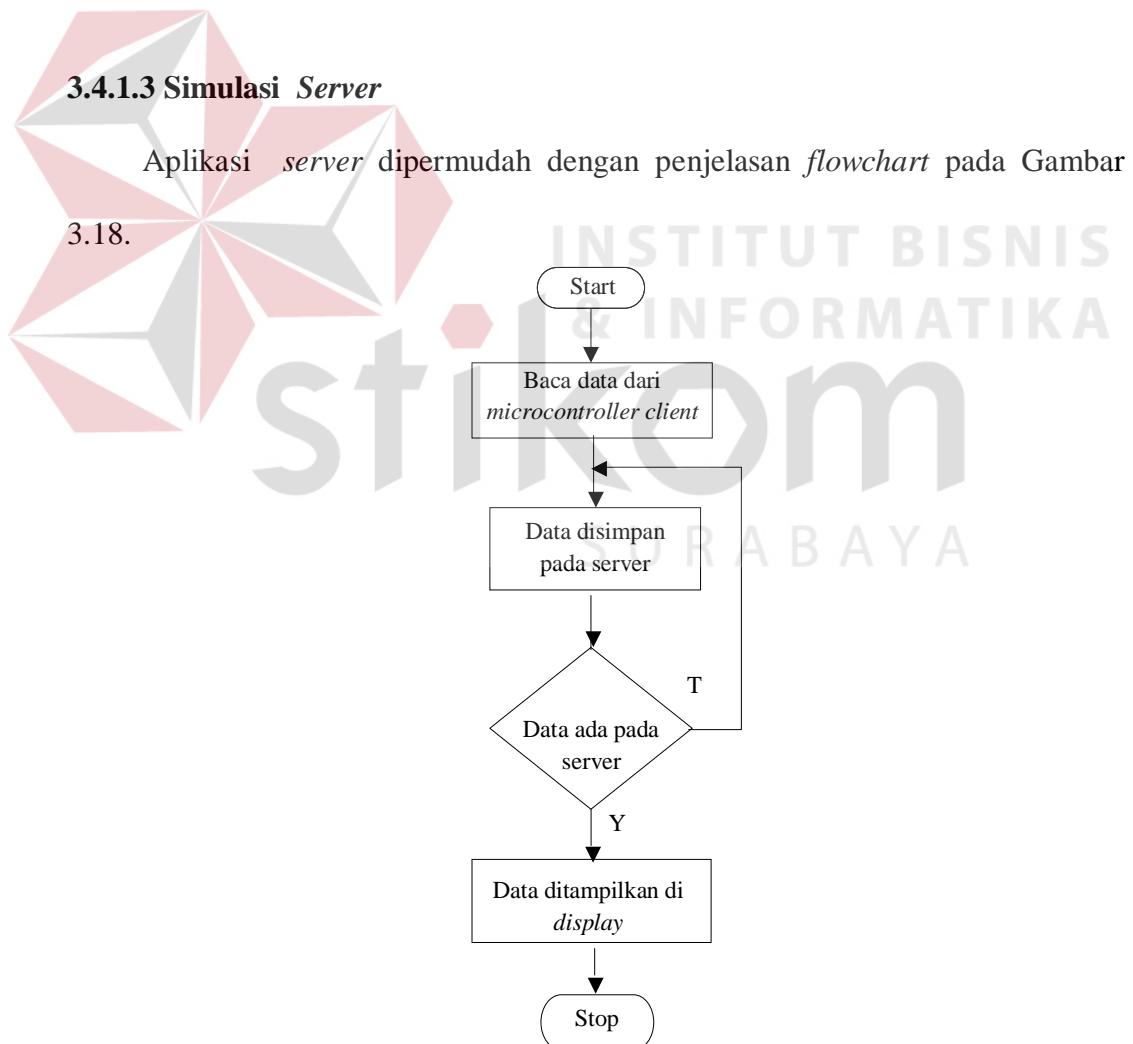
3.17.



