RANCANG BANGUN KWH-METER DIGITAL BERBASIS SISTEM TELEMETRI

TUGAS AKHIR

Untuk memperoleh gelar Ahli Madya pada
Program Studi Diploma III Teknik Elektro
Jurusan Teknik Elektro – Fakultas Teknik
Universitas Negeri Semarang

Oleh
Nur Huda
5350308003

PERPUSTAKAAN UNNES

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS NEGERI SEMARANG

2011

PENGESAHAN

Tugas Akhir ini telah dipertahankan dihadapan Panitia Ujian Tugas Akhir Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang pada hari Selasa tanggal 26 Juli 2011.

Panitia :

Ketua	Sekretaris
Drs. Djoko Adi Widodo, M.T.	**Drs. Agus Murnomo, M.T.**
NIP. 195909271986011001	NIP. 195506061986031002
Penguji I	Penguji II/Pembimbing
Tatyantoro Andrasto, S.T., M.T.	**Drs. Agus Murnomo, M.T.**
NIP. 196803161999031001	NIP. 195506061986031002

Mengetahui

Dekan Fakultas Teknik

Drs. Abdurrahman, M.Pd.

NIP. 196009031985031002

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN UNNES

MOTTO DAN PERSEMBAHAN

MOTTO

My life is my adventure

Don't look book from just a cover

Experience is the best teacher

I do and I understand

PERSEMBAHAN

Tugas akhir ini saya persembahkan untuk:

Kedua orang tua tercinta yang senantiasa menyayangi dan mendukungku. Terima kasih atas do'a dan segala yang telah diberikan.

Kakak dan Adik tercinta yang selalu memberi semangat dan dorongan.

Seseorang yang sangat saya sayangi dan cintai yang selalu menemani disaat suka maupun duka, dan senantiasa selalu setia dan mau menerima saya apa adanya.

Teman-teman yang telah menberi semangat, dorongan dan dukungan.

Serta orang-orang yang senantiasa membantu saya dalam menyelesaikan Tugas Akhir ini.

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN UNNES

KATA PENGANTAR

Puja dan puji syukur tak lupa saya panjatkan kehadirat Allah SWT yang senantiasa melimpahkan rahmat, taufiq dan hidayah-Nya, sehingga tugas akhir dengan judul **"Rancang Bangun KWH-meter Digital Berbasis Sistem Telemteri"** ini dapat terselesaikan dengan baik.

Saya menyadari bahwa dalam pembuatan tugas akhir ini, keberhasilan bukan semata-mata diraih dengan sendirinya, melainkan berkat dorongan dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini saya bermaksud menyampaikan ucapan terimakasih kepada pihak-pihak yang telah membantu dalam pembuatan tugas akhir ini yang mungkin mana tidak bisa saya sebut namanya satu-persatu. Dengan penuh kerendahan hati, saya mengucapkan terimakasih kepada yang terhormat:

1. Prof. Dr. H. Sudijono Sastroatmodjo, M.Si. selaku Rektor Universitas Negeri Semarang.
2. Drs. Abdurrahman, M.Pd. selaku Dekan Fakultas Teknik.
3. Drs. Djoko Adi Widodo, M.T. selaku Ketua Jurusan Teknik Elektro.
4. Drs. Agus Murnomo, M.T. selaku Ka.Prodi DIII Teknik Elektro dan sekaligus dosen pembimbing.
5. Bapak Ibu tercinta yang telah mendoakan dan memberikan semangat.
6. Serta teman-teman dan orang-orang yang telah memberikan bantuannya.

Saya menyadari dalam tugas akhir ini masih banyak terdapat kekurangan dan kesalahan, oleh karena itu kritik dan saran yang mendukung sangat saya butuhkan sebagai upaya perbaikan dan semoga tugas akhir ini dapat bermanfaat.

Semarang,

Penulis

ABSTRAK

Nur Huda. 2011. *"Rancang Bangun KWH-meter Digital Berbasis Sistem Telemetri"*. Tugas Akhir, Diploma III Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang. Drs. Agus Murnomo, M.T.

Kata kunci: KWH-meter Digital, Mikrokontroller, Telemetri, Pemrograman

Pencatatan dan penghitungan daya pemakaian energi listrik pada pelanggan yang masih secara manual, sehingga sering terjadi ketidak sesuaian pemakaian energi listrik dengan biaya yang harus dibayarkan. Alternatif penyelesaian permasalahan ini adalah penggunaan KWH-meter digital sistem telemetri yang dapat menampilkan daya dan biaya pemakaian energi listrik serta dapat dipantau jarak jauh. Tujuan dari alat ini adalah unjuk kerja dari KWH-meter digital sistem telemetri tersebut, dan manfaat dari alat ini untuk mengurangi kesalahan pencatatan dan penghitungan secara manual dan mempercepat informasi daya serta biaya pemakaian energi listrik di pelanggan kepada PLN.

Metode yang dipakai adalah metode laboratoris, pengumpulan data, instrumen dan teknik analasis data. Laboratoris merupakan metode eksperimen atau uji coba beberapa literatur untuk mengetahui tingkat keberhasilan. Pengumpulan data merupakan pencarian beberapa materi dan literatur penunjang pembuatan alat. Instrumen adalah pemakaian beberapa alat ukur sebagai patokan, standar dan kalibrasi alat. Terakhir teknik analisis data adalah uraian atau analisis pengujian alat berlandaskan teori-teori yang relevan.

KWH-meter digital sistem telemetri merupakan inovasi dari KWH-meter konvensional PLN yang dapat menampilkan daya dan biaya pemakaian energi lsitrik setiap satuan waktu serta dapat dipantau dari jarak jauh. Alat ini lebih akurat dalam pembacaan daya pemakaian listrik, karena penghitungan dilakukan secara program memakai IC mikrokontroller ATmega8535 dan ditampilkan secara digital pada LCD per *watt*. Setiap satuan waktu daya penakaian energi listrik akan dikirim ke stasiun penerima melalui sistem telemetri RF *transceiver* YS1100UB yang akan ditampilkan pada komputer dengan penampil program bahasa antar muka Borland Delphi 7.0.

