UNIT SENTRAL DATA PADA SISTEM kWh METER PRABAYAR YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Teknik

Disusun oleh :

HERMAWAN RAHMAN SHOLEH

NIM. 0210630059

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2007

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

UNIT SENTRAL DATA PADA SISTEM kWh METER PRABAYAR YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :

HERMAWAN RAHMAN SHOLEH

NIM. 0210630059

Telah Diperiksa dan Disetujui

Dosen Pembimbing :

Ir. Ponco Siwindarto, MS.	**Adharul Muttaqin, ST, MT.**
NIP. 131 837 966	**NIP.132 311 886**

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

UNIT SENTRAL DATA PADA SISTEM kWh METER PRABAYAR YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

Disusun Oleh:

HERMAWAN RAHMAN SHOLEH

NIM. 0210630059

Skripsi ini telah diuji dan dinyatakan lulus pada

tanggal 14 Juni 2007

DOSEN PENGUJI

Panca Mudjirahardjo, ST, MT
NIP. 132 288 163

M. Rifan, ST, MT
NIP. 132 283 659

Ir. Primantara H.T.
NIP. 132 090 390

Waru Djuriatno, ST, MT
NIP. 132 158 733

Mengetahui

Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, MKom
NIP. 131 879 033

PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat, hidayah, serta karunia–Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Unit Sentral Data Pada Sistem kWh Meter Prabayar Yang Menggunakan Jala-Jala Listrik Sebagai Media Komunikasi Data" dengan baik dan lancar.

Penyusunan skripsi ini diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik, yang merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa Jurusan Teknik Elektro di lingkungan Fakultas Teknik Universitas Brawijaya Malang.

Dengan kesungguhan dan rasa rendah hati, penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Ponco Siwindarto, MS dan Bapak Adharul Muttaqin ST, MT selaku dosen pembimbing yang telah banyak bimbingan, motivasi dan pengarahan serta saran dalam penyusunan skripsi ini
2. Ir. Heru Nurwarsito, MKom selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
3. Rudy Yuwono, ST, MT. selaku Sekertaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
4. Ir. Ponco Siwindarto, MS selaku KKDK Elektronika Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
5. Seluruh dosen dan karyawan Teknik Elektro Universitas Brawijaya.
6. Kedua orang tua penulis yang telah banyak berkorban dan membesarkan penulis.
7. Adik-adik tercinta (Wike dan Ferry) yang selalu penulis sayangi.
8. Cah Ayu yang selalu memberi semangat untuk penulis.
9. Keluarga Besar Mahasiswa Teknik Elektro Universitas Brawijaya.
10. Rekan Kerja Laboratorium Dasar Elektrik dan Pengukuran Teknik Elektro Universitas Brawijaya.
11. Serta semua pihak yang telah membantu menyelesaikan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan serta masih jauh dari sempurna. Oleh karena itu kritik serta saran sangat penulis harapkan demi kesempurnaan penyusunan skripsi ini.

Akhir kata, penulis mengharapkan semoga skripsi ini dapat bermanfaat bagi rekan – rekan mahasiswa dan bagi seluruh pembaca pada umumnya.

Malang, Juli 2007

Penulis

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGANTAR i
DAFTAR ISI iii
DAFTAR TABEL vi
DAFTAR GAMBAR vii
DAFTAR LAMPIRAN x
RINGKASAN xi
I. PENDAHULUHAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Ruang Lingkup 2
1.5 Sistematika Penulisan 2
II. TINJAUAN PUSTAKA 4
2.1 Listrik Prabayar 4
2.2 *PLC (Power Line Carrier)* 5
2.3 Modem *PLC (Power Line Carrier)* Menggunakan *IC* LM1893 6
2.3.1 Prinsip Kerja 6
2.3.2 Pemilihan Komponen Pendukung 8
2.4 Mikrokontroler ATMEGA8515 14
2.4.1 Struktur dan Operasi Port 15
2.4.2 Sistem *Interupt* 17
2.4.3 Komunikasi Serial 18
2.5 Delphi dan Database 20
2.5.1 Delphi 20
2.5.2 Database 20
2.6 Standard Komunikasi RS232 21
III. METODOLOGI 24
3.1 Lingkup Perancangan dan Pembuatan Alat 24
3.2 Persiapan Perancangan dan Pembuatan Alat 24
3.3 Variabel Perancangan dan Pembuatan Alat 24
3.4 Langkah-langkah perancangan dan pembuatan alat 25
3.4.1 Studi Literatur 25

3.4.2 Penentuan Spesifikasi Alat..... 25
3.4.3 Perancangan Alat..... 25
3.4.4 Pembuatan Alat..... 25
3.4.5 Pengujian dan Analisis Alat..... 27
3.4.6 Pengambilan Kesimpulan dan Saran..... 27
3.4.7 Penulisan Laporan..... 27
IV. PERANCANGAN DAN PEMBUATAN ALAT..... 29
4.1 Gambaran Umum Sistem Kwh Meter Prabayar yang Menggunakan Jala-Jala Listrik Sebagai Media Komunikasi Data 29
4.2 Spesifikasi dan Blok Diagram Unit Sentral Data..... 30
4.3 Prinsip Kerja Unit Sentral Data..... 32
4.4 Perancangan Perangkat Keras..... 33
4.4.1 Rangkaian Modem *Power Line Carrier (PLC)*..... 33
4.4.2 Rangkaian Mikrokontroler ATMEGA8515..... 40
4.4.3 Rangkaian Antarmuka RS232..... 41
4.4.4 Rangkaian *Relay*..... 41
4.5 Perancangan Protokol Komunikasi..... 43
4.6 Perancangan Perangkat Lunak..... 46
4.7 Perancangan Database..... 47
V. PENGUJIAN DAN ANALISIS..... 54
5.1 Pengujian Rangkaian LM1893..... 54
5.1.1 Tujuan..... 54
5.1.2 Peralatan Pengujian..... 54
5.1.3 Prosedur Pengujian..... 54
5.1.4 Data Pengujian..... 56
5.1.5 Analisis Data Pengujian..... 57
5.2 Pengujian Komunikasi Antar Mikrokontroler Melalui Rangkaian LM1893..... 58
5.2.1 Tujuan..... 58
5.2.2 Peralatan Pengujian..... 58
5.2.3 Prosedur Pengujian..... 58
5.2.4 Data Pengujian..... 63
5.2.5 Analisis Data Pengujian..... 64
5.3 Pengujian Rangkaian RS232..... 64

5.3.1 Tujuan……64
5.3.2 Peralatan Pengujian…… 64
5.3.3 Prosedur Pengujian……64
5.3.4 Data Pengujian……65
5.3.5 Analisis Data Pengujian……..65
5.4 Pengujian Komunikasi Antara Mikrokontroler dengan *Personal Computer* Melalui Rangkaian RS232.....66
5.4.1 Tujuan.....66
5.4.2 Peralatan Pengujian.....66
5.4.3 Prosedur Pengujian.....66
5.4.4 Data Pengujian.....67
5.4.5 Analisis Data Pengujian.....68
5.5 Pengujian Keseluruhan Sistem kWh Meter Prabayar..... 68
5.1.1 Pengujian Pengiriman Nominal kWh.....68
5.1.1.1 Tujuan.....68
5.1.1.2 Peralatan Pengujian.....69
5.1.1.3 Prosedur Pengujian.....69
5.1.1.4 Data Pengujian..... 73
5.1.1.5 Analisis Data Pengujian..... 73
5.1.2 Pengujian Pengurangan dan Pemantauan Nominal kWh.....74
5.1.1.1 Tujuan.....74
5.1.1.2 Peralatan Pengujian……...74
5.1.1.3 Prosedur Pengujian……74
5.1.1.4 Data Pengujian……..74
5.1.1.5 Analisis Data Pengujian……75
VI. KESIMPULAN DAN SARAN……...76
6.1 Kesimpulan……...76
6.2 Saran……76
DAFTAR PUSTAKA ……..78
LAMPIRAN

DAFTAR TABEL

Tabel 2.1 Fungsi tambahan dari port B……..16
Tabel 2.2 Fungsi tambahan port D……16
Tabel 2.3 Fungsi Tambahan Port E……17
Tabel 2.4. Alamat vector interupt dari ATMEGA8515..... 18
Tabel 2.5. Rumus menghitung baudrate……. 19
Tabel 4.1 Tipe data atribut-atribut tabel dalam database…….....47
Tabel 5.1 Data pegujian frekuensi keluaran rangkaian LM1893 mode *transmit*.……57
Tabel 5.2 Hasil pegujian rangkaian LM1893 mode *receive*…….. 57
Tabel 5.3 Data yang dikirim oleh mikrokontroler master……..50
Tabel 5.4 Data yang diterima oleh mikrokontroler slave (dalam heksa desimal)……63
Tabel 5.5 Data pengujian dengan masukan logika TTL..... 65
Tabel 5.6 Data pengujian dengan masukan logika TTL..... 65
Tabel 5.7 Hasil pengiriman data pertama..... 73
Tabel 5.8 Hasil pengiriman data kedua.....73
Tabel 5.9 Perhitungan dan pemantauan nominal kWh 15 menit pertama.....75
Tabel 5.10 Perhitungan dan pemantauan nominal kWh 15 menit kedua.....75
Tabel 5.11 Perhitungan dan pemantauan nominal kWh 15 menit ketiga.....75
Tabel 5.12 Perhitungan dan pemantauan nominal kWh 15 menit keempat.....75

DAFTAR GAMBAR

Gambar 2.1. *PLC* dengan kapasitor jenis penyaring..... 6
Gambar 2.2 Konfigurasi pin LM1893..... 6
Gambar 2.3 Blok diagram prinsip kerja LM1893.....7
Gambar 2.4. IC LM1893 beserta komponen pendukungnya.....8
Gambar 2.5. Hubungan antara Nilai C_0 dan F_0.....9
Gambar 2.6. Hubungan antara nilai C_Q dengan F_0..... 10
Gambar 2.7. Hubungan antara nilai C_C dengan Impedansi C_C.....11
Gambar 2.8. Hubungan antara nilai C_C dengan arus jala-jala..... 11
Gambar 2.9. Penentuan nilai C_L berdasarkan F_0.....12
Gambar 2.10. Penentuan nilai C_I berdasarkan data *rate*.....13
Gambar 2.11. Penentuan nilai C_F berdasarkan F_0.....13
Gambar 2.12. Penentuan nilai R_F berdasarkan F_0.....14
Gambar 2.13. Konfigurasi pin ATMEGA8515..... 15
Gambar 2.14. Format data pengiriman..... 19
Gambar 2.15 Konfigurasi pin-pin konektor DB9..... 22
Gambar 2.16 Konfigurasi pin *IC* MAX232 23
Gambar 3.1 Diagram alir pelaksanaan perancangan dan pembuatan alat..... 28
Gambar 4.1 Gambaran umum dari sistem kWh Meter prabayar..... 29
Gambar 4.2 Blok diagram unit sentral data..... 31
Gambar 4.3 *IC* LM1893 beserta komponen pendukungnya..... 33
Gambar 4.4 Rangkaian kopling transformer.....35
Gambar 4.5 Rangkaian Mikrokontroler..... 40
Gambar 4.6 Rangkaian Antar Muka RS232.....41
Gambar 4.7 Rangkaian *Relay*..... 42
Gambar 4.8 Format data dari *PC* ke mikrokontroler.....45
Gambar 4.9 Format data antara unit sentral data dengan unit penghitung pemakaian energi listrik.....45
Gambar 4.10 Format data dari mikrokontroler ke *PC*.....46
Gambar 4.11 Diagram alir perangkat lunak Mikrokontroler ATMEGA8515.....48

Gambar 4.12 Diagram alir perangkat lunak PC untuk meminta data sisa nominal kWh.....50
Gambar 4.13 Diagram alir perangkat lunak PC untuk memberi data nominal kWh.....52
Gambar 4.14 Relasi tabel database.....53
Gambar 5.1 Diagram blok pengujian Rangkaian LM1893 Mode Transmit.....55
Gambar 5.2 Diagram blok pengujian Rangkaian LM1893 Mode Receive.....56
Gambar 5.3 Bentuk sinyal keluaran pemancar LM1893.....56
Gambar 5.4 Diagram blok pengujian komunikasi data antar mikrokontroler melalui rangkaian LM1893.....58
Gambar 5.5 Rangkaian LED.....59
Gambar 5.6 Rangkaian *SW DIP-8*.....59
Gambar 5.7 Rangkaian *Push Button*.....59
Gambar 5.8 Diagram alir program mikrokontroler master.....60
Gambar 5.9 Diagram alir program mikrokontroler slave.....62
Gambar 5.10 Diagram blok pengujian dengan level tegangan masukan logika TTL.....64
Gambar 5.11 Diagram blok pengujian dengan level tegangan masukan logika serial.....64
Gambar 5.12. Diagram blok pengujian komunikasi antara mikrokontroler dengan *personal computer*.....66
Gambar 5.13 Diagram alir pengujian komunikasi antara mikrokontroler dengan *personal computer* melalui rangkaian RS232.....67
Gambar 5.14 Data pengujian komunikasi antara mikrokontroler dengan *personal computer* melalui rangkaian RS232.....68
Gambar 5.15 Diagram blok pengujian sistem keseluruhan.....69
Gambar 5.16 Tampilan utama program *PC* pada unit sentral data.....69
Gambar 5.17 Tampilan untuk memasukkan atau mengubah identitas Pelanggan.....70
Gambar 5.18 Tampilan untuk memasukkan atau mengubah identitas pelanggan setelah data pelanggan dimasukkan.....70
Gambar 5.19 Tampilan untuk mengganti nominal kwh yang tersimpan di EEPROM unit penghitung pemakaian energi listrik.....71

Gambar 5.20 Tampilan pengganti nominal kWh yang telah terisi ID unit penghitung pemakaian energi listrik tujuan dan nominal kwh yang baru..... 71
Gambar 5.21 Tampilan untuk mengirim nominal kWh ke unit penghitung pemakaian energi listrik..... 72
Gambar 5.22. Tampilan *window* daftar harga energi listrik..... 72

DAFTAR LAMPIRAN

LAMPIRAN A	Foto alat
LAMPIRAN B	Kode program delphi dan mikrokontroler
LAMPIRAN C	Datasheet

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

RINGKASAN

Hermawan Rahman Sholeh, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijiya, Mei 2007, *Unit Sentral Data Pada Sistem kWh Meter Prabayar Yang Menggunakan Jala-Jala Listrik Sebagai Media Komunikasi Data,* Dosen Pembimbing : Ir. Ponco Siwindarto, MS. dan Adharul Muttaqin, ST, MT.

Penggunaan kartu isi ulang untuk pengisian nominal kWh pada sistem listrik prabayar saat ini masih memiliki kekurangan. Kekurangan tersebut yaitu sulit untuk dikembangkan kearah pembelian nominal kWh tanpa harus datang ke loket pengisian. Oleh karena itu dibuat sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data. Dengan pengembangan lebih lanjut, sistem ini bisa dikembangkan kearah pembelian nominal kWh melalui *SMS* atau ATM karena terdapat komunikasi secara langsung antara loket pengisian dengan kWh meter di rumah pelanggan.

Sistem kWh meter prabayar terdiri atas unit sentral data dan unit penghitung pemakaian energi listrik. Unit sentral data berfungsi untuk mengirimkan nominal kWh dari loket pengisian ke kWh Meter pelanggan yang membeli nominal kWh serta untuk memantau pemakaian energi listrik setiap pelanggan. Komunikasi antara loket pengisian dengan kWh Meter pelanggan dilakukan melalui jala-jala listrik menggunakan modem *PLC(Power Line Carrier)* dengan *IC* tipe LM1893 sebagai komponen utamanya. Data dari *personal computer* yang terdiri kode perintah, nominal kWh yang dibeli, dan ID tujuan diumpankan ke mikrokontroler untuk ditambah dengan header dan ID sumber. Kemudian mikrokontroler mengubah mode modem *PLC* menjadi mode *transmit* dan mengirim data ke unit penghitung pemakaian energi listrik. Selanjutnya, mikrokontroler mengubah mode modem *PLC* menjadi mode *receive* dan menunggu data jawaban dari unit penghitung pemakaian energi listrik. Data jawaban yang telah diterim mikrokontroler diumpankan ke *personal computer* untuk diolah dan disimpan kedalam *database*.

Analisis data pengujian sistem menunjukkan bahwa nominal kWh yang dikirim sama dengan nominal kWh yang diterima oleh unit penghitung pemakaian energi listrik yang dituju serta sisa nominal kWh yang tercatat di semua kWh meter pelanggan sama dengan sisa nominal kWh yang tercatat di *database* saat dilakukan pemantauan sisa nominal kWh. Hal ini berarti bahwa unit sentral data dapat mengirimkan nominal kWh ke kWh meter pelanggan yang dituju dengan tepat serta dapat memantau sisa nominal kWh yang tercatat di setiap kWh meter pelanggan.

UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

PT. PLN (persero) telah meluncurkan sistem pembayaran tagihan listrik secara prabayar. Untuk mendukung sistem prabayar, PLN telah meluncurkan sebuah produk yang diberi nama "Meter Prabayar". Meter prabayar adalah sebuah produk layanan yang memberi pilihan kepada pelanggan untuk berlangganan energi listrik dengan sistem kartu prabayar melalui kartu isi ulang (Nia, 2005). Prinsip kerjanya hampir sama dengan kartu prabayar telepon seluler. Kartu isi ulang ini dibawa ke loket PLN untuk diisi dengan nilai nominal energi listrik (berupa nominal kWh atau nominal uang) yang dapat dimanfaatkan untuk mendapatkan energi listrik, sehingga pada produk meter prabayar telah dilengkapi dengan mesin pembaca kartu. Nominal pada kartu isi ulang akan berkurang sesuai pemakaian listrik, sama seperti penggunaan kartu telepon prabayar (Anonymous, 2003).

Penggunaan kartu isi ulang masih memiliki kelemahan yaitu mengharuskan pelanggan untuk datang langsung ke loket pengisian nominal energi listrik untuk mengisi ulang kartu sehingga sulit untuk dikembangkan ke arah peningkatan pelayanan berupa pembelian nominal energi listrik tanpa harus datang ke loket PLN, seperti pembelian melalui ATM atau SMS. Selain itu, pihak PLN tidak dapat memantau penggunaan energi listrik oleh pelanggan, kecuali dengan mendatangkan petugas ke setiap rumah pelanggan.

Jika pada perkembangan selanjutnya diinginkan bahwa pelanggan tidak perlu datang ke loket pengisian kartu untuk membeli nominal energi listrik, tetapi cukup melalui ATM atau SMS, maka diperlukan suatu metode lain yang dapat mengirim nominal energi listrik secara langsung ke rumah pelanggan yang membeli nominal energi listrik. Oleh karena itu, melalui tugas akhir ini dirancang dan dibuat *prototype* kWh meter prabayar untuk mendukung sistem pembayaran tagihan listrik secara prabayar tanpa menggunakan kartu isi ulang. Nominal energi listrik (berupa nominal kWh) yang dibeli akan dikirimkan ke rumah pelanggan secara langsung melalui jala-jala listrik. Sistem yang dirancang dan dibuat terdiri dari dua bagian, yaitu unit penghitung pemakaian energi listrik yang berfungsi untuk menghitung pemakaian energi listrik dan unit sentral data yang berfungsi sebagai pengontrol komunikasi untuk pengiriman dan pemantauan nominal kWh.

Alat ini dapat dikembangkan untuk pembelian nominal kWh melalui ATM karena komputer pada loket pengisian dapat dihubungkan dengan jaringan ATM dengan mudah. Selain itu, perusahaan penyedia jasa listrik dapat terus memantau pemakaian energi listrik dan sisa nominal kWh yang dapat dipakai oleh pelanggan tanpa harus mendatangkan petugas ke setiap rumah pelanggan karena antara loket pengisian dan rumah-rumah pelanggan terdapat hubungan komunikasi secara langsung.

1.2 Rumusan Masalah

Dalam skripsi ini hanya dibahas unit sentral data, karena unit penghitung pemakaian energi listrik telah dibahas pada judul skripsi yang lain. Pada unit sentral data dapat dirumuskan beberapa permasalahan sebagai berikut :

1. Bagaimana merancang dan membuat modem *PLC* (*Power Line Carrier*) menggunakan *IC* tipe LM1893.
2. Bagaimana merancang protokol komunikasi antara unit sentral data dan unit penghitung pemakaian energi listrik pada sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data.
3. Bagaimana merancang dan membuat software mikrokontroler ATMEGA8515 untuk mengontrol jalannya komunikasi data.
4. Bagaimana merancang dan membuat antarmuka antara tranceiver data dan *PC (Personal Computer)*.
5. Bagaimana membuat program komputer dan *database* untuk mengolah data pemakaian listrik setiap pelanggan dan mengirim nominal kWh kepada pelanggan yang membeli nominal kWh.

1.3. Tujuan

Tugas akhir ini bertujuan untuk merealisasikan unit sentral data pada sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data. Dengan pengembangan lebih lanjut, alat ini diharapkan dapat menjadi salah satu solusi bagi perusahaan penyedia jasa listrik seperti PLN untuk mengisi nominal energi listrik pelanggan agar dapat meningkatkan pelayanan kepada pelanggan.

1.4. Ruang Lingkup

Karena begitu luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Permasalahan dalam tugas akhir ini dibatasi pada beberapa hal sebagai berikut :

1. Sistem dirancang untuk dapat bekerja pada jala-jala listrik 1 fasa dengan tegangan sebesar 220 V dan frekuensi sebesar 50 Hz.
2. Jalur komunikasi tidak melalui transformer.
3. Belum memperhitungkan pengaruh panjang, jenis kabel, dan sambungan-sambungan pada jala-jala listrik yang digunakan sebagai media komunikasi.
4. Belum memperhitungkan pengaruh harmonisa dan *noise* pada jala-jala lstrik.
5. Belum memperhitungkan keamanan data.
6. Parameter keberhasilan alat adalah dapat melakukan komunikasi dengan unit penghitung pemakaian energi listrik dalam area terbatas yang telah ditentukan sebagai simulasi, dapat mengirimkan nominal kWh ke pelanggan yang dituju, serta dapat menyimpan sisa nominal kWh setiap pelanggan kedalam *database*.

1.5. Sistematika Penulisan

Sistematika penulisan tugas akhir ini adalah sebagai berikut:

Bab I : Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, ruang lingkup, dan sistematika penulisan.

Bab II : Tinjauan Pustaka

Membahas teori-teori dasar penunjang perancangan dan pembuatan alat.

Bab III: Metode Penelitian

Membahas metodologi yang digunakan dalam penulisan skripsi.

Bab IV: Perancangan dan Pembuatan

Memuat prinsip kerja, perancangan dan pembuatan alat.

Bab V : Hasil dan Pembahasan

Memuat hasil pengujian dan analisis sistem yang dibuat.

Bab VI: Kesimpulan dan Saran

Berisi kesimpulan dan saran dari skripsi ini.

BAB II
TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan unit sentral data pada sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data, maka perlu penjelasan dan uraian teori penunjang yang digunakan dalam tugas akhir ini.

Teori-teori penunjang yang dijelaskan dalam bab ini meliputi:

1. Listrik prabayar.
2. *PLC (Power Line Carrier).*
3. Mikrokontroler ATMEGA8515.
4. Delphi dan *database*.
5. Standar komunikasi RS232

2.1 Listrik Prabayar

Saat ini, sistem pembayaran rekening listrik di Indonesia sebagian besar masih menggunakan metode pasca bayar. Pada sistem pembayaran listrik pascabayar terdapat beberapa kelemahan. Yang pertama yaitu adanya kemungkinan terjadinya kesalahan pembacaan kWh meter oleh petugas sehinggga tagihan yang dibayarkan tidak sesuai dengan daya listrik yang dipakai (Hilmy Wanamulya, 2005). Yang kedua yaitu terjadinya pemutusan sementara ketika terlambat dalam membayar tagihan dan pelanggan harus membayar denda.

Untuk itu, saat ini mulai dikembangkan metode pembelian energi listrik secara prabayar. Pelanggan dapat membeli nominal energi listrik seperti membeli pulsa telepon seluler. Nominal energi listrik dikemas dalam bentuk nominal uang atau nominal kWh yang kemudian diisikan ke meteran pelanggan sehungga secara otomatis, energi listrik bisa digunakan oleh pelanggan. Pengisian nominal energi listrik ke pelanggan bisa melalui media yang bermacam-macam. Seperti menggunakan kartu isi ulang, dikirim melalui frekuensi radio, atau dikirim melalui jala-jala listrik. Dengan sistem ini, diharapkan pelanggan bisa menggunakan energi listrik secara cermat mengingat jumlah nominal pulsa yang ada pada meteran sesuai dengan nilai yang dibayar.

Nominal energi listrik yang dibeli oleh pelanggan akan dikurangi oleh meteran yang ada dirumah pelanggan sesuai dengan beban yang digunakan. Untuk itu perlu dilakukan suatu perhitungan energi listrik yang digunakan oleh pelanggan.

2.2 *PLC (Power Line Carrier)*

PLC (Power Line Carrier) adalah salah satu sistem telekomunikasi yang menggunakan jala-jala listrik sebagai media komunikasi. Sinyal yang ditransmisikan dapat berupa sinyal analog maupun sinyal digital. Sebelum dihubungkan ke jaringan jala-jala listrik, sinyal analog maupun sinyal digital perlu dimodulasi agar bisa dipancarkan. Modulasi untuk sinyal analog dapat berupa modulasi amplitudo, modulasi frekuensi, maupun modulasi fasa. Sedangkan modulasi untuk sinyal digital dapat berupa *FSK (Frequency Shift Keying)* maupun *ASK (Amplitude Shift Keying)*. Peralatan komunikasi yang bisa digunakan untuk komunikasi sinyal analog melalui jala-jala listrik adalah *transceiver* FM atau *transceiver* AM. Sedangkan peralatan komunikasi untuk sinyal digital dapat menggunakan Modem *PLC*.

Sinyal yang telah termodulasi dapat dihubungkan dengan jaringan jala-jala listrik melalui 2 metode, yaitu :

1. PLC terhubung induktif, yaitu menempatkan penghantar yang berfungsi sebagai antena dengan arah paralel terhadap penghantar jaringan tenaga listrik untuk jarak tertentu.
2. PLC terhubung kapasitif, yaitu menghubungkan peralatan komunikasi dengan jaringan tenaga listrik lewat kapasitor.

Metode yang sering digunakan adalah *PLC* terhubung kapasitif. Fungsi kapasitor adalah sebagai penahan arus ac dari jala-jala listrik yang memiliki frekuensi berbeda dengan frekuensi pembawa dari peralatan komunikasi. Impedansi kapasitif ditentukan oleh Persamaan 2-1:

$$Xc = \frac{1}{2\pi f.C} \qquad (2\text{-}1)$$

Dengan: Xc = Impedansi kapasitif (Ω)

f = frekuensi (Hz)

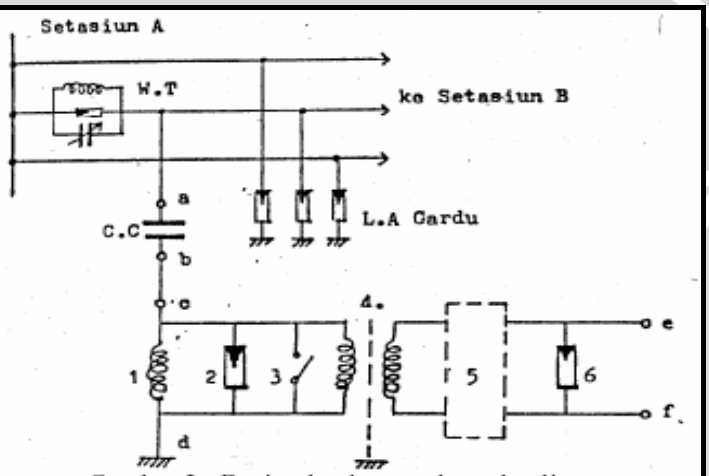
C = Kapasitansi (F)

Nilai X_c dapat ditentukan dengan cara mengatur nilai C sehingga didapatkan nilai X_c yang sesuai dengan kebutuhan. Untuk frekuensi arus jala-jala sebesar 50 Hz, X_c bernilai sangat besar sehingga arus yang mengalir ke peralatan komunikasi sangat kecil dan tidak dapat merusak rangkaian. Untuk frekuensi pembawa dari peralatan komunikasi bagian pemancar yang bernilai sangat tinggi, maka X_c bernilai sangat rendah sehingga arus gelombang pembawa dari bagian pemancar dapat dialirkan ke rangkaian peralatan komunikasi bagian penerima untuk diterjemahkan.

Ada dua jenis hubungan dengan kapasitor, yaitu:

1. Penghubung kapasitor jenis penala (*tunning type*), yaitu kapasitor merupakan bagian dari alat penala yang dihubungkan seri dengan jaringan tenaga listrik.
2. Penghubung kapasitor jenis penyaring (*filter type*), yaitu kapasitor penggandeng merupakan jaringan berkutub 4 dan menggandengkan peralatan komunikasi dengan jaringan tenaga listrik. *IC* tipe LM1893 merupakan komponen untuk modem *PLC* yang menggunakan penghubung kapasitor jenis ini.

Gambar 2.1 menunjukan rangkaian penghubung antara peralatan komunikasi dengan jala-jala listrik.



Gambar 2. Bagian-bagian peralatan kopling.

Keterangan gambar :
1. drain coil
2. arrester pertama.
3. kontak pentanahan
5. peralatan penala
4. transformator penyeimbang dan pengisolasi.
6. arrester kedua.

a. terminal tegangan tinggi kapasitor kopling
b. terminal tegangan rendah CC
c. terminal utama peralatan kopling
d. terminal pentanahan
e,f. terminal peralatan kopling ke terminal PLC (SSB).

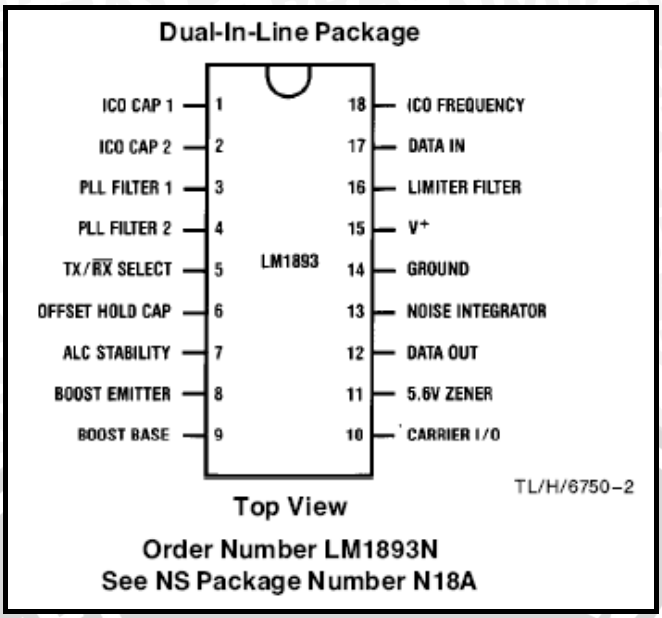
Gambar 2.1. Rangkaian penghubung antara peralatan komunikasi dengan jala-jala listrik

Sumber: Agung 2005

2.3 Modem *PLC (Power Line Carrier)* Menggunakan *IC* LM1893

2.3.1 Prinsip Kerja

LM1893 merupakan *IC* buatan National Semiconductor yang didesain secara khusus sebagai modulator dan demodulator *power line carrier*. Gambar 2.2 menunjukkan konfigurasi pin IC LM1893.

