

Sistem Kendali Dan Monitoring Jarak Jauh Peralatan Elektronik Rumah Tangga Melalui Media Transmisi Jala-Jala Listrik

Oleh :

Erik Mubarak / L2F 000 599

e-mail : mubarak.erik@gmail.com

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro Semarang

Abstrak - Transmisi data atau informasi dari satu tempat ke tempat lain dapat dilakukan dalam berbagai cara dan media. Penggunaan jala-jala listrik sebagai media komunikasi merupakan salah satu alternatif yang memiliki beberapa kelebihan, seperti penghematan penggunaan media kabel. Penerapan teknologi ini secara lebih luas adalah pada pengintegrasian sistem kontrol/kendali pada suatu bangunan. Namun demikian, jaringan listrik masih merupakan media komunikasi yang karakteristik listriknya berbeda dengan media lainnya. Pengaruh redaman maupun noise yang timbul pada jaringan listrik dapat mengakibatkan terganggunya proses pengiriman data.

Realisasi sistem kendali dengan media komunikasi jala-jala listrik pada tugas akhir ini merupakan salah satu usaha untuk memanfaatkan media tersebut sebagai penyalur data/informasi yang terutama ditujukan pada sistem kontrol jarak jauh, (*remote control system*).

Dalam tugas akhir ini dibuat suatu aplikasi khusus dari komunikasi data melalui jaringan listrik, yaitu penggunaan komputer sebagai pengendali utama untuk melakukan kontrol ON/OFF terhadap beban listrik serta monitoring suhu ruangan yang dilakukan dari jarak jauh. Data dikirim oleh komputer melewati jaringan listrik, kemudian pada bagian penerima, data didistribusikan pada beban listrik sesuai dengan informasi yang diberikan oleh komputer pengendali. Untuk proses pendistribusian data pada penerima, digunakan mikrokontroler. Mikrokontroler merespon data yang dikirim oleh komputer untuk melakukan proses pengendalian beban listrik.

Kata kunci : SNAP Protokol, CRC-16, Driver Lampu, Detektor Arus, Pengkondisi Sinyal, Sensor Suhu, Mikrokontroler AT89S51.

I. PENDAHULUAN

I.1 Latar Belakang

Perkembangan teknologi komputer digital yang sangat pesat, mempunyai dampak yang luas terhadap sistem komunikasi dan kendali secara keseluruhan. Dewasa ini, istilah komunikasi data atau komunikasi komputer bukan lagi merupakan sesuatu yang asing.

Komunikasi data tidak terlepas dari media transmisi yang harus ada, agar komunikasi dapat dilakukan. Penggunaan jalur telepon untuk komunikasi data sudah lama dilakukan. Muncullah istilah modem, yang merupakan perantara (*interface*) antara komputer dan saluran transmisi. Modem ini mengubah data digital dari komputer ke dalam bentuk sinyal analog yang akan dikirim lewat saluran telepon.

Komunikasi data lewat jala-jala listrik pun sudah dijangkau kemungkinannya. Telah tersedia *chip* khusus di pasaran yang didesain sebagai modem untuk *interface* ke jala-jala listrik.

Penggunaan jaringan listrik sebagai media transmisi untuk menyalurkan informasi telah lama dilakukan. Di Indonesia sendiri PLN

(Perusahaan Listrik Negara) telah lama menggunakan sistem ini untuk kepentingan komunikasi yang lebih populer dikenal dengan nama PLC (*Power Line Carrier*). Hanya saja perkembangannya di Indonesia tidak secepat di negara lain, yang sudah menggunakan media ini untuk mengakses internet dengan kecepatan tinggi.

I.2 Tujuan

Tujuan yang hendak dicapai pada Tugas Akhir ini adalah:

- Mempelajari karakteristik dari jaringan listrik yang akan digunakan sebagai media komunikasi/pengendalian.
- Mempelajari, memahami dan kemudian mengimplementasikan protokol komunikasi yang digunakan untuk menghubungkan bagian pengendali dan yang dikendalikan agar bisa saling berkomunikasi satu sama lain.

I.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini penulis membatasi permasalahan sebagai berikut :

- 1) Karakter Jaringan Listrik yang digunakan yaitu pada jaringan listrik 1 fasa 220V/50Hz.

- 2) Pembahasan dan Penelitian Tidak Mencakup Pembuatan Modem.
- 3) Tidak Membahas Modulasi Digital yang digunakan
- 4) Penelitian dititikberatkan pada pemanfaatan protokol SNAP pada pengendalian jarak jauh melalui media jaringan listrik.
- 5) Komputer pada bagian server/pengirim digunakan sebagai pengendali utama untuk memonitor suhu ruangan, dan meng-ON/OFF-kan lampu ruangan.
- 6) Mikrokontroller pada bagian *client*/penerima digunakan sebagai *interfacing*/antarmuka untuk merespon permintaan komputer pengendali untuk mengaktifkan peralatan yang dikehendaki.
- 7) Pembahasan tidak mencakup arsitektur perangkat komputer.
- 8) Bahasa Pemrograman yang digunakan yaitu Delphi.

II. DASAR TEORI

2.1 Sistem PLC

Saluran distribusi daya, selain dapat digunakan untuk menyalurkan energi listrik, ia dapat juga digunakan untuk menyalurkan sinyal informasi baik berupa suara maupun data. Penggunaan saluran distribusi daya sebagai media transmisi untuk keperluan komunikasi dikenal dengan istilah *Power Line Carrier*(PLC).

Pada dasarnya PLC hanya menumpangkan sinyal informasi dan sinyal pembawa pada kawat listrik, oleh sebab itu sistem ini masih bisa berfungsi meskipun tidak ada tegangan listrik yang mengalir asalkan kabelnya tidak terputus. Pada sistem ini tegangan listrik diblokir oleh rangkaian kopling tetapi yang diloloskan adalah sinyal-sinyal yang dikehendaki saja. Selanjutnya sinyal-sinyal tersebut akan dipisahkan antara sinyal informasi dengan sinyal pembawanya.