Pembacaan alat tersebut stabil terhadap beban yang sama dalam kurun waktu tertentu. Tingkat kesalahan pembacaan relatif kecil dan keakurasian penghitungan cukup tinggi. Respon pengiriman data cukup baik, data yang dikirim dan diterima sama pada jarak kurang dari 200 m dengan *baudrate* 9600 bps, frekuensi 433 MHz memakai antena bawaan di dalam ruangan. Kedepannya diharapkan alat dapat digunakan pada beberapa pelanggan dengan tarif berbeda dan alat bisa diakses melalui internet.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN ii
MOTTO DAN PERSEMBAHAN iii
KATA PENGANTAR iv
ABSTRAK v
DAFTAR ISI vi
DAFTAR GAMBAR viii
DAFTAR TABEL x
BAB I PENDAHULUAN 1
A. Latar Belakang 1
B. Rumusan Masalah 2
C. Batasan Masalah 2
D. Tujuan 3
E. Manfaat 3
BAB II PEMBAHASAN 4
A. Landasan Teori 4
1. ACS712 4
2. Mikrokontroller (ATmega8535) 5
a. Diskripsi Pin ATmega8535 6
b. Organisasi Memori 8
3. Liquid Crystal Display (M1632) 9
a. Spesifikasi LCD M1632 10
4. Pemancar dan Penerima 11
a. Mode Transmisi Serial 11
5. Serial to USB (PL2303HX) 14
4. Program Bahasa Antarmuka (Borland Delphi 7.0) 15
a. Mengenal IDE (Integrated Development Environment) Delphi 16
b. Main Window 17

c. Object Inspector 19
d. Code Editor 20
e. Code Explorer 21
f. Object Tree View 22
g. Tipe Data 22
B. Metode atau Prosedur 26
1. Metode Laboratoris 26
2. Teknik Pengumpulan Data 26
3. Instrumen 26
4. Teknik Analisis Data 27
5 Prosedur 27
a. Diagram Blok Rangkaian 27
b. Rangkaian ACS712 28
c. Rangkaian Sistem Minimum Mikrokontroller 29
d. Pemancar dan Penerima 30
e. Penampil Liquid Crystal Display (LCD) 32
f. Rangkaian Serial to USB PL3203HX 33
g. Perancangan Perangkat Lunak Mikrokontroller 33
C. Pengujian Alat 35
D. Pembahasan 38
BAB III PENUTUP 41
1. Kesimpulan 41
2. Saran 41
DAFTAR PUSTAKA 42
LAMPIRAN

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

DAFTAR GAMBAR

2.1 Konfigurasi Pin ACS712 5
2.2 ATmega8535 Minimum System Modul 6
2.3 Konfigurasi Pin ATmega8535 6
2.4 Organisasi Memori ATmega8535 8
2.5 Penampang LCD tampak dari depan 10
2.6 Transmisi serial, serial dalam bit, serial dalam karakter 12
2.7 Contoh transmisi data serial tak sinkron 13
2.8 Penampang Modul Serial to USB PL2303HX 14
2.9 Pin out Diagram Modul Serial to USB PL2303HX 15
2.10 Lembar kerja Borland Delphi 17
2.11 Main menu Borland Delphi 18
2.12 Toolbar 18
2.13 Component palette 18
2.14 Form designer 19
2.15 a. (Tab properties) dan b. (Tab events object inspector) 20
2.16 Code Editor 21
2.17 Code explorer 22
2.18 Object tree view 22
2.19 Diagram blok rangkaian pembuatan alat 27
2.20 Rangkaian pembaca arus IC ACS712 28
2.21 Rangkaian sistem minimum mikrokontroller 29
2.22 YS-1100UB RF Data Transceiver 30
2.23 Definisi pin YS-1100UB RF Data Transceiver 31
2.24 Hubungan port C ATmega8535 dengan LCD 32
2.25 Rangkaian Serial to USB PL2303HX 33
2.26 Flowchart Program 35
2.27 Data Pengujian Alat yang Dipararel dengan KWH-meter PLN 37
2.28 Tampilan LCD Stasiun Pengirim 37

2.29 Tampilan Pada Komputer di Stasiun Penerima 38
2.30 Perbedaan tampilan KWH-meter Digital dengan KWH-meter PLN 39
2.31 Alat penerima dan pengirim (KWH-meter Digital Telemetri) 40

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

DAFTAR TABEL

2.1 Penjelasan pena-pena LCD 10
2.2 Range dan Format Tipe Data Integer Fundamental 23
2.3 Range dan Format Penyimpanan tipe Data Real 23
2.4 Hubungan tipe Boolean dengan ByteBool, WordBool, dan LongBool ... 24
2.5 Macam-Macam Tipe Data String 25
2.6 Metode eksperimen 26
2.7 Pin-Pin YS-1100UB RF Data Transceiver 31
2.8 Data pengujian alat pertama 36
2.9 Data pengujian alat kedua 37

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

BAB I
PENDAHULUAN

A. Latar Belakang

Kemajuan ilmu pengetahuan dan teknologi telah mendorong manusia untuk berusaha mengatasi segala permasalahan yang timbul dan meringankan pekerjaan yang ada (Sapiie, Nishino. 1994), seperti permasalahan sistem admistrasi pembayaran jasa listrik. Sistem administrasi pembayaran jasa listrik dimulai dari kedatangan petugas pencatat daya pemakaian energi listrik dari satu rumah ke rumah yang lain untuk mencatat berapa besar penggunaan daya energi listrik setiap bulannya. Hasil pencatatan tersebut, kemudian akan dilakukan penghitungan dengan mengalikan besar daya pemakaian energi listrik dengan tarif dasar listrik pelanggan.

Proses penghitungan penggunaan daya energi listrik seperti inilah yang sering menimbulkan protes oleh konsumen mengenai nominal rupiah yang harus dibayar setiap bulannya, karena tidak sesuai dengan pemakaian energi listrik yamg sebenarnya. Faktor yang menyebabkan antara lain: proses penghitungan daya listrik yang setiap bulan yang masih manual, sehingga menyebabkan kemungkinan kesalahan entri data yang besar. Protes oleh konsumen semakin lama akan menurunkan tingkat kepercayaan konsumen terhadap penyedia jasa listrik.

Alternatif penyelesaian dalam permasalahan ini adalah penggunaan alat pencatat digital yang mampu mengkonversi nilai penggunaan daya listrik dalam waktu tertentu kedalam nilai nominal rupiah yang akan ditampilkan dalam LCD dan dipantau melalui PC dengan sistem telemetri, sehingga pelanggan bisa mengetahui biaya yang harus dibayar setiap bulan yang ditampilkan di LCD disetiap rumah pelanggan. Petugas pencatat daya pemakaian energi listrik pun tidak perlu mendatangi rumah-rumah pelanggan, tetapi cukup melihat melalui PC di stasiun pusat *(server)*.

KWH-meter digital sistem telemetri merupakan solusi yang tepat untuk merealisasikan penggunaan alat pencatat digital yang dapat menampilkan besar

daya pemakaian energi listrik pelanggan, biaya pemakaian energi listrik serta daya dan biaya pemakaian tersebut dapat dipantau secara jarak jauh tanpa memakai kabel. Alat ini merupakan inovasi atau pengembangan dari KWH-meter konvensional milik PLN yang sudah umum dipakai di rumah-rumah.