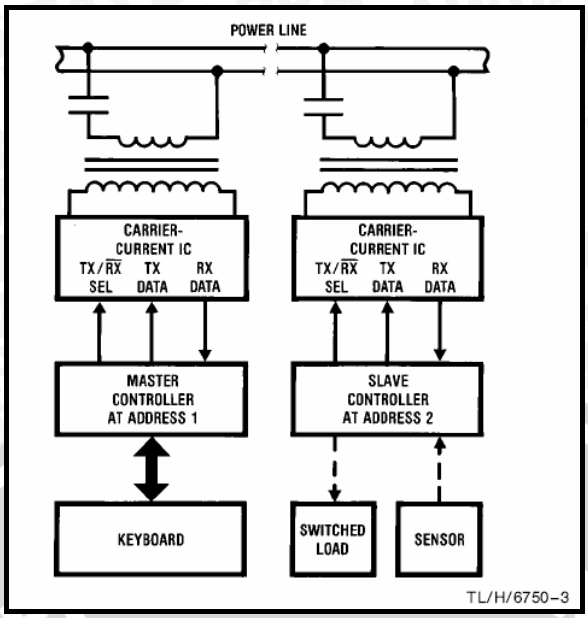


Gambar 2.2 Konfigurasi pin LM1893
Sumber : Datasheet LM1893 , 2003

Gambar 2.3 menunjukkan prinsip kerja dari LM1893. *Master controller* dan *slave controller* berfungsi sebagai pengirim dan penerima data serta untuk mengubah mode kerja LM1893 menjadi mode *transmit* atau mode *receive*. *Controler* dapat berupa mikrokontroler atau *personal computer*.

Ketika mengirim data, pertama-tama *controler* mengubah mode kerja LM1893 menjadi mode *transmit* dengan memberi logika 1 pada pin 5. Kemudian data dikirim secara serial melalui pin 17. Data serial tersebut akan membangkitkan arus untuk mengontorol *oscillator*. Kemudian keluaran *oscillator* dikuatkan dan dikeluarkan melalui pin 10. Keluaran di pin 10 ini merupakan data digital yang telah termodulasi. Keluarannya berupa arus dengan frekuensi tertentu yang dapat membangkitkan tegangan berosilasi setelah dihubungkan dengan rangkaian penghubung kapasitor jenis penyaring.

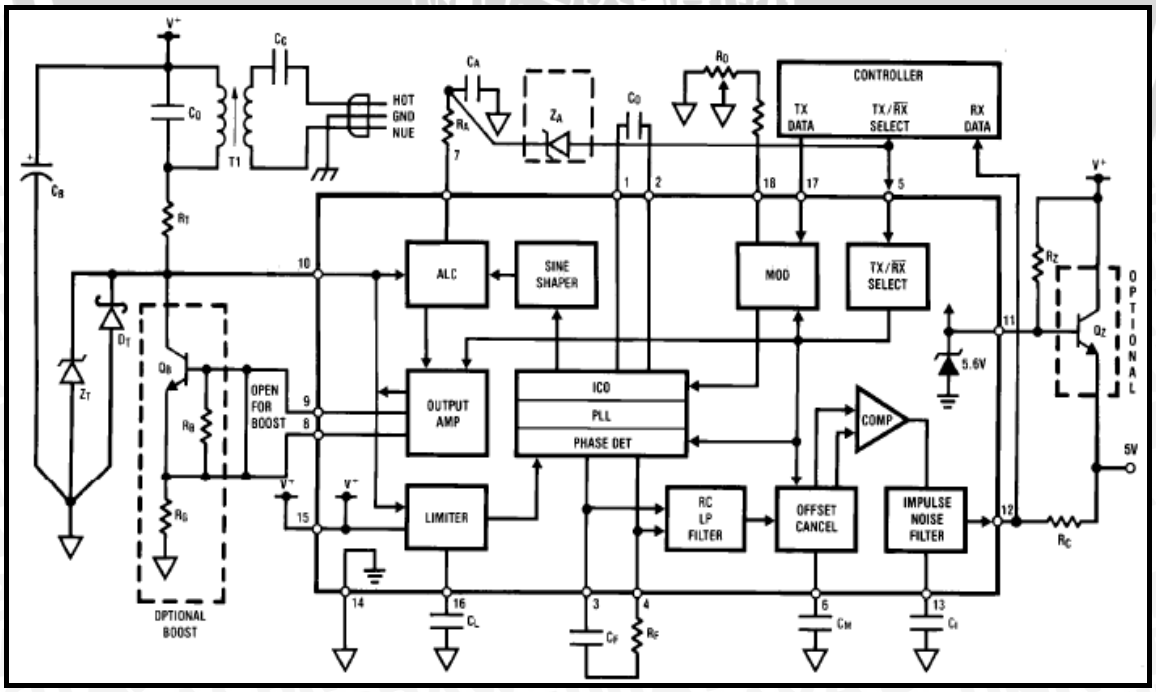
Ketika akan menerima data, *controller* mengubah mode kerja LM1893 menjadi mode *receiver* dengan memberi logika pada pin 5. Sinyal dari jala-jala listrik dikopel untuk melewatkan frekuensi sinyal dan menahan frekuensi jala-jala, serta *PLL (Phase Locked Loop)* untuk memisahkan frekuensi sinyal dengan frekuensi pembawa. Sinyal yang telah terpisah dengan gelombang pembawa masuk ke bagian *FSK* demodulator yang terdapat dalam chip *IC* LM1893 untuk diubah menjadi data digital.



Gambar 2.3 Blok diagram prinsip kerja LM1893
Sumber : Datasheet LM1893 , 2003

2.3.2 Pemilihan Komponen Pendukung

Agar *IC* LM1893 dapat bekerja, maka *IC* LM1893 harus dihubungkan dengan beberapa rangkaian pendukung sesuai dengan yang ditunjukkan dalam Gambar 2.4. Karena *IC* LM1893 dapat bekerja sebagai pemancar sekaligus sebagai penerima, maka komponen pendukungnya terdiri dari dua bagian, yaitu komponen pendukung bagian pemancar dan komponen pendukung bagian penerima.



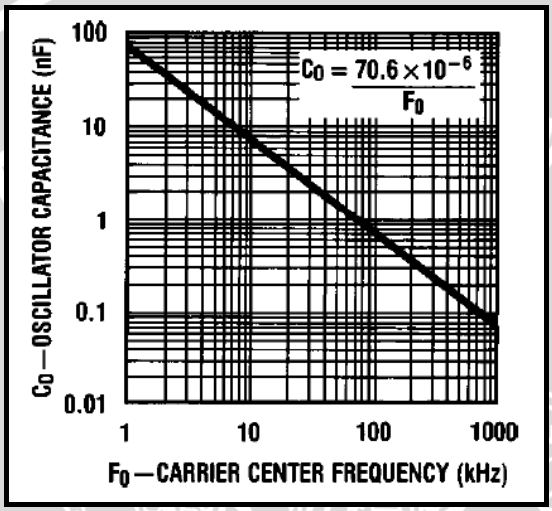
Gambar 2.4. IC LM1893 beserta komponen pendukungnya
Sumber: Datasheet LM1893 , 2003

2.3.2.1 Bagian Pemancar

Komponen pendukung bagian pemancar terdiri dari C_0, R_0, C_A, R_A, T_1, C_Q, Z_T, dan R_T. Penjelasan fungsi dan nilai masing-masing komponen pendukung bagian pemancar adalah sebagai berikut.

- **C_0**

C_0 berfungsi untuk menentukan frekuensi pembawa yang digunakan. Penentuan nilai C_0 dilakukan berdasarkan *datasheet IC* LM1893 yaitu seperti yang ditunjukkan dalam Gambar 2.5.



Gambar 2.5. Hubungan antara Nilai C_0 dan F_0
Sumber: Datasheet LM1893 , 2003

Penentuan nilai C_0 menggunakan persamaan sebagai berikut :

$$C_0 = \frac{70{,}6 \times 10^{-6}}{F_0} \qquad (2\text{-}2)$$

Dengan F_0 adalah frekuensi pembawa yang digunakan.

- **R_0**

Resistor R_0 digunakan untuk membangkitkan arus V_{BE} yang akan dikalikan 2 untuk menghasilkan arus sebesar 200 µA pada ICO yang menentukan nilai F_0. Semakin kecil nilai R_0 akan meningkatkan F_0. Nilai yang dianjurkan untuk R_0 adalah antara 5,6 kΩ sampai 7,6 kΩ. (Sumber: *Datasheet LM1893 , 2003*). Oleh karena itu digunakan nilai potensio 2 kΩ dirangkai seri dengan resistor 5,6 kΩ untuk R_0.

- **C_A dan R_A**

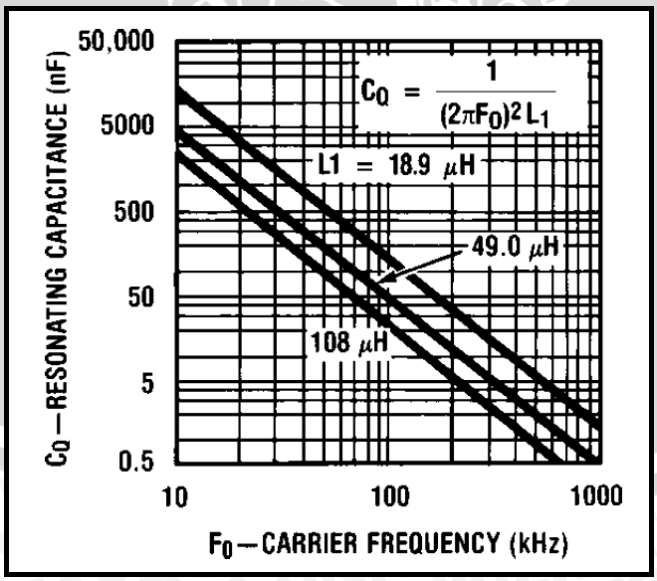
Besarnya nilai C_A dan R_A akan mempengaruhi ALC IC LM 1893. Semakin kecil nilai C_A dan R_A maka ALC akan tidak stabil, sedangkan semakin besar nilainya respon ACL akan lambat tetapi semakin stabil. Nilai yang dianjurkan untuk C_A dan R_A adalah C_A = 0,1 µF dan R_A = 10 µΩ.

- **T_1**

T_1 digunakan bersama C_Q untuk membentuk frekuensi resonansi F_Q. T_1 juga digunakan sebagai *transformer coupling* pada penumpangan sinyal informasi ke jala-jala listrik. Nilai induktansi kumparan primer L_1 bersama-sama dengan C_Q akan menentukan frekuensi resonansi. T_1 juga harus mempunyai perbandingan jumlah lilitan kumparan primer dan sekunder sebesar mungkin agar dapat menumpangkan sinyal ke jala-jala listrik dengan baik.

- **C_Q**

C_Q digunakan bersama T_1 sebagai frekuensi resonansi yang sama dengan F_0 supaya sinyal informasi dapat ditumpangkan ke jala-jala listrik. Frekuensi resonansi yang dihasilkan oleh C_Q dan induktansi T_1 harus sama dengan F_0. Penentuan nilai C_Q didasarkan pada Gambar 2.6.



Gambar 2.6. Hubungan antara nilai C_Q dengan F_0

Sumber: Datasheet LM1893 , 2003

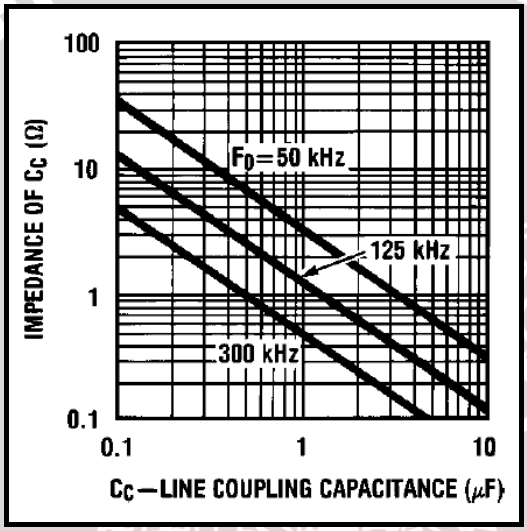
Persamaan untuk menetukan nilai C_Q adalah sebagai berikut :

$$C_Q = \frac{1}{(2\pi F_0)^2 L_1} \qquad (2\text{-}3)$$

Dengan L_1 adalah induktansi primer T_1.

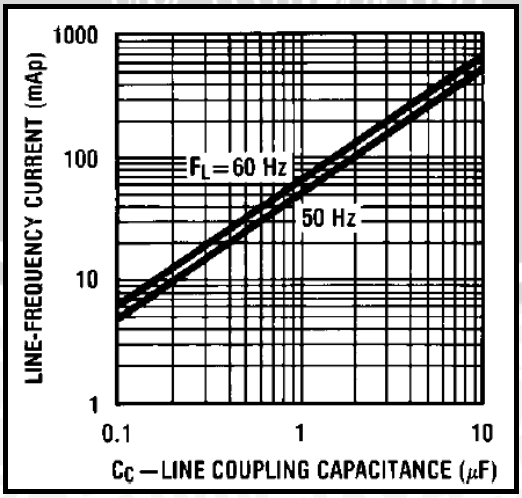
- **C_C**

Kapasitor C_C digunakan agar tegangan jala-jala listrik tidak masuk ke lilitan T_1. C_C juga digunakan sebagai L_C *highpass filter*. C_C harus mempunyai impedansi yang cukup besar untuk memfilter tegangan jala-jala listrik 50 Hz, dan impedansi untuk F_0 harus sekecil mungkin dengan acuan Gambar 2.7. Arus jala-jala yang melewati C_C harus lebih kecil dari arus maksimum 10 *Amp-turn* (1 Amp melalui 10 lilitan), dengan acuan Gambar 2.8.



Gambar 2.7. Hubungan antara nilai C_C dengan Impedansi C_C

Sumber: *Datasheet LM1893 , 2003*



Gambar 2.8. Hubungan antara nilai C_C dengan arus jala-jala

Sumber: *Datasheet LM1893 , 2003*

- **Z_T**

Z_T merupakan dioda zener yang digunakan untuk mencegah adanya sinyal transien yang melalui transformator T_1 menuju ke pin *carrier I/O*. sinyal transien tersebut dapat juga berasal dari pengosongan tegangan di kapasitor C_C pada saat menghubungkan atau melepas rangkaian dari jala-jala listrik. Untuk itu Z_T menggunakan dioda *avalance* yang biasanya sering digunakan untuk mencegah sinyal transien. Z_T harus mempunyai tegangan dadal yang sedikit lebih besar dari 44 V, karena pin *carrier I/O* mempunyai dioda zener internal dengan BV 44 V.

- **R_T**

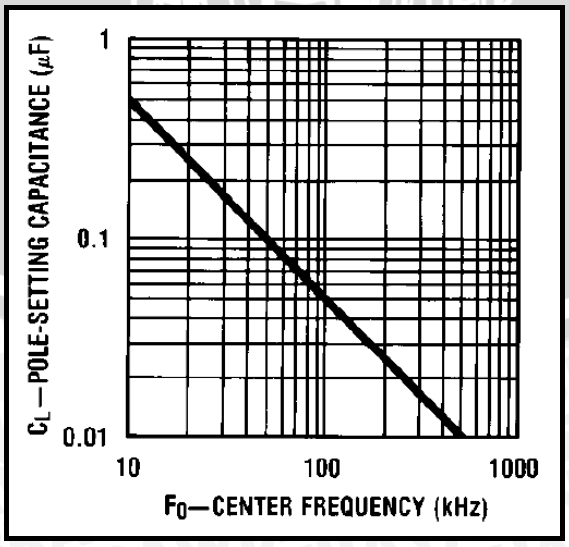
R_T berfungsi sebagai pembagi tegangan dengan Z_T, yang akan menyerap tegangan transien supaya tegangan yang masuk pada pin *carrier I/O* tetap dibawah 44 Volt. Nilai yang dianjurkan adalah sebesar 47 Ω.

2.3.2.2 Bagian Penerima

Komponen pendukung bagian penerima terdiri dari C_L, C_I, C_F, R_F, dan Z_A. Penjelasan fungsi dan nilai masing-masing komponen pendukung bagian penerima adalah sebagai berikut:

- **C_L**

LM1893 mempunyai *band pass filter* untuk meningkatkan sensivitas penerimaan sinyal. Frekuensi *cutt off* atas ditentukan sebesar 300 kHz. Sedangkan frekuensi *cut off* bawah ditentukan oleh nilai C_L.

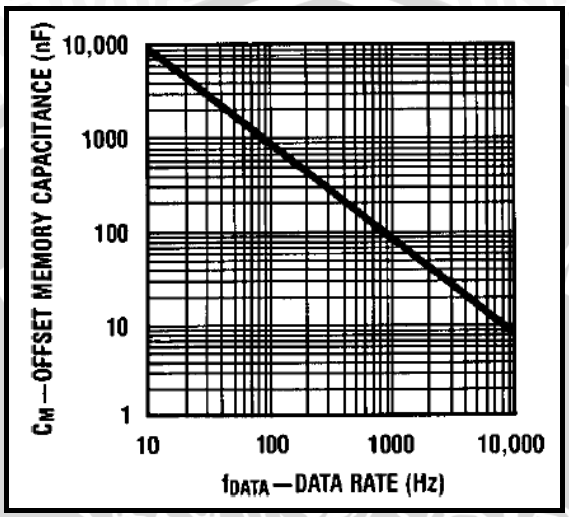


Gambar 2.9. Penentuan nilai C_L berdasarkan F_0

Sumber: Datasheet LM1893 , 2003

- **C_I**

C_I digunakan pada bagian filter *noise* yang berfungsi untuk mencegah lewatnya sinyal pulsa yang mempunyai waktu lebih pendek dari waktu pengisian integrator chip LM 1893. Waktu pengisian integrator ditentukan sebesar ½ waktu data bit. Jadi nilai C_I ditentukan besarnya data *rate* yang digunakan. Gambar 2.10 menunjukkan hubungan data *rate* dengan nilai C_I.

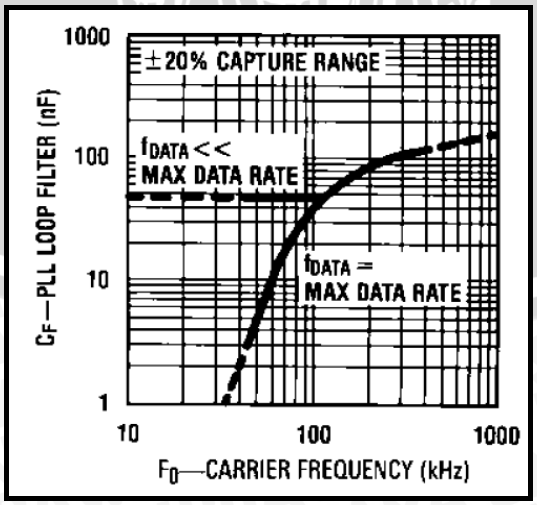


Gambar 2.10. Penentuan nilai C_I berdasarkan data *rate*

Sumber: Datasheet LM1893 , 2003

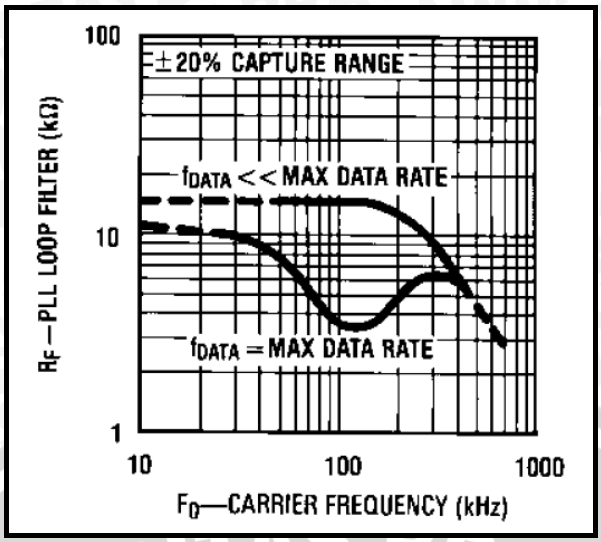
- **C_F dan R_F**

C_F dan R_F adalah komponen yang digunakan pada *PLL IC* LM1893 sebagai filter untuk menghilangkan *noise*. Besarnya nilai C_F dan R_F ditentukan oleh frekuensi pembawa (F_0) yang digunakan. Gambar 2.11 dan 2.12 merupakan Gambar hubungan nilai C_F dan R_F dengan F_0.



Gambar 2.11. Penentuan nilai C_F berdasarkan F_0

Sumber: Datasheet LM1893 , 2003



Gambar 2.12. Penentuan nilai R_F berdasarkan F_0

Sumber: Datasheet LM1893 , 2003

- **Z_A**

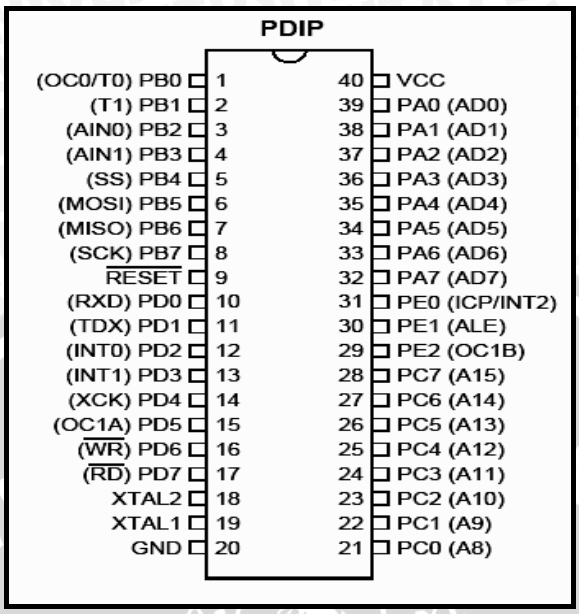
Z_A digunakan jika sinyal yang masuk pada *input* R_X(pin 10) melebihi tegangan catu bagian penerima, yang disebabkan tegangan catu pemancar lebih tinggi dari penerima, jarak T_X dan R_X yang terlalu dekat, atau perbandingan lilitan *step-up* T_1 pada R_X lebih tinggi dari perbandingan lilitan *step-down* T_1 pada T_X.

2.4 Mikrokontroler ATMEGA8515

Mikrokontroler ATMEGA8515 yang diproduksi oleh ATMEL *Company* merupakan salah satu anggota keluarga dari jenis AVR. Mikrokontroler ATMEGA8515 mempunyai karakteristik utama sebagai berikut:

1. CPU dengan lebar data 8 bit
2. Empat buah I/O 8 bit dan sebuah I/O 3 bit.
3. Ruang memori program sebesar 8 Kbyte
4. Ruang memori data sebesar 512 byte
5. EEPROM sebesar 512 byte untuk memori program pada chip
6. Satu buah timer/counter 8 bit dan 16 bit
7. Tiga buah pin PWM
8. UART (Universal Asyncronous Receiver Transmitter) full duplex
9. 3 jalur interupsi
10. Oscilator internal terdapat dalam chip

Konfigurasi pin mikrokontroler ATMEGA8515 dapat dilihat dalam Gambar 2.13.



Gambar 2.13. Konfigurasi pin ATMEGA8515

Sumber : Datasheet ATMEGA8515, 2003

2.4.1 Struktur dan Operasi Port

Agar dapat berhubungan dengan *peripheral* lain, Mikrokontroler ATMEGA8515 memiliki 5 buah port. Empat buah port memiliki 8 buah jalur I/O dan sebuah port memiliki 3 buah jalur I/O yang bersifat bidirectional. Beberapa karakteristik port mikrokontroler ATMEGA8515 dijelaskan secara singkat berikut ini :

1. Unit I/O dapat dialamati perjalur atau per port
2. Setiap jalur I/O memiliki buffer, penahan (latch), kemudi input dan kemudi output.
3. Setiap jalur I/O terdapat regieter pengatur apakah dijadikan input atau dijadikan output.
4. Port A merupakan port I/O 8 bit dua arah dengan pull up internal. Fungsi tambahan dari port A adalah sebagai jalur alamat dan data pada saat mengakses memori ekternal yaitu AD0-AD7.
5. Port B merupakan port I/O dua arah dengan pull up internal. Fungsi tambahan dari port B seperti yang terlihat dalam Tabel 2.1.

Tabel 2.1. Fungsi tambahan dari port B

Sumber : Datasheet ATMEGA8515, 2003

Port pin	Fungsi tambahan
PB7	SCK (Bus serial clok SPI)
PB6	MISO (Bus Master Input/Slave Output SPI)
PB5	MOSI (Bus Master Output/Slave Input SPI)
PB4	SS (Pemilih input slave SPI)
PB3	AN1 (Input inferting analog komparator)
PB2	AN2 (Input noninferting analog komparator)
PB1	T1 (Input counter pada timer/counter 1)
PB0	T0 (Input counter pada timer/counter 0)
	OCO (output compare match pada timer/counter 0)

6. Port C merupakan port I/O dua arah dengan pul up internal. Fungsi tambahan dari port C adalah sebagai jalur alamat atas pada saat akses memori eksternal yaitu A8-A15.
7. Port D merupakan port I/O dua arah dengan pull up internal. Fungsi tambahan dari port D seperti yang terlihat dalam Tabel 2.2.

Tabel 2.2. Fungsi tambahan port D

Sumber : Datasheet ATMEGA8515, 2003

Port pin	Fungsi tambahan
PD7	RD (Sinyal stobe pada saat membaca memori ekternal)
PD6	WR (Sinyal strobe pada saat menulis memori eksternal)
PD5	OC1A (Output compare A pada timer/counter 1)
PD4	XCK (Input/output clok eksternal dari USART)
PD3	INT1 (Interupt eksternal 1)
PD2	INT0 (Interupt eksternal 0)
PD1	TXD (output pin pada USART)
PD0	RDX (Input pin pada USART)

8 Port E merupakan port I/O dua arah dengan pull up internal. Fungsi tambahan dari port E seperti yang terlihat dalam Tabel 2.3.

Tabel 2.3 Fungsi Tambahan Port E

Sumber : Datasheet ATMEGA8515, 2003

Port pin	Fungsi Tambahan
PE0	OC1B (Output compare 1 B pada timer/counter 1)
PE1	ALE (Address Latch Enable pada saat akses memori ekternal)
PE2	ICP (Input capture pada timer/counter 1) INT 2 (Input interupt eksternal 2)

2.4.2 Sistem *Interupt*

Pada saat-saat tertentu, mikrokontroler harus melakukan kontrol ketika terjadi kejadian-kejadian tak terduga. Untuk mengatasi hal ini, mikrokontroler memiliki sistem *interrupt*. Jumlah kejadian-kejadian tak terduga yang bisa ditangani oleh mikrokontroler sebanyak 17 kejadian, masing-masing kejadian ditempatkan pada *interupt vector* dengan alamat tertentu. Mikrokontroller ATMEGA8515 memiliki 17 alamat *interupt vector* dimana nomor urut dari *interupt vector* tadi menyatakan prioritas dari interupt tersebut. Alamat *interupt vector* dari mikrokontroller ATMEGA8515 dapat dilihat dalam Tabel 2.4.

Tabel 2.4. Alamat vector interupt dari ATMEGA8515

Sumber : Datasheet ATMEGA8515, 2003

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0 OVF	Timer/Counter0 Overflow
9	$008	SPI, STC	Serial Transfer Complete
10	$009	USART, RXC	USART, Rx Complete
11	$00A	USART, UDRE	USART Data Register Empty
12	$00B	USART, TXC	USART, Tx Complete
13	$00C	ANA_COMP	Analog Comparator
14	$00D	INT2	External Interrupt Request 2
15	$00E	TIMER0 COMP	Timer/Counter0 Compare Match
16	$00F	EE_RDY	EEPROM Ready
17	$010	SPM_RDY	Store Program memory Ready

2.4.3 Komunikasi Serial

Mikrokontroller ATMEGA8515 dilengkapi dengan fasilitas komunikasi serial USART dengan fitur sebagai berikut:

- Komunikasi full-duplex dengan register serial untuk penerima dan pengirim data.
- Dapat dioperasikan pada mode komunikasi sinkron dan asinkron
- Pada operasi sinkron, *clock* berasal dari master atau slave
- Mempunyai resolusi tinggi untuk generator baudrate
- Layanan pengiriman data terdiri dari 5,6,7,8, dan 9 bit dan 1 atau 2 bit stop.
- Paritas genap atau ganjil dan didukung dengan pengecekan paritas oleh hardware.
- Pendeteksi pengiriman kelebihan data.
- Pendeteksi kesalahan pada format data yang dikirim
- Memiliki filter noise yang terdiri dari pendeteksi kesalahan bit start dan low pass filter.

- Memiliki 3 layanan interupt yaitu TX complite, TX data empity, dan RX complite.
- Mode komunikasi multi processor
- Mode komunikasi asinkron dengan dua kecepatan.

Untuk menghitung baudrate dari komunikasi serial digunakan rumus seperti yang terlihat dalam Tabel 2.5.

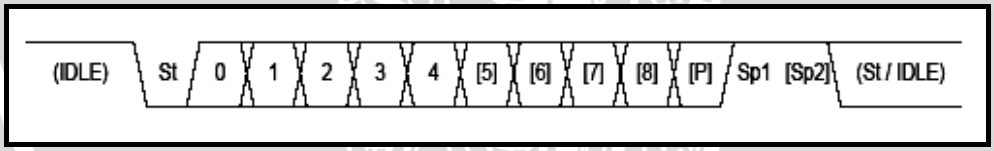
Tabel 2.5. Rumus menghitung baudrate
Sumber : Datasheet ATMEGA8515, 2003

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Dimana :

f_{OSC} = Frekuensi clok dari sistem osilator
UBRR = Register baudrate yang terdiri dari UBRRL dan UBRRH
BAUD = Baudrate dalam bit per secon (bps)

Pada pengiriman data secara serial menggunakan ATMEGA8515 memakai format seperti dalam Gambar 2.14.