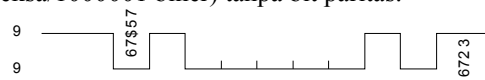
Keuntungan sistem PLC dibandingkan dengan sistem komunikasi lainnya adalah tidak diperlukannya pembangunan media transmisi lagi karena memanfaatkan jaringan kawat distribusi daya yang sudah ada. Selain itu sistem distribusi daya mencakup daerah yang sangat luas dan merupakan salah satu bagian terpenting dari kehidupan modern sehingga menyimpan potensi lain yang dapat terus dikembangkan.

2.2 Komunikasi Serial Asinkron

Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara

asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama-sama dengan data serial, sedangkan komunikasi data serial asinkron, clock tidak dikirimkan bersama-sama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receiver*). Komunikasi data serial asinkron ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial dan menerima data serial yang kemudian diubah kembali menjadi data paralel.

Pada UART, kecepatan pengiriman data (*baud rate*) dan fase clock pada sisi *transmitter* dan pada sisi *receiver* harus sinkron. Untuk itu diperlukan sinkronisasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit 'Start' dan bit 'Stop'. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika '1'. Ketika *transmitter* ingin mengirimkan data, output UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal 'Start' yang digunakan untuk mensinkronkan fase clocknya sehingga sinkron dengan fase clock *transmitter*. Selanjutnya data akan dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirim sinyal 'Stop' sebagai akhir dari pengiriman data serial. Cara pemberian kode data yang disalurkan tidak ditetapkan secara pasti. Berikut ini adalah contoh pengiriman huruf 'A' dalam format ASCII (41 heksa/1000001 biner) tanpa bit paritas.



Gambar 1 Pengiriman huruf 'A' tanpa bit paritas

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. Baud rate yang umum dipakai adalah 110,135, 150,300,600,1200,2400,9600 (bit/detik). Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya, harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil atau tanpa paritas), dan jumlah bit 'Stop' (1, 1 1/2, atau 2 bit).

2.3 Konfigurasi Komunikasi Serial Dalam BASCOM

BASCOM memberikan kemudahan dalam hal komunikasi secara serial, sehingga tidak perlu mengkonfigurasi register secara langsung. Hanya ada dua hal yang harus dikonfigurasi ketika menggunakan komunikasi serial dengan BASCOM, yaitu kristal dan *baud rate* yang digunakan.

Untuk mendeklarasikan *baud rate* dan kristal tambahkan ke dalam program perintah berikut :

```
$Baud = 9600
$Crystal = 11059200
```

Perintah diatas dapat dijelaskan sebagai berikut, *baud rate* yang digunakan yaitu 9600 dengan menggunakan kristal 11.059.200 Hz

2.4 Rutin-rutin Untuk Komunikasi Serial

BASCOSM telah menyediakan beberapa rutin untuk berkomunikasi secara serial, antara lain:

A. WAITKEY

Berfungsi untuk menunggu hingga sebuah karakter diterima pada buffer serial, adapun cara penulisannya adalah sebagai berikut,

```
Var = WAITKEY()
```

Keterangan:

Var Variabel untuk menampung karakter dari buffer serial dalam bentuk ASCII (var:byte, integer, word, long,string)

B. INPUTBIN

Berfungsi untuk membaca nilai biner dari port serial. Cara penulisannya adalah,

```
INPUTBIN var1[,var2]
```

Keterangan :

Var1 Variabel yang ditentukan untuk menampung karakter pertama dari port serial

Var2 Variabel yang ditentukan untuk menampung karakter kedua dari port serial (opsional)

- Jumlah *byte* yang akan dibaca tergantung pada variabel yang digunakan. ketika variabel yang digunakan hanya 1 *byte*, maka hanya 1 karakter yang terbaca dari port serial. *Integer* akan menunggu hingga 2 karakter dan jika menggunakan variabel *array*, maka jumlah karakternya sebanyak anggota *array*.
- Pernyataan INPUTBIN tidak menunggu tanda <enter> atau karakter 13H, tetapi hanya menunggu jumlah *byte*.
- Data yang disimpan berupa kode ASCII

C. PRINT

Berfungsi untuk mengirim data ke port serial. Cara penulisannya adalah,

```
PRINT var; "constant"
```

Keterangan:

var Variabel atau konstanta yang akan di print.

- Untuk mengirim lebih dari satu variabel, gunakan titik koma (;)

D. PRINTBIN

Berfungsi untuk mengirim isi biner dari variabel ke port serial. Cara penulisannya adalah,

```
PRINTBIN var [; varn]
```

```
PRINTBIN #dev,var; [,varn]
```

Keterangan:

Var Variabel yang dikirim ke port serial.

Varn Variabel opsional yang akan dikirim dipisahkan dengan tanda ;
#dev Jumlah komponen untuk digunakan dengan open dan close.

- Printbin sama dengan perintah `print chr(var)`, namun seluruh array dapat dikirim dengan cara demikian.
- Ketika menggunakan tipe data *long*, maka data yang dikirim sebanyak 4 *byte*.