Telemetri merupakan salah satu metode pengukuran yang dilakukan tanpa menyentuh obyek. Telemetri sebenarnya adalah salah satu bentuk pengembangan teknologi telekomunikasi. Telekomunikasi sendiri dapat diartikan sebagai hubungan komunikasi jarak jauh dengan menggunakan sinyal-sinyal listrik. Unsur-unsur yang terdapat dalam telekomunikasi antara lain: informasi (data), media komunikasi, jarak, metode komunikasi dan waktu.

Pemrograman Borland Delphi dapat dibuat suatu program antarmuka perangkat keras (*hardware*) dengan perangkat lunak (*software*). Tampilan hasil biaya pemakaian energi listrik oleh pelanggan setiap hari sampai satu bulannya dapat dilihat langsung pada PC di stasiun pusat *(server)* dengan aplikasi Delphi tersebut tanpa harus memantau langsung ke obyek.

B. Rumusan Masalah

Berdasarkan uraian dalam latar belakang tersebut maka munculah permasalahan bagaimana unjuk kerja dari KWH-meter digital berbasis sistem telemetri ini, yang meliputi kehandalan pengiriman data, respon, tingkat kesalahan dan keakurasian penghitungan, serta jangkauan pengiriman data?

C. Batasan Masalah

Pembatasan masalah dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. KWH-meter digital berbasis sistem telemetri ini digunakan untuk pelanggan rumah tangga daya satu fasa.
2. Penghitungan konversi daya listrik ke nominal rupiah menggunakan IC mikrokontroller ATmega8535.
3. Daya dan biaya pemakian energi listrik di pelanggan ditampilkan pada LCD.

4. Sistem telemetri dua arah menggunakan modul *transmitter* dan *receiver* YS-1100U RF *Data Transceiver* sebagai pemancar dan penerima.
5. *Interfacing* ke komputer menggunakan protokol serial ke USB *(serial to USB).*
6. KWH-meter digital berbasis sistem telemetri ini hanya untuk pemakaian tunggal (satu pelanggan).

D. Tujuan

Adapun tujuan dari penyusunan Tugas Akhir ini adalah untuk mengetahui bagaimana unjuk kerja KWH meter digital, seperti; kehandalan pengiriman data, respon, tingkat kesalahan dan keakurasian penghitungan, serta jangkauan pengiriman data.

E. Manfaat

Beberapa manfaat dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Mengurangi kesalahan pencatatan dan penghitungan pemakaian daya listrik secara manual.
2. Mempermudah dan mempercepat pengiriman informasi pemakaian energi listrik.
3. Membantu konsumen dalam pengontrolan pemakaian daya energi listrik, karena dapat dipantau setiap saat.
4. Menghemat tenaga dan waktu petugas PLN dalam melakukan pencatatan daya listrik yang terpakai pelanggan.

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

BAB II
PEMBAHASAN

A. Landasan Teori

1. ACS712

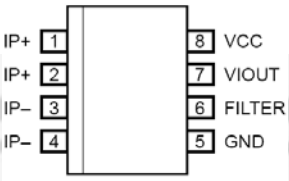
IC ACS712 adalah sebuah IC yang dapat mendeteksi arus searah maupun arus bolak balik. Prinsip kerja dari ACS712 hampir sama dengan rangkaian *hall effect* lainnya yaitu dengan memanfaatkan medan magnetik disekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Trafo *step down* digunakan sebagai pendeteksi perubahan tegangan pada beban. Arus bolak-balik atau *altenating current* (AC) dari jala-jala listrik PLN diturunkan ke tegangan yang lebih rendah dan arus yang lebih kecil, diikuti proses penyearahan dan filterisasi. Nilai varibel dari ACS712 dan trafo merupakan masukan bagi mikrokontroler, maka dapat dihitung besar daya yang digunakan. Perhitungan dilakukan di dalam mikrokontroler yaitu dengan mengalikan secara program nilai arus dan tegangan yang terdeteksi.

IC ACS712 yang dipakai pada tugas akhir ini adalah tipe ACS712ELCTR-05B-T, dimana IC ini dapat mendeteksi arus hingga 5A dengan sensitivitas rata-rata 185mV/A (www.allegromicro.com, 2011). Prinsip kerja dari ACS712 hampir sama dengan sensor *hall effect* yaitu dengan memanfaatkan medan magnetik di sekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Arus boalk-balik (AC) dari jala-jala listrik PLN diturunkan ke tegangan yang lebih rendah dan arus yang lebih kecil, diikuti proses penyearahan dan filterisasi.

Pengukuran arus biasanya membutuhkan sebuah resistor shunt yaitu resistor yang dihubungkan secara seri pada beban dan mengubah aliran arus menjadi tegangan. Tegangan tersebut biasanya diumpankan ke transformator arus (CT) terlebih dahulu sebelum masuk ke rangkaian pengkondisi sinyal.

Teknologi *hall effect* yang diterapkan pada IC ini menggantikan fungsi resistor *shunt* dan transformator arus menjadi sebuah sensor dengan ukuran yang relatif jauh lebih kecil. Aliran arus listrik yang mengakibatkan medan magnet dan

menginduksi bagian *dynamic offset cancellation* dari ACS712ELCTR-05B-T. Bagian ini akan dikuatkan oleh bagian amplifier dan melalui proses filter sebelum dikeluarkan melalui pin 6 dan 7. Gambar 2.1 memperlihatkan gambar konfigurasi pin IC ACS712.