Gambar 2.14. Format data pengiriman
Sumber : ATMEGA8515 Data sheet , 2003

Dimana :

St = Bit start selalu berlogika rendah
(n) = Banyaknya data yang dikirim (0-8)
P = Bit paritas (ganjil atau genap)
Sp = Bit stop selalu berlogika tinggi (bit stop bisa berjumlah 1 atau 2)
IDLE = Tidak ada data yang ditransfer pada RX dan TX, IDLE selalu berlogika tinggi

2.5 Delphi dan Database

Untuk melakukan komunikasi antara mikrokontroler dengan *PC*, mengolah data, dan menyimpan data pada komputer, diperlukan suatu program aplikasi dan *database*. Program aplikasi yang bisa digunakan adalah Borland Delphi 7.

2.5.1 Delphi

Borland Delphi atau biasa disebut Delphi merupakan perangkat lunak pengembangan aplikasi yang sangat populer di lingkungan Windows. Delphi telah dilengkapi dengan komponen-komponen untuk mengakses *database*.

Beberapa istilah dasar pada delphi antara lain *application*, *form*, dan *component*. *Application* adalah sederetan kode yang digunakan untuk mengatur komputer agar melakukan sesuatu sesuai dengan keinginan pembuat program. Secara umum, sebuah *application* melibatkan satu atau lebih *form*. *Form* adalah tampilan berbentuk jendela untuk interaksi antara pengguna program dengan komputer. Sebuah *form* melibatkan satu atau lebih *component*. Beberapa contoh *component* antara lain tombol tekan, tombol radio, edit, memo, cek box, dan lain-lain.

2.5.2 *Database*

Secara sederhana, *database* dapat diungkapkan sebagai pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data, seperti menambah dan menghapus data.

Manajemen modern mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, uang, mesin, dan material. Informasi adalah suatu bentuk penyajian data yang melalui mekanisme pemrosesan, yang berguna bagi pihak tertentu, misalnya manajer. Bagi pihak manajemen, informasi merupakan bahan untuk pengambilan keputusan.

Dengan adanya komputer, data dapat disimpan dalam media pengingat yang disebut *hard disk*. Dengan menggunakan media ini, kehadiran kertas yang digunakan untuk menyimpan data dapat dikurangi. Selain itu data menjadi lebih cepat untuk diakses, terutama kalau dikemas dalam bentuk *database*. Perangkat lunak yang digunakan untuk operasi *database* disebut DBMS.

DBMS singkatan dari *Database Management System*. DBMS merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk

memudahkan pengelolaan *database*. Salah satu DBMS yang populer dewasa ini berupa RDBMS (*Relational Database Management System*), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling terhubungkan. Salah satu contoh produk RDBMS yang sangat populer adalah MySQL. MySQL dapat diakses melalui web browser atau melalui program seperti Borland Delphi.

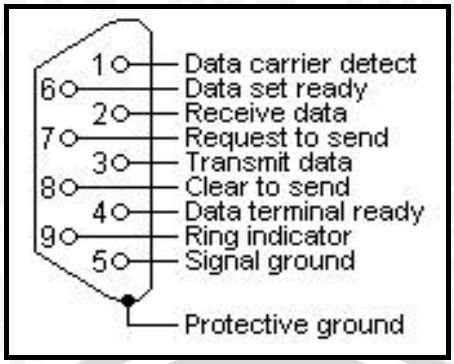
2.6 Standard Komunikasi RS232

Standard komunikasi RS232 diperlukan agar terjadi komunikasi antara mikrokontroler dengan *PC (Personal Computer)*. RS232 merupakan salah satu jenis antar muka (*interface*) dalam proses transfer data antar komputer dalam bentuk serial. RS232 merupakan kependekan dari *Recommended Standart Number 232*. RS232 dibuat untuk interface antara peralatan terminal data dan peralatan komunikasi data, dengan menggunakan data biner serial sebagai data yang ditransmisikan. Serial interface RS232 memberi ketentuan logic level sebagai berikut:

- Logic 1 disebut "*mark*" terletak antara –3Volt hingga –15 Volt.
- Logic 0 disebut "*space*" terletak antara + 3Volt hingga +15Volt.

Daerah tegangan antara -3Volt hingga +3Volt adalah *invaled level*, yaitu daerah tegangan yang tidak memiliki *logic* sehingga daerah itu harus dihindari. Suatu saluran data RS232 yang memberi keadaan tegangan ini berarti ada kesalahan. Demikian pula pada saluran pada daerah lebih negatif dari -15Volt dan daerah lebih positif dari +15Volt.

Pada standard RS232 ditentukan jenis-jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE dan DCE. Ada 24 jenis sinyal, tetapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakaipun ditentukan dalam standar RS232, untuk sinyal yang lengkap dipakai konektor DB25, sedangkan konektor yang memakai 9 sinyal digunakan konektor DB9. Susunan sinyal RS232 pada konektor DB9 dan DB25 berlainan susunan kakinya. Konfigurasi pin-pin konektor DB9 seperti ditunjukkan dalam Gambar 2.15.



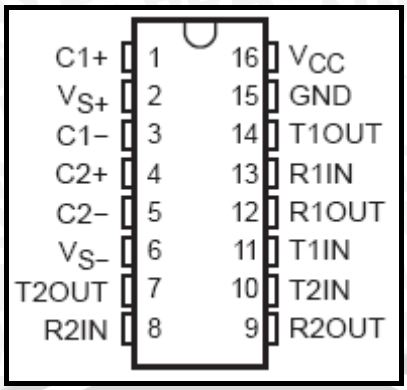
Gambar 2.15 Konfigurasi pin-pin konektor DB9

Sumber: Lammert, 1997

Adapun guna masing-masing pin pada DB9 adalah sebagai berikut:

1) Pin 1: *Data Carier Detect*.
   Berguna bagi DTE untuk tidak memperbolehkan penerimaan data
2) Pin 2 : *Received Data (RxD/RD)*
   Berguna sebagai jalur pengiriman data dari DCE ke DTE.
3) Pin 3 : *Transmitted Data (TxD/TD)*
   Berguna untuk jalur pengiriman data dari DTE ke DCE.
4) Pin 4: *Data Transmitted Ready*
   Berguna untuk memberitahu DCE bahwa DTE telah aktif dan siap untuk bekerja.
5) Pin 5 : *Signal Ground*.
   Berguna untuk referensi semua tegangan interface.
6) Pin 6 : *Data Set Ready*
   Berguna untuk memberitahu DTE bahwa DCE telah aktif dan sipa untuk bekerja.
7) Pin 7 : *Request To Send (RTS)*
   Berguna untuk memberitahu DCE bahwa DTE akan mengirim data. RTS merupakan sebuah protokol perangkat keras yang mendahului pengirman data dari DTE ke DCE
8) Pin 8 : *Clear To Send*
   Berguna untuk memberitahu DTE bahwa DCE siap untuk menerima data. CTS merupakan sebuah protokol perangkat keras yang mendahului pengiriman data dari DTE ke DCE.
9) Pin 9 : *Ring Indikator*

Selanjutnya saluran tersebut dipasangkan pada *IC* MAX232, yang berfungsi sebagai penyetara level tegangan. Konfigurasi pin *IC* MAX232 seperti ditunjukkan dalam Gambar 2.16.



Gambar 2.16 Konfigurasi pin *IC* MAX232

Sumber: Datasheet MAX232

Spesifikasi dari EIA/TIA-232 E dan V.28 menentukan bahwa level tegangan yang lebih dari 3V adalah berlogika 0. Jadi, semua *receiver* adalah terbalik. *Input threshold* ditentukan pada 0,8V dan 2,4V sehingga keluaran dari *receiver* akan sesuai dengan level tegangan dari TTL.

BAB III
METODOLOGI

3.1 Lingkup Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat bertujuan agar sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data dapat menjadi salah satu solusi metode pengisian nominal energi listrik pada sistem listrik prabayar. Perancangan dan pembuatan alat bersifat eksperimental yang pengujiannya hanya dalam lingkup laboratorium.

3.2 Persiapan Perancangan dan Pembuatan Alat

Perancangan, pembuatan, dan pengujian alat dilakukan di Laboratorium Dasar Elektrik dan Pengukuran Jurusan Elektro Fakultas Teknik Universitas Brawijaya. Adapun alat-alat yang digunakan dalam perancangan, pembuatan, dan pengujian alat adalah:

1. Voltmeter.
2. Ampmeter digital.
3. *Frequency counter*.
4. *Oscilloscope*.
5. *Function generator*.
6. Catu daya DC.

3.3 Variabel Perancangan dan Pembuatan Alat

Variabel merupakan faktor-faktor yang mempengaruhi perancangan dan pembuatan sistem. Dalam perancangan dan pembuatan unit sentral data pada sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data, terdapat variabel-variabel yang ditetapkan, diubah, dan diukur, untuk selanjutnya dianalisa. Variabel-variabel tersebut adalah sebagai berikut:

1. Frekuensi keluaran pemancar LM1893
2. Tegangan keluaran pemancar LM1893
3. Level Tegangan masukan pemancar LM1893
4. Baudrate
5. Data nominal kWh

6. Level tegangan TTL
7. Level Tegangan RS232

3.4 Langkah-langkah perancangan dan pembuatan alat.

Agar proses perancangan dan pembuatan alat dapat berjalan dengan sistematis, maka perlu ditetapkan langkah-langkah yang diambil. Langkah-langkah yang diambil dalam perancangan dan pembuatan alat adalah sebagai berikut:

3.4.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan dasar teori tentang segala sesuatu yang mendukung perancangan dan pembuatan alat. Dasar teori tersebut diambil dari buku-buku maupun internet untuk mengetahui karakteristik komponen, prinsip kerja serta teori yang menunjang, antara lain:

1. Prinsip dasar transmisi data melalui jala-jala listrik (*power line carrier)*
2. Perangkat keras dan perangkat lunak mikrokontroler ATMEGA8515
3. Prinsip dasar pemrograman dengan Delphi 7
4. Prinsip dasar *DBMS (Database Management System)*
5. Prinsip dasar komunikasi serial RS232

3.4.2 Penentuan Spesifikasi Alat

Setelah mempelajari berbagai literatur yang menunjang perancangan dan pembuatan alat, kemudian dari permasalahan yang ada dibuat spesifikasi alat yang dibuat.

3.4.3 Perancangan Alat

Perancangan alat meliputi perancangan perangkat keras dan perangkat lunak yang keduanya dikombinasikan untuk memenuhi spesifikasi alat yang telah ditentukan sebelumnya.

3.4.3.1 Perancangan Perangkat Keras

Perancangan alat disesuaikan dengan fungsi dari komponen-komponen yang akan digunakan sehingga terbentuk rangkaian elektronik yang siap direalisasikan.

Untuk perancangan alat, dilakukan langkah-langkah sebagai berikut:

1. Penyusunan blok diagram sistem untuk mempermudah pemahaman tentang alur kerja dari alat yang dibuat.
2. Menentukan nilai dan jenis komponen yang sesuai dengan hasil perhitungan serta yang ada di pasaran sehingga mudah dalam mencari komponen jika terjadi kerusakan.
3. Perencanaan dan pembuatan rangkaian dari masing-masing blok, yang meliputi:
   a. Rangkaian LM1893.
   b. Minimum sistem ATMEGA8515.
   c. Rangkaian antarmuka komunikasi serial RS232.
4. Menggabungkan beberapa blok menjadi keseluruhan sistem yang direncanakan.
5. Mengadakan pengujian-pengujian per blok rangkaian yang direncanakan.

3.4.3.2 Perancangan Perangkat Lunak

Setelah perangkat keras dirancang sesuai dengan perencanaan, maka langkah selanjutnya adalah perancangan perangkat lunak. Perangkat lunak ini difungsikan untuk mengatur keseluruhan sistem yang terdiri atas beberapa perangkat keras sehingga sistem ini dapat bekerja dengan baik. Perancangan perangkat lunak dilakukan dengan pembuatan diagram alir.

3.4.4 Pembuatan Alat

Pembuatan alat dilakukan sesuai dengan perancangan alat. Tahap ini dimulai dengan pembuatan rangkaian per blok menggunakan *project board*. Hal ini bertujuan untuk melihat unjuk kerja dari setiap blok dan jika terjadi kesalahan, perubahan rancangan dapat dilakukan dengan mudah. Jika setiap blok sudah menunjukkan kerja yang sesuai dengan perancangan, tahap selanjutnya adalah pembuatan PCB yang memuat seluruh blok rangkaian. Tahap terakhir yaitu pemasangan komponen-komponen elektronika pada PCB.

Pembuatan perangkat lunak dilakukan dengan mengimplementasikan diagram alir program yang telah direncanakan. Pembuatan perangkat lunak terdiri dari 2 macam yaitu perangkat lunak mikrokontroler dan perangkat lunak *PC*. Pembuatan perangkat lunak mikrokontroler menggunakan bahasa C. Pembuatan perangkat lunak *PC* menggunakan bahasa pemrograman Delphi 7 dan pembuatan database menggunakan Microsoft Access 2003.

3.4.5 Pengujian dan Analisis Alat

Tahap pengujian meliputi pengujian tiap blok rangkaian yang bertujuan untuk mengetahui unjuk kerja dari masing-masing blok serta kesesuaian dengan spesifikasi perancangan kemudian pengujian dilakukan dengan menggabungkan blok-blok rangkaian yang telah dibuat dengan tujuan untuk mengetahui tingkat keberhasilan dari sistem yang telah dibuat.

Pengujian yang dilakukan menghasilkan data-data yang nantinya akan dianalisis untuk mengetahui tingkat keberhasilan perancangan. Data yang dihasilkan dianalisis dengan cara membandingkannya dengan hasil perhitungan. Hasil analisis tersebut selanjutnya akan disusun suatu kesimpulan. Analisis akhir dilakukan untuk mengetahui serta memastikan bekerjanya alat ini.

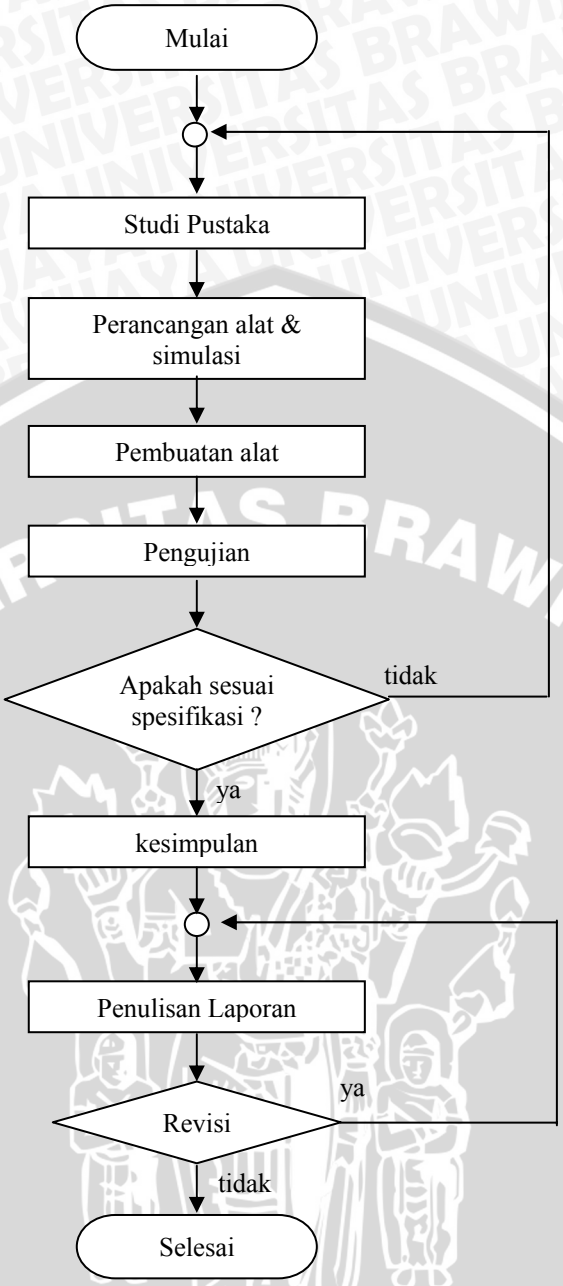
3.4.6 Pengambilan Kesimpulan dan Saran

Tahap berikutnya adalah pengambilan kesimpulan dari peralatan yang dibuat. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penelitian.

3.4.7 Penulisan Laporan

Setelah semua data hasil penelitian didapat maka disusun laporan agar penelitian dapat didokumentasikan dan dapat digunakan sebagai referensi untuk pengembangan penelitian dimasa yang akan datang.

Gambar 3.1 menggambarkan urutan kegiatan yang dilakukan dalam perancangan, pembuatan, dan pengujian unit sentral data pada sistem kwh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data.



Gambar 3.1 Diagram alir pelaksanaan perancangan dan pembuatan alat

BAB IV
PERENCANAAN DAN PEMBUATAN ALAT

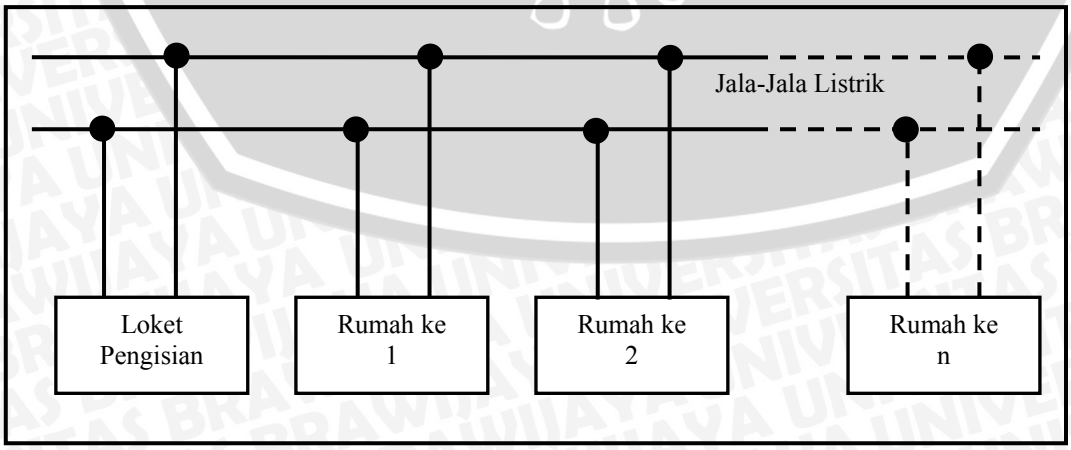
Pada bab ini menjelaskan proses perancangan dan pembuatan Unit Sentral Data pada Sistem kWh Meter Prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data. Proses perancangan dan pembuatan tersebut meliputi perencanaan sistem, penentuan spesifikasi alat, penyusunan blok diagram, penyusunan skema rangkaian dan perhitungan nilai komponennya, perancangan perangkat lunak mikrokontroler, serta perancangan perangkat lunak PC dan database.

4.1 Gambaran Umum Sistem Kwh Meter Prabayar yang Menggunakan Jala-Jala Listrik Sebagai Media Komunikasi Data

Gambaran umum dari sistem kWh Meter prabayar yang akan dibuat ditunjukkan dalam Gambar 4.1. Dalam sistem tersebut, akan terjadi komunikasi dua arah antara setiap rumah dengan loket pengisian. Komunikasi yang terjadi terdiri dari komunikasi untuk memberikan nominal kWh dari loket pengisian ke rumah pelanggan yang membeli nominal kWh dan komunikasi untuk memberikan informasi sisa nominal kWh dari setiap rumah ke loket pengisian. Komunikasi terjadi melalui jala-jala listrik sehingga diperlukan suatu modem *power line carrier (PLC).*

Dari gambaran umum tersebut dapat ditentukan spesifikasi umum dari sistem, antara lain :

1. Jala-jala listrik yang digunakan sebagai media komunikasi adalah satu fasa dengan tegangan sebesar 220 V dan frekuensi sebesar 50Hz.
2. Jalur komunikasi tidak melalui transformator.



Gambar 4.1 Gambaran umum dari sistem kWh Meter prabayar

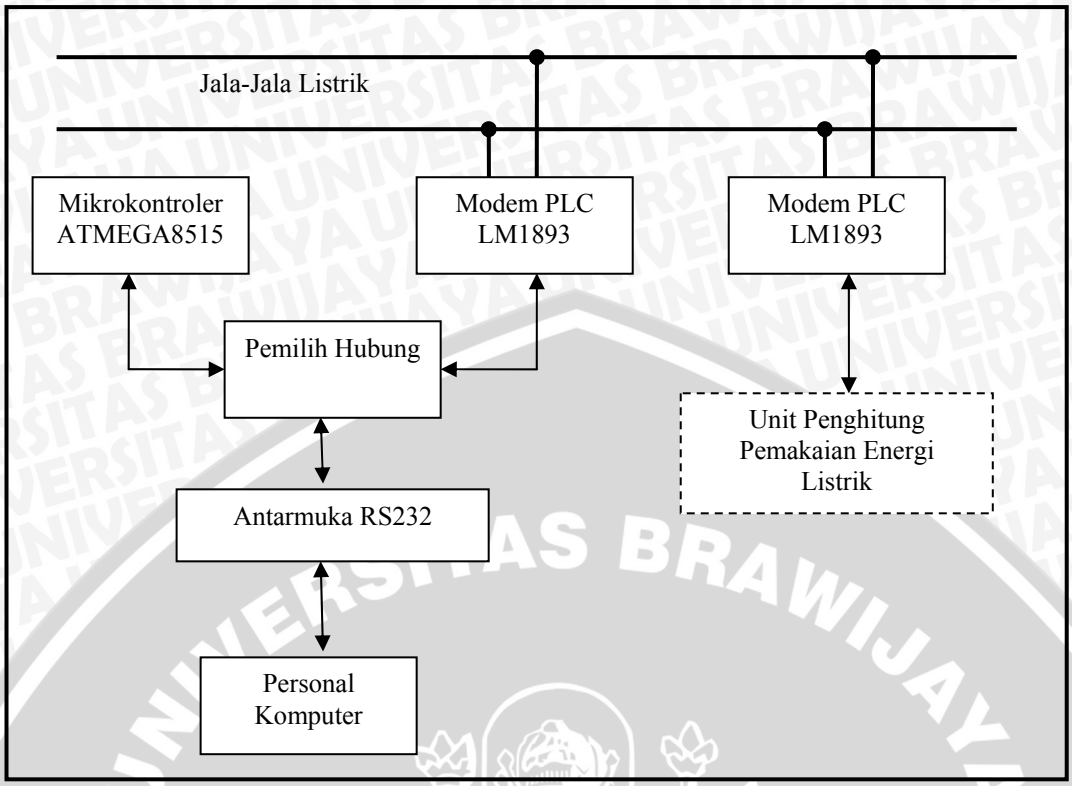
4.2 Spesifikasi dan Blok Diagram Unit Sentral Data

Unit sentral data merupakan salah satu bagian dari sistem kWh Meter prabayar yang berfungsi sebagai pengontrol utama komunikasi dalam sistem, penyimpan dan pengolah data pemakaian energi listrik setiap pelanggan, serta memberi nominal kWh kepada pelanggan yang membeli nominal kWh.

Spesifikasi Unit Sentral Data ditentukan sebagai berikut :

1. Menggunakan Mikrokontroler ATMEGA8515 sebagai pengontol utama komunikasi dalam sistem. Komunikasi data menggunakan mode serial Asinkronus dengan baudrate sebesar 360 bps.
2. Menggunakan modem *PLC* dengan *IC* tipe LM1893 sebagai *tranceiver* data dengan frekuensi pembawa sebesar 125 kHz.
3. Menggunakan PC sebagai penyimpan dan pengolah data pemakaian daya listrik setiap pelanggan serta memberi nominal kWh kepada pelanggan yang membeli *pulsa* dengan delphi 7 sebagai bahasa pemrogramannya. Jumlah rumah yang bisa diproses adalah 255 buah dan jumlah maksimum nominal kWh yang bisa diproses adalah 16777,215 kWh.
4. Antarmuka antara PC dengan mikrokontoler menggunakan RS232.
5. Menggunakan relay 5 V sebagai pemilih hubungan antara Mikrokontroler dengan Modem *PLC* atau antara Mikrokontroler dengan PC.

Blok diagram Unit Sentral Data yang disusun ditunjukkan dalam Gambar 4.2



Gambar 4.2 Blok diagram unit sentral data

Modulator *Power Line Carrier* (PLC) LM1893 berfungsi untuk mengirimkan data digital yang telah termodulasi secara serial melalui jala-jala. Modulasi yang digunakan adalah *Frequency Shift Keying (FSK)*. FSK mengubah kode digital menjadi sinyal sinusoida. Disini logika 1 dan 0 diubah kedalam 2 frekuensi yang berbeda. Untuk logika 1 diubah dalam gelombang sinusoida berfrekuensi f_1 dan logika 0 diubah dalam gelombang sinusoida berfrekuensi f_2. Data digital yang telah termodulasi, ditumpangkan ke gelombang pembawa agar dapat ditransmisikan. Frekuensi gelombang pembawa yang digunakan yaitu 125 kHz.

Sinyal yang dipancarkan, diterima oleh demodulator *Power Line Carrier* (PLC) LM1893. Demodulator terdiri dari pengkopel untuk melewatkan frekuensi sinyal dan menahan frekuensi jala-jala, serta *Phase Locked Loop* (PLL) untuk memisahkan frekuensi sinyal dengan frekuensi pembawa. Sinyal yang telah terpisah dengan gelombang pembawa masuk ke bagian FSK demodulator yang terdapat dalam chip IC LM1893 untuk diubah menjadi data digital.

Pemilih hubung berfungsi sebagai pemilih hubungan antara mikrokontroler dengan PC atau antara mikrokontroler dengan modem *PLC*.

Mikrokontroler ATMEGA8515 adalah pengontrol utama dari sistem yang berfungsi untuk mengatur komunikasi dengan setiap rumah (unit penghitung pemakaian daya listrik).

Antarmuka RS232 berfungsi sebagai antarmuka untuk komunikasi serial antara personal komputer dengan mikrokontroler.

Personal komputer berfungsi sebagai penyimpan dan pengolah data pelanggan, termasuk menyimpan dan mengolah sisa nominal kWh setiap pelanggan.

4.3 Prinsip Kerja Unit Sentral Data

Proses kerja unit sentral data dimulai dari mikrokontoler mengontrol relay agar menghubungkan port serial mikrokontroler dengan PC melalui *interface* MAX232. Kemudian PC mengirimkan data ke mikrokontroler yang berisi nominal kWh, kode perintah dan alamat tujuan (nomor *ID* unit penghitung pemakaian energi listrik). Jika kode perintah sama dengan 0, berarti unit sentral data hanya meminta sisa nominal kWh ke unit penghitung pemakaian daya listrik. Jika kode perintah sama dengan 1, berarti unit sentral data akan mengirim nominal kWh ke unit penghitung pemakaian energi listrik kemudian meminta data hasil penjumlahan nominal kWh yang dikirim dengan sisa pulsa setelah pengiriman nominal kWh. Dan jika kode perintah sama dengan 2, berarti unit sentral data akan mengganti nilai nominal kWh yang tersimpan di unit penghitung pemakaian energi listrik dengan nilai yang baru.

Setelah data diterima oleh mikrokontroler, relay dikontrol agar menghubungkan port serial mikrokontroler dengan modem *Power Line Carrier* LM1893. Selanjutnya data yang diterima oleh mikrokontroler dari PC dikirimkan ke unit penghitung pemakaian energi listrik yang dituju. Data yang dikirimkan tersebut ditambah dengan nomor ID unit sentral data yaitu 0. Kemudian mikrokontroler menunggu jawaban dari unit penghitung daya listrik. Jawaban dari unit penghitung daya listrik berupa data yang terdiri dari data nominal kWh yang dimiliki unit penghitung pemakaian energi listrik saat memberikan data ke unit sentral data, nomor *ID* unit sentral data yaitu 0, dan nomor *ID* unit penghitung pemakaian energi listrik yang memberi jawaban.

Setelah mikrokontroler menerima jawaban dari unit penghitung energi listrik, relay dikontrol agar menghubungkan port serial mikrokontroler dengan PC melalui *interface* MAX232. Data jawaban tersebut kemudian diberikan ke PC untuk diolah dan disimpan. Data yang diberikan ke PC berupa nominal nominal kWh dan nomor *ID* unit penghitung pemakaian energi listrik yang memberi jawaban.

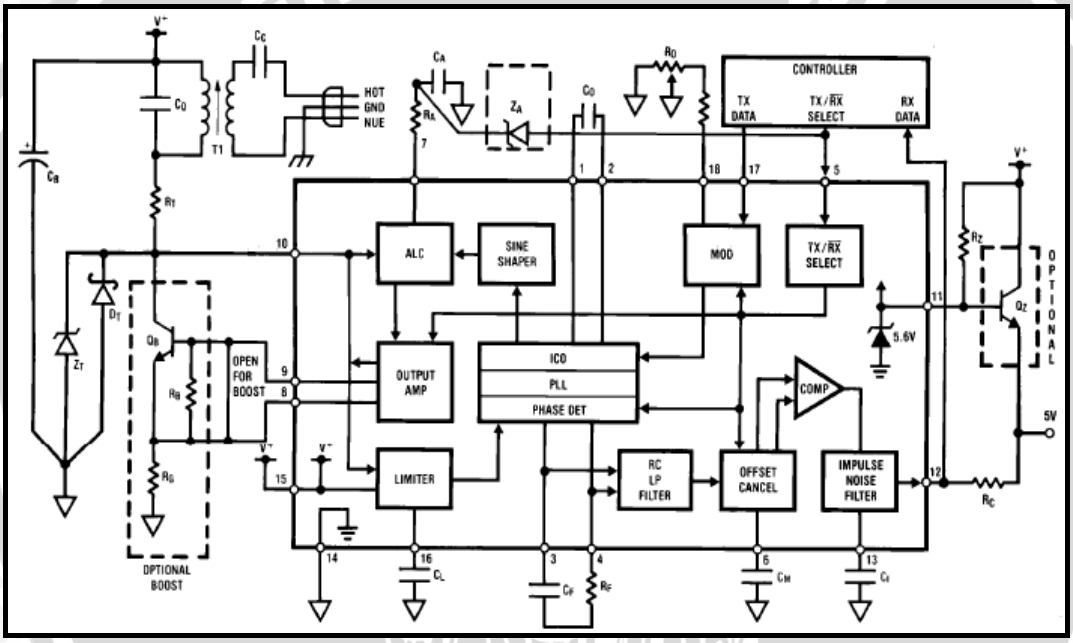
4.4 Perancangan Perangkat Keras

Perangkat keras yang digunakan pada unit sentral data terdiri dari beberapa bagian yaitu:

1. Rangkaian Modem *Power Line Carrier (PLC)*
2. Rangkaian Mikrokontroler ATMEGA8515
3. Rangkaian RS232
4. Rangkaian Relay

4.4.1 Rangkaian Modem *Power Line Carrier (PLC)*

Rangkaian modem *Power Line Carrier (PLC)* LM1893 beserta komponen pendukungnya ditunjukkan dalam Gambar 4.3.