2.5 Protokol SNAP (Scaleable Node Address Protocol)

Protokol SNAP pertama kali dikembangkan oleh perusahaan HTH untuk keperluan komunikasi data dari produk PLM-24(Power Line Modem) yang digunakan untuk mengendalikan peralatan listrik rumah tangga. Protokol ini dapat digunakan untuk keperluan yang cukup luas dan tidak terbatas hanya pada komunikasi melalui media jaringan listrik, tetapi dapat digunakan oleh berbagai media seperti, kabel, RF(Radio Frekuensi), IR(Infra Red), dan lainnya. Fleksibilitasnya terhadap jenis dan tipe mikrokontroller, protokol ini dapat digunakan sebagai alternatif untuk keperluan komunikasi data, khususnya dalam sistem pengendalian. Peralatan SNAP berkomunikasi dengan teknik *server-client*, dimana hanya salah satu peralatan (*server*) yang dapat menginisiasi komunikasi tersebut. Inisiasi ini disebut "SNAP Packet". Peralatan yang lain (*client*) memberikan respon dengan mengirimkan data yang diminta ke server atau melakukan aksi yang diminta dalam *SNAP Packet*. *Client*, yang dapat berupa I/O, transduser, *valve*, peralatan ukur lainnya, memproses informasi dan mengirimkannya ke server melalui protokol SNAP. Jumlah *client*

maksimum yang dapat ditangani oleh sebuah server adalah sebanyak 16 777 215 buah.

SNAP Packet yang dikirimkan *server* terdiri atas alamat *server*, alamat *client*, kode fungsi yang mendefinisikan aksi yang diminta, data yang dibutuhkan, dan bagian pengecekan kesalahan (*error checking*). Respon dari *client* terdiri atas konfirmasi aksi yang dilakukan, data yang harus dikirimkan, dan bagian pengecekan kesalahan. Jika terdapat suatu kesalahan pada penerimaan *query* atau jika *client* tidak dapat melakukan aksi yang diminta, *client* akan mengirimkan pesan *exception*. Bagian pengecekan kesalahan akan membantu server untuk melihat apakah pesan yang diterima valid atau terdapat suatu kesalahan.

Panjang paket data yang akan dikirimkan bisa berbeda-beda, tergantung dari jumlah *client*/alamat yang akan dituju(0-6 Bytes), banyaknya data/informasi yang akan ditransmisikan (0-512 Bytes), juga metoda deteksi kesalahan yang akan digunakan. Semua informasi diatas didefinisikan oleh suatu kode fungsi yang dinamakan *Header Definition Bytes* (HDB2 dan HDB1).

Setiap paket diawali dengan suatu *byte* yang dinamakan *preamble bytes*. Fungsinya yaitu sebagai kalibrasi *byte-byte* dari paket yang akan dikirimkan. Kemudian diikuti dengan *byte* yang unik(01010100₂) yang dinamakan *Synchronization Byte*, sebagai sinkronisasi untuk pengiriman suatu paket. Tipe dari karakter *preamble bytes* tersebut bisa didefinisikan sendiri asalkan tidak sama dengan karakter dari *Synchronization Byte* (01010100₂).

Sebagai contoh, dibawah ini merupakan *SNAP Packet* yang terdiri dari beberapa bagian kode fungsi, diantaranya:

SYNC *Synchronization Byte*
HDB2 *Header Definition Byte 2*
HDB1 *Header Definition Byte 1*
DAB *Destination Address Byte*
SAB *Source Address Byte*
DB *Data Byte1*
CRC2 *High Byte of CRC-16 (Kode Deteksi Kesalahan)*
CRC1 *Low Byte of CRC-16 (Kode Deteksi Kesalahan)*

SYNC HDB2 HDB1 DAB SAB DB CRC2 CRC1

Paket data diatas, memiliki panjang 8 *byte*(tidak termasuk *byte preamble*). Proses Pengiriman data dimulai dari *byte* yang paling kiri (LSB, *Least Significant Byte*) ke *byte* paling kanan (MSB, *Most Significant Byte*)

Sebagai contoh, akan dikirimkan paket data dengan panjang 8 *byte*. Alamat asal yang digunakan yaitu 00000001₂ dan alamat yang dituju yaitu 00000010₂. Sedangkan data yang dikirimkan berisi 11111111₂. Sinyal ACK tidak digunakan, sehingga tidak ada respon dari penerima. Maka, kode fungsi HDB2 dan HDB1 adalah sebagai berikut,

DD = 01 1 Byte alamat tujuan
SS = 01 1 Byte alamat asal
PP = 00 PFB tidak digunakan
AA = 00 Sinyal ACK tidak
C = digunakan
EEE = 0 Command mode tidak
NNNN = digunakan
100 Deteksi kesalahan yang digunakan 16-bit CRC
0001 1 Byte data

Paket Data yang dikirim,

SYNC = 01010100
HDB2 = 01010000
HDB1 = 01000001
DAB = 00000010
SAB = 00000001
DB = 11111111
CRC2 = 01001110
CRC1 = 10111011

2.6 CRC-16

Kode pendeteksian kesalahan yang paling umum serta paling hebat adalah *Cyclic Redundancy Check* (CRC) yang dapat digambarkan sebagai berikut, dengan adanya blok k-bit, atau pesan, transmitter mengirimkan suatu deretan n-bit, disebut sebagai *Frame Check Sequence* (FCS), sehingga frame yang dihasilkan, terdiri dari k+n bit, dapat dibagi dengan jelas oleh beberapa nomor yang sebelumnya sudah ditetapkan. Kemudian receiver membagi frame yang datang dengan nomor tersebut dan, bila tidak ada sisa, maka diasumsikan tidak terdapat kesalahan.