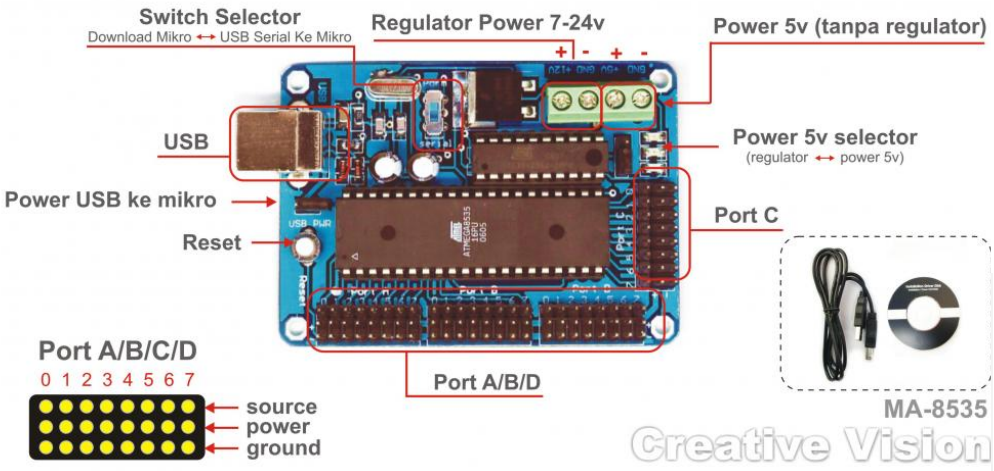


Gambar 2.1 Konfigurasi Pin ACS712

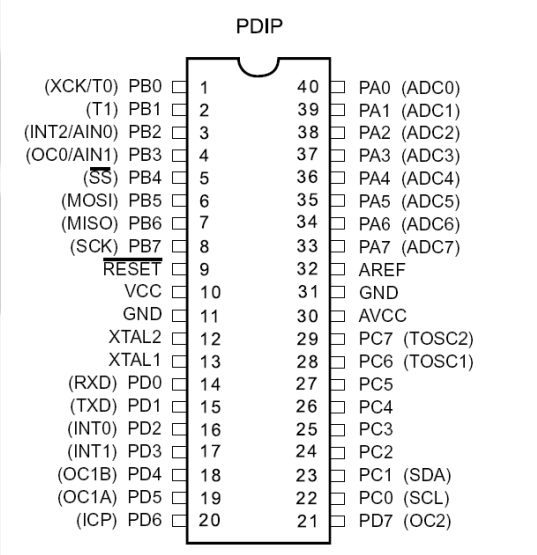
2. Mikrokontroller (ATmega8535)

AVR *(Alf, Vegard and RISC)* merupakan seri mikrokontroller CMOS *(Complementary Metal Oxide Semiconductor)* 8-bit buatan Atmel, berbasis arsitektur RISC *(Reduced Instruction Set Computer)*. Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 *register general-purpose*, *timer/counter fleksibel* dengan mode *compare*, *interupt internal* dan *eksternal*, serial UART *(Universal Aysnchronous Receiver Trasmitter)*, *programmable Watchdog Timer*, dan mode *power saving*. Beberapa diantaranya mempunyai ADC *(Analog to Digital Converter)* dan PWM *(Pulse With Modulating)* internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI *(Serial Protocol Input)*. Chip AVR yang digunakan untuk tugas akhir ini adalah ATmega8535. ATmega8535 adalah mikrokontroller CMOS8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan.

Mikrokontroller ATmega8535 digunakan pada tugas akhir ini dikarenakan mempunyai memori yang cukup untuk menyimpan program yang digunakan dan juga mempunyai banyak *port* yang sesuai untuk rangkaian *relay*. Berikut gambar 2.2 dan 2.3 merupakan modul sistem minimum ATmega8535 yang digunakan dan konfigurasi pin ATmega8535 tersebut.



Gambar 2.2 ATmega8535 *Minimum System Modul (Creative Vision, 2010)*



Gambar 2.3 Konfigurasi Pin ATmega8535

a. Deskripsi Pin ATmega8535

1) VCC *(power supply)*
2) GND *(ground)*
3) Port A (PA7 – PA0)

Port A berfungsi sebagai *input* analog pada A/D Konverter. Port Ajuga berfungsi sebagai suatu Port I/O 8-bit dua arah, jika A/D Konverter tidak digunakan. Pin-pin Port dapat menyediakan resistor *internal pull-up*

(yang dipilih untuk masing-masing bit). Port A *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai *input* dan secara *eksternal* ditarik rendah, pin-pin akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pin Port A adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

4) Port B (PB7 – PB0)

Port B adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port B *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin port B yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin PortB adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

5) Port C (PC7 – PC0)

Port C adalah suatu Port I/O 8-bit dua arah dengan resistor internal *pull-up* (yang dipilih untuk beberapa bit). Port C *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin port C yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin PortC adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

6) Port D (PD7 – PD0)

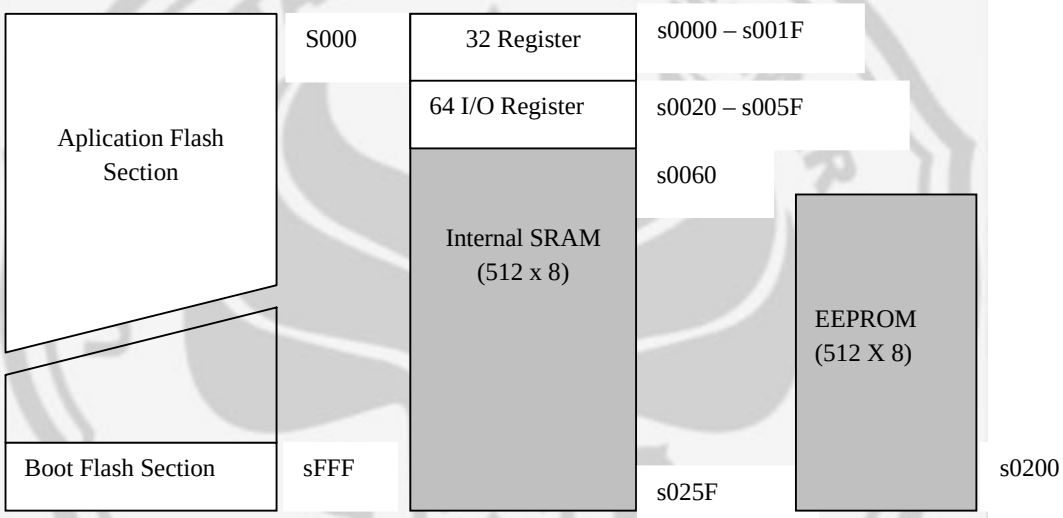
Port D adalah suatu Port I/O 8-bit dua arah dengan resistor internal pull-up (yang dipilih untuk beberapa bit). Port D *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin port D yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin PortD adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN UNNES

7) RESET *(Reset input)*
8) XTAL1 *(Input Oscillator)*
9) XTAL2 *(Output Oscillator)*

b. Organisasi Memori

Mikrokontroller ATMega 8535 memiliki 3 jenis memori yaitu memori program, memori data dan memori EEPROM *(Electrically Erasble Program Read-Only Memory)*. Ketiganya memilki ruang sendiri dan terpisah seperti terlihat pada gambar 2.4 (Bejo, 2008).



Gambar 2.4 Organisasi Memori ATmega8535

1) Memori Program

ATmega8535 memiliki 32 *Kilobyte flash memory* untuk menyimpan program. Lebar intruksi 16 bit atau 32 bit maka *flash memory* dibuat berukuran 16K x 16K. Artinya ada 16K alamat pada *flash memory* yang bisa dipakai, dimulai dari alamat 0 heksa sampai alamat 3FFF heksa dan setiap alamatnya menyimpan 16 *bit* instruksi.