Gambar 4.3. IC LM1893 beserta komponen pendukungnya
Sumber: Datasheet LM1893 , 2003

Penentuan nilai masing-masing komponen pendukungnya dengan frekuensi *carrier* (F_0) yang digunakan adalah 125 kHz, frekuensi data 180 Hz (360 baud), tegangan catu daya 18 Volt dan untuk jala-jala listrik 220 V 50 Hz adalah sebagai berikut:

4.4.1.1 Bagian Pemancar

Komponen pendukung bagian pemancar terdiri dari C_0, R_0, C_A, R_A, T_1, C_Q, Z_T, dan R_T. Penentuan nilai masing-masing komponen pendukung bagian pemancar adalah sebagai berikut.

- **C_0**

C_0 berfungsi untuk menentukan frekuensi *carrier* yang digunakan. Penentuan nilai C_0 dilakukan berdasarkan *datasheet* IC LM1893 yaitu seperti yang ditunjukkan dalam Gambar 2.2.

Frekuensi *carrier* yang akan digunakan adalah 125 kHz. Dengan persamaan 2-2 dapat diperoleh nilai C_0 sebagai berikut :

$$C_0 = \frac{70{,}6 \times 10^{-6}}{F_0}$$

$$C_0 = \frac{70{,}6 \times 10^{-6}}{125 \times 10^{3}} = 0{,}5648 \times 10^{-9}\ \text{F}$$

Dari hasil perhitungan tersebut, maka C_0 dibuat sebesar 560 pF, disesuaikan dengan yang ada di pasaran.

- **R_0**

Resistor R_0 digunakan untuk membangkitkan arus V_{BE} yang akan dikalikan 2 untuk menghasilkan arus sebesar 200 µA pada ICO yang menentukan nilai F_0. Semakin kecil nilai R_0 akan meningkatkan F_0. Nilai yang dianjurkan untuk R_0 adalah antara 5,6 kΩ sampai 7,6 kΩ. (Sumber: *Datasheet LM1893 , 2003*). Oleh karena itu digunakan nilai potensio 2 kΩ dirangkai seri dengan resistor 5,6 kΩ untuk R_0.

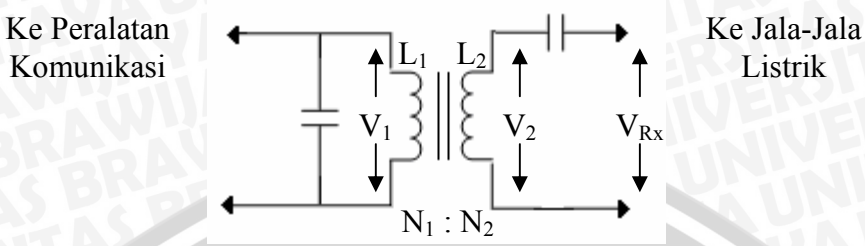
- **C_A dan R_A**

Besarnya nilai C_A dan R_A akan mempengaruhi ALC IC LM 1893. Semakin kecil nilai C_A dan R_A maka ALC akan tidak stabil, sedangkan semakin besar nilainya respon ALC akan lambat tetapi semakin stabil. Nilai yang dianjurkan untuk C_A dan R_A adalah C_A = 0,1 µF dan R_A = 10 µΩ.

- **T_1**

T_1 digunakan bersama C_Q untuk membentuk frekuensi resonansi F_Q. T_1 juga digunakan sebagai *transformator coupling* pada penumpangan sinyal informasi ke jala-jala listrik. Nilai induktansi kumparan primer L_1 bersama-sama dengan C_Q akan menentukan frekuensi resonansi. T_1 juga harus mempunyai perbandingan jumlah lilitan kumparan primer dan sekunder sebesar mungkin agar dapat menumpangkan sinyal ke jala-jala listrik dengan baik serta dapat menguatkan sinyal dari pemancar sehingga

tegangan sinyal minimum yang diperlukan *IC* LM1893 saat mode *receive* dapat terpenuhi. Gambar 4.4 menunjukkan rangkaian kopling transformer yang digunakan oleh *IC* LM1893.



Gambar 4.4 Rangkaian kopling transformer

Untuk mencari perbandingan lilitan T_1 minimum, maka harus diketahui nilai V_{Rx} minimum terlebih dahulu. V_{Rx} adalah tegangan sinyal dari pemancar (V_{Tx}) berfrekuensi 125 kHz yang telah mengalami pelemahan karena proses transmisi sinyal. Berdasarkan datasheet, pelemahan total maksimum yang diijinkan adalah sebesar 70 dB dan tegangan sinyal dari pemancar adalah sebesar 13,3 V. Dengan persamaan 4-1, maka nilai V_{Rx} minimum adalah :

$$dB = 20\log\frac{V_{Rx}}{V_{Tx}}\,dB \qquad (4\text{-}1)$$

$$-70dB = 20\log\frac{V_{Rx}}{13{,}3}\,dB$$

$$\frac{-70}{20} = \log\frac{V_{Rx}}{13.3}$$

$$-3{,}5 = \log V_{Rx} - \log 13{,}3$$

$$\log V_{Rx} = -2{,}38$$

$$V_{Rx} = 4{,}2\times10^{-3}\ \text{volt}$$

Nilai Induktansi kumparan sekunder (L_2) untuk transformer yang akan digunakan (tipe Toko 707VX-A042YUK) adalah sebesar 1 µH. Untuk frekuensi pembawa 125 kHz, nilai V_2 adalah :

$$V_2\,|_{f=125kHz} = \frac{Z_{L2}}{Z_{L2}+Z_C}V_{Rx} \qquad (4\text{-}2)$$

$$V_2 |_{f=125kHz} = \frac{jX_{L2}}{jX_{L2} - jX_C} V_{Rx}$$

$$V_2 |_{f=125kHz} = \frac{X_{L2}}{X_{L2} - X_C} V_{Rx}$$

$$V_2 |_{f=125kHz} = \frac{2\pi f L_2}{2\pi f L_2 - \frac{1}{2\pi f C}} V_{Rx}$$

$$V_2 |_{f=125kHz} = \frac{2 \times 3{,}14 \times 125.10^3 \times 1.10^{-6}}{2 \times 3{,}14 \times 125.10^3 \times 1.10^{-6} - \frac{1}{2 \times 3{,}14 \times 125.10^3 \times 220.10^{-9}}} 4{,}2.10^{-3}$$

$$V_2 |_{f=125kHz} = 6{,}57.10^{-4} \text{ volt} = 0{,}66 \text{ mV}$$

Untuk tegangan jala-jala sebesar 220 V dengan frekuensi sebesar 60 Hz, nilai V_2 adalah :

$$V_2 |_{f=60Hz} = \frac{Z_{L2}}{Z_{L2} + Z_C} V_{Jala-Jala} \quad \text{(4-3)}$$

$$V_2 |_{f=60Hz} = \frac{jX_{L2}}{jX_{L2} - jX_C} V_{Jala-Jala}$$

$$V_2 |_{f=60Hz} = \frac{X_{L2}}{X_{L2} - X_C} V_{Jala-Jala}$$

$$V_2 |_{f=60Hz} = \frac{2\pi f L_2}{2\pi f L_2 - \frac{1}{2\pi f C}} V_{Jala-Jala}$$

$$V_2 |_{f=60Hz} = \frac{2 \times 3{,}14 \times 60 \times 1.10^{-6}}{2 \times 3{,}14 \times 60 \times 1.10^{-6} - \frac{1}{2 \times 3{,}14 \times 60 \times 220.10^{-9}}} 220$$

$$V_2 |_{f=60Hz} = 6{,}88.10^{-6} \text{ volt} = 6{,}88 \ \mu\text{V}$$

Karena nilai V_2 akibat tegangan jala-jala sangat kecil, maka tegangan ini dapat diabaikan.

Berdasarkan datasheet, tegangan minimum yang diperlukan *IC* LM1893 saat mode *receive* adalah sebesar 1,8 mV, sehingga perbandingan lilitan minimum T_1 yang diperbolehkan adalah :

$$\frac{V_1}{V_2} = \frac{N_1}{N_2} \qquad (4\text{-}4)$$

$$\frac{1{,}8}{0{,}66} = \frac{N_1}{N_2}$$

$$\frac{N_1}{N_2} = 2{,}73 \approx 3$$

Ditetapkan N_2 =1, maka perbandingan lilitan T_1 minimum (N_1 : N_2) adalah 3 : 1.

Dengan pertimbangan tersebut maka untuk T_1 dianjurkan menggunakan transformer tipe Toko 707VX-A042YUK yang mempunyai induktansi kumparan primer sebesar 49 µH dan induktansi kumparan sekunder sebesar 0,98 µH dengan perbandingan lilitan sebesar 7 : 1.

- **C_Q**

C_Q digunakan bersama T_1 sebagai frekuensi resonansi yang sama dengan F_0 supaya sinyal informasi dapat ditumpangkan ke jala-jala listrik. Frekuensi resonansi yang dihasilkan oleh C_Q dan induktansi T_1 harus sama dengan F_0. Penentuan nilai C_Q didasarkan pada Gambar 2.6.

Dari Gambar 2.6 dapat dihitung nilai C_Q menggunakan persamaan 2-3. Dengan L_1 adalah nilai induktansi T_1 yang telah ditentukan sebesar 49 µH, Maka nilai C_Q adalah:

$$C_Q = \frac{1}{(2\pi F_0)^2 L_1}$$

$$C_Q = \frac{1}{(2\pi \times 125 \times 10^3)^2 \times 49 \times 10^{-6}}$$

$$C_Q = 33{,}01 \times 10^{-9} \text{ nF}$$

Sehingga nilai C_Q ditentukan sebesar 33 nF.

- **C_C**

Kapasitor C_C digunakan agar tegangan jala-jala listrik tidak masuk ke lilitan T_1. C_C juga digunakan sebagai L_C *highpass* filter. C_C harus mempunyai impedansi yang cukup besar untuk memfilter tegangan jala-jala listrik 50 Hz, dan impedansi untuk F_0 (125 kHz) harus sekecil mungkin dengan acuan Gambar 2.7. Arus jala-jala yang

melewati C_C harus lebih kecil dari arus maksimum 10 *Amp-turn* (1 Amp melalui 10 lilitan), dengan acuan Gambar 2.8.

Dari Gambar 2.7 dan 2.8 ditentukan nilai C_C sebesar 0,22 µF. Dengan persamaan 2-1, impedansi C_C untuk frekuensi jala-jala 50 Hz adalah sebesar:

$$Z = \frac{1}{2 \cdot \pi \cdot f \cdot C}$$

$$Z = \frac{1}{2 \cdot \pi \cdot 50 \cdot 0{,}22 \times 10^{-6}}$$

$$Z = 0{,}724 \times 10^{6}\ \Omega$$

Nilai impedansi untuk frekuensi 125 kHz sebesar:

$$Z = \frac{1}{2 \cdot \pi \cdot f \cdot C}$$

$$Z = \frac{1}{2 \cdot \pi \cdot 125 \times 10^{3} \cdot 0{,}22 \times 10^{-6}}$$

$$Z = 5{,}8\Omega$$

- **Z_T**

Z_T merupakan dioda zener yang digunakan untuk mencegah adanya sinyal transien yang melalui transformator T_1 menuju ke pin *carrier I/O*. sinyal transien tersebut bias juga berasal dari pengosongan tegangan di kapasitor C_C pada saat menghubungkan atau melepas rangkaian dari jala-jala listrik. Untuk itu Z_T menggunakan dioda *avalance* yang biasanya sering digunakan untuk mencegah sinyal transien. Z_T harus mempunyai tegangan dadal yang sedikit lebih besar dari 44 V, karena pin *carrier I/O* mempunyai dioda zener internal dengan BV 44 V.

- **R_T**

R_T berfungsi sebagai pembagi tegangan dengan Z_T, yang akan menyerap tegangan transien supaya tegangan yang masuk pada pin *carrier I/O* tetap dibawah 44 Volt.

4.4.1.2 Bagian Penerima

Komponen pendukung bagian penerima terdiri dari C_L, C_I, C_F, R_F, dan Z_A. Penenetuan nilai masing-masing komponen pendukung bagian penerima adalah sebagai berikut:

- **C_L**

LM 1893 mempunyai *filter band pass* untuk meningkatkan sensivitas penerimaan sinyal. Frekuensi *cutt off* atas ditentukan sebesar 300 kHz. Sedangkan frekuensi *cut off* bawah ditentukan oleh nilai C_L. Berdasarkan Gambar 2.9 untuk F_0=125 kHz, maka nilai C_L sebesar 0,047 µF.

- **C_I**

C_I digunakan pada bagian filter *noise* yang berfungsi untuk mencegah lewatnya sinyal pulsa yang mempunyai waktu lebih pendek dari waktu pengisian integrator chip LM 1893. Waktu pengisian integrator ditentukan sebesar ½ waktu data bit. Jadi nilai C_I ditentukan besarnya data *rate* yang digunakan. Gambar 2.10 menunjukkan hubungan data *rate* dengan nilai C_I. Untuk frekuensi data maksimum 180 Hz, maka nilai C_I sebesar 0,047 µF.

- **C_F dan R_F**

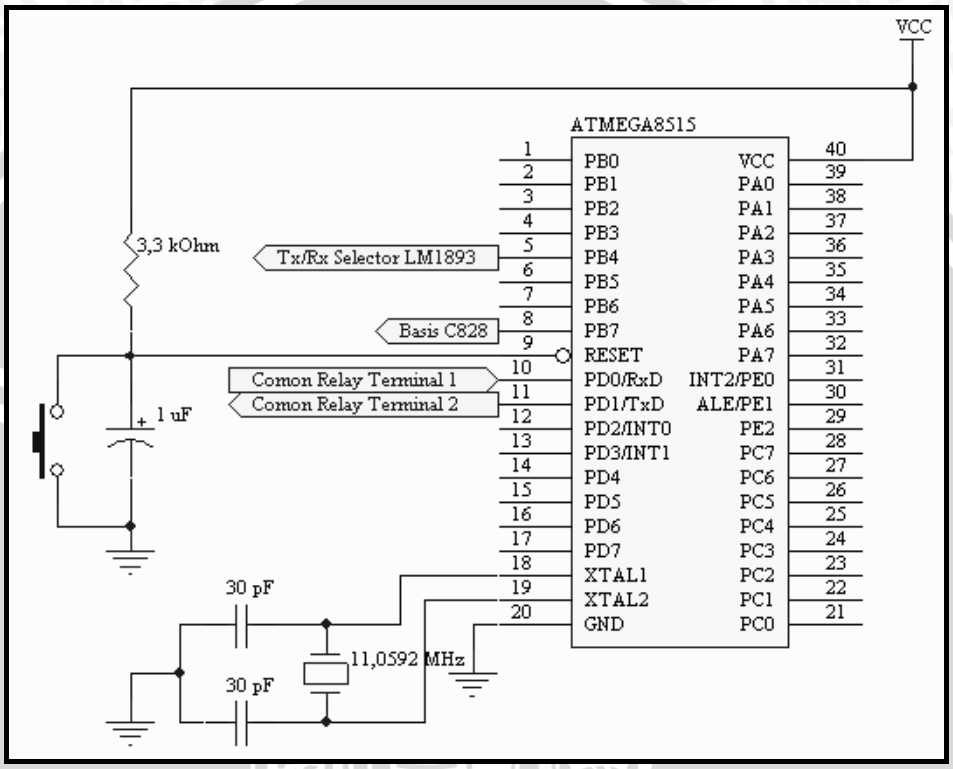
C_F dan R_F adalah komponen yang digunakan pada PLL IC LM 1893 sebagai filter untuk menghilangkan *noise*. Besarnya nilai C_F dan R_F ditentukan oleh frekuensi *carrier* F_0 yang digunakan. Gambar 2.11 dan 2.12 merupakan Gambar hubungan nilai C_F dan R_F dengan F_0. Untuk F_0 = 125 kHz, maka nilai C_F = 0,047 µF dan R_F = 3,3 kΩ.

- **Z_A**

Z_A digunakan jika sinyal yang masuk pada *input* R_X(pin 10) melebihi tegangan catu bagian penerima, yang disebabkan tegangan catu pemancar lebih tinggi dari penerima, jarak T_X dan R_X yang terlalu dekat, atau perbandingan lilitan *step-up* T_1 pada R_X lebih tinggi dari perbandingan lilitan *step-down* T_1 pada T_X.

4.4.2 Rangkaian Mikrokontroler ATMEGA8515

Mikrokontroler yang digunakan pada rangkaian ini adalah mikrokontroler tipe ATMEGA8535 yang merupakan keluarga dari AVR. Komponen ini merupakan sebuah chip tunggal yang berfungsi sebagai pengontrol komunikasi data antara unit sentral data dan unit penghitung pemakaian daya listrik. Pemilihan mikrokontroler jenis ini karena mudah diperoleh di pasaran. Pin-pin Mikrokontroler ATMEGA8535 dihubungkan pada rangkaian pendukung seperti yang ditunjukkan dalam Gambar 4.5.



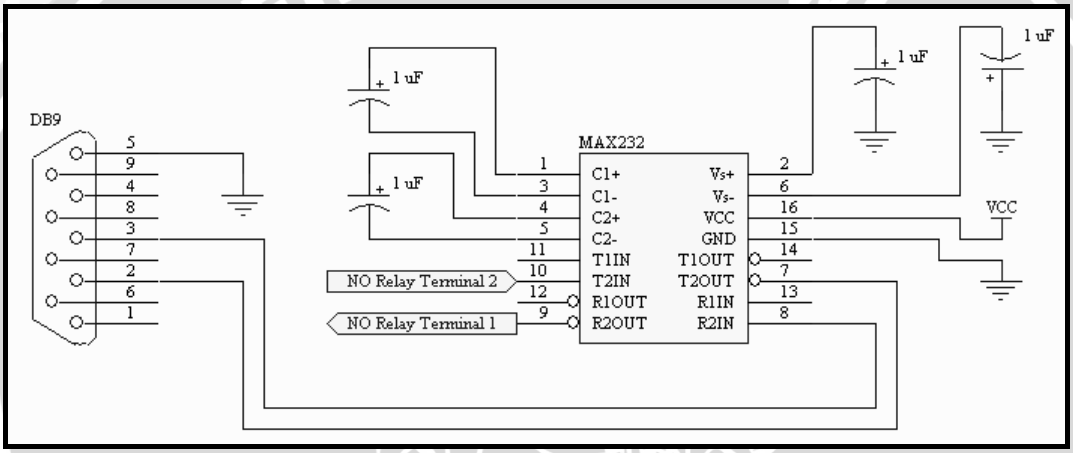
Gambar 4.5. Rangkaian Mikrokontroler

Pin-pin yang digunakan yaitu:

1. PORTB
   - PORTB.4 digunakan sebagai pemilih mode *transmit* atau *receive* pada modem *Power Line Carrier (PLC)* LM1893
   - PORTB.7 digunakan sebagai pengontrol relay untuk memilih hubungan antara mikrokontroler dengan PC atau antara mikrokontroler dengan modem *Power Line Carrier (PLC)* LM1893
2. PORTD
   - PORTD.0 digunakan sebagai masukan data serial
   - PORTD.1digunakan sebagai keluaran data serial

4.4.3 Rangkaian Antarmuka RS232

Karena port serial komputer tidak bekerja pada level tegangan TTL, melainkan level tegangan RS-232. untuk itu diperlukan komponen tambahan yang berfungsi untuk mengkonversi level tegangan TTL ke level tegangan RS-232, begitu juga sebaliknya. Komponen yang digunakan adalah MAX232 yang memerlukan beberapa komponen tambahan berupa empat buah kapasitor yang nilainya telah ditentukan *datasheet*, yaitu 1µF untuk seluruh kapasitor. MAX232 memiliki sepasang terminal masukan level tegangan TTL yang berkorespondensi dengan sepasang terminal keluaran level tegangan RS-232, juga sepasang terminal masukan level RS-232 yang berkorespondensi dengan sepasang terminal keluaran level tegangan TTL. Rangkaian RS232 ditunjukkan dalam Gambar 4.6.

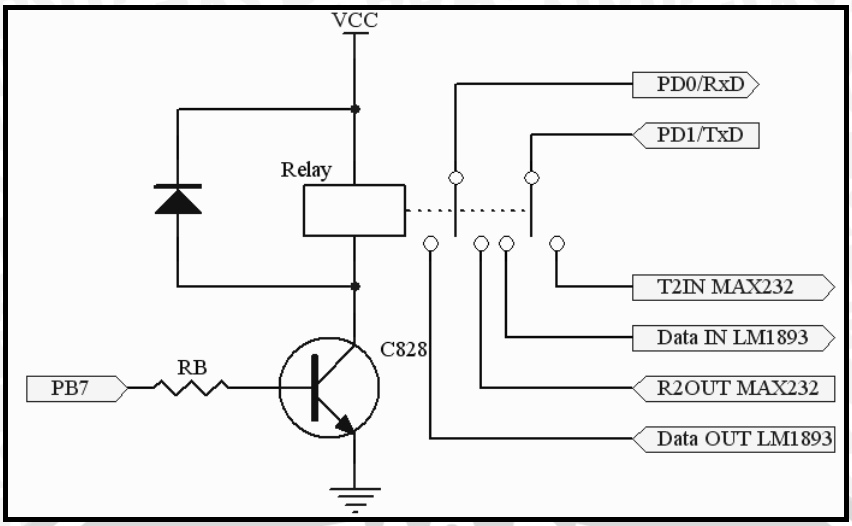


Gambar 4.6. Rangkaian Antar Muka RS232

Pin T2OUT dan pin R2IN dari MAX232 dihubungkan dengan DB9, sedangkan pin R2OUT dan pin T2IN dihubungkan dengan rangakaian relay. Pin-pin MAX232 yang lain berdasarkan *datasheet.*

4.4.4 Rangkaian *Relay*

Rangkaian *relay* berfungsi sebagai pemilih hubungan antara mikrokontroler dengan PC atau antara mikrokontroler dengan modem *Power Line Carrier (PLC)* LM1893. Dalam perencanaan ini rangkaian *relay* tersusun dari transistor NPN C828, resistor, diode dan relay DPDT 5V. Rangkaian relay ditunjukkan dalam Gambar 4.7.



Gambar 4.7. Rangkaian *Relay*

Perhitungan dari rangkaian *relay* pada Gambar 4.6. terlebih dahulu harus dicari nilai R_B. Data yang diperlukan untuk mencari besar resistansi R_B adalah sebagai berikut:

Data transistor C828 yang diperoleh dari *datasheet* adalah:

- Besar pengukuran tahanan dalam *relay* ($R_{relay} = R_C$) = 87 Ω.
- V_{CE} saturasi = 0,14 volt.
- hfe minimum = 130.
- hfe maximum = 520.
- V_{BE} = 0,8 volt.

Dengan resistansi *relay* sebesar 87 Ω, tegangan catu sebesar 5 V, dan V_{CE} saturasi sebesar 0,14 V maka dengan menggunakan Persamaan (4-5) besar arus I_{relay} adalah :

$$I_{relay} = \frac{V_{CC} - V_{CE}}{R_{relay}} \qquad (4\text{-}5)$$

$$I_{relay} = \frac{5V - 0{,}14V}{87\Omega} = 55{,}86 \text{ mA}$$

Dalam perancangan digunakan *hfe* sebesar 230 agar transistor berkerja optimal. Dengan menggunakan Persamaan (4-6) maka besar arus basis :

$$I_B = \frac{I_{relay}}{hfe} \qquad (4\text{-}6)$$

$$I_B = \frac{55{,}86mA}{230} = 0{,}24 \text{ mA}$$

Jika V_{bb} adalah tegangan keluaran dari mikrokontroler saat logika tinggi yaitu sebesar 4,2V dan V_{BE} = 0,8 V maka dengan menggunakan Persamaan (4-7) besar resistansi R_b adalah :

$$R_B = \frac{V_{BB} - V_{BE}}{I_B} \qquad (4\text{-}7)$$

$$R_B = \frac{4{,}2V - 0{,}8V}{0{,}24mA} = 14{,}2\,k\Omega \cong 14\,k\Omega$$

Dengan resistansi R_B sebesar 14 kΩ, arus I_B menjadi 0,24 mA sehingga telah diperoleh arus I_B yang dibutuhkan.

4.5 Perancangan Protokol Komunikasi

Agar terjadi komunikasi antara loket pengisian dengan kWh meter pelanggan, maka diperlukan suatu protokol komunikasi. Protokol komunikasi dirancang sebagai berikut :

1. Proses komunikasi dimulai dengan *PC* mengirim data ke mikrokontroler yang terdiri dari nomor ID kWh meter yang dituju, kode perintah, nilai nominal kWh, dan validasi.
2. PC menunggu respon dari mikrokontroler berupa kode apakah mikrokontroler dapat menerima data dengan benar atau tidak. Jika dalam waktu tertentu tidak ada respon, maka *PC* menganggap terjadi *error* pada hubungan komunikasi dengan mikrontroler dan proses tidak dilanjutkan.
3. Jika data yang dikirim oleh *PC* tidak sama dengan data yang diterima oleh mikrokontroler yang ditandai dengan ketidaksamaan data validasi, maka mikrokontroler akan memberikan kode kepada *PC* yang menyatakan bahwa data yang diterima mikrokontroler salah dan *PC* akan mengirim data kembali. Jika dalam dua kali pengiriman, data yang diterima mikrokontroler masih tidak sama dengan data yang dikirim oleh *PC*, maka *PC* menganggap terjadi error pada hubungan komunikasi dengan mikrontroler dan proses tidak dilanjutkan.
4. Jika data dapat diterima oleh mikrokontroler dengan benar, maka mikrokontroler mengirim data yang telah dikirim dari *PC* ke unit penghitung pemakaian energi listrik ditambah dengan header dan nomor ID loket. Saat ini *PC* dalam kondisi menunggu pengiriman data dari mikrokontroler berupa hasil proses komunikasi dengan unit penghitung pemakaian energi listrik.

5. Mikrokontroler menunggu jawaban dari unit penghitung pemakaian energi listrik yang berupa kode apakah data sudah diterima oleh unit penghitung pemakaian energi listrik dengan benar atau tidak serta nilai sisa nominal kWh yang tersimpan di unit penghitung pemakaian energi listrik. Jika dalam waktu tertentu tidak ada jawaban, maka mikrokontroler menganggap terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik. Dalam kondisi ini, mikrokontroler akan mengirim data ke *PC* berupa kode yang menyatakan bahwa terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik dan proses tidak dilanjutkan.
6. Jika data yang dikirm oleh mikrokontroler tidak sama dengan data yang diterima oleh unit penghitung pemakaian energi listrik ditandai dengan ketidaksamaan validasi, maka unit penghitung pemakaian energi listrik akan mengirim kode ke mikrokontroler yang menyatakan bahwa data yang diterima unit penghitung pemakaian energi listrik salah dan mikrokontroler akan mengirim data kembali. Jika dalam dua kali pengiriman, data yang dikirim oleh mikrokontroler masih tidak sama dengan data yang diterima oleh unit penghitung pemakaian energi listrik, maka mikrokontroler menganggap terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik. Dalam kondisi ini, mikrokontroler akan mengirim data ke *PC* berupa kode yang menyatakan bahwa terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik dan proses tidak dilanjutkan.
7. Jika data sudah dapat diterima oleh unit penghitung pemakaian energi listrik dengan benar, maka unit penghitung pemakaian energi listrik akan mengirim data sisa nominal kWh yang tersimpan ke mikrokontroler. Jika data yang diterima mikrokontroler tidak sama dengan data yang dikirim oleh unit penghitung pemakaian energi listrik ditandai dengan ketidaksamaan validasi, maka mikrokontroler akan mengirim kode ke unit penghitung pemakaian energi listrik yang menyatakan bahwa data yang diterima mikrokontroler salah dan unit penghitung pemakaian energi listrik akan mengirim data kembali. Jika dalam dua kali pengiriman, data yang diterima mikrokontroler masih tidak sama dengan data yang dikirim oleh unit penghitung pemakaian energi listrik, maka mikrokontroler menganggap terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik. Dalam kondisi ini, mikrokontroler akan mengirim data ke

PC berupa kode yang menyatakan bahwa terjadi *error* pada hubungan komunikasi dengan unit penghitung pemakaian energi listrik dan proses tidak dilanjutkan.

8. Jika mikrokontroler sudah menerima data dari unit penghitung pemakaian energi listrik dengan benar, maka mikrokontroler mengirim data yang diterima dari unit penghitung pemakaian energi listrik ke *PC* untuk disimpan ke dalam database. Data terdiri dari nomor ID unit penghitung pemakaian energi listrik yang dituju, kode-kode apakah terjadi error atau tidak, dan sisa nominal kWh yang tersimpan di unit penghitung pemakaian energi listrik dan proses telah selesai.

Gambar 4.8, 4.9, dan 4.10 menunjukkan format data yang digunakan saat melakukan komunikasi, baik antara *PC* dengan mikrokontroler maupun antara mikrokontroler dengan unti penghitung pemakaian energi listrik. Dengan format data tersebut, maka jumlah rumah yang bisa ditangani adalah 255 rumah dan nominal kWh maksimal yang bisa dibeli adalah 16777,215 kWh

Tujuan	Kode Perintah	Atas	Tengah	Bawah	Validasi

Gambar 4.8. Format data dari *PC* ke mikrokontroler

keterangan :

Tujuan = 8 bit alamat tujuan pengiriman data

Kode Perintah = 8 bit kode perintah untuk unit penghitung pemakaian energi listrik

Atas, Tengah, Bawah = 24 bit nominal nominal kWh

Validasi = 8 bit penanda kebenaran pengiriman data

3 byte Header	Kode Perintah	Tujuan	Sumber	Atas	Tengah	Bawah	Validasi

Gambar 4.9. Format data antara mikrokontroler dengan unit penghitung pemakaian energi listrik

keterangan :

Kode Perintah = 8 bit kode perintah untuk unit penghitung pemakaian energi listrik

Tujuan = 8 bit nomor ID penerima data

Sumber = 8 bit nomor ID pengirim data

Atas, Tengah, Bawah = 24 bit nominal nominal kWh

Validasi = 8 bit penanda kebenaran pengiriman data

1 byte Header	'I'/'e'	Sumber	'n'/'e'	Nominal kWh

Gambar 4.10. Format data dari mikrokontroler ke *PC*

keterangan :

'i'/'e' = Sebagai kode apakah terjadi *error* atau tidak di unit sentral data

Sumber = 8 bit nomor ID kWh meter yang memberi konfirmasi

'n/e' = Sebagai kode apakah terjadi *error* atau tidak di unit penghitung pemakaian energi listrik

Nominal kWh = Nominal kWh dalam bentuk string

4.6 Perancangan Perangkat Lunak

Pengatur komunikasi data pada alat ini dikendalikan oleh perangkat lunak dari mikrokontroler ATMEGA8515 sedangkan pengolahan dan penyimpanan data dilakukan oleh perangkat lunak PC. Perangkat lunak ini tersusun dari instruksi-intruksi yang membentuk sebuah listing program (*source code*).