Ada beberapa cara mengamati proses CRC. Diantaranya adalah dengan menyatakan seluruh nilai sebagai polinomial dalam suatu model variabel X, dengan koefisien-koefisien biner. Koefisien berhubungan dengan bit-bit dalam angka biner. Jadi untuk M = 110011, kita peroleh $M(X) = X^5 + X^4 + X + 1$, dan untuk P = 11001, kita peroleh $P(X) = X^4 + X^3 + 1$. Operasi aritmatik lagi-lagi berupa modulo 2. sekarang proses CRC digambarkan sebagai:

$$\frac{X^n M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)} \quad (1)$$

$$T(X) = X^n M(X) + R(X) \quad (2)$$

Empat versi P(X) yang telah digunakan secara luas adalah:

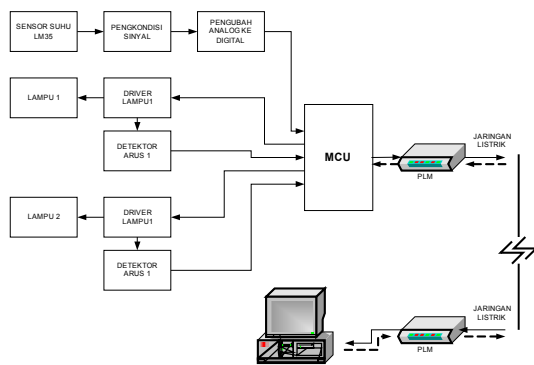
CRC-12	=	$X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16	=	$X^{16} + X^{15} + X^2 + 1$
CRC-CCITT	=	$X^{16} + X^{12} + X^5 + 1$
CRC-32	=	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Sistem CRC-12 dipergunakan untuk transmisi sederetan sebesar 6-bit karakter dan membangkitkan 12-bit FCS. Baik CRC-16 maupun CRC-CCITT populer untuk 8 bit karakter, masing-masing di Amerika Serikat dan Eropa, di mana kedua-duanya sama-sama menghasilkan 16-bit FCS. Nampaknya ini sesuai untuk sebagian besar aplikasi, meskipun CRC-32 ditentukan sebagai salah satu pilihan untuk standar transmisi data.

III. PERANCANGAN SISTEM

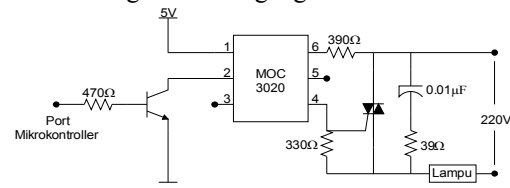
III.1 Perancangan Perangkat Keras

Perancangan perangkat keras ini meliputi pembuatan driver lampu yang digunakan untuk mengendalikan lampu AC tegangan bolak-balik 220V/50, pembuatan detektor arus untuk mengetahui kondisi lampu, sensor suhu LM35, pengkondisi sinyal, pengubah analog ke digital (ADC), dan mikrokontroler sebagai pengendali peralatan yang di-set agar dapat berkomunikasi secara serial. Mikrokontroler inilah yang merespon permintaan dari komputer untuk melakukan proses monitoring suhu dan kendali lampu.

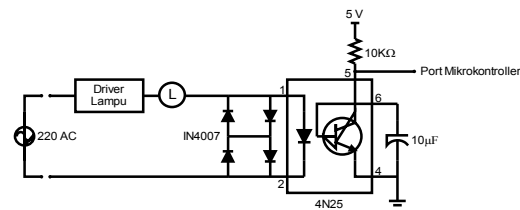


Gambar 2 Blok Sistem Komunikasi

Secara umum, perancangan perangkat keras ini meliputi, perancangan blok *driver* lampu yang digunakan untuk menghidupkan dan mematikan lampu 220V/50Hz dengan tegangan rendah dari mikrokontroler. Blok ini menggunakan TRIAC untuk proses pensaklarannya. Kemudian blok detektor arus. blok ini digunakan untuk mendeteksi masalah pada lampu. Jika terjadi masalah pada lampu, blok ini akan aktif, dan menginformasikan kepada komputer. Pada perancangan blok detektor arus ini digunakan komponen *optocoupler* yang berfungsi untuk mengisolasi rangkaian dari jaringan listrik. Dan dioda clipper untuk menghasilkan tegangan rendah.



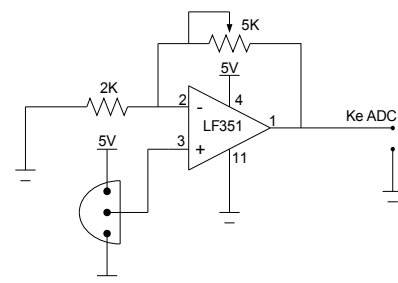
Gambar 3 Driver Lampu



Gambar 4 Detektor Arus

Pada blok akuisisi data suhu, ada beberapa komponen yang digunakan yaitu diantaranya sensor suhu LM35, pengkondisi sinyal, dan ADC. Sensor ini yang akan merubah besaran fisik seperti besaran temperatur ke dalam besaran listrik. Karena tegangan keluaran sensor ini masih rendah, maka diperlukan suatu penyesuaian agar keluarannya menjadi optimal.

Rangkaian pengkondisi sinyal ini merupakan rangkaian penguat tegangan dengan menggunakan penguat operasional tak membalik seperti gambar 3.4 berikut ini,



Gambar 5 Pengkondisi Sinyal dengan IC LF351

Tegangan keluaran maksimum yang diinginkan adalah 5 Volt, sedangkan tegangan

masukan pengkondisi sinyal yang merupakan tegangan keluaran sensor hanya memiliki nilai maksimum sebesar 1,5 V. Sehingga diperlukan penguatan sebesar:

$$A = \frac{V_{out}}{V_{in}} = \frac{5}{1,5} = 3,333$$

Sehingga dari persamaan penguat tak membalik diperoleh persamaan sebagai berikut,

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1} + 1 = 3,333$$

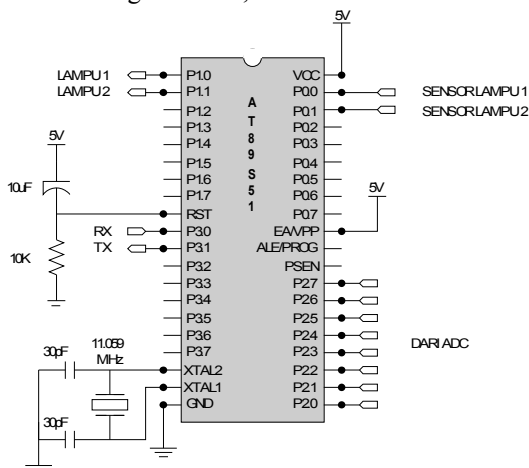
Dari persamaan diatas didapatkan nilai perbandingan R_2 dan R_1 ,

$$\frac{R_2}{R_1} = 2,333$$

Ditentukan nilai R_1 sebesar 2K Ω . Untuk menghasilkan perbandingan yang sesuai maka nilai R_2 diatur sebesar 4,666K Ω .