2) Memori Data

ATmega8535 memiliki 2 *Kilobyte* SRAM *(Static Random Acces memory)*. Memori ini dipakai untuk menyimpan variabel. Tempat khusus di SRAM yang senantiasa ditunjuk register SP disebut *stack*. *Stack* berfungsi untuk menyimpan nilai yang di masukan.

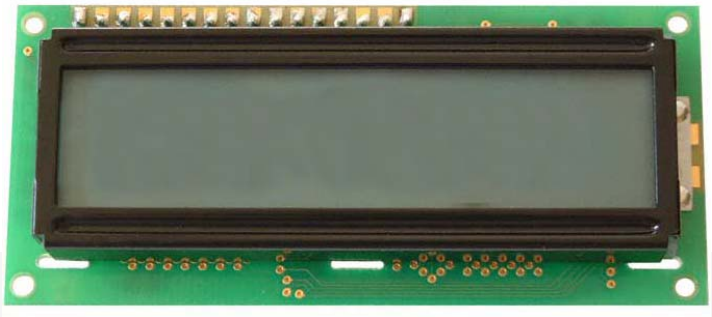
3) Memori EEPROM

ATmega8535 memiliki 1024 *byte* data EEPROM. Data di EEPROM tidak akan hilang walaupun *catu daya* ke sistem mati. Parameter sistem yang penting disimpan di EEPROM. Saat sistem pertama kali menyala paramater tersebut dibaca dan system diinisialisasi sesuai dengan nilai parameter tersebut (Bejo, 2008).

3. *Liquid Crystal Display* (M1632)

Modul *display* LCD dibuat dalam bentuk pengontrol LSI *(large scale integration)*, dimana pengontrol ini mempunyai dua register masing-masing 8-bit, yaitu *Instruction Register (IR)* dan *Data Register (DR)*. IR menyimpan kode-kode instruksi yang berupa: bersihkan layar, geser kursor, kursor kembali, kontrol display *ON/OFF* dan alamat informasi untuk *Display Data RAM (DDRAM)* dan *Character Generator (CGRAM)*. *IR* hanya dapat ditulis dari MPU. Register data (DR) kadang kala menyimpan data yang akan ditulis atau dibaca dari *DDRAM* atau *CGRAM*. Ketika alamat informasi ditulis dalam *IR*, data disimpan kedalam *DR* dari *DDRAM* atau *CGRAM*. Menggunakan sinyal *Register Selector (RS)*, dua register tersebut dapat terseleksi.

LCD *display module* M1632 terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing-masing baris bisa menampung 16 huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempel dibalik pada panel LCD, berfungsi mengatur tampilan LCD. Pemakaian LCD M1632 menjadi sederhana, sistem lain cukup mengirimkan kode-kode ASCII dari informasi yang ditampilkan. Gambar 2.5 memperlihatkan pena-pena LCD dan penjelasannya ada pada tabel 2.1.



Gambar 2.5 Penampang LCD tampak dari depan

Tabel 2.1 Penjelasan pena-pena LCD

No. Pena	Nama Pena	Penjelasan
1	Vss	Catu daya Gnd 0 V
2	V_{DD}	Catu daya + 5V
3	Vo	Untuk mengatur kekontrasan LCD
4	RS	Sinyal pemilih register, bila 0 sebagai data masukan, 1 sebagai instruksi masukan
5	R/W	Sinyal pemilih baca (R) atau tulis (W), 0 = tulis, 1 = baca
6	E	Untuk mengaktifkan sinyal instruksi
7 – 14	$DB_0 – DB_7$	Sebagai masukan atau keluaran data
15	A	Catu daya positif lampu belakang *(back light)*
16	K	Catu daya negatif (GND) lampu belakang *(back light)*

a. Spesifikasi LCD M1632

1) Tampilan 16 karakter2 baris dengan matrik 5 x 7 + kursor.
2) ROM pembangkit karakter 192 jenis.
3) RAM pembangkit karakter 8 jenis (diprogram pemakai).
4) RAM data tampilan 80 x 8 bit (8 karakter).
5) *Duty ratio* 1/16.
6) RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit mikroprosesor.
7) Beberapa fungsi perintah antara lain: penghapusan tampilan (*display clear*), posisi krusor awal (*cursor home*), tampilan karakter kedip (*display character blink*), pengeseran krusor (*cursor shift*), dan penggeseran tampilan (*display shift*).

8) Rangkaian pembangkit detak.
9) Rangkaian otomatis reset saat daya dinyalakan.
10) Catu daya tunggal + 5 volt.

4. Pemancar dan Penerima

Pemancar merupakan suatu pesawat yang digunakan untuk memancarkan informasi kepada penerima untuk keperluan tertentu. Gelombang modulasi dapat diperoleh dari penumpangan frekuensi informasi pada frekuensi *carrier*, pada FM amplitudo dari frekuensi informasi mengubah-ubah frekuensi *carrier*. Dalam sistem modulasi frekuensi, amplitudo dari sinyal pembawa dibuat konstan, sedangkan frekuensinya dinamis sebanding dengan sinyal yang memodulasinya.

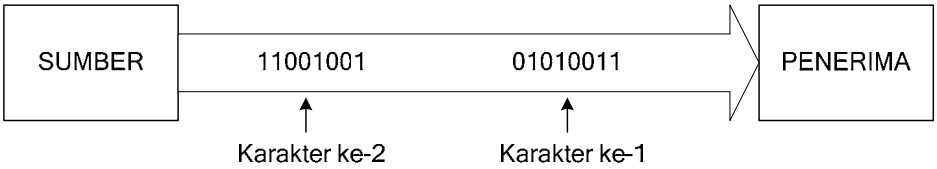
Pesawat radio penerima merupakan suatu pesawat yang dipergunakan untuk menerima frekuensi radio yang dipancarkan oleh pemancar, dan selanjutnya diubah menjadi frekuensi rendah yang dapat diterima dan diproses kembali menjadi data-data dengan baik.

Tipe transmisi pemancar dan penerima yang diapakai yaitu tipe radio *dueplex* (dua arah). Pemancar dan penerima ini bisa dilakukan dalam dua arah bolak balik. Tipe transmisi dua arah merupakan jalur transmisi yang dapat membawa informasi data dalam dua arah atau bolak balik. Sinyal yang dikirim dari stasiun pemancar dapat diterima oleh pesawat penangkap siaran, dan penangkap siaran dapat mengirimkan kembali informasi ke stasiun pemancar.