Program dibuat secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu agar memudahkan dalam pembuatan dan pencarian kesalahan serta pengujian program sehingga dapat bekerja dengan baik. Perangkat lunak mikrokontroler dibuat menggunakan *Code Vision AVR* buatan HP *info tech*. Bahasa pemrograman yang digunakan adalah bahasa C. Sedangkan perangkat lunak PC menggunakan bahasa pemrograman Delphi 7.

Perancangan perangkat lunak mikrokontroler ditunjukan oleh diagram alir pada Gambar 4.11. Perangkat lunak PC terdiri dari 2 bagian yaitu perangkat lunak yang berfungsi untuk meminta data sisa nominal kWh pada unit penghitung pemakaian daya listrik dan perangkat lunak yang berfungsi untuk memberi data nominal kWh kepada unit penghitung pemakaian daya listrik. Perancangan perangkat lunak PC masing-masing ditunjukan oleh diagram alir pada Gambar 4.12 dan 4.13.

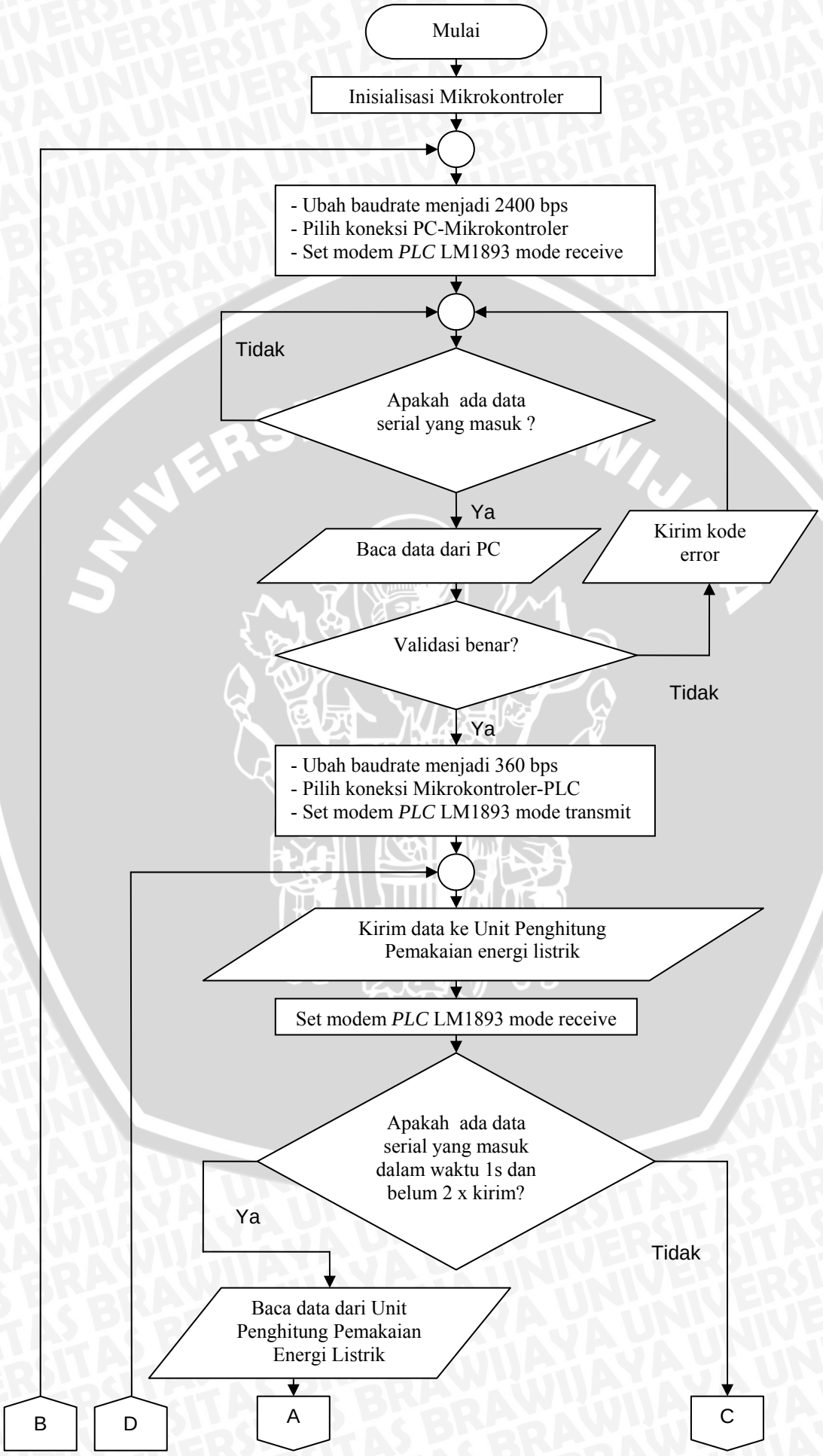
4.7 Perancangan Database

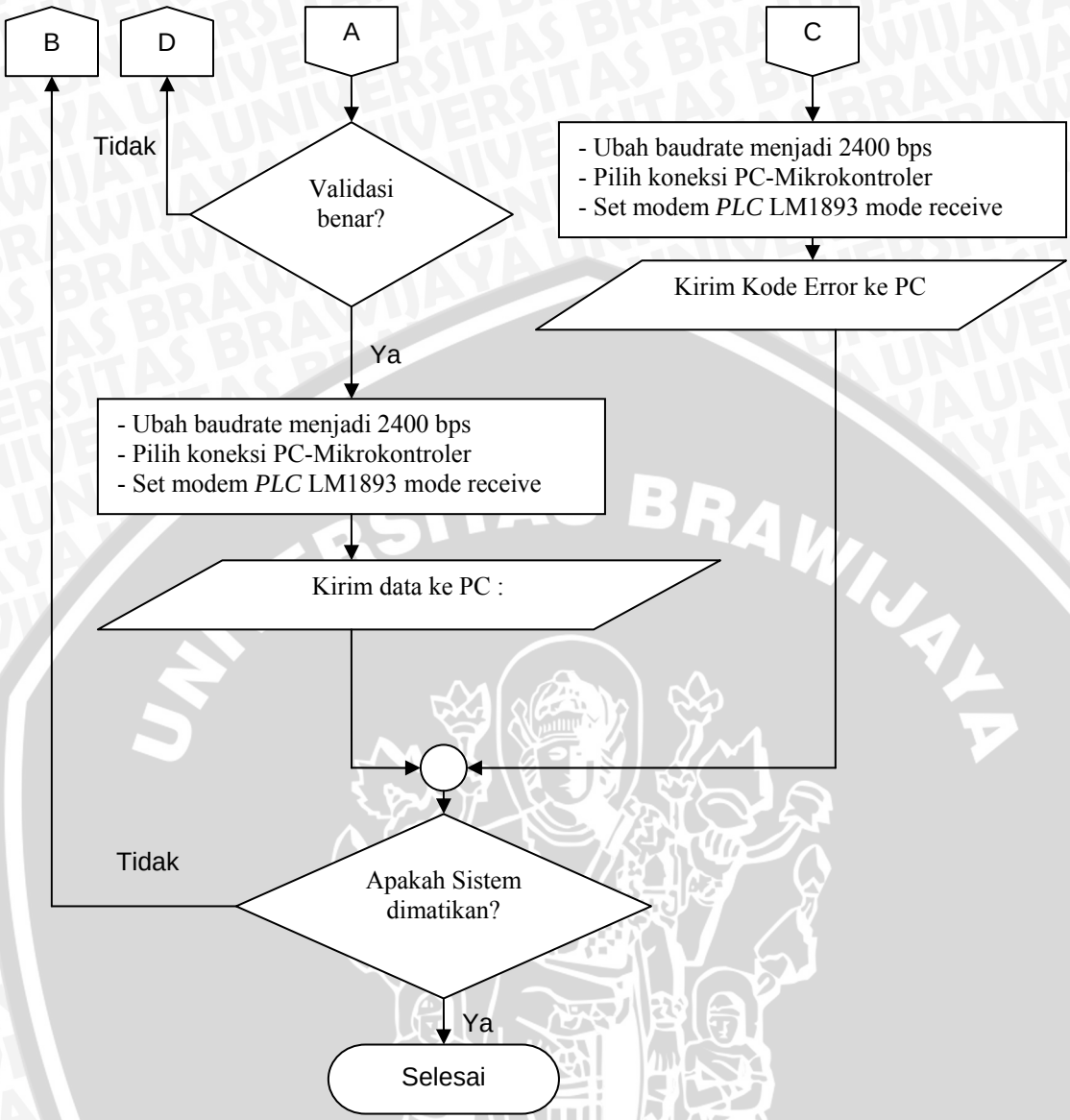
Database berfungsi untuk menyimpan sisa nominal kWh setiap pelanggan pada waktu tertentu. Database dibuat dalam 4 tabel yaitu tabel identitas, tabel harga, tabel sisa dan tabel pengiriman_kWh. Tabel identitas berisi biodata pelanggan dan golongan dayanya. Tabel harga berisi daftar harga untuk setiap golongan daya. Tabel sisa berisi sisa nominal kWh setiap pelanggan pada waktu tertentu saat dilakukan pengiriman dan pemantauan nominal kWh. Tabel pengiriman_kWh berisi data pengiriman kWh.

Database yang dibuat menggunakan Microsoft Access 2003. Tipe data atribut-atribut setiap tabel dalam database ditunjukkan dalam Tabel 4.1 dan relasi setiap tabel dalam database ditunjukkan dalam Gambar 4.13.

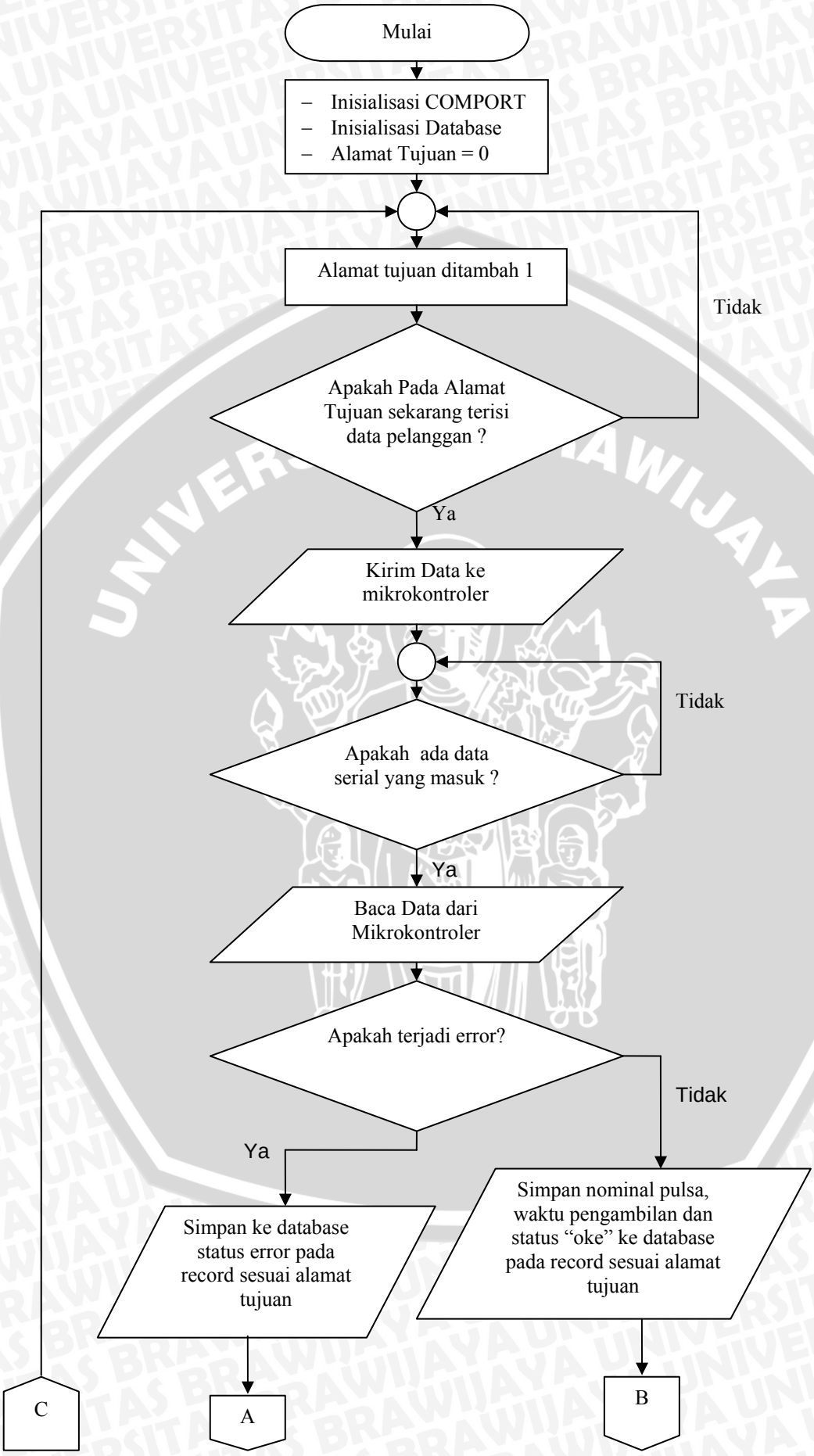
Tabel 4.1 Tipe data atribut-atribut setiap tabel dalam database

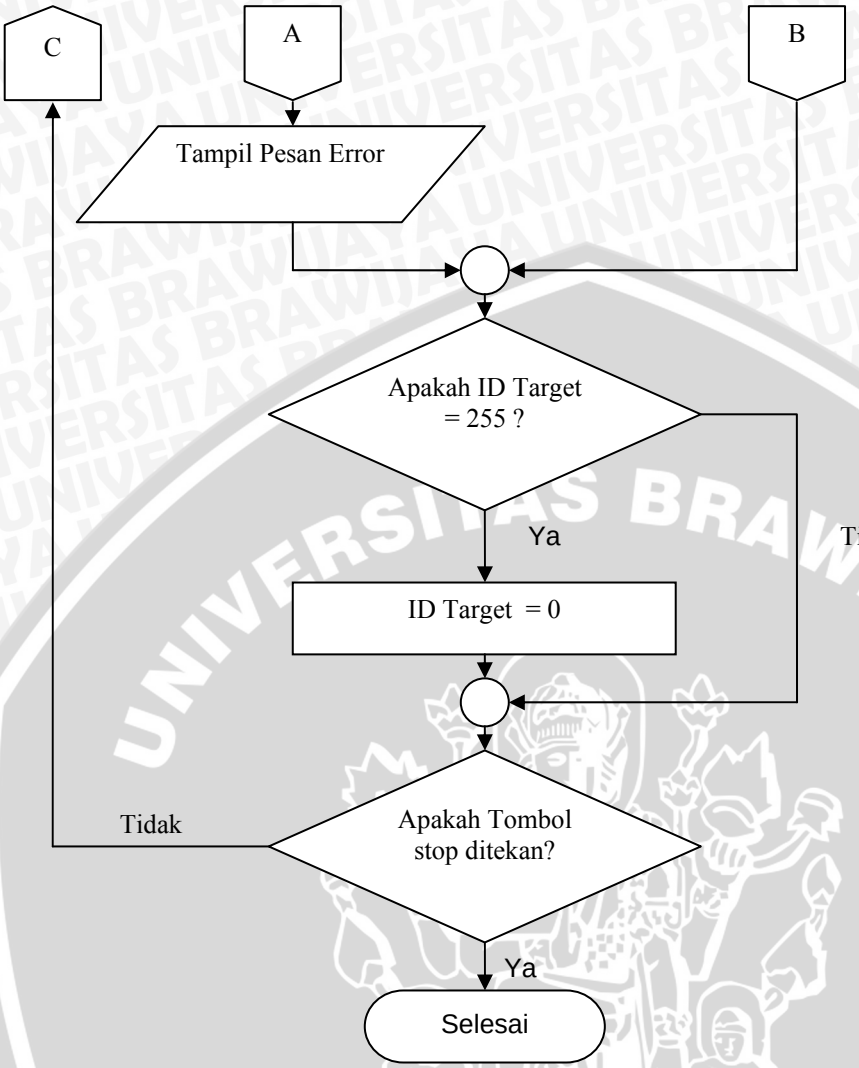
No	Nama Tabel	Nama Kolom	*Data Type*	*Field Size*	Keterangan
1.	Identitas	ID	Number	Byte	Primary key
		Nama	Text	25	
		Alamat	Text	66	
		Pekerjaan	Text	15	
		Golongan_Daya	Text	3	
2.	Harga	Golongan _Daya	Text	3	Primary key
		Harga_Per_kWh	Number	Integer	
3.	Sisa	ID	Number	Byte	Primary key
		Waktu _Data_Terambil	Date/Time	-	
		Sisa_kWh	Number	Decimal	
		Status	Text	5	
4.	Pengiriman_kWh	ID	Number	Byte	Primary key
		Sebelum	Date/Time	-	
		Besar_kWh	Number	Decimal	
		Sekarang	Date/Time	-	
		Besar_kWh_Sekarang	Number	Decimal	


Mulai
Inisialisasi Mikrokontroler
- Ubah baudrate menjadi 2400 bps
- Pilih koneksi PC-Mikrokontroler
- Set modem *PLC* LM1893 mode receive
Tidak
Apakah ada data serial yang masuk ?
Ya
Baca data dari PC
Kirim kode error
Validasi benar?
Tidak
Ya
- Ubah baudrate menjadi 360 bps
- Pilih koneksi Mikrokontroler-PLC
- Set modem *PLC* LM1893 mode transmit
Kirim data ke Unit Penghitung Pemakaian energi listrik
Set modem *PLC* LM1893 mode receive
Apakah ada data serial yang masuk dalam waktu 1s dan belum 2 x kirim?
Ya
Tidak
Baca data dari Unit Penghitung Pemakaian Energi Listrik
B
D
A
C

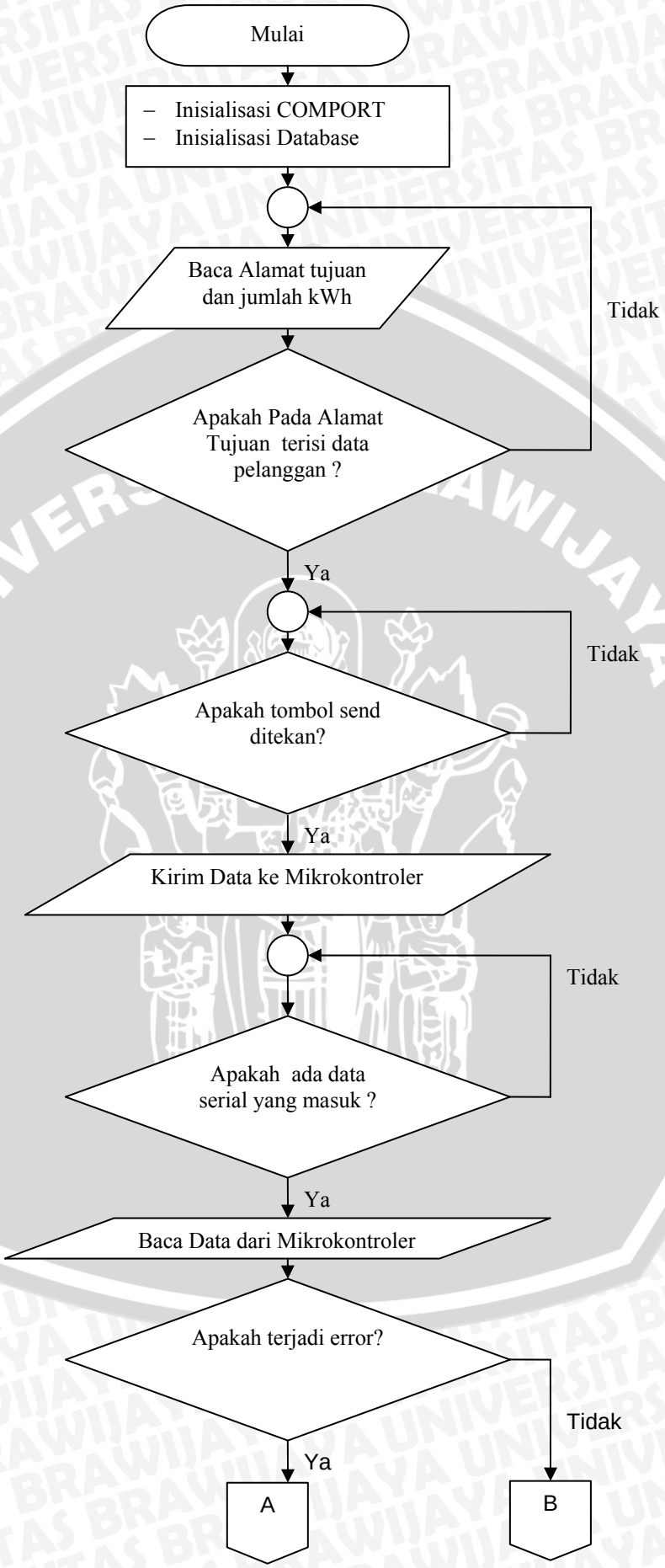


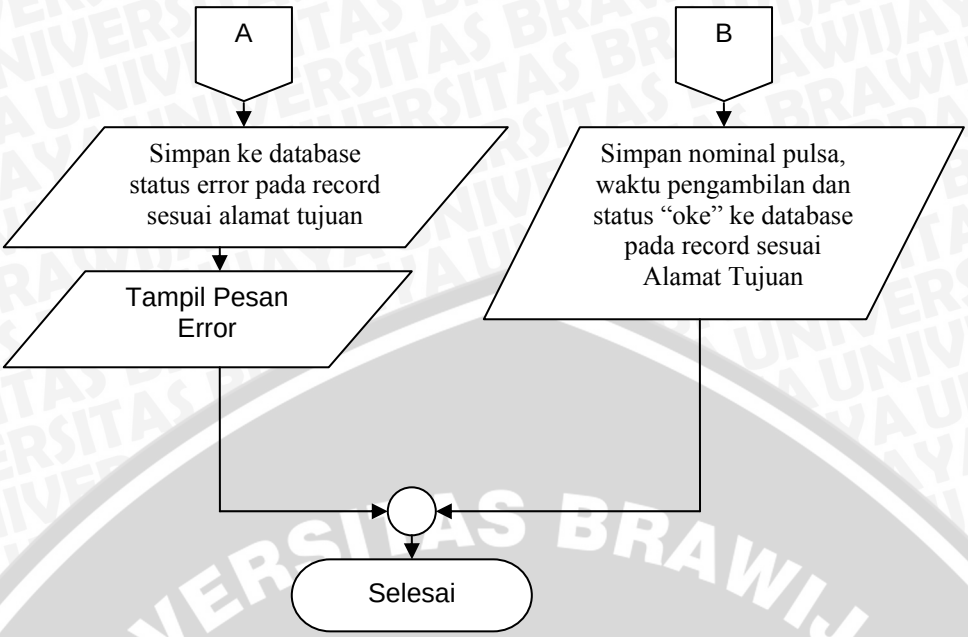
Gambar 4.11. Diagram alir perangkat lunak Mikrokontroler ATMEGA8515


Mulai
– Inisialisasi COMPORT
– Inisialisasi Database
– Alamat Tujuan = 0
Alamat tujuan ditambah 1
Tidak
Apakah Pada Alamat Tujuan sekarang terisi data pelanggan ?
Ya
Kirim Data ke mikrokontroler
Tidak
Apakah ada data serial yang masuk ?
Ya
Baca Data dari Mikrokontroler
Apakah terjadi error?
Tidak
Ya
Simpan nominal pulsa, waktu pengambilan dan status "oke" ke database pada record sesuai alamat tujuan
Simpan ke database status error pada record sesuai alamat tujuan
C
A
B

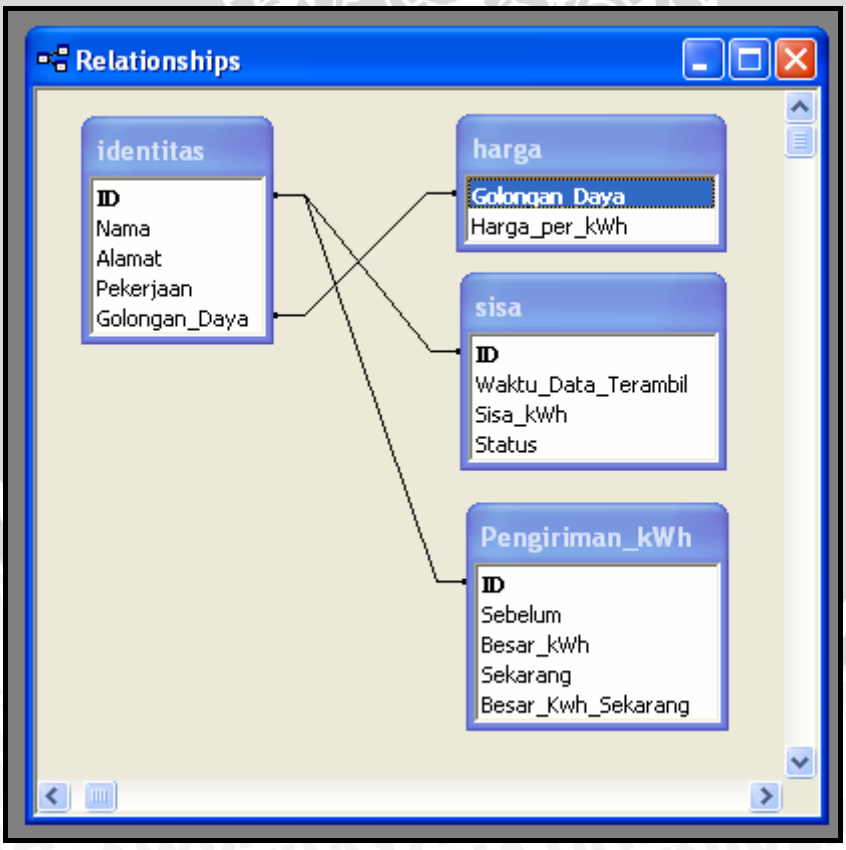


Gambar 4.12. Diagram alir perangkat lunak PC untuk meminta data sisa nominal kWh


Mulai
– Inisialisasi COMPORT
– Inisialisasi Database
Baca Alamat tujuan dan jumlah kWh
Apakah Pada Alamat Tujuan terisi data pelanggan ?
Tidak
Ya
Apakah tombol send ditekan?
Tidak
Ya
Kirim Data ke Mikrokontroler
Apakah ada data serial yang masuk ?
Tidak
Ya
Baca Data dari Mikrokontroler
Apakah terjadi error?
Tidak
Ya
A
B



Gambar 4.13. Diagram alir perangkat lunak PC untuk memberi data nominal kWh



Gambar 4.14 Relasi tabel database.

BAB V
PENGUJIAN DAN ANALISIS

Bab ini berisi penjelasan tentang pengujian dari alat yang telah dirancang untuk mengetahui apakah sistem dapat bekerja dengan baik sesuai dengan perencanaan. Pengujian dilakukan dengan memberikan perubahan pada masukan blok rangkaian dan mengamati keluaran dari blok rangkaian yang diuji tersebut. Data pengujian dianalisis untuk dijadikan acuan dalam mengambil kesimpulan.

5.1 Pengujian Rangkaian LM1893

5.1.1 Tujuan

Secara umum, tujuan pengujian rangkaian LM1893 adalah untuk mengetahui apakah rangkaian ini dapat bekerja sesuai dengan spesifikasi yang ditunjukkan oleh *datasheet*. Untuk mencapai tujuan tersebut, pengujian dilakukan dalam dua keadaan, yaitu pengujian saat mode *transmit* dan pengujian saat mode *receive*.

Tujuan pengujian rangkaian LM1893 saat mode *transmit* adalah untuk mengetahui bentuk sinyal keluaran, tegangan keluaran, dan frekuensi sinyal keluaran saat diberi masukan berupa logika 1 dan 0. Sedangkan tujuan pengujian rangkaian LM1893 saat mode *receive* adalah untuk mengetahui logika keluaran saat diberi masukan berupa sinyal sinusoida dengan frekuensi 122,5 kHz dan 127,7 kHz.