Kemudian keluaran dari pengkondisi sinyal ini akan dirubah ke dalam data digital dengan menggunakan ADC 0804. ADC ini mempunyai resolusi 8 bit dengan perhitungan $5/255 = 19.6$ mV. Jadi setiap bit mempunyai resolusi tegangan 19.6 mV. Data digital inilah yang akan diproses oleh mikrokontroler dan kemudian dikirim ke komputer untuk ditampilkan nilai sesungguhnya dari data suhu.

Sedangkan untuk sistem pengendalian utama, digunakan mikrokontroler yang akan memproses data masukan dan keluaran dari rangkaian diatas. Adapun sistem minimumnya adalah sebagai berikut,



Gambar 6 Sistem Minimum Mikrokontroler AT89S51

III.2 Perancangan Perangkat Lunak

Dalam Tugas Akhir ini, diimplementasikan protokol SNAP (*Scaleable Node Address Protocol*) agar komputer dan mikrokontroler dapat saling berkomunikasi dan mengirim informasi. Sedangkan untuk

mendeteksi kesalahan dalam proses pengiriman data, digunakan metode deteksi kesalahan CRC-16.

Pada perancangan perangkat lunak ini bahasa pemrograman yang digunakan yaitu *Delphi* dan *Basic*. Program *Delphi* digunakan komputer untuk mengirim data kendali dan memonitor data yang dikirim oleh mikrokontroler, yang ditampilkan secara grafis sebagai antar muka untuk mempermudah pengguna dalam mengoperasikan sistem ini. Sedangkan bahasa *Basic* digunakan oleh mikrokontroler untuk menerima data dari komputer dan mengirim data ke komputer.

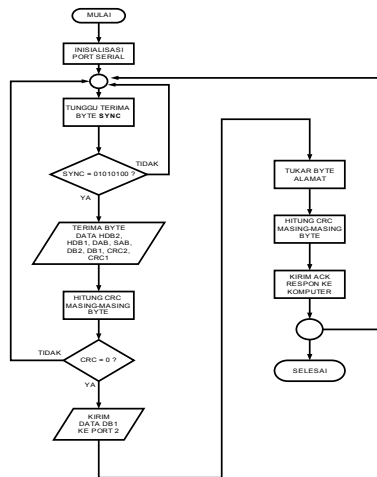
Perancangan sistem komunikasi dalam tugas akhir ini, dikirimkan 1 paket data yang berisi 9 byte yang terdiri dari byte SYNC, byte HDB2, byte HDB1, byte DAB, byte SAB, byte DB2, byte DB1, byte CRC2, byte CRC1.

Perangkat lunak yang digunakan untuk memprogram mikrokontroler yaitu bahasa *Basic*. Pada perancangan ini terdapat 3 proses yaitu, mikrokontroler menerima data dari komputer yang berisi informasi untuk mengendalikan lampu. Proses kedua, mikrokontroler mengirimkan data ke komputer yang berisi informasi akuisisi data suhu dan proses ketiga, mikrokontroler mengirim data yang berisi status lampu. Pada proses kedua dan ketiga, data dikirim secara bersamaan secara terus menerus.

Pada proses ini mikrokontroler menunggu data yang dikirim oleh komputer yang berisi informasi pengendalian lampu. Lampu yang digunakan sebanyak 2 buah.

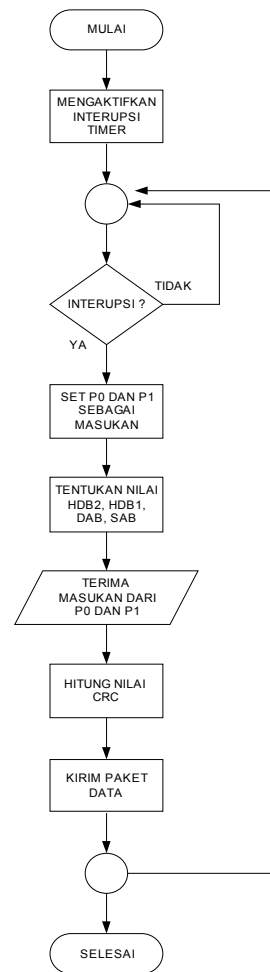
Pertama-tama mikrokontroler mendeteksi adanya byte SYNC sebagai awal proses pengiriman data oleh komputer ke mikrokontroler. Byte SYNC ditetapkan yaitu 01010100 (54 Heksa). Mikrokontroler melakukan pengecekan apakah byte SYNC yang dikirim oleh komputer sama. Jika sama, maka mikrokontroler selanjutnya akan menerima byte HDB2, HDB1, DAB, SAB, DB2, DB1, CRC2, CRC1.