PERPUSTAKAAN UNNES

a. Mode Transmisi Serial

Transmisi secara serial merupakan mode transmisi yang umum dipergunakan. Pada mode ini, masing-masing bit pada suatu karakter dikirim secara berurutan, yaitu bit per bit, satu bit diikuti bit berikutnya. Penerima kemudian merakit kembali arus bit-bit yang datang ke dalam bentuk karakter. Mode transmisi serial ini dapat berbentuk transmisi yang tak tersinkronisasi. Gambar 2.6 menunjukkan transmisi serial, serial dalam bit, serial dalam karakter.



Gambar 2.6 Transmisi serial, serial dalam bit, serial dalam karakter

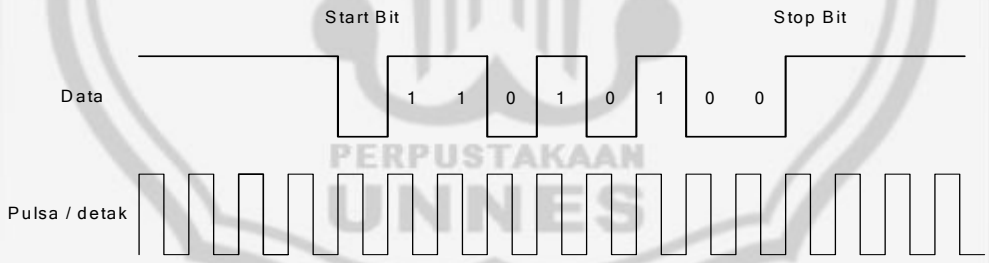
Pada transmisi tersinkronisasi waktu pengiriman bit-bit di sumber pengiriman harus sinkron (sesuai) dengan waktu penerimaan bit-bit yang diterima oleh penerima. Transmisi data yang menggunakan transmisi tersinkronisasi menghadapi permasalahan dalam sinkronisasi yang berhubungan dengan sinkronisasi bit dan karakter yang dikirim dengan yang diterima. Sinkronisasi bit berhubungan dengan waktu kapan sumber informasi harus mengirimkan bit-bit ke jalur transmisi dan kapan penerima harus mengetahui dengan tepat untuk mengambil bit-bit yang dikirim tersebut.

Permasalahan tersebut dapat teratasi dengan *clock* yang ada disumber pengirim dan *clock* yang ada pada penerima. *Clock* yang ada di sumber pengirim akan memberitahu sumber kapan harus meletakkan bit-bit yang akan dikirim ke jalur transmisi dan *clock* yang ada di penerima akan memberitahu kapan harus mengambil bit-bit yang dikirim. Misalnya kalau diinginkan untuk mengirim dengan kapasitas 100 bps, *clock* di sumber harus diatur untuk bekerja dengan kecepatan 100 bps dan *clock* di penerima harus diatur untuk mengambil dari jalur transmisi 100 kali setiap detiknya, sehingga bit-bit yang dikirim akan sinkron dengan bit-bit yang diterima.

Metode pengaturan *clock* di sumber dan di penerima dapat mengatasi masalah sinkronisasi bit, namun masih timbul permasalahan yang lain, yaitu sinkronisasi karakter. Permasalahan ini berupa penentuan sejumlah bit-bit mana saja yang merupakan bentuk sebuah karakter. Permasalahan ini dapat diatasi dengan mendahulukan masing-masing blok data yang hendak dikirim dengan suatu bentuk karakter kontrol transmisi tertentu. Bentuk karakter kontrol transmisi pada kode ASCII *(American Standard Code for Information Interchange)* tersebut adalah SYN *(Syncronous)* dengan bentuk dalam bilangan adalah 00010110,

umumnya dua atau lebih karakter kontrol SYN diletakkan di muka blok data yang akan dikirimkan. Karakter kontrol transmisi SYN yang digunakan harus lebih dari satu, karena untuk menghindari kemungkinan terjadi kesalahan sinkronisasi. Dua buah karakter kontrol SYN dapat digunakan di awal blok data yang ditransmisikan. Penerima, setelah mengidentifikasikan bentuk SYN yang pertama, kemudian mengidentifikasikan delapan bit berikutnya, kalau berupa karakter kontrol SYN yang kedua, maka dimulai menghitung tiap-tiap delapan bit menjadi sebuah karakter.

Transmisi tak tersinkronisasi merupakan transmisi dari data yang ditransmisikan satu karakter setiap waktu yang tertentu. Pengirim dapat mentransmisikan karakter-karakter pada interval waktu yang berbeda atau dalam kata lain tidak harus dalam waktu yang sinkron antara pengiriman satu karakter dengan karakter berikutnya. Tiap-tiap karakter dikirimkan sebagai satu kesatuan yang berdiri sendiri dan penerima harus dapat mengenal masing-masing karakter tersebut. Masing-masing karakter harus diawali dengan bit-bit tambahan, yaitu *start bit* atau *start pulse* yang berupa nilai bit 0 dan *stop bit* atau *stop pulse* yang berupa nilai bit 1 diletakkan pada akhir dari masing-masing karakter, agar tidak terjadi kesalahan pembaaan karakter yang dikirimkan.



Gambar 2.7 Contoh transmisi data serial tak sinkron

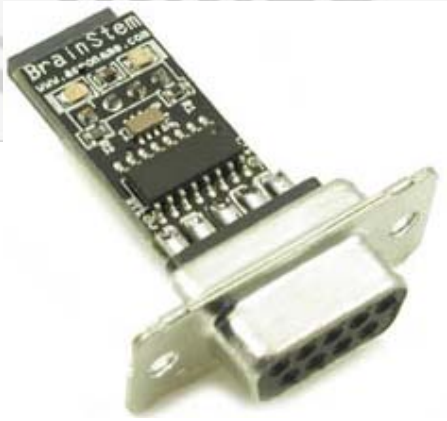
Terlihat pada gambar 2.7, setiap karakter diawali dengan *start bit* dan diakhiri dengan *stop bit*, sehingga transmisi tak tersinkronisasi disebut juga dengan transmisi *star/stop*. Transmisi tak tersinkronisasi lebih aman dibandingkan dengan transmisi tersinkronisasi. Pada transmisi tak tersinkronisasi bila terjadi kesalahan pada data yang ditransmisikan, hanya akan merusak sebuah karakter saja, sedangkan pada transmisi tersinkronisasi akan merusak satu blok dari data.

Namun transmisi tak tersinkronisasi kurang efisien bila dibandingkan dengan transmisi tersinkronisasi karena diperlukannya tambahan untuk tiap-tiap karakter, yaitu *start bit* dan *stop bit* (Rahman, 2003).