5.1.2 Peralatan Pengujian

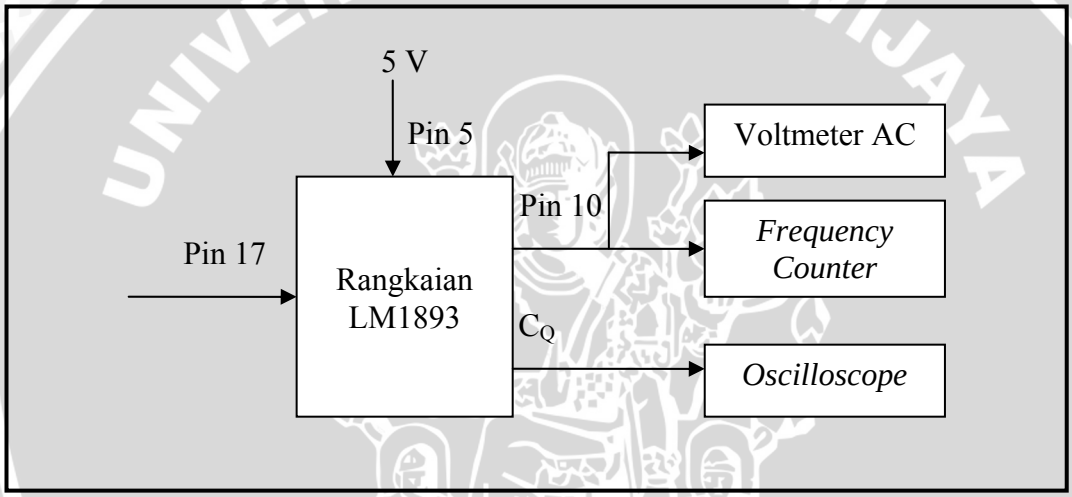
Peralatan yang digunakan terdiri dari :

1. *Oscilloscope*
2. Voltmeter
3. *Frequency Counter*
4. *Logic probe*
5. *Function Generator*
6. Rangkaian LM1893 dalam Gambar 4.4.
7. Catu daya 18 V dan 5 V

5.1.3 Prosedur Pengujian

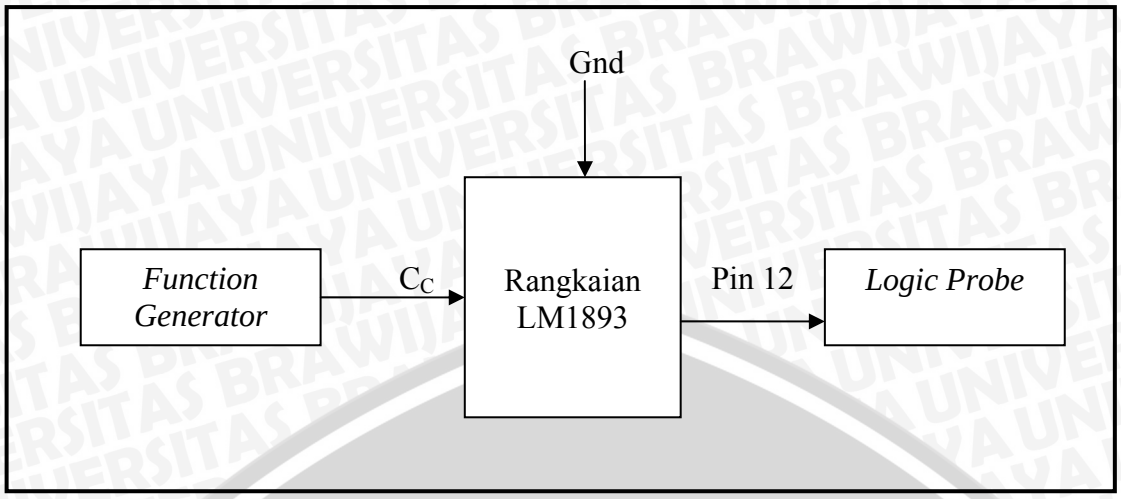
Langkah-langkah yang dilakukan untuk pengujian rangkaian LM1893 adalah sebagai berikut :

1. Rangkaian dikalibrasi dengan prosedur seperti yang ditunjukkan oleh *datasheet*, yaitu sebagai berikut :
   a. Pin 17 diberi logika rendah.
   b. Pin 5 diberi logika tinggi.
   c. *Frequency counter main probe* diletakkan pada pin 10 dan *frequency counter common probe* diletakkan pada pin *ground*.
   d. Variable resistor (R_0) pada pin 18 diputar sampai *frequency counter* menunjukkan angka frekuensi sebesar $1{,}022f_0$, yaitu sebesar 127.7 kHz
2. Untuk pengujian saat mode *transmit*, langkah-langkah yang dilakukan adalah sebagai berikut :
   a. Rangkaian disusun seperti Gambar 5.1



Gambar 5.1 Diagram blok pengujian Rangkaian LM1893 Mode Transmit

   b. Pin 5 diberi logika tinggi (1)
   c. Pin 17 diberi masukan berupa logika tinggi (1) dan logika rendah (0)
   d. Bentuk keluaran sinyal diamati melalui *oscilloscope*, besar tegangan keluaran diukur menggunakan voltmeter, dan besar frekuensi keluaran diukur menggunakan *frequency counter*.
3. Untuk pengujian saat mode *receive*, langkah-langkah yang dilakukan adalah sebagai berikut :
   a. Rangkaian disusun seperti Gambar 5.2



Gambar 5.2 Diagram blok pengujian Rangkaian LM1893 Mode Receive

b. Pin 5 diberi logika rendah (0)

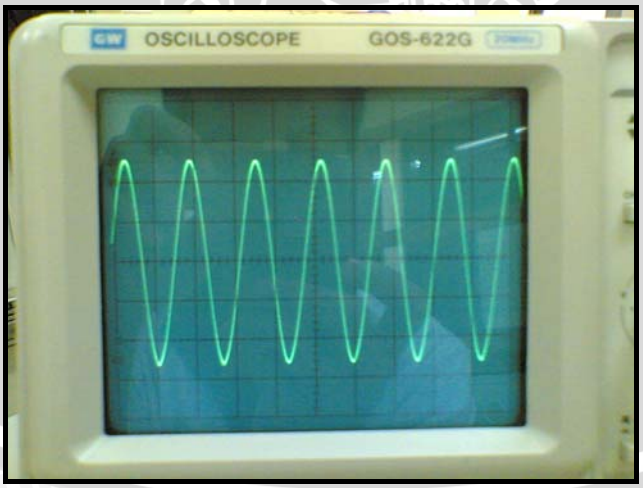
c. Rangkaian diberi masukan berupa sinyal sinuoida menggunakan *function generator* dengan tegangan maksimum sebesar 3 V berfrekuensi 122,5 kHz dan 127,7 kHz

d. Logika keluaran pada pin 12 diamati melalui *logic probe*

5.1.4 Data pengujian

Data pengujian rangkaian LM1893 saat mode *transmit* adalah sebagai berikut :

1. Bentuk sinyal keluaran pemancar LM1893 ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Bentuk sinyal keluaran pemancar LM1893

2. Tegangan keluaran yang diukur melalui pin 10 adalah 5,2 volt.
3. Frekuensi sinyal keluaran pemancar LM1893 dapat dilihat dalam tabel 5.1

Tabel 5.1 Data pegujian frekuensi keluaran rangkaian LM1893 mode *transmit*

No.	Logika Masukan Pada Pin 17	Frekuensi keluaran (kHz)
1.	1	122,4
2.	0	127,7

Data pengujian rangkaian LM1893 saat mode *receive* dapat dilihat dalam Tabel 5.2.

Tabel 5.2 Hasil pegujian rangkaian LM1893 mode *receive*

No.	Frekuensi Sinyal Masukan	Logika Keluaran Pada Pin 12
1.	122,5	*high*
2.	127,7	*low*

5.1.5 Analisis Data pengujian

Saat mode *transmit*, data pengujian menunjukkan bahwa sinyal keluarannya berupa sinyal sinusoida, tegangan sinyal kelurannya sebesar 5,2 V, dan frekuensi sinyal keluarannya adalah sebesar 127,7 kHz saat diberi masukan logika 0 dan 122,4 kHz saat diberi masukan logika 1. Sedangkan spesifikasi yang ditunjukkan oleh *datasheet* adalah sinyal keluarannya berupa sinyal sinusoida, tegangan sinyal keluaran minimum sebesar 4,0 V, frekuensi sinyal keluaran sebesar 127,7 kHz saat diberi masukan logika 0 dan 122,5 kHz saat diberi masukan logika 1. Toleransi penyimpangan frekuensi sinyal keluaran yang diijinkan adalah sebesar ± 20%. Oleh karena itu dapat disimpulkan bahwa saat mode transmit, rangkaian ini dapat bekerja sesuai dengan spesifikasi yang ditunjukkan oleh *datasheet*.

Saat mode *receive*, data pengujian menunjukkan bahwa logika keluaranya adalah *high* saat diberi masukan berupa sinyal sinusoida dengan frekuensi sebesar 122,5 kHz dan *low* saat diberi masukan berupa sinyal sinusoida dengan frekuensi sebesar 127,7 kHz. Data pengujian tersebut sama dengan spesifikasi yang ditunjukkan oleh *datasheet* sehingga dapat disimpulkan bahwa rangkaian ini dapat bekerja dengan baik saat mode *receive*.

5.2 Pengujian Komunikasi Antar Mikrokontroler Melalui Rangkaian LM1893

5.2.1 Tujuan

Tujuan pengujian komunikasi antar mikrokontroler melalui rangkaian LM1893 adalah untuk mengatahui baudrate-baudrate yang bisa digunakan agar tercapai keberhasilan komunikasi data antara mikrokontroler master dan mikrokontroler slave.

5.2.2 Peralatan Pengujian

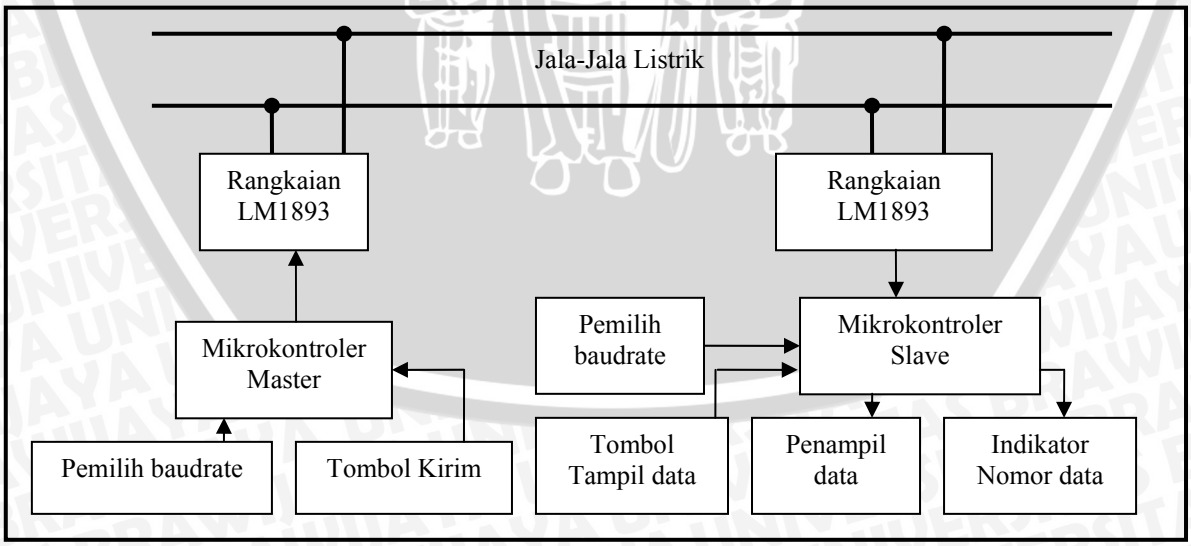
Peralatan yang digunakan terdiri dari :

1. Rangkaian LM1893 yang telah teruji
2. *Minimum system* mikrokontroler ATMEGA8515
3. Catu daya 18 V dan 5 V
4. Rangkaian indikator LED
5. Rangkaian *SW DIP-8*
6. Rangkaian *push button*
7. *Software* CVAVR
8. *Software* Ponyprog versi 2.06

5.2.3 Prosedur Pengujian

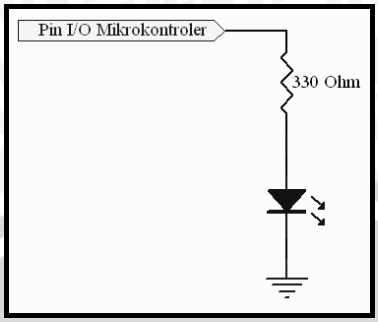
Langkah-langkah yang dilakukan untuk pengujian komunikasi antar mikrokontroler melalui rangkaian LM1893 adalah sebagai berikut :

1. Rangkaian disusun seperti Gambar 5.4.



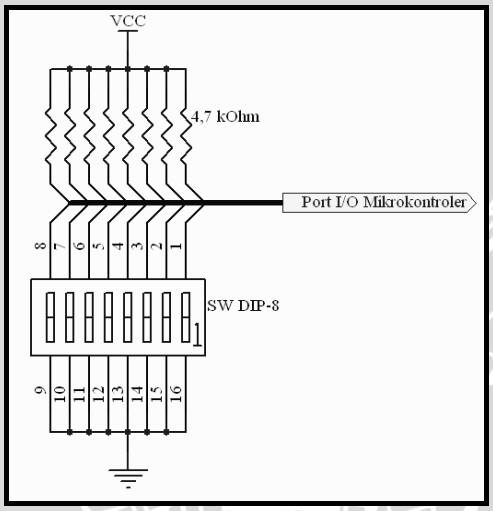
Gambar 5.4. Diagram blok pengujian komunikasi data antar mikrokontroler melalui rangkaian LM1893

Penampil data dan Indikator Nomor urut data masing-masing terdiri dari 8 buah rangkaian LED. Skema rangkaian LED yang digunakan ditunjukkan dalam Gambar 5.5.



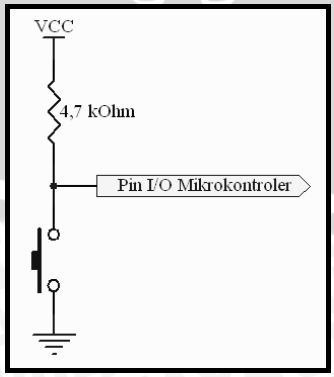
Gambar 5.5. Rangkaian LED

Pemilih baudrate menggunakan Rangkaian *SW DIP-8*. Rangkaian ini ditunjukkan dalam Gambar 5.6.



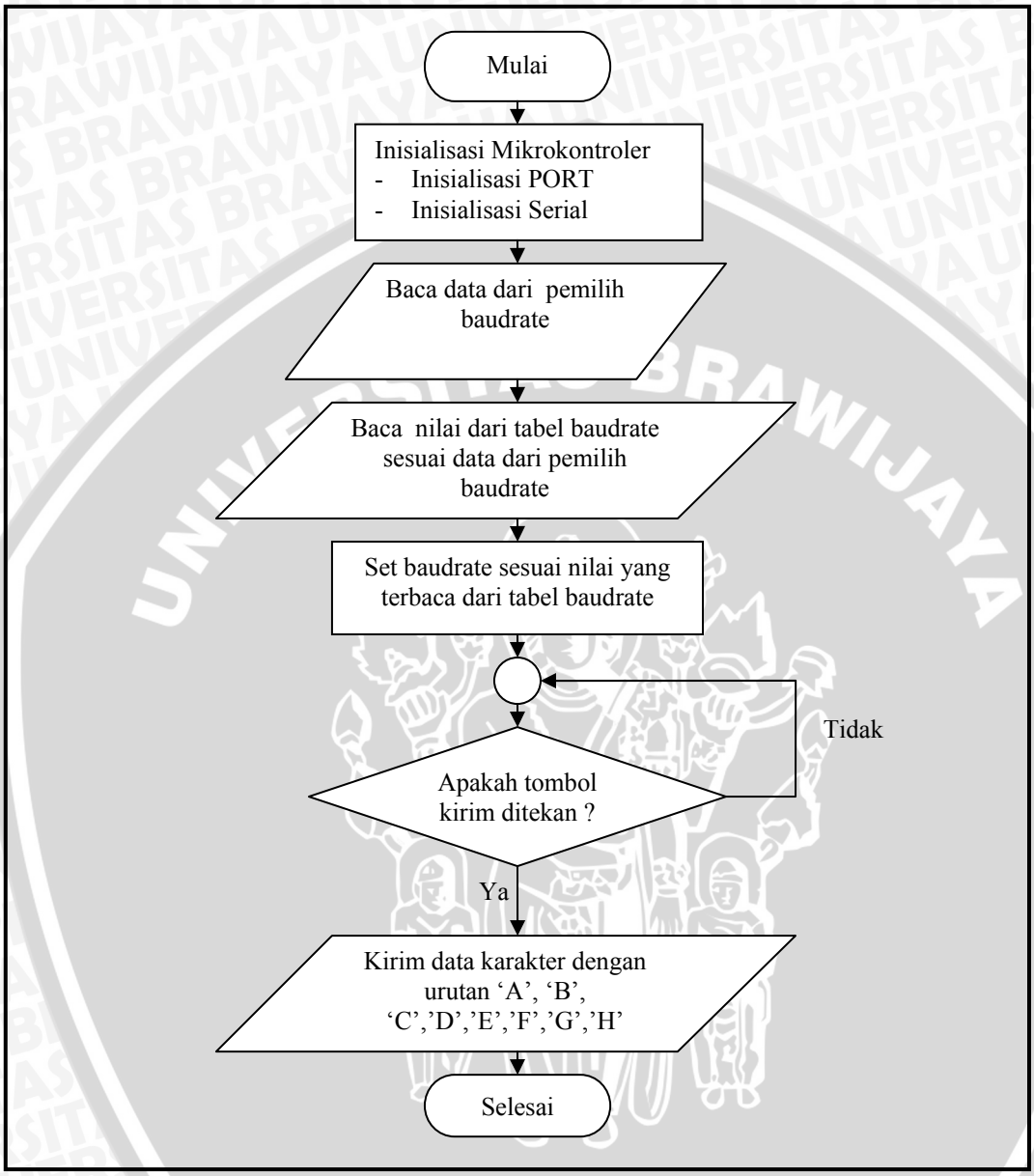
Gambar 5.6. Rangkaian *SW DIP-8*

Tombol kirim dan tombol tampil data menggunakan rangkaian *Push Button* seperti yang ditunjukkan dalam Gambar 5.7.



Gambar 5.7. Rangkaian *Push Button*

2. Program mikrokontroler master dibuat dengan algoritma seperti yang ditunjukkan oleh Gambar 5.8. Sedangkan program mikrokontroler slave dibuat dengan algoritma seperti yang ditunjukkan oleh Gambar 5.9.

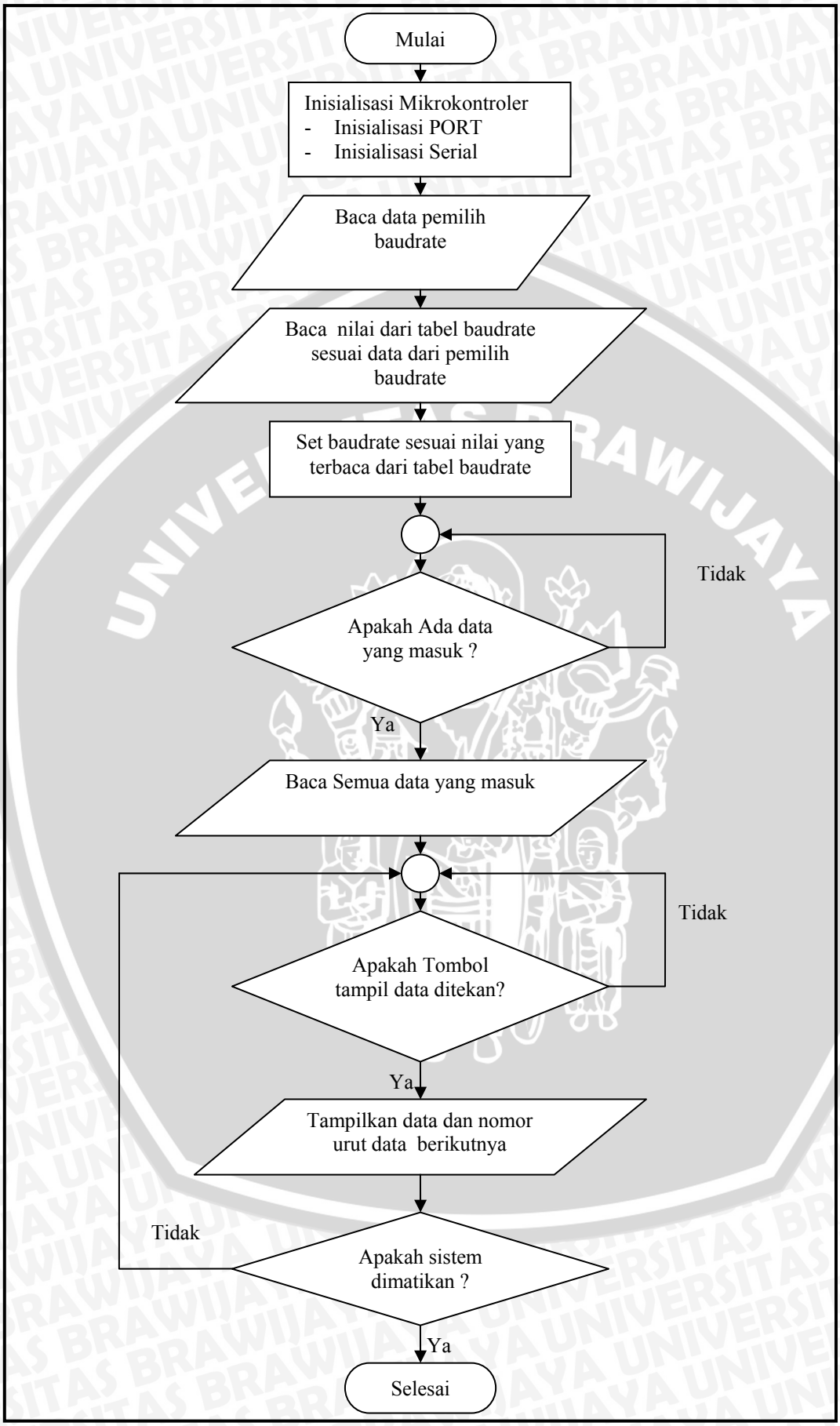


Gambar 5.8. Diagram alir program mikrokontroler master

Berdasarkan diagram alir dalam Gambar 5.8, data yang dikirim adalah seperti yang ditunjukkan dalam Tabel 5.3.

Tabel 5.3. Data yang dikirim oleh mikrokontroler master

Nomor Urut	Karakter terkirim	Data terkirim dalam Heksa
0	A	41
1	B	42
2	C	43
3	D	44
4	E	45
5	F	46
6	G	47
7	H	48



Gambar 5.9. Diagram alir program mikrokontroler slave

3. Baudrate pada sisi master dan sisi slave ditetapkan melalui pemilih baudrate.
4. Tombol kirim data pada sisi master ditekan
5. Tombol tampil data ditekan untuk melihat tampilan data dan nomor data pertama, ditekan lagi untuk melihat tampilan data dan nomor data kedua, dan seterusnya.
6. Data yang diterima dan indikator nomor urutannya diamati melalui nyala LED penampil data dan LED penampil nomor data

5.2.4 Data pengujian

Data pengujian komunikasi data antara mikrokontroler master dengan mikrokontroler slave melalui rangkaian LM1893 ditunjukkan oleh tabel 5.4. Baris dibawah *cell* yang bernama " Indikator Nomor Urut Data (Dalam Heksa Desimal)" adalah nyala led nomor urut dalam heksadesimal yang merepresentasikan urutan penerimaan data. *Cell-cell* dibawah nyala led nomor urut data adalah nyala LED data dalam heksadesimal yang merepresentasikan data yang diterima. Data yang diterima harus memiliki kesamaan nilai dan urutan dengan data yang dikirirm. Data yang dikirim dan urutannya ditunjukkan dalam Tabel 5.3.

Tabel 5.4. Data yang diterima oleh mikrokontroler slave (dalam heksa desimal)

No.	Baudrate (bps)	Indikator Nomor Urut Data (Dalam Heksa Desimal)							
		00	01	02	03	04	05	06	07
1.	60	41	42	43	44	45	46	47	48
2.	120	41	42	43	44	45	46	47	48
3.	180	41	42	43	44	45	46	47	48
4.	240	41	42	43	44	45	46	47	48
5.	300	41	42	43	44	45	46	47	48
6.	360	41	42	43	44	45	46	47	48
7.	480	41	42	43	44	45	46	47	48
8.	540	41	42	43	44	45	46	47	48
9.	600	41	42	43	44	45	46	47	48
10.	720	41	42	43	44	45	46	47	48
11.	900	41	42	43	44	45	46	47	48
12.	960	41	42	43	44	45	46	47	48
13.	1080	F0	3C	F0	3C	00	30	80	78
14.	1200	C0	00	80	C0	00	00	00	80

15.	1440	00	00	C0	00	00	00	C0	00
16.	1800	00	00	C0	00	00	00	C0	00
17.	1920	00	00	C0	00	00	00	C0	00
18.	2160	00	00	C0	00	00	00	C0	00
19.	2400	00	00	00	00	00	00	00	00

5.2.5 Analisis Data pengujian

Berdasarkan data yang ditunjukkan oleh Tabel 5.3 dan Tabel 5.4, kesamaan nilai dan urutan antara data yang dikirim dengan data yang diterima terjadi hanya sampai baudrate sebesar 960 bps. Sehingga dapat disimpulkan bahwa pemilihan baudrate sebesar 360 bps pada perancangan sudah cocok digunakan dalam sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data. Baudrate yang dipilih adalah baudrate yang tidak terlalu cepat dan tidak terlalu lambat supaya dapat dihasilkan kinerja sistem yang paling optimal.

5.3 Pengujian Rangkaian RS232

5.3.1 Tujuan

Tujuan pengujian rangkaian RS232 adalah untuk mengetahui besarnya level tegangan keluaran rangkaian antarmuka RS232 dengan masukan berupa level tegangan TTL dan RS232.

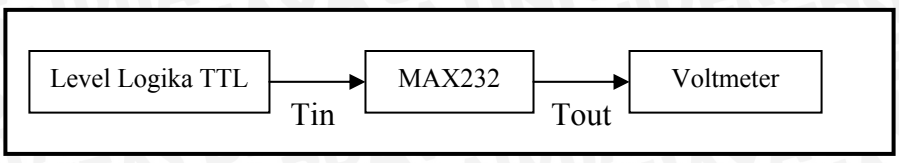
5.3.2 Peralatan Pengujian

Peralatan yang digunakan antara lain :

1. Voltmeter
2. IC MAX232 dan minimum sistemnya
3. Catu daya +5V, -15V dan +15V

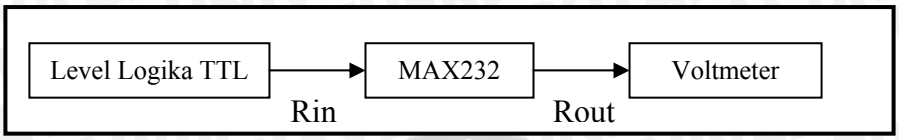
5.3.3 Prosedur Pengujian

1. Untuk pengujian dengan level tegangan masukan logika TTL, rangkaian disusun seperti dalam Gambar 5.10.



Gambar 5.10. Diagram blok pengujian dengan level tegangan masukan logika TTL

2. Untuk pengujian dengan level tegangan masukan logika serial, rangkaian disusun seperti dalam Gambar 5.11.



Gambar 5.11. Diagram blok pengujian dengan level tegangan masukan logika serial

3. Mengukur tegangan keluaran IC MAX232 dari masing-masing pengujian menggunakan Voltmeter.

5.3.4 Data pengujian

Data pengujian dengan masukan logika TTL dapat dilihat dalam Tabel 5.5.

Tabel 5.5 Data pengujian dengan masukan logika TTL

Tegangan Masukan (volt)	Tegangan Keluaran (volt)
0	7.94
5	-8.51

Data pengujian dengan masukan logika Serial dapat dilihat dalam Tabel 5.6

Tabel 5.6 Data pengujian dengan masukan logika TTL

Tegangan Masukan (volt)	Tegangan Keluaran (volt)
+15	0
-15	4.73

5.3.5 Analisis Data pengujian

Data pengujian pada Tabel 5.5 menunjukkan bahwa saat diberi tegangan masukan sebesar 0 V menghasilkan keluaran dengan nilai yang berada dalam *range* 3 – 15 V, dan saat diberi tegangan masukan sebesar 5 V menghasilkan tegangan keluaran dengan nilai yang berada dalam *range* (-3) – (-15) V. Data pengujian pada TAbel 5.6 menunjukkan bahwa saat diberi tegangan masukan sebesar 15 V menghasilkan tegangan keluaran berlogika rendah, dan saat diberi masukan sebesar -15 V menghasilkan tegangan keluaran berlogika tinggi. Sehingga dapat disimpulkan bahwa rangkaian hasil perancangan mampu mengubah level tegangan TTL menjadi level tegangan RS232 serta mampu mengubah level tegangan RS232 menjadi level tegangan TTL.

5.4 Pengujian Komunikasi Antara Mikrokontroler dengan *Personal Computer* Melalui Rangkaian RS232

5.4.1 Tujuan

Tujuan pengujian komunikasi antara mikrokontroler dengan *Personal* Computer adalah untuk mengetahui keberhasilan komunikasi data bertipe karakter antara mikrokontroler dengan komputer melalui rangkaian RS232.