Diagram Alir program pengendalian lampu seperti diperlihatkan pada gambar 7 di bawah ini



Gambar 7 Diagram Alir Program Pengendalian Lampu

Pada proses *monitoring* lampu dan akuisisi data suhu, data dikirimkan secara terus menerus. Data suhu disimpan di DB2, sedangkan data *monitoring* lampu, disimpan di DB1. Data suhu didapatkan dari ADC yang merubah data masukan analog dari sensor suhu yang telah disesuaikan nilainya dengan menggunakan pengkondisi sinyal menjadi masukan ke ADC. Sedangkan data *monitoring* lampu didapat dari masukan *port* mikrokontroler P1.0 dan P1.1, keluaran rangkaian detektor arus. proses ini memanfaatkan interupsi timer 0 untuk melakukan pengiriman. Jadi data dikirimkan pada saat terjadi *overflow* pada timer 0.



Gambar 8 Diagram Alir Program Monitoring Lamu dan Akuisisi Data Suhu

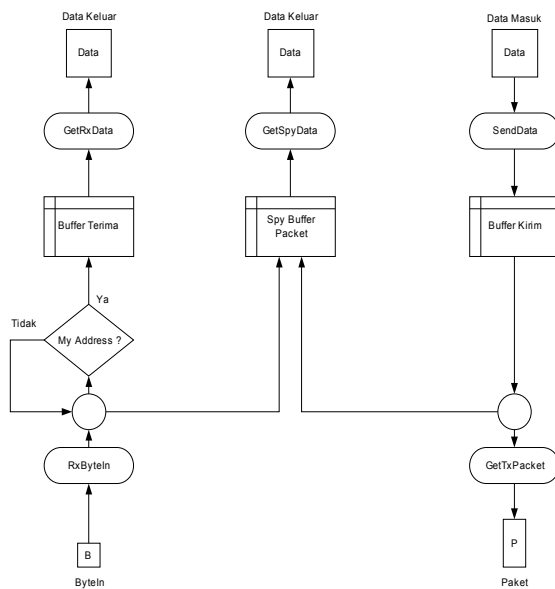
Pada perancangan perangkat lunak di komputer, digunakan Bahasa pemrograman visual Delphi. Program ini digunakan sebagai antar muka pengguna untuk mempermudah pengguna dalam mengoperasikan sistem ini. Tampilan pada program ini diantaranya yaitu, jendela kontrol lampu dan tampilan data suhu ruangan yang ditampilkan berupa *seven segment*. Jendela berikutnya yaitu, hasil akuisisi data suhu yang ditampilkan ke dalam bentuk grafik. Pada jendela ini, data hasil akuisisi dapat direkam, dan aktivitas proses akuisisi dapat dilihat pada *file* yang berisi hasil rekaman aktivitas akuisisi data suhu. Jendela terakhir yaitu, aktivitas statistik data baik pada proses pengiriman dan penerimaan data oleh komputer. Pada jendela ini dapat dilihat secara *detail*, proses komunikasi yang terjadi, proses pengiriman setiap *byte*-nya, juga jika terjadi kesalahan dalam proses pengiriman maupun penerimaan data. Pada perancangan perangkat lunak ini, dibagi menjadi 3 bagian diantaranya yaitu,

2. Proses Penerimaan Data oleh Komputer dari Mikrokontroler
3. Proses Pengiriman Data oleh Komputer ke Mikrokontroler
4. Proses Aktivitas Komunikasi

Pada proses penerimaan data, data akan diterima apabila alamat tujuan yang dikirim bersama paket data sesuai dengan alamat komputer. Data akan dikirim setiap *byte* dan masing masing dihitung nilai CRC-nya.

Pada proses pengiriman data, paket data akan ditampung pada *buffer* kirim sebelum data dikirimkan satu persatu.

Pada proses aktivitas komunikasi, ditampilkan statistik data yang menginformasikan paket data yang dikirim dan diterima dalam rentang waktu tertentu. Data ini dapat ditampilkan setiap waktu.



Gambar 9 Diagram Alir Program Delphi

IV. PENGUJIAN DAN ANALISIS

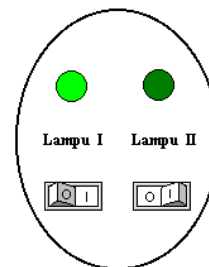
Pada perangkat lunak ini ada beberapa aktivitas yang dapat dilakukan. Pada Jendela Monitoring terdapat bagian pengendalian 2 lampu beserta indikator status lampu, dan tampilan suhu ruangan.



Gambar 10 Jendela Monitoring

IV.1 Driver Lampu

Pada pengujian lampu pertama, menghidupkan lampu 1 dengan menekan tombol saklar lampu 1 pada layar, hasilnya menjadi,



Gambar 11 Tampilan Ketika Saklar Lampu 1 Ditekan

Dari gambar diatas dapat dilihat perubahan posisi saklar lampu 1 dan indikator lampu diatasnya menyala. Hal ini menandakan lampu tersebut aktif. Untuk mengetahui apakah data lampu 1 sampai ke mikrokontroler, dapat dilihat pada proses aktivasi dibawah,

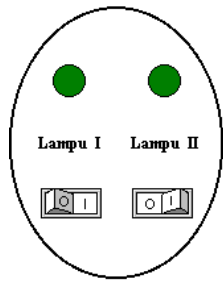
AKTIVITAS KOMUNIKASI							
Waktu	Sumber	Tujuan	ACK	CMD	EDM	NNN	Data
23/08/07, 20:47:59	BC	CB	1	0	4	2	FE00
23/08/07, 20:48:00	CB	BC	1	0	4	2	FF01
23/08/07, 20:48:00	BC	CB	2	0	4	2	FF01

Gambar 12 Aktivitas Komunikasi Menghidupkan Lampu 1

Pada gambar diatas dapat dilihat, komputer mengirimkan data lampu 01 untuk menghidupkan lampu 1. Kemudian mikrokontroler merespon dengan mengirim kembali data dengan nilai ACK (*Acknowledge*) 2 yang berarti mikrokontroler mengirim ACK *response*.