Gambar 3.17. *form microcontroller client*

Keterangan Gambar 3.17 :

- Tombol Start : Menerima data dari *keypad*
- Tombol Stop : Menghentikan penerimaan data dari *keypad*
- Tombol Exit : Keluar dari simulasi
- Tombol Connect : Menghubungkan koneksi dan mengirimkan data ke *server*
- Tombol Disconnect : Memutuskan atau menhentikan koneksi pada *server*
- Kotak Merah : Hasil yang diterima dari *keypad*



Gambar 3.18. *Flowchart Simulasi Server*

Penjelasan *flowchart* pada Gambar 3.18 bahwa program simulasi *server* menerima data dari *microcontroller client*. Data yang diinputkan dari *microcontroller client* kemudian data disimpan ke *server*. Data yang masuk pada *server* disimpan dan ditampilkan pada PC *sever*. Sebelum data dikirim kembali ke *microcontroller* data tersebut harus dipastikan sudah tersimpan di *server* dan kemudian ditampilkan ke *moving sign display*.

Program untuk baca data dari *microcontroller client* adalah :

```

For ex = 1 To Len(file) 'memisahkan data dari file
    If Mid(file, ex + 1, 1) = "," Then
        koma = koma + 1
    If koma = 1 Then
        var = ""
    ElseIf koma = 2 Then
        var1 = Mid(var, 3, Len(var))
        var = ""
    ElseIf koma = 3 Then
        var2 = Mid(var, 3, Len(var))
        koma = 1
        var = ""
    End If
    ElseIf Mid(file, ex + 1, 1) = "" Then
        var3 = Mid(file, ((Len(file) - Len(var)) + 1), Len(var))
        var = ""
    Else
        var = var + Mid(file, ex, 1)
    End If
Next ex

```

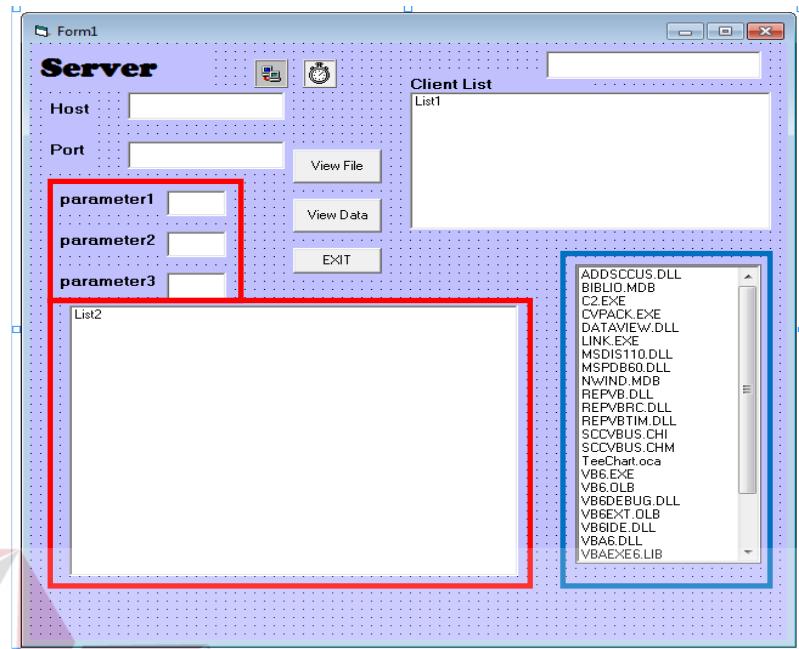
Program untuk menyimpan data pada *server* adalah :

```

If "D:\TA-VB\TA\server\" & nama & ".txt" = Null Then
    Open "D:\TA-VB\TA\server\" & nama & ".txt" For Output
    As #1
    Write #1, oke, nama, par1, par2, par3
    Close #1
Else
    Open "D:\TA-VB\TA\server\" & nama & ".txt" For Append
    As #1
    Write #1, oke, nama, par1, par2, par3
    Close #1
End If

```

Desain dan kegunaan *form server* dapat dilihat pada Gambar 3.19.



Gambar 3.19. *form microcontroller client*

Keterangan Gambar 3.19 :

- Tombol View File : Merefresh list yang ada pada kotak berwarna biru
- Tombol View data : Menampilkan data yang telah diterima dari *microcontroller client*
- Tombol Exit : Keluar dari simulasi