5.4.2 Peralatan Pengujian

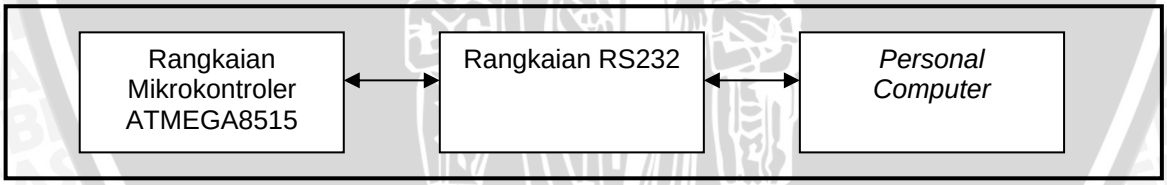
Peralatan yang digunakan terdiri dari :

1. *Personal Computer*
2. *Minimum System* Mikrokontroler ATMEGA8515
3. Rangkaian RS232
4. Software *Hyper Terminal*
5. Software CVAVR
6. Software Ponyprog versi 2.06

5.4.3 Prosedur Pengujian

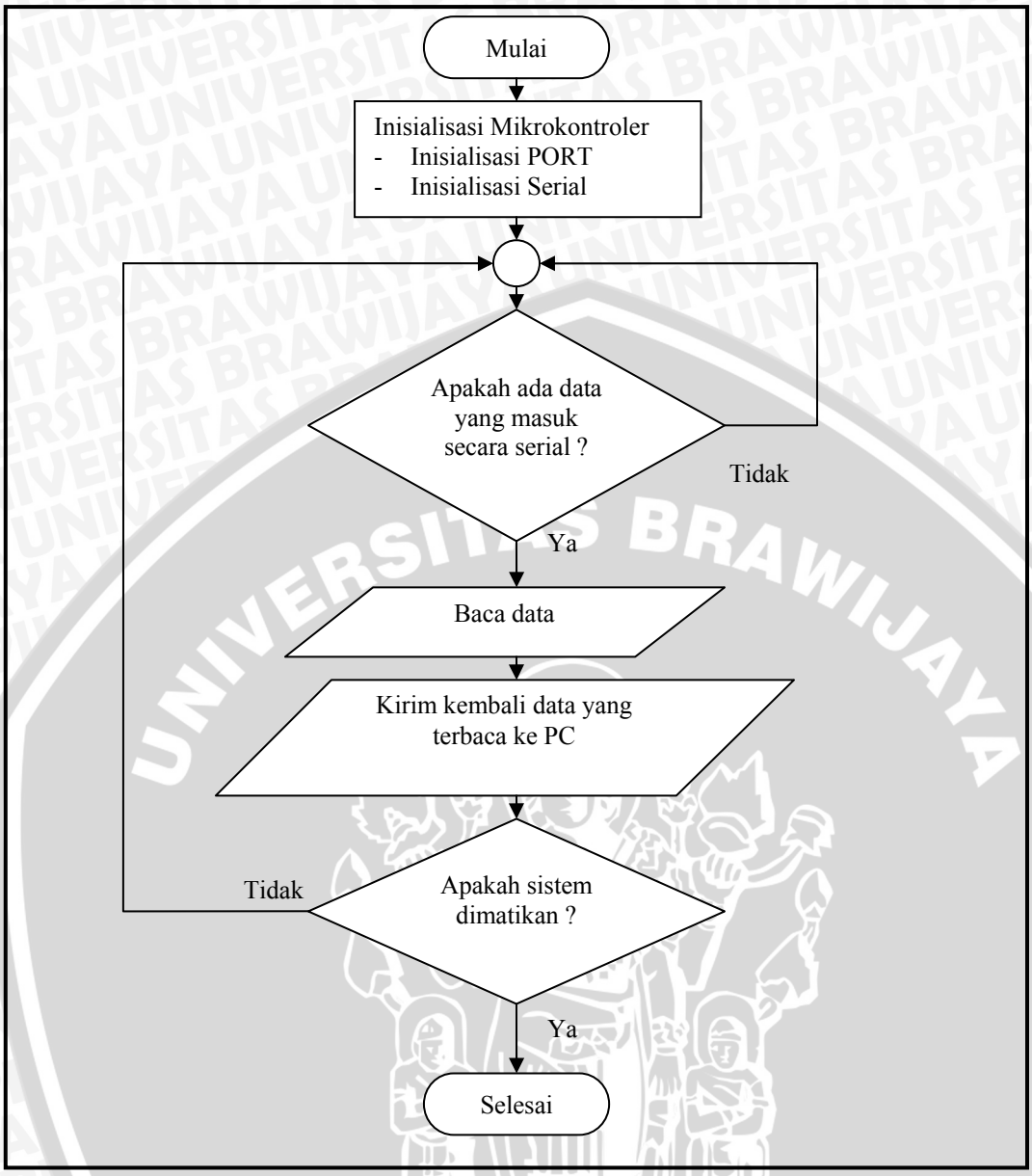
Langkah-langkah yang dilakukan untuk pengujian komunikasi antara mikrokontroler dengan *personal computer* adalah sebagai berikut :

1. Rangkaian disusun seperti Gambar 5.12.



Gambar 5.12. Diagram blok pengujian komunikasi antara mikrokontroler dengan *personal computer*

2. Program mikrokontroler dibuat dengan algoritma seperti yang ditunjukkan oleh Gambar 5.13.

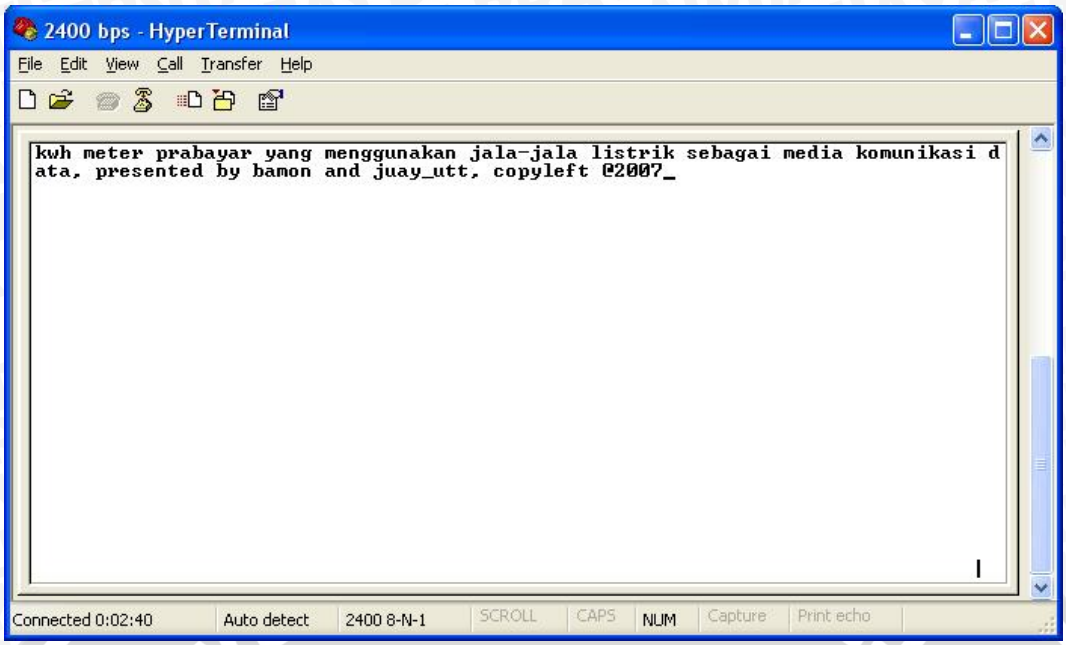


Gambar 5.13. Diagram alir pengujian komunikasi antara mikrokontroler dengan *personal computer* melalui rangkaian RS232

3. Software *Hyper Terminal* diaktifkan dengan baudrate sebesar 2400 bps.
4. Karakter **"kwh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data, presented by bamon and juay_utt, copyleft @2007"** diketik melalui keyboard.
5. Hasilnya diamati melalui tampilan *hyper terminal*

5.4.4 Data pengujian

Data pengujian komunikasi antara mikrokontroler dengan *personal computer* melalui rangkaian RS232 ditunjukkan dalam Gambar 5.14.



Gambar 5.14. Data pengujian komunikasi antara mikrokontroler dengan *personal computer* melalui rangkaian RS232

5.4.5 Analisis Data pengujian

Dalam pengujian komunikasi antara mikrokontroler dengan *personal* computer, karakter yang diketik melalui *keyboard* dikirim ke mikrokontroler secara serial melalui rangkaian RS232. Kemudian karakter yang diterima oleh mikrokontroler dikirim kembali ke *personal computer* dan karakter yang diterima ditampilkan oleh *software hyper terminal*. Gambar 5.14. menunjukkan bahwa karakter yang diketik melalui *keyboard* sama dengan karakter yang ditampilkan oleh *software hyper terminal*, sehingga dapat disimpulkan bahwa komunikasi antara mikrokontroler dan personal komputer dapat terjadi dengan baik.

5.5 Pengujian Keseluruhan Sistem kWh Meter Prabayar

5.5.1 Pengujian Pengiriman Nominal Kwh

5.5.1.1 Tujuan

Pengujian pengiriman nominal kWh bertujuan untuk mengetahui keberhasilan sistem dalam mengirim nominal kWh ke ID yang dituju.

5.5.1.2 Peralatan Pengujian

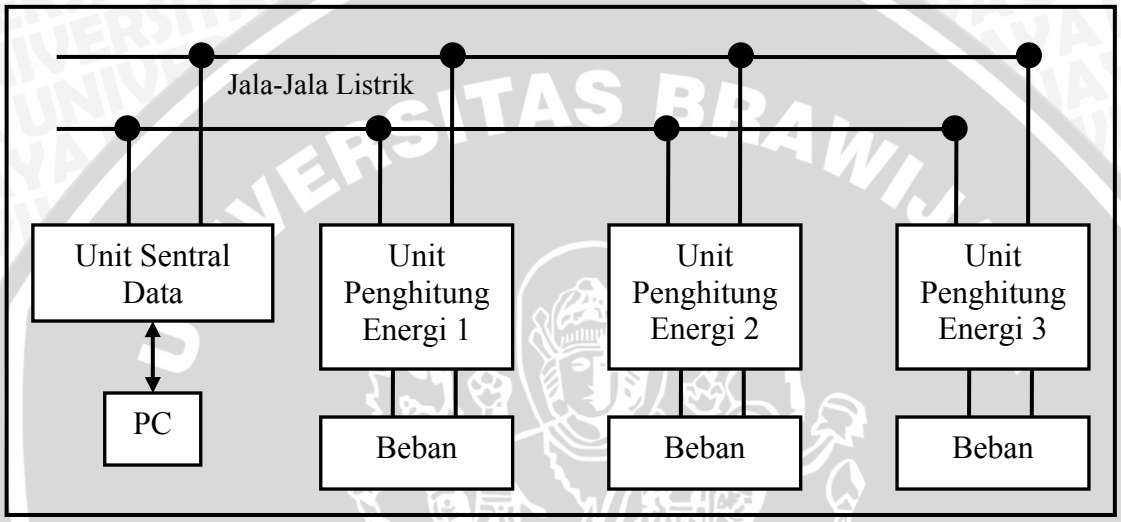
Peralatan yang digunakan terdiri dari :

1. Rangkaian unit penghitung pemakaian energi listrik
2. Rangkaian unit sentral data

5.5.1.3 Prosedur Pengujian

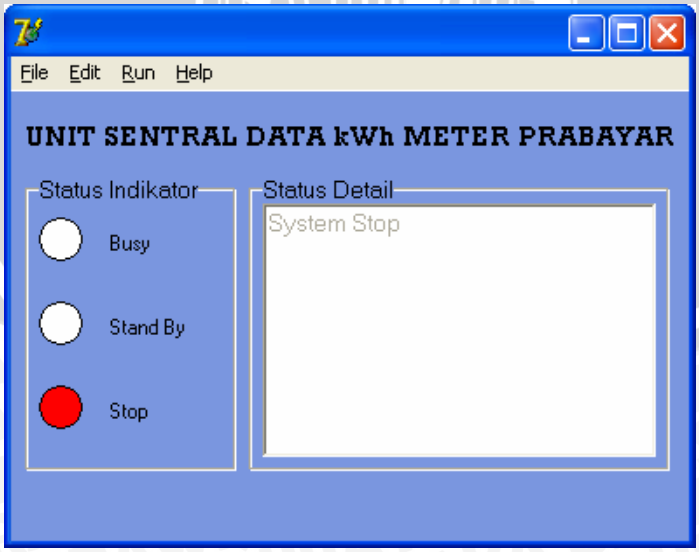
Prosedur pengujian dalam pengiriman nominal kWh adalah sebagai berikut:

1. Sistem disusun seperti yang ditunjukkan oleh diagram blok dalam Gambar 5.15.



Gambar 5.15. Diagram blok pengujian sistem keseluruhan

2. Program *PC* pada unit sentral data diaktifkan sehingga muncul tampilan seperti yang ditunjukkan dalam Gambar 5.16.



Gambar 5.16. Tampilan utama program *PC* pada unit sentral data

3. Identitas pelanggan dimasukkan dengan ID 1, 2 dan 3 dengan langkah-langkah sebagai berikut :
   a. Pada tampilan utama, di-klik menu ”Edit > Edit Pelanggan” sehingga muncul tampilan seperti yang ditunjukkan dalam Gambar 5.17.

Gambar 5.17. Tampilan untuk memasukkan atau mengubah identitas pelanggan

   b. Identitas pelanggan dimasukkan sebagaimana yang ditunjukkan oleh Gambar 5.18.

Gambar 5.18. Tampilan untuk memasukkan atau mengubah identitas pelanggan setelah data pelanggan dimasukkan

   c. Tombol Save di-klik untuk menyimpan data pelanggan yang sudah diisikan
   d. Tombol Next di-klik untuk mengisi data pelanggan pada ID berikutnya.
4. Mengosongkan data pulsa di EEPROM pada semua Unit Penghitung Energi Listrik dengan langkah-langkah sebagai berikut :

a. Pada tampilan utama, di-klik menu ”Run > Edit kWh” sehingga muncul tampilan seperti yang ditunjukkan dalam Gambar 5.19.

Gambar 5.19. Tampilan untuk mengganti nominal kwh yang tersimpan di EEPROM unit penghitung pemakaian energi listrik

b. ID unit penghitung pemakaian energi listrik dan nominal kWh dimasukkan kedalam *form* pada *window* seperti Gambar 5.19 sehingga tampilan menjadi seperti yang ditunjukkan dalam Gambar 5.20.

Gambar 5.20. Tampilan pengganti nominal kWh yang telah terisi ID unit penghitung pemakaian energi listrik tujuan dan nominal kwh yang baru.

c. Tombol ”Send” di-klik

d. Poin a, b, dan c diulangi untuk pengososongan EEPROM pada ID-ID berikutnya.

5. Pusat data mengirim data nominal kWh sebanyak dua kali ke masing-masing unit penghitung pemakaian energi listrik. Untuk pengiriman pertama, unit dengan ID ”1” diberi nominal kWh sebesar 1 kWh, ID ”2” diberi nominal kWh sebesar 2 kWh, dan ID ”3” diberi nominal kWh sebesar 0,06 kWh. Untuk pengiriman kedua, unit dengan ID ”1” diberi nominal kWh sebesar 3 kWh, ID ”2” diberi nominal kWh sebesar 1 kWh, dan ID ”3” diberi nominal kWh sebesar 0,06 kWh. Prosedur untuk

mengirimkan kWh ke unit penghitung pemakaian energi listrik adalah sebagai berikut :

a. Pada tampilan utama, di-klik ”Run > Kirim kWh” sehingga muncul tampilan seperti yang ditunjukkan dalam Gambar 5.21.

Gambar 5.21. Tampilan untuk mengirim nominal kWh ke unit penghitung pemakaian energi listrik

b. ID tujuan dan jumlah uang dimasukkan kedalam *form* yang tersedia pada *window* yang ditunjukkan oleh Gambar 5.21. Untuk mengetahui nilai jumlah uang yang harus dimasukkan supaya dapat mengirim nominal kWh sesuai dengan yang diinginkan, maka perlu dilihat daftar harga dan golongan daya dari ID tujuan. Golongan daya ID tujuan dapat dilihat melalui *window* seperti yang ditunjukkan dalam Gambar 5.18. Sedangkan daftar harga dapat dilihat dengan cara mengklik menu ”Edit > Edit Tarif” sehingga muncul tampilan seperti yang ditunjukkan dalam Gambar 5.22.

Gambar 5.22. Tampilan *window* daftar harga energi listrik

c. Tombol ”Send” pada *window* yang seperti ditunjukkan dalam Gambar 5.21 di-klik.

6. Mencatat nominal kWh yang ditampilkan oleh LCD pada masing-masing Unit Penghitung Pemakaian Energi Lisrik.

5.5.1.4 Data pengujian

Data pengujian pengiriman pulsa listrik yang telah dilakukan ditunjukkan dalam Tabel 5.7. yang merupakan pengiriman nominal kWh yang pertama dimana nominal kWh awal yang dimiliki masing-masing ID adalah 0 kWh.

Tabel 5.7 Hasil pengiriman data pertama.

No	ID Pelanggan	Nominal kWh yang dikirim dari pusat	Nominal kWh yang diterima pelanggan	Nominal kWh tercatat di database pusat
1.	1	1 kWh	1 kWh	1 kWh
2.	2	2 kWh	2 kWh	2 kWh
3.	3	0,06 kWh	0,06 kWh	0,06 kWh

Tabel 5.8. menunjukkan hasil pengiriman data yang kedua dimana pulsa yang tercatat pada pelanggan merupakan hasil penjumlahan antara nominal kWh yang pertama dengan nominal kWh yang baru.

Tabel 5.8. Hasil pengiriman data kedua.

No	ID Pelanggan	Nominal kWh yang dikirim dari pusat	Nominal kWh hasil akumulasi	Nominal kWh tercatat di database pusat
1.	1	3 kWh	4 kWh	4 kWh
2.	2	1 kWh	3 kWh	3 kWh
3.	3	0,06 kWh	0,12 kWh	0,12 kWh

5.5.1.5 Analisis Data pengujian

Dalam pengujian pengiriman nominal kWh, unit sentral data mengirim nominal kWh sebesar 1 kWh untuk ID pelanggan ”1”, 2 kWh untuk ID pelanggan ”2”, dan 0.06 kWh untuk ID pelanggan ”3”. Berdasarkan data pengujian pada Tabel 5.7, nominal kWh dapat dikirimkan ke ID pelanggan yang dituju dengan tepat. Kemudian unit sentral data mengirim nominal kWh lagi sebesar 3 kWh untuk ID pelanggan ”1”, 1 kWh untuk ID pelanggan ”2”, dan 0.06 kWh untuk ID pelanggan ”3”. Berdasarkan data pengujian pada Tabel 5.8, nominal kWh dapat dikirimkan ke ID pelanggan yang dituju dengan tepat sehingga nominal kWh yang disimpan oleh masing-masing unit penghitung

pemakaian energi listrik merupakan akumulasi antara pengiriman nominal kWh yang pertama dengan pengiriman nominal kWh yang kedua. Oleh karena itu dapat disimpulkan bahwa proses pengiriman nominal kWh dapat berjalan dengan baik.

5.5.2 Pengujian Pengurangan dan Pemantauan Nominal kWh

5.5.2.1 Tujuan

Pengujian pengiriman nominal kWh bertujuan untuk mengetahui keberhasilan unit penghitung pemakaian energi listrik dalam menghitung pemakaian energi listrik serta untuk mengetahui keberhasilan unit sentral data dalam memantau sisa nominal kWh pada setiap unit penghitung pemakaian energi listrik .

5.5.2.2 Peralatan Pengujian

Peralatan yang digunakan terdiri dari :

1. Rangkaian unit penghitung pemakaian energi listrik
2. Rangkaian unit sentral data
3. Beban 40W, 60W, dan 100W.

5.5.2.3 Prosedur Pengujian

Prosedur dalam pengujian pengurangan dan pemantauan nominal kWh adalah:

1. Sistem disusun seperti yang ditunjukkan oleh diagram blok dalam Gambar 5.15
2. Masing-masing Unit Penghitung Pemakaian Energi Listrik diberi beban yang berbeda yaitu:
   a. Unit 1 dengan ID 1 diberi beban 40 W.
   b. Unit 2 dengan ID 2 diberi beban 60 W.
   c. Unit 3 dengan ID 3 diberi beban 120 W.
3. Unit penghitung pemakaian energi listrik akan mengurangi nominal kWh yang sudah dikirim pada pengujian pengiriman nominal kWh sesuai dengan beban yang dipasang. Sisa nominal kWh yang terdapat pada unit penghitung pemakaian energi listrik dipantau oleh unit sentral data setiap 15 menit selama 1 jam.
4. Mencatat pulsa yang diperoleh Unit sentral data

5.5.2.4 Data pengujian

Data pengujian pengurangan dan pemantauan nominal kWh ditunjukkan Dalam Tabel 5.9, 5.10, 5.11, dan 5.12.

Tabel 5.9. Perhitungan dan pemantauan nominal kWh 15 menit pertama

No	ID Pelanggan	Nominal kWh setelah energi dihitung	Nominal kWh yang terpantau oleh pusat
1.	1	3,99 kWh	3,99 kWh
2.	2	2,99 kWh	2,99 kWh
3.	3	0,09 kWh	0,09 kWh

Tabel 5.10. Perhitungan dan pemantauan nominal kWh 15 menit kedua

No	ID Pelanggan	Nominal kWh setelah energi dihitung	Nominal kWh yang terpantau oleh pusat
1.	1	3,97 kWh	3,97 kWh
2.	2	2,98 kWh	2,98 kWh
3.	3	0,06 kWh	0,06 kWh

Tabel 5.11. Perhitungan dan pemantauan nominal kWh 15 menit ketiga.

No	ID Pelanggan	Nominal kWh setelah energi dihitung	Nominal kWh yang terpantau oleh pusat
1.	1	3,96 kWh	3,96 kWh
2.	2	2,97 kWh	2,97 kWh
3.	3	0,03 kWh	0,03 kWh

Tabel 5.12. Perhitungan dan pemantauan nominal kWh 15 menit keempat.

No	ID Pelanggan	Nominal kWh setelah energi dihitung	Nominal kWh yang terpantau oleh pusat
1.	1	3,94 kWh	3,94 kWh
2.	2	2,96 kWh	2,96 kWh
3.	3	0,00 kWh	0,00 kWh

5.5.5 Analisis Data pengujian

Untuk pemantauan pemakaian energi listrik, unit sentral data meminta unit penghitung pemakaian energi listrik untuk mengirimkan sisa nominal kWh yang tersimpan ke unit sentral data. Berdasarkan Tabel 5.9, 5.10, 5.11, dan 5.12, terdapat kesamaan antara sisa nominal kWh yang tesimpan di unit penghitung pemakaian daya listrik dengan nominal kWh yang terpantau oleh unit sentral data. Sehingga dapat disimpulkan bahwa proses pemantauan pemakaian energi listrik dapat berjalan dengan baik.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil pengujian tiap blok rangkaian dan pengujian sistem keseluruhan yang telah dilakukan pada bab V, dapat ditarik kesimpulan sebagai berikut :

1. Dari pengujian rangkaian LM1893 dapat diketahui bahwa rangkaian tersebut dapat bekerja dengan baik sesuai dengan spesifikasi yang ditunjukkan oleh *datasheet*.
2. Dari pengujian komunikasi antar mikrokontroler melalui rangkaian LM1893 dapat diketahui bahwa nilai baudrate terbesar yang bisa digunakan adalah 960 bps. Oleh karena itu dipilih baudrate sebesar 360 bps agar komunikasi dapat terjadi secara optimal.
3. Dari pengujian komunikasi antara mikrokontroler dengan *PC* dapat diketahui bahwa karakter yang diketik melalui *keyboard* dan dikirim ke mikrokontroler sama dengan karakter yang dikirim kembali oleh mikrokontroler ke *PC*. Hal ini menunjukkan komunikasi antara mikrokontroler dan *PC* dapat terjadi dengan baik
4. Sistem kWh meter prabayar yang menggunakan jala-jala listrik sebagai media komunikasi data dapat bekerja dengan baik karena mampu melakukan perhitungan energi listrik dan mampu melakukan pengiriman dan pemantauan nominal kWh sesuai dengan spesifikasi yang ditetapkan.

6.2 Saran

Proses perancangan, pembuatan dan pengujian alat ini dilakukan sesuai dengan spesifikasi yang telah ditentukan sehingga masih ada faktor-faktor lain yang diabaikan. Faktor-faktor tersebut mungkin dapat mempengaruhi kinerja sistem jika alat diaplikasikan pada kondisi sesungguhnya. Faktor-faktor tersebut seperti panjang dan jenis kabel yang digunakan, adanya sambungan-sambungan, serta adanya harmonisa dan *noise*. Oleh sebab itu perlu dilakukan penelitian tentang pengaruh faktor-faktor tersebut terhadap kinerja sistem. Besarnya daya pemancar pada alat ini sesuai dengan spesifikasi *IC* LM1893 yang memiliki jangkauan yang terbatas sehingga perlu dilakukan penelitian tentang daya pemancar agar dapat digunakan untuk proses pengiriman data dalam jangkauan yang lebih jauh. Selain itu perlu juga dilakukan

penambahan sistem keamanan data karena dalam perancangan dan pembuatan alat ini belum memperhitungkan masalah keamanan data yang ditransmisikan.

Untuk pengembangan lebih lanjut bisa ditambahkan pembelian nominal kWh melalui sms atau melalui ATM agar pelanggan dapat lebih mudah memperoleh nominal kWh yang akan digunakan.

DAFTAR PUSTAKA

Anonymous. 2003. *Kendali Diri dan Kendali Sistem*. http:// www.pikiran-rakyat.com/ cetak/ 0303/ 13/ cakrawala/ lainnya04.htm, diakses tanggal 27 Mei 2006

Atmel. 2003. *8-bit AVR microcontroller with 8Kbytes In Sistem Programable Flash.* www. atmel.com/ dyn/ resources/ prod_documents/ doc8515.pdf, diakses tanggal 10 April 2005.

Bies, Lammert. 1997. *RS232 Serial Cable Pinout*. http:// www. lammertbies. nl/ comm/ cable/ RS-232.html, diakses tanggal 4 Desember 2005.

Malvino, Albert Paul. 1987. *Prinsip-Prinsip Elektronika*. *Jilid 1, terjemahan Prof. M. Barmawi, Ph.D.* Jakarta : Erlangga.

Marcus, Teddy, et al. 2005.*Pemrograman Delphi dengan ADOExpress*. Bandung : Informatika.

Martina, Inge. 2004. *Pemrograman Visual Borland Delphi 7*. Jakarta : PT. Elex Media Computindo

National Semiconductor. 1995. *LM1893/LM2893 Carrier-Current Transciever*. www.national.com

Nia. 2005. Meter *Prabayar, Produk Baru Layanan PLN*. http://www.balipost.co.id/balipostcetak/2005/4/8/e2.htm , diakses tanggal 27 Mei 2006.

Nugroho, Agung. 2005. ”Peralatan Kopling Power Line Carrier,” *Transmisi*. No. 2, Vol.10, 2005, hal. 14-15.

Ridwan, Agus Dwi M. 2002. *Pengatur Beban Listrik Portable Menggunakan Power Line Carrier Berbasis Mikrokontroler AT89C51*. *Skripsi* Tidak Diterbitkan. Malang : Jurusan Teknik Elektro UNIBRAW.

Texas Instrument. 1998. MAX232, MAX232I Dual EIA-232 Driver/Receiver. www.texasinstrument.com, diakses tanggal 4 Desember 2005.

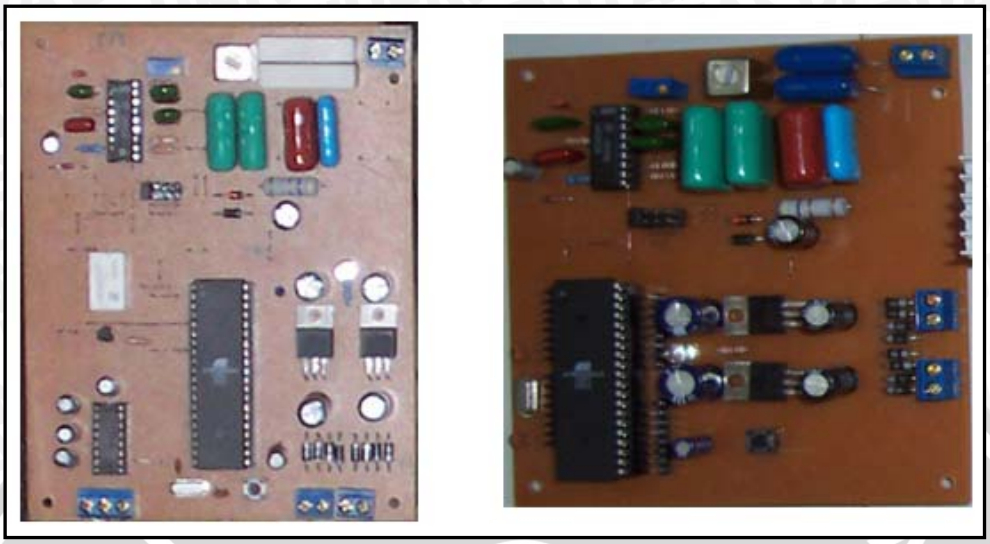
Wanamulya, Hilmy. 2005. *PLN Bali Luncurkan Layanan Meter Prabayar*. http:// www.mail-archive.com/bumi-serpong@yahoogroups.com/msg00418.html, diakses tanggal 27 Mei 2006.

Yuswanto. 2001. *Panduan Belajar Microsoft Access 2002*. Surabaya : Prestasi Pustaka.