IV.2 Detektor Arus

Pada pengujian detektor arus lampu pertama, pada saat tombol saklar lampu 1 ditekan dan lampu 1 dilepas dari rangkaian, maka indikator lampu menjadi tidak aktif. Prosesnya dapat dilihat pada gambar dibawah,



Gambar 12 Indikator Lampu 1 Mati

Dari gambar dapat dilihat bahwa meskipun saklar lampu 1 ditekan, indikator lampu 1 tidak aktif. Hal ini menandakan terjadi masalah pada lampu 1. Mikrokontroler menerima masukan dari detektor arus lampu 1, kemudian dikirimkan ke komputer. Untuk mengetahui aktivitas komunikasinya dapat dilihat pada gambar dibawah ini,

AKTIVITAS KOMUNIKASI							
Waktu	Sumber	Tujuan	ACK	CMD	EDM	NNN	Data
23/08/07, 20:54:44	BC	CB	1	0	4	2	FE01
23/08/07, 20:54:45	CB	BC	1	0	4	2	FF01
23/08/07, 20:54:45	BC	CB	2	0	4	2	FF01
23/08/07, 20:54:45	BC	CB	1	0	4	2	FF00

Gambar 13 Aktivitas Komunikasi Pada Saat Detektor Arus Lampu 1 Bekerja

Pada saat lampu masih terpasang, komputer mengirim data 01 untuk menghidupkan lampu1. mikrokontroler merespon dengan mengirim kembali data dengan memberikan sinyal *ACK response*. Ketika lampu dilepas, mikrokontroler mengirim data 00 yang mengindikasikan bahwa detektor arus aktif.

IV.3 Akuisisi Data Suhu

Pada pengujian perangkat lunak proses akuisisi data suhu, karena pengkondisi sinyal tidak dapat digunakan, pengujian dilakukan dengan memberikan tegangan masukan langsung ke ADC sebesar 4.98 V. Tampilan pada jendela program monitoring suhu adalah seperti pada gambar dibawah ini,



Gambar 14 Tampilan Monitoring Suhu

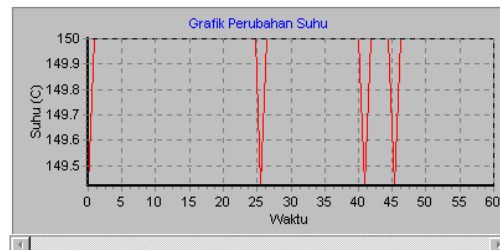
Proses konversi dari tegangan menjadi keluaran suhu dapat dihitung dengan menggunakan persamaan:

$$\begin{aligned} \text{penguatan} &= \text{tegangan masukan} * (1.48/5) \\ \text{keluaran suhu} &= 2 + \text{penguatan} * 100 \end{aligned}$$

pada pengujian diatas tegangan masukan 4.98 V, maka dihasilkan

$$\begin{aligned} \text{penguatan} &= 4.98 * (1.48/5) \\ &= 147.4 \\ \text{keluaran suhu} &= 2 + (147.4 * 100) \\ &= 149.4 \text{ Celcius} \end{aligned}$$

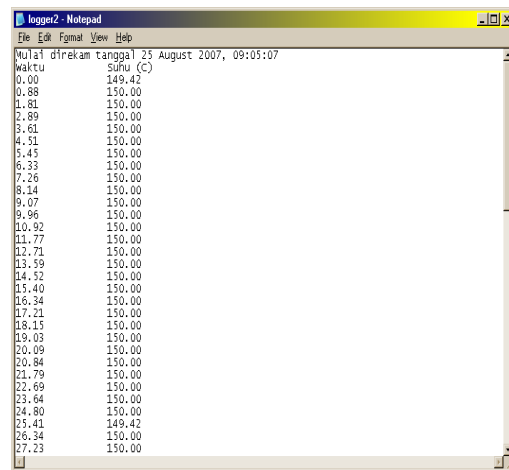
Pada jendela grafik dapat dilihat hasil akuisisi data suhu. Pada tegangan masukan 4.98 V dapat dilihat pada gambar dibawah ini,



Gambar 15 Grafik Akuisisi Suhu

Pada grafik dapat dilihat terjadi perubahan suhu. Hal ini disebabkan tegangan masukan yang diberikan nilainya berubah-ubah, sehingga data yang ditampilkan mengikuti perubahan tegangan.

Untuk merekam data suhu dapat juga pada data dalam bentuk *file* yang berisi informasi pengambilan data suhu pada setiap waktunya.



Gambar 16 Rekaman Aktifitas Akuisisi Data

IV.4 Pengujian Statistik Komunikasi Data

Pada perangkat lunak Tugas Akhir ini dapat dilihat statistik komunikasi data baik data yang dikirim maupun yang diterima komputer. Pada pengujian ini komputer mengirim data lampu ke mikrokontroler sebanyak 5. proses pengiriman. Data ini dapat dilihat pada jendela statistik dibawah ini,

STATISTIK	
Rejected Packets.....	0
Transmitted Packets....	5
Received Packets.....	66
Total Packets.....	71
Rejected Bytes.....	0
Transmitted Bytes.....	45
Received Bytes.....	594
Total Bytes.....	639
Statistics since.....	23 August 2007, 20:43:42