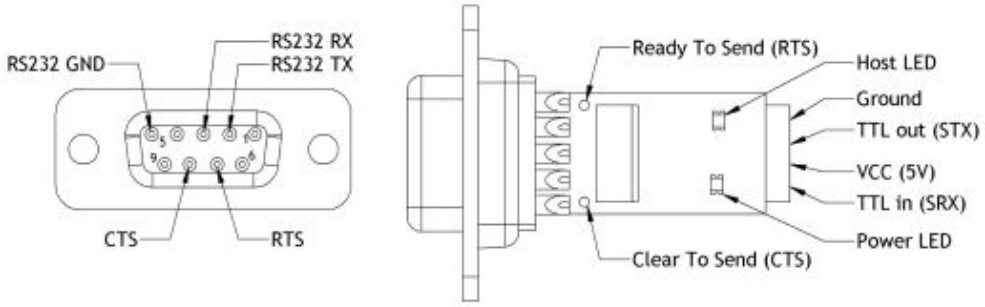
5. *Serial to USB* (PL2303HX)

Komunikasi serial adalah pengiriman data secara serial (data dikirim satu per satu secara berurutan). Serial port lebih sulit ditangani karena peralatan yang dihubungkan ke serial port harus berkomunikasi dengan menggunakan transmisi serial, sedangkan data di komputer diolah secara paralel. Data dari dan ke serial port harus dikonversikan ke dan dari bentuk paralel terlebih dahulu, agar bisa digunakan.

Sekarang ini sangat sulit menemukan laptop yang memakai port serial maupun pararel. Kebanyakan hanya memiliki port USB, sehingga perlu mengkonversi dari serial ke USB. *Serial to USB* adalah suatu perangkat yang bisa menjembatani pengiriman data dari *hardware* (Mikrokontroller) ke laptop. *Serial to USB* PL2303HX merupakan salah satu konverter serial ke USB yang memiliki keunggulan seperti; TTL atau CMOS untuk shifter RS232, 5V dan 3.3V TTL toleran, standar DB9 konektor, memenuhi persyaratan S*tandar Acroname Serial Interface*, LED indikator koneksi, Power LED status, support pada semua OS komputer (windows XP, Win7, Vista, Mac OS, dll) (www.acroname.com, 2011). Berikut gambar 2.8 adalah penampang modul *Serial to USB* PL2303HX dan 2.9 Pin out diagram.



Gambar 2.8 Penampang Modul *Serial to USB* PL2303HX



Gambar 2.9 Pin out Diagram Modul *Serial to USB* PL2303HX

6. Program Bahasa Antarmuka (Borland Delphi 7.0)

Borland Delphi adalah suatu bahasa pemrograman visual dengan menggunakan bahasa pemrograman Pascal. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas. Berbagai jenis aplikasi dapat dibuat menggunakan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, basis data, dan aplikasi web. Bahkan dengan *source code* tertentu, Borland Delphi dapat digunakan untuk menggerakkan perangkat keras (*hardware*).

Untuk mempermudah pemrograman dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang lengkap. Fasilitas pemrograman itu terbagi atas dua kelompok, yaitu *object* dan bahasa pemrograman. *Object* adalah suatu komponen yang mempunyai bentuk fisik dan dapat dilihat *(visual)*. *Object* biasanya dipakai untuk melakukan tugas tertentu dan memiliki batasan-batasan tertentu. Sedangkan bahasa pemrograman adalah sekumpulan teks yang memiliki arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu.

Program antarmuka dapat menjalankan transfer data masukan atau keluaran melalui bus data, bus alamat dan bus kontrol. Dalam hal ini diperlukan program yang menginstruksikan perangkat keras untuk melakukan kegiatan tersebut. Program yang akan disusun menggunakan bahasa pemrograman Delphi 7.0.

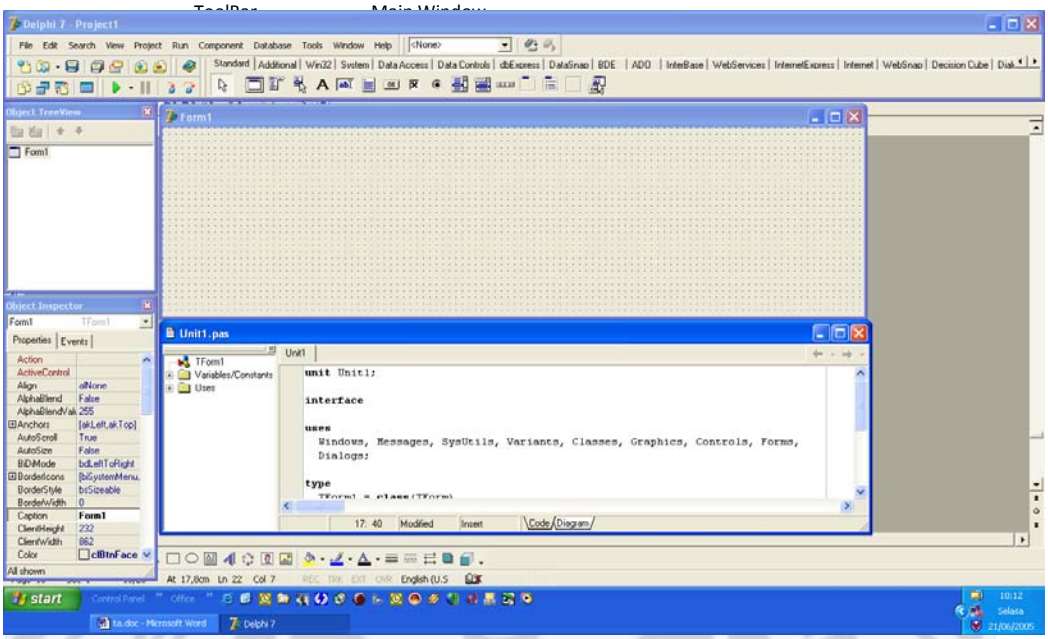
Delphi merupakan perangkat pengembangan aplikasi yang sangat terkenal di lingkungan Windows. Dengan menggunakan program ini maka dapat dibangun berbagai aplikasi Windows dengan cepat dan mudah. Delphi menggunakan bahasa Pascal sebagai bahasa dasar. Dengan pendekatan visual, maka dapat diciptakan aplikasi yang canggih tanpa banyak menulis kode. Delphi mengandung komponen-komponen siap pakai, sehingga akan mengurangi penulisan program dan lebih efektif dalam pembuatan aplikasi (Kadir, 2000).

Delphi 7.0 menyediakan fasilitas pemrograman yang lebih lengkap bila dibandingkan dengan versi pendahulunya, fasilitas pemrograman tersebut dibagi dalam dua kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). *Object* biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa program yaitu sekumpulan text yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu (Agus, 2001).

Untuk program database, Delphi menyediakan object yang sangat kuat, canggih dan lengkap. Format *database* yang digunakan untuk program aplikasi ini adalah paradox yang merupakan produk asli dari Delphi selain dBase. Delphi juga dapat menangani data dalam berbagai format *database*, misalnya format *Ms Access*, *SyBase*, *Oracle*, *FoxPro*, *Informix*, *DB2*, dan lain-lain.

a. Mengenal IDE *(Integrated Development Environment)* Delphi

Lingkungan pengembangan terpadu atau *Integrated Development Environment* (IDE) dalam program Delphi terbagi menjadi delapan bagian utama, yaitu *Main Window*, *Toolbar*, *Component Palette*, *Form Designer*, *Code Editor*, *Object Inspector*, *Exploring*, dan *Object Tree View* terlihat jelas seperti pada gambar 2.10.



Gambar 2.10 Lembar kerja Borland Delphi

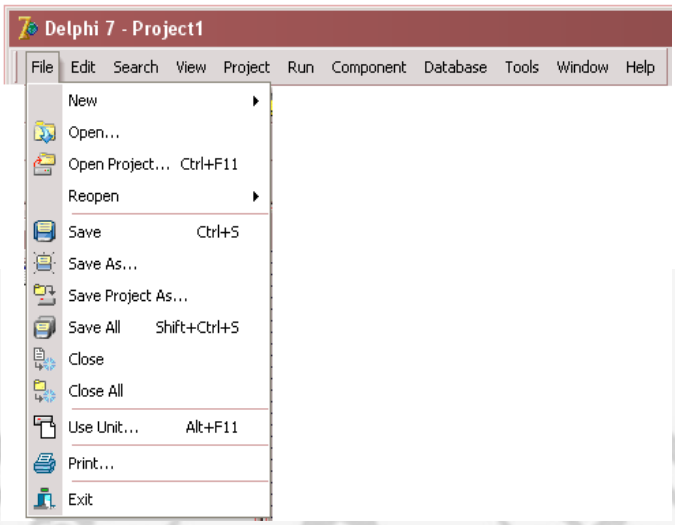
IDE merupakan sebuah lingkungan dimana semua tombol perintah yang diperlukan untuk mendesain aplikasi, menjalankan dan menguji sebuah aplikasi disajikan dengan baik untuk memudahkan pengembangan program.

b. *Main Windows*

Jendela utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, berupa *Main Menu*, *Toolbar* dan *Component Palette*.

1) *Main Menu*

Menu utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi *Windows* lainnya. Dengan menggunakan fasilitas menu program dapat dipanggil atau disimpan, seperti diperlihatkan pada gambar 2.11.



Gambar 2.11 *Main menu* Borland Delphi

2) *Toolbar*

Delphi memiliki beberapa *toolbar* yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian yang berfungsi sebagai pengganti suatu menu perintah yang sering digunakan. *Toolbar* sering disebut juga *Speedbar*, gambar 2.12.

Gambar 2.12 *Toolbar*

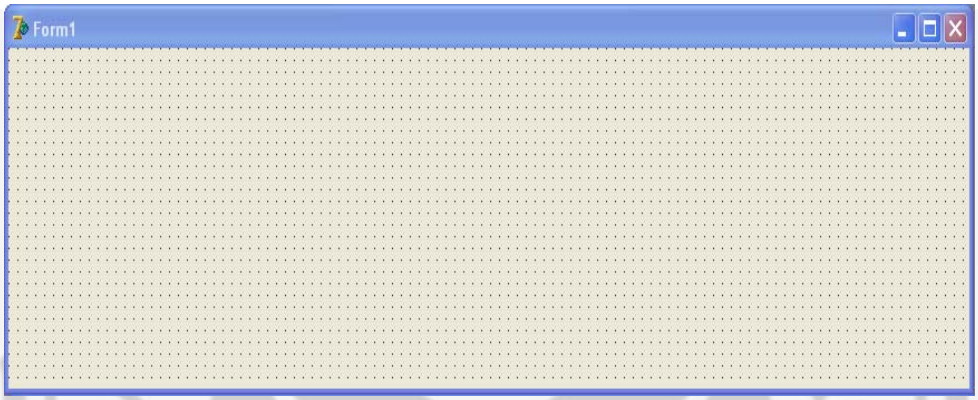
3) *Component Palette*

Component Palette berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada VCL *(Visual Component Library)*. Icon tombol pointer terdapat di setiap *page control*, gambar 2.13.

Gambar 2.13 *Component palette*

4) *Form Designer*

Form designer, terlihat seperti pada gambar 2.14 merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari Component Palette.



Gambar 2.14 *Form designer*

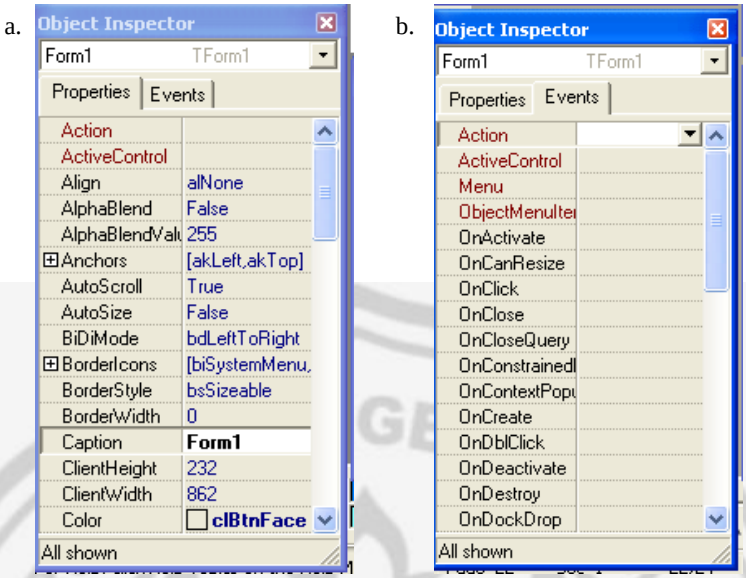
c. *Object Inspector*

Object Inspector digunakan untuk mengubah properti atau karakteristik dari sebuah komponen. *Object Inspector* terdiri dari dua tab, yaitu *Properties* dan *Event*.

1) *Tab Properties*

Tab *Properties*, gambar 2.15a, digunakan untuk mengubah properti komponen. Properti dengan tanda "+" menunjukkan bahwa properti tersebut mempunyai subproperti. Klik tanda + untuk membuka subproperti.

PERPUSTAKAAN UNNES



Gambar 2.15a. (*Tab properties*) dan b