LAMPIRAN A

FOTO ALAT DAN SKEMA RANGKAIAN

FOTO ALAT



Rangkaian Unit Sentral Data

Alat di Loket Pengisian Nominal kWh

Alat di Rumah Pelanggan

repository.ub.ac.id

LAMPIRAN B

KODE PROGRAM DELPHI DAN
MIKROKONTROLER

UNIVERSITAS BRAWIJAYA

KODE PROGRAM DELPHI UNIT 1 (UNIT UTAMA)

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, Grids, StdCtrls, CPort, Menus, ExtCtrls, DB, ADODB;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Edit1: TMenuItem;
    LihatData: TMenuItem;
    EditPelanggan: TMenuItem;
    EditTarif: TMenuItem;
    Run1: TMenuItem;
    Start: TMenuItem;
    Stop: TMenuItem;
    KirimkWh: TMenuItem;
    Help1: TMenuItem;
    PetunjukPenggunaan: TMenuItem;
    Pembuat: TMenuItem;
    Label1: TLabel;
    GroupBox1: TGroupBox;
    ShapeBusy: TShape;
    ShapeStandBy: TShape;
    ShapeStop: TShape;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    GroupBox2: TGroupBox;
    MemoStatus: TMemo;
    ComPort1: TComPort;
    TimerRequest: TTimer;
    Keluar: TMenuItem;
    EditKode: TEdit;
    EditIDTarget: TEdit;
    EditkWh: TMenuItem;
    TimerConnection: TTimer;
    TimerPenerimaan: TTimer;
    procedure KeluarClick(Sender: TObject);
    procedure EditPelangganClick(Sender: TObject);
    procedure EditTarifClick(Sender: TObject);
    procedure StartClick(Sender: TObject);
    procedure TimerRequestTimer(Sender: TObject);
    procedure StopClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure LihatDataClick(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure KirimkWhClick(Sender: TObject);
    procedure TimerPenerimaanTimer(Sender: TObject);
    procedure TimerConnectionTimer(Sender: TObject);
    procedure EditkWhClick(Sender: TObject);
    procedure PembuatClick(Sender: TObject);
  private
    { Private declarations }
  public
```

```
    { Public declarations }
  end;

var
  Form1: TForm1;
  KodePerintah,Validasi,NominalHigh,NominalMedium,NominalLow:byte;
  IDTarget,IDSource:byte;
  NominalkWh,header,IDHeader,NominalHeader,status:string;
  BusyCode,StandByCode,StopCode: boolean;
  JumlahTarget, NomorAntrian, NomorPermintaan : integer;
  AngkakWh : Real;

implementation

uses
  unit2,unit3,unit4,unit5,unit6;
{$R *.dfm}

procedure TForm1.KeluarClick(Sender: TObject);
begin
  comport1.Close;
  application.Terminate;
end;

procedure TForm1.EditPelangganClick(Sender: TObject);
begin
  form2.Show;
end;

procedure TForm1.EditTarifClick(Sender: TObject);
begin
  form3.Show;
end;

procedure TForm1.StartClick(Sender: TObject);
begin
  (* Persiapan *)
  TimerRequest.Interval:=100;
  start.Enabled:=false;
  kirimKWH.Enabled:=false;
  EditPelanggan.Enabled:=false;
  EditTarif.Enabled:=false;
  keluar.Enabled:=false;
  EditkWh.Enabled:=false;
  KodePerintah:=0;
  editkode.Text:='Request';
  NominalHigh:=0;
  NominalMedium:=0;
  NominalLow:=0;
  IDSource:=0;
  header:='';
  IDHeader:='';
  NominalHeader:='';
  NominalkWh:='';

  (* Pengambilan dari database data-data yang akan diproses *)
  form2.ADOQueryIdentitas.Close;
  form2.ADOQueryIdentitas.SQL.Clear;
  form2.ADOQueryIdentitas.SQL.Add('SELECT * FROM Identitas WHERE Nama
<> ""');
  form2.ADOQueryIdentitas.Open;
```

```
  JumlahTarget:=form2.ADOQueryIdentitas.RecordCount;
  NomorAntrian:=1;
  form2.ADOQueryIdentitas.First;

  (* pengaktifan timer dan port serial, tanda mulainya sistem bekerja
*)
  comport1.Open;
  TimerRequest.Enabled:=true;
end;

procedure TForm1.TimerRequestTimer(Sender: TObject);
begin
  (* Matikan timer dan pemberian nomor permintaan *)
  TimerRequest.Enabled:=false;
  If IDHeader='e' then NomorPermintaan:=NomorPermintaan+1 else
NomorPermintaan:=1;

  (* Pengambilan ID yang akan diproses *)
  with form2.ADOQueryIdentitas do
  begin
    IDTarget:=FieldByName('ID').AsInteger;
    EditIDTarget.Text:=inttostr(IDTarget);
  end;

  (* Pembuatan data validasi *)
  Validasi:= IDTarget xor KodePerintah xor NominalHigh xor
NominalMedium xor NominalLow;

  (* Pengiriman data serta tampilan indikator dan memo *)
  ShapeBusy.Brush.Color:=clred;
  ShapeStandBy.Brush.Color:=clWhite;
  ShapeStop.Brush.Color:=clwhite;
  MemoStatus.Clear;
  MemoStatus.Lines.Add('Target = '+ inttostr(IDTarget));
  MemoStatus.Lines.Add('Command = Request');
  MemoStatus.Lines.Add('Request Number = '+inttostr(NomorPermintaan));
  MemoStatus.Lines.Add('Please Wait ...');
  comport1.WriteStr(char(IDTarget));
  sleep(25);
  comport1.WriteStr(char(KodePerintah));
  sleep(25);
  comport1.WriteStr(char(NominalHigh));
  sleep(25);
  comport1.WriteStr(char(NominalMedium));
  sleep(25);
  comport1.WriteStr(char(NominalLow));
  sleep(25);
  comport1.WriteStr(char(Validasi));
  stop.Enabled:=false;
  TimerConnection.Enabled:=true;

  (* Persiapan berikutnya *)
  NomorAntrian:=NomorAntrian+1;
  if NomorAntrian > JumlahTarget then
  begin
    form2.ADOQueryIdentitas.First;
    NomorAntrian:=1;
  end
  else form2.ADOQueryIdentitas.Next;
end;
```

```
procedure TForm1.StopClick(Sender: TObject);
begin
  (* Kembalikan seperti semula *)
  start.Enabled:=true;
  TimerRequest.Enabled:=false;
  TimerConnection.Enabled:=false;
  TimerPenerimaan.Enabled:=false;
  keluar.Enabled:=true;
  stop.Enabled:=false;
  kirimkwh.Enabled:=true;
  EditkWh.Enabled:=true;
  ShapeStop.Brush.Color:=clRed;
  ShapeStandBy.Brush.Color:=clWhite;
  ShapeBusy.Brush.Color:=clWhite;
  EditPelanggan.Enabled:=true;
  EditTarif.Enabled:=true;
  MemoStatus.Clear;
  MemoStatus.Lines.Add('System Stop');
  comport1.Close;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  (* Tampilan awal *)
  ShapeStop.Brush.Color:=clRed;
  MemoStatus.Lines.Add('System Stop');
end;

procedure TForm1.LihatDataClick(Sender: TObject);
begin
  form4.Show;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  kata,s:string;
begin
  if header='$' then
  begin
    if (IDHeader='i') or (IDHeader='e') then
    begin
      if IDSource<>0  then
      begin
        if (NominalHeader='n') or (NominalHeader='e') then
        begin
          if length(NominalkWh)= 7 then
          begin
            comport1.ReadStr(kata,count);
            NominalkWh:=NominalkWh+kata;
            TimerConnection.Enabled:=false;
            TimerPenerimaan.Enabled:=true;
            exit;
          end
          else
          begin
            comport1.ReadStr(kata,count);
            NominalkWh:=NominalkWh+kata;
            exit;
          end;
        end
        else
```

```
        begin
          comport1.ReadStr(NominalHeader,count);
          exit;
        end;
      end
      else
      begin
        comport1.ReadStr(s,count);
        IDSource:=byte(s[1]);
        exit;
      end
    end
    else
    begin
      comport1.ReadStr(IDHeader,count);
      exit;
    end;
  end
  else
  begin
    comport1.ReadStr(header,count);
    exit;
  end;
end;

procedure TForm1.KirimkWhClick(Sender: TObject);
begin
  IDSource:=0;
  header:='';
  IDHeader:='';
  NominalHeader:='';
  NominalkWh:='';
  form5.Show;
end;

procedure TForm1.TimerPenerimaanTimer(Sender: TObject);
begin
  TimerPenerimaan.Enabled:=false;
  if (IDHeader='e') and (NomorPermintaan<>2) then
  begin
    form2.ADOQueryIdentitas.Prior;
    NomorAntrian:=NomorAntrian-1;
    TimerRequest.Interval:=500;
    TimerRequest.Enabled:=true;
    exit;
  end
  else
  begin
    if (IDHeader='e') then
    begin
      (* Kembalikan seperti semula *)
      start.Enabled:=true;
      TimerRequest.Enabled:=false;
      TimerConnection.Enabled:=false;
      TimerPenerimaan.Enabled:=false;
      keluar.Enabled:=true;
      stop.Enabled:=false;
      kirimkwh.Enabled:=true;
      EditkWh.Enabled:=true;
      ShapeStop.Brush.Color:=clRed;
      ShapeStandBy.Brush.Color:=clWhite;
```

```
        ShapeBusy.Brush.Color:=clWhite;
        EditPelanggan.Enabled:=true;
        EditTarif.Enabled:=true;
        MemoStatus.Clear;
        MemoStatus.Lines.Add('Error on Sentral Connection');
        MemoStatus.Lines.Add('System Stop');
        comport1.Close;
        exit;
      end
      else
      begin
        memostatus.Lines.Add('Source = '+inttostr(IDSource));
        AngkakWh:=strtofloat(NominalkWh)/1000;
        NominalkWh:=floattostr(AngkakWh);
        if NominalHeader='n' then Status:='Oke' else Status:='Error';
        (* Penyimpanan ke dalam database *)
        with form4.ADOQuerySisa do
        begin
          close;
          SQL.Clear;
          SQL.Add('SELECT sisa.ID, identitas.Nama,
sisa.Waktu_Data_Terambil, sisa.Sisa_kWh, sisa.Status FROM sisa,
identitas WHERE sisa.ID=identitas.ID AND identitas.nama <> "";');
          open;
          Locate('ID',Strtoint(EditIDTarget.Text),[loPartialKey]);
          Edit;
          FieldByName('Waktu_Data_Terambil').Value:=now;
          if NominalHeader='n' then
FieldByName('Sisa_kWh').AsFloat:=AngkakWh;
          FieldByName('Status').AsString:=Status;
          Post;
        end;
        (* Pra Persiapan Berikutnya *)
        if editkode.Text='Request' then
        begin
          ShapeStop.Brush.Color:=clWhite;
          ShapeStandBy.Brush.Color:=clRed;
          ShapeBusy.Brush.Color:=clWhite;
          stop.Enabled:=true;
          if NominalHeader='n' then
          begin
            memostatus.Lines.Add('Nominal = '+NominalkWh+' kWh');
            memostatus.Lines.Add('Status = '+Status);
          end
          else
          begin
            memostatus.Lines.Add('System Error On PLC Connection');
          end;
          MemoStatus.Lines.Add('System Stand By From REQUEST');
          NominalHigh:=0;
          NominalMedium:=0;
          NominalLow:=0;
          IDSource:=0;
          header:='';
          IDHeader:='';
          NominalHeader:='';
          NominalkWh:='';
          TimerRequest.Interval:=5000;
          TimerRequest.Enabled:=true;
          exit;
        end;
```

```
        if (editkode.Text='Send') or (editkode.Text='Edit') then
        begin
          sleep(2000);
          start.Enabled:=true;
          keluar.Enabled:=true;
          stop.Enabled:=true;
          kirimkwh.Enabled:=true;
          ShapeStop.Brush.Color:=clWhite;
          ShapeStandBy.Brush.Color:=clRed;
          ShapeBusy.Brush.Color:=clWhite;
          EditPelanggan.Enabled:=true;
          EditTarif.Enabled:=true;
          if NominalHeader='n' then
          begin
            with form4.ADOQueryPengirimanKWH do
            begin
              close;
              SQL.Clear;
              SQL.Add('SELECT Pengiriman_kWh.ID, identitas.Nama,
Pengiriman_kWh.Sebelum, Pengiriman_kWh.Besar_kWh,
Pengiriman_kWh.Sekarang, Pengiriman_kWh.Besar_kWh_Sekarang FROM
Pengiriman_kWh, identitas WHERE Pengiriman_kWh.ID=identitas.ID AND
identitas.nama <> "";');
              open;
              Locate('ID',Strtoint(EditIDTarget.Text),[loPartialKey]);
              Edit;

FieldByName('Sebelum').Value:=FieldByName('Sekarang').Value;

FieldByName('Besar_kWh').Value:=FieldByName('Besar_kWh_Sekarang').Valu
e;
              FieldByName('Sekarang').Value:=now;
              FieldByName('Besar_kWh_Sekarang').AsFloat:=AngkakWh;
              Post;
            end;
            memostatus.Lines.Add('Nominal = '+NominalkWh+' kWh');
            memostatus.Lines.Add('Status = '+Status);
          end
          else
          begin
            memostatus.Lines.Add('System Error On PLC Connection');
          end;
          editkWh.Enabled:=true;
          MemoStatus.Lines.Add('System Stand By From SEND or EDIT');
          exit;
        end;
      end;
    end;
end;

procedure TForm1.TimerConnectionTimer(Sender: TObject);
begin
  (* Kembalikan seperti semula *)
  start.Enabled:=true;
  TimerRequest.Enabled:=false;
  TimerConnection.Enabled:=false;
  TimerPenerimaan.Enabled:=false;
  keluar.Enabled:=true;
  stop.Enabled:=false;
  kirimkwh.Enabled:=true;
  EditkWh.Enabled:=true;
```

```
  ShapeStop.Brush.Color:=clRed;
  ShapeStandBy.Brush.Color:=clWhite;
  ShapeBusy.Brush.Color:=clWhite;
  EditPelanggan.Enabled:=true;
  EditTarif.Enabled:=true;
  MemoStatus.Clear;
  MemoStatus.Lines.Add('System Error on Sentral Connection');
  MemoStatus.Lines.Add('System Stop');
  comport1.Close;
end;

procedure TForm1.EditkWhClick(Sender: TObject);
begin
  IDSource:=0;
  header:='';
  IDHeader:='';
  NominalHeader:='';
  NominalkWh:='';
  form6.Show;
end;

procedure TForm1.PembuatClick(Sender: TObject);
begin
  MessageDlg('Presented by Bamon and Juay_UTT, Copyleft @2007
Reserved',mtInformation, [mbOk], 0);
end;

end.
```

KODE PROGRAM DELPHI UNIT 5 (KIRIM kWh)

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls;

type
  TForm5 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    EditID: TEdit;
    Label3: TLabel;
    EditJumlahUang: TEdit;
    ButtonSend: TButton;
    ButtonCancel: TButton;
    procedure ButtonCancelClick(Sender: TObject);
    procedure ButtonSendClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation
uses unit1,unit2,unit3;
{$R *.dfm}

procedure TForm5.ButtonCancelClick(Sender: TObject);
begin
  form5.Close;
end;

procedure TForm5.ButtonSendClick(Sender: TObject);
var
  s1,s2,GolonganDaya:string;
  JumlahUang,Harga:integer;
  JumlahkWh,JumlahWh,pembanding:real;
  byte2,byte1,byte0:real;
  KodePerintah,Validasi,NominalHigh,NominalMedium,NominalLow:byte;
begin
  s1:=EditID.Text;
  s2:=EditJumlahUang.Text;
  if (length(s1)=0) or (length(s2)=0) then
  begin
    MessageDlg('Data is not complete',mtError, [mbOk], 0);
    exit;
  end
  else
  begin
    form2.ADOQueryIdentitas.Close;
    form2.ADOQueryIdentitas.SQL.Clear;
```

```
    form2.ADOQueryIdentitas.SQL.Add('SELECT * FROM Identitas WHERE ID
='+s1);
    form2.ADOQueryIdentitas.Open;
    if (form2.ADOQueryIdentitas.RecordCount=0) or
(form2.ADOQueryIdentitas.FieldByName('Nama').AsString='') then
    begin
      MessageDlg('Data is not found',mtInformation, [mbOk], 0);
      EditID.Text:='';
      EditJumlahUang.Text:='';
      EditID.SetFocus;
      exit;
    end
    else
    begin
      with form2.ADOQueryIdentitas do
      begin
        close;
        SQL.Clear;
        SQL.Add('SELECT Golongan_Daya FROM identitas WHERE ID='+s1);
        open;
        GolonganDaya:=FieldByName('Golongan_Daya').AsString;
      end;
      with form3.ADOQueryHarga do
      begin
        close;
        SQL.Clear;
        SQL.Add('SELECT Harga_Per_kWh FROM Harga WHERE
Golongan_Daya='+'"'+GolonganDaya+'"');
        open;
        Harga:=FieldByName('Harga_Per_kWh').AsInteger;
      end;
      JumlahkWh:=strtoint(s2)/Harga;
      JumlahWh:=round(JumlahkWh*1000);
      (* Ubah desimal ke biner *)
      Pembanding:=8388608;
      byte2:=0;
      repeat
        if JumlahWh>=pembanding then
                begin
                      JumlahWh:=JumlahWh-pembanding;
                      byte2:=byte2+(pembanding/65536);
                end
        else;
        pembanding:=pembanding/2;
      until(pembanding=32768);
      byte1:=0;
      repeat
        if JumlahWh>=pembanding then
                begin
                      JumlahWh:=JumlahWh-pembanding;
                      byte1:=byte1+(pembanding/256);
                end
        else;
        pembanding:=pembanding/2;
      until(pembanding=128);
      byte0:=JumlahWh;
      NominalHigh:=round(byte2);
      NominalMedium:=round(byte1);
      NominalLow:=round(byte0);
      KodePerintah:=1;
      (* Persiapan *)
```

```
        form1.comport1.Open;
        form1.start.Enabled:=false;
        form1.EditPelanggan.Enabled:=false;
        form1.EditTarif.Enabled:=false;
        form1.keluar.Enabled:=false;
        form1.EditKode.Text:='Send';
        form1.KirimkWh.Enabled:=false;
        form1.EditkWh.Enabled:=false;
        form1.EditIDTarget.Text:=s1;
        (* Pembuatan data validasi *)
        Validasi:= IDTarget xor KodePerintah xor NominalHigh xor
NominalMedium xor NominalLow;
        (* Send *)
        form1.ShapeBusy.Brush.Color:=clred;
        form1.ShapeStandBy.Brush.Color:=clWhite;
        form1.ShapeStop.Brush.Color:=clwhite;
        form1.MemoStatus.Clear;
        form1.MemoStatus.Lines.Add('Target = '+s1);
        form1.MemoStatus.Lines.Add('Command = Send');
        form1.MemoStatus.Lines.Add('Jumlah kWh =
'+FloatToStr(JumlahkWh));
        form1.MemoStatus.Lines.Add('Please Wait ...');
        form1.comport1.WriteStr(char(strtoint(s1)));
        sleep(25);
        form1.comport1.WriteStr(char(KodePerintah));
        sleep(25);
        form1.comport1.WriteStr(char(NominalHigh));
        sleep(25);
        form1.comport1.WriteStr(char(NominalMedium));
        sleep(25);
        form1.comport1.WriteStr(char(NominalLow));
        sleep(25);
        form1.comport1.WriteStr(char(Validasi));
        form1.TimerConnection.Enabled:=true;
        form5.Close;
    end;
  end;
end;

procedure TForm5.FormShow(Sender: TObject);
begin
  EditID.Text:='';
  EditJumlahUang.Text:='';
  EditID.SetFocus;
end;

end.
```

KODE PROGRAM DELPHI UNIT 6 (EDIT kWh)

```
unit Unit6;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls;

type
  TForm6 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    EditID: TEdit;
    EditJumlahkWh: TEdit;
    ButtonSend: TButton;
    ButtonCancel: TButton;
    procedure ButtonCancelClick(Sender: TObject);
    procedure ButtonSendClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

implementation
uses unit1,unit2,unit3, Unit5;

{$R *.dfm}

procedure TForm6.ButtonCancelClick(Sender: TObject);
begin
  form6.Close;
end;

procedure TForm6.ButtonSendClick(Sender: TObject);
var
  s1,s2:string;
  JumlahkWh,JumlahWh,pembanding:real;
  byte2,byte1,byte0:real;
  KodePerintah,Validasi,NominalHigh,NominalMedium,NominalLow:byte;
begin
  s1:=EditID.Text;
  s2:=EditJumlahkWh.Text;
  if (length(s1)=0) or (length(s2)=0) then
  begin
    MessageDlg('Data is not complete',mtError, [mbOk], 0);
    exit;
  end
  else
  begin
    form2.ADOQueryIdentitas.Close;
    form2.ADOQueryIdentitas.SQL.Clear;
```

```
    form2.ADOQueryIdentitas.SQL.Add('SELECT * FROM Identitas WHERE ID
='+s1);
    form2.ADOQueryIdentitas.Open;
    if (form2.ADOQueryIdentitas.RecordCount=0) or
(form2.ADOQueryIdentitas.FieldByName('Nama').AsString='') then
    begin
      MessageDlg('Data is not found',mtInformation, [mbOk], 0);
      EditID.Text:='';
      EditJumlahkWh.Text:='';
      EditID.SetFocus;
      exit;
    end
    else
    begin
      JumlahkWh:=strtofloat(s2);
      JumlahWh:=round(JumlahkWh*1000);
      (* Ubah desimal ke biner *)
      Pembanding:=8388608;
      byte2:=0;
      repeat
        if JumlahWh>=pembanding then
                begin
                        JumlahWh:=JumlahWh-pembanding;
                        byte2:=byte2+(pembanding/65536);
                end
        else;
        pembanding:=pembanding/2;
      until(pembanding=32768);
      byte1:=0;
      repeat
        if JumlahWh>=pembanding then
                begin
                        JumlahWh:=JumlahWh-pembanding;
                        byte1:=byte1+(pembanding/256);
                end
        else;
        pembanding:=pembanding/2;
      until(pembanding=128);
      byte0:=JumlahWh;
      NominalHigh:=round(byte2);
      NominalMedium:=round(byte1);
      NominalLow:=round(byte0);
      KodePerintah:=2;
      (* Persiapan *)
      form1.comport1.Open;
      form1.start.Enabled:=false;
      form1.EditPelanggan.Enabled:=false;
      form1.EditTarif.Enabled:=false;
      form1.keluar.Enabled:=false;
      form1.EditKode.Text:='Edit';
      form1.KirimkWh.Enabled:=false;
      form1.EditkWh.Enabled:=false;
      form1.EditIDTarget.Text:=s1;
      (* Pembuatan data validasi *)
      Validasi:= IDTarget xor KodePerintah xor NominalHigh xor
NominalMedium xor NominalLow;
      (* Send *)
      form1.ShapeBusy.Brush.Color:=clred;
      form1.ShapeStandBy.Brush.Color:=clWhite;
      form1.ShapeStop.Brush.Color:=clwhite;
      form1.MemoStatus.Clear;
```

```
        form1.MemoStatus.Lines.Add('Target = '+s1);
        form1.MemoStatus.Lines.Add('Command = Edit');
        form1.MemoStatus.Lines.Add('Jumlah kWh =
'+FloatToStr(JumlahkWh));
        form1.MemoStatus.Lines.Add('Please Wait ...');
        form1.comport1.WriteStr(char(strtoint(s1)));
        sleep(25);
        form1.comport1.WriteStr(char(KodePerintah));
        sleep(25);
        form1.comport1.WriteStr(char(NominalHigh));
        sleep(25);
        form1.comport1.WriteStr(char(NominalMedium));
        sleep(25);
        form1.comport1.WriteStr(char(NominalLow));
        sleep(25);
        form1.comport1.WriteStr(char(Validasi));
        form1.TimerConnection.Enabled:=true;
        form6.Close;
      end;
    end;

end;

procedure TForm6.FormShow(Sender: TObject);
begin
  EditID.Text:='';
  EditJumlahkWh.Text:='';
  EditID.SetFocus;
end;

end.
```

KODE PROGRAM MIKROKONTROLER SENTRAL

```
/* File include */
#include <mega8515.h>
#include <delay.h>


/* Pendefinisian */
#define tranceive_selector      PORTB.4
#define connection_selector     PORTB.7


/* Inisialisasi variabel global */
bit receive_code,validasi_code;
unsigned char header, header2, header3, nominal_high, nominal_medium,
nominal_low, source, target, kode, validasi;
unsigned char nomor_pengiriman;
unsigned char tempat_sampah,source_awal;

/* Setting port direction */
void init_port_direction()
{
        /* port b.4-b.7 sebagai keluaran*/
        DDRB=0xff;
        DDRD.0=0;
        DDRD.1=1;
}


/* Setting timer 0 */
void set_timer0()
{
        /* Timer ini akan digunakan sebagai pewaktu tunda biasa */
        /* Mode yang digunakan adalah Clear Timer On Compare */
        TCCR0=0x08;
        TCNT0=0;
}


/* Setting timer 0 */
void set_timer1()
{
        /* Timer ini akan digunakan sebagai pewaktu tunda biasa */
        /* Mode yang digunakan adalah Clear Timer On Compare */
        /* Dari BOTTOM sampai COMPARE MATCH di setting selama 50 ms
           dengan fosc ==11,0592 MHz */
        TCCR1A=0x00;
        TCCR1B=0x08;
        TCNT1H=0;
        TCNT1L=0;
}


/* Untuk komunikasi serial */
void set_serial()
{
        /* Baudrate awal = 360 bps  dengan fosc = 11.0592 MHz */
        /* Interupt receive complete aktif */
        UBRRH=UBRRH&0b01111111;
        UBRRH=0x07;
        UBRRL=0x7f;
```

```
        UCSRB=(UCSRB&0b00000011)|0b10011000;
        UCSRC=UCSRC|0b10000000;
        UCSRC=0b10000110;
}


/* Ubah baudrate */
void ubah_baudrate(unsigned char new_baudrate_high, unsigned char
new_baudrate_low)
{
        UBRRH=UBRRH&0b01111111;
        UBRRH=new_baudrate_high;
        UBRRL=new_baudrate_low;
}


/* Fungsi penunda 20 us */
void delay_20us(unsigned long int pengali)
{
        /* Dari BOTTOM sampai COMPARE MATCH di setting selama 20 us
           dengan fosc=11,0592 MHz */
        OCR0=221;

        /* Nyalakan timer */
        TCCR0=TCCR0|0x01;

        /* Tunggu selesai masa penantian */
        while (pengali!=0)
        {
                while ((TIFR&0x01)!=0x01);
                TIFR=TIFR|0x01;
                pengali--;
        }

        /* Matikan timer */
        TCCR0=TCCR0&0xfe;
}


/* Fungsi penunda 5 ms */
void delay_5ms(int pengali)
{
        /* Dari BOTTOM sampai COMPARE MATCH di setting selama 5 ms
           dengan fosc=11,0592 MHz */
        OCR1AH=0xd9;
        OCR1AL=0x03;

        /* Nyalakan timer */
        TCCR1B=TCCR0|0x01;

        /* Tunggu selesai masa penantian */
        while (pengali!=0)
        {
                while ((TIFR&0x40)!=0x40);
                TIFR=TIFR|0x40;
                pengali--;
        }

        /* Matikan timer */
        TCCR1B=TCCR0&0xfe;
}
```

```
/* Fungsi penunda 1 us */
void delay_1us(unsigned long int pengali)
{
        /* Dari BOTTOM sampai COMPARE MATCH di setting selama 1 us
           dengan fosc=11,0592 MHz */
        OCR0=11;

        /* Nyalakan timer */
        TCCR0=TCCR0|0x01;

        /* Tunggu selesai masa penantian */
        while (pengali!=0)
        {
                while ((TIFR&0x01)!=0x01);
                TIFR=TIFR|0x01;
                pengali--;
        }

        /* Matikan timer */
        TCCR0=TCCR0&0xfe;
}


/* Fungsi kirim data secara serial */
void USART_Transmit( unsigned char data )
{
        /* Clear TXC */
        UCSRA=UCSRA|0b01000000;

        /* Wait for empty transmit buffer */
        while ((UCSRA&0b01100000)!=0b00100000);

        /* Put data into buffer, sends the data */
        UDR = data;
}


/* Fungsi terima data secara serial */
unsigned char USART_Receive( void )
{
        /* Wait for data to be received */
        while ((UCSRA&0b10000000)!=0b10000000);

        /* Get and return received data from buffer */
        return UDR;
}


/* Fungsi untuk mengirim data string ke PC */
void kirim_string(long int biner)
{
    long int ascii;

    ascii=biner;
    ascii=(ascii/10000000)|0x30;
    USART_Transmit(ascii);
    delay_5ms(5);
    ascii=biner;
    ascii=((ascii/1000000)%10)|0x30;
```

```
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=((ascii/100000)%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=((ascii/10000)%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=((ascii/1000)%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=((ascii/100)%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=((ascii/10)%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
        ascii=biner;
        ascii=(ascii%10)|0x30;
        USART_Transmit(ascii);
        delay_5ms(5);
}


/* Fungsi mengubah biner ke desimal */
long int biner_to_desimal(unsigned char high, unsigned char medium,
unsigned char low)
{
        long int pembanding,desimal;
        int i;
        unsigned char temp;

        desimal=0;
        pembanding=65536;
        for (i=1;i<=8;i++)
        {
                temp=high&0x01;
                desimal=desimal+(temp*pembanding);
                high=high>>1;
                pembanding*=2;
        }

        pembanding=256;
        for (i=1;i<=8;i++)
        {
                temp=medium&0x01;
                desimal=desimal+(temp*pembanding);
                medium=medium>>1;
                pembanding*=2;

        }

        desimal=desimal+low;
        return desimal;
}
```

```
/* Fungsi saat terjadi interrupt Rx Complete */
/* Turn registers saving off */
#pragma savereg-
/* interrupt handler */
interrupt [10] void rx_complete(void)
{
        /* Pemanasan */
        #asm
        push r30
        push r31
        in   r30,SREG
        push r30
        #endasm

        /* Inti */
        header=UDR;
        if (header=='$')
        {
                tempat_sampah=UDR;
                header2=USART_Receive();
                if (header2=='*')
                {
                        tempat_sampah=UDR;
                        header3=USART_Receive();
                        if (header3=='#')
                        {
                                tempat_sampah=UDR;
                                kode=USART_Receive();
                                target=USART_Receive();
                                source=USART_Receive();
                                nominal_low=USART_Receive();
                                nominal_medium=USART_Receive();
                                nominal_high=USART_Receive();
                                validasi=USART_Receive();
                                if ((source_awal==source)&&(target=0))
                                receive_code=1;
                        }
                }

        }
        else;

        /* Pendinginan */
        #asm
        pop r30
        out SREG,r30
        pop r31
        pop r30
        #endasm

}
/* re-enable register saving for the other interrupts */
#pragma savereg+


/* Program Utama */
void main()
{
        /* Inisialisasi variabel local MAIN */
        long int nominal_pulsa;
```

```
/* Proses inisialisasi MK */
init_port_direction();
set_timer0();
set_timer1();
set_serial();

/* Lets Go */
do
{
        /* Menunggu perintah dari PC */
        ubah_baudrate(0x01,0x1F);
        tranceive_selector=0;
        connection_selector=0;
        validasi_code=0;
        do
        {
                tempat_sampah=UDR;
                target=USART_Receive();
                kode=USART_Receive();
                nominal_high=USART_Receive();
                nominal_medium=USART_Receive();
                nominal_low=USART_Receive();
                validasi=USART_Receive();
                if  (validasi == target ^ kode ^ nominal_high
                ^ nominal_medium ^ nominal_low) validasi_code
                = 1;
                else
                {
                        USART_Transmit('$');
                        delay_5ms(5);
                        USART_Transmit('e');
                        delay_5ms(5);
                        USART_Transmit(target);
                        delay_5ms(5);
                        USART_Transmit('e');
                        delay_5ms(5);
                        nominal_pulsa =
                        biner_to_desimal(nominal_high,
                        nominal_medium, nominal_low);
                        kirim_string(nominal_pulsa);
                }
        }while(validasi_code==0);

        /* Kirim data ke rumah */
        source_awal=target;
        nomor_pengiriman=0;
        do
        {
                receive_code=0;
                nomor_pengiriman++;
                validasi = '$' ^ kode ^ target ^ 0 ^
                nominal_low ^ nominal_medium ^ nominal_high;
                ubah_baudrate(0x07,0x7f);
                tranceive_selector=1;
                connection_selector=1;
                delay_5ms(400);
                USART_Transmit('x');
                delay_5ms(10);
                USART_Transmit('x');
                delay_5ms(10);
```

```
                USART_Transmit('$');
                delay_5ms(10);
                USART_Transmit('*');
                delay_5ms(10);
                USART_Transmit('#');
                delay_5ms(10);
                USART_Transmit(kode);
                delay_5ms(10);
                USART_Transmit(target);
                delay_5ms(10);
                USART_Transmit(0);
                delay_5ms(10);
                USART_Transmit(nominal_low);
                delay_5ms(10);
                USART_Transmit(nominal_medium);
                delay_5ms(10);
                USART_Transmit(nominal_high);
                delay_5ms(10);
                USART_Transmit(validasi);
                delay_5ms(10);

                /* Menunggu kiriman dari rumah */
                tranceive_selector=0;
                #asm("sei");
                delay_5ms(1000);
                #asm("cli");
        }while((nomor_pengiriman!=2)&&(kode==3));

        /* Lempar kiriman dari rumah ke PC */
        ubah_baudrate(0x01,0x1f);
        connection_selector=0;
        delay_5ms(600);
        if ((receive_code==1)||(kode==4))
        {
                USART_Transmit('x');
                delay_5ms(5);
                USART_Transmit('$');
                delay_5ms(5);
                USART_Transmit('i');
                delay_5ms(5);
                USART_Transmit(source);
                delay_5ms(5);
                USART_Transmit('n');
                delay_5ms(5);
                nominal_pulsa = biner_to_desimal(nominal_high,
                nominal_medium, nominal_low);
                kirim_string(nominal_pulsa);
        }
        else
        {
                USART_Transmit('x');
                delay_5ms(5);
                USART_Transmit('$');
                delay_5ms(5);
                USART_Transmit('i');
                delay_5ms(5);
                USART_Transmit(target);
                delay_5ms(5);
                USART_Transmit('e');
                delay_5ms(5);
                nominal_pulsa = biner_to_desimal(nominal_high,
```

```
                    nominal_medium, nominal_low);
                    kirim_string(nominal_pulsa);
                }
        }while(1);
}
```

LAMPIRAN C

DATASHEET