Gambar 17 Statistik Data

Dari gambar dapat dianalisa, yaitu *Rejected Packets* 0 menunjukkan paket yang dikirim tidak terdapat kesalahan. Pada *Transmitted Packet* terdapat nilai 5 yang berarti data yang dikirim komputer sebanyak 5 kali pengiriman. Kemudian Pada *Received Packets* menunjukkan bahwa paket yang diterima oleh komputer dari mikrokontroler sebanyak 66 paket data. *Total Packets* menunjukkan keseluruhan paket baik yang dikirim maupun yang diterima oleh komputer, yaitu sebanyak 71 paket. Pada *Rejected bytes* menunjukkan nilai 0 yang berarti tidak ada data yang salah pada proses pengiriman. *Transmitted bytes* menunjukkan jumlah *byte* data yang dikirim oleh komputer yaitu 45 *byte* data. Karena setiap paket berisi 9 *byte* data, maka total *byte* data yang dikirim yaitu 45. Pada *Received Bytes*, nilai 594 menunjukkan jumlah *byte* data yang diterima komputer. Pada *Total bytes*, nilai 639 menunjukkan jumlah keseluruhan *byte* yang dikirim maupun yang diterima komputer. Dan *Statistic Since* menunjukkan proses komunikasi data direkam mulai tanggal 23 Agustus 2007 pada pukul 20 Menit 43 Detik 42.

V. PENUTUP

V.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan didapatkan beberapa kesimpulan sebagai berikut:

1. Pengendalian Melalui jaringan listrik dilakukan secara simulasi dengan menggunakan kabel.
2. Kesalahan terbesar pengubahan data dari analog ke digital oleh ADC0804 adalah 1.5 bit.
3. Akuisisi data suhu tidak dapat dilakukan karena terjadi kesalahan dalam perancangan blok pengkondisi sinyal. Tegangan keluaran terlalu tinggi yaitu pada saat tidak ada tegangan masukan, tegangan keluaran sebesar 5,6 volt. Seharusnya pada waktu tidak ada tegangan masukan, keluaran menghasilkan tegangan yang sangat kecil sekitar 3 mV.
4. Pengujian perangkat lunak dilakukan dengan memberikan tegangan masukan 4.98 V

langsung pada blok ADC. Didapatkan suhu keluaran 149.4 °C

5. Komunikasi data dilakukan dengan menggunakan *baud rate* 1200 bps antara mikrokontroler dengan komputer. Protokol yang digunakan yaitu Protokol SNAP dengan paket data yang dikirim sebanyak 9 *byte* dan paket yang diterima sebanyak 9 *byte*.
6. Pengecekan kesalahan pengiriman data dilakukan dengan menggunakan metode CRC-16.
7. Selama proses komunikasi data, data dapat dikirim dan diterima dengan baik tanpa terjadi kesalahan.
8. *Driver* lampu dapat berfungsi dengan baik. *Driver* lampu aktif pada saat diberi masukan logika 1.
9. Detektor arus dapat digunakan untuk mendeteksi kondisi lampu. Pada saat lampu terjadi masalah, detektor dapat mendeteksi, kemudian mengirim data pada komputer bahwa terjadi masalah pada lampu.
10. Pengolahan data dilakukan oleh komputer dengan menggunakan bahasa pemrograman Delphi dengan menggunakan komponen serial *comdrv32*.

V.2 Saran

1. Untuk merealisasikan sistem kendali menggunakan media jaringan listrik, dapat menggunakan modem khusus yang digunakan untuk komunikasi melalui jaringan listrik, yang sering disebut *Powerline Modem*.
2. Untuk menguji kehandalan protokol SNAP, dapat dilakukan komunikasi data pada beberapa perangkat komputer/mikrokontroler.
3. Merealisasikan Protokol SNAP pada media komunikasi data yang lain seperti infra merah, gelombang radio, *bluetooth*, fiber optik dan sebagainya.

DAFTAR PUSTAKA

1. Erwin, Robert M. *Pengantar Telekomunikasi*, Elex Media Komputindo, Jakarta, 1986
2. Ibnu Malik, Muhammad, Anistardi, *Bereksperimen dengan Mikrokontroler 8031*, Elex Media Komputindo, Jakarta, 1997
3. Malvino, Paul Albert, *Prinsip – Prinsip Elektronika Jilid I*, Diterjemahkan oleh Sahat Pakpahan, Erlangga, Jakarta, 1996.



4. Millman, Halkias, *Elektronika Terpadu*, Erlangga , Jakarta, 1993
5. Budiharto, Widodo, *Interfacing Komputer dan Mikrokontroler*, Elex Media Komputindo
6. Tocci, Ronald J, *Digital Systems : Principles and Applications*, 5th Edition, Prentice – Hall, New Jersey, 1991
7. Wasito, S, *Vademekum Elektronika Edisi Kedua*, PT Gramedia Pustaka Utama, Jakarta 2001
8. Malik, Ibnu, Moh, *Belajar Mikrokontroler ATMEL AT89S8252*, Gava Media, Yogyakarta 2003.
9. Wahyudin, Didin, *Belajar Mudah Mikrokontroler AT89S52 dengan Bahasa BASIC Menggunakan BASCOM-8051*, Andi Offset, Yogyakarta 2007
10. Prasetya, Retna, *Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*, Andi Offset, Yogyakarta 2004.
- [11], *ADC0804 Datasheet*, <http://www.national.com>
- [12], *AT89S51 Datasheet*, <http://www.atmel.com>
- [13], *LM35 Datasheet*, <http://www.national.com>
- [14], *MAX232 Datasheet*, <http://www.maxim-ic.com>
- [15], *SNAP Protocol*, <http://www.hth.com>
- [16], *4N25 Datasheet*, <http://www.datasheetcatalog.com>
- [17], *MOC3021 Datasheet*, <http://www.alldatasheet.com>

BIOGRAFI

Erik Mubarak. Lahir Di Bandung 26 tahun silam. Saat ini sedang menempuh pendidikan tinggi di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro pada bidang konsentrasi Kontrol.

Mengetahui / Mengesahkan

Pembimbing I

Pembimbing II

Sumardi, ST. MT. **Trias Andromeda, ST. MT.**
NIP. 132 125 670 **NIP. 132 283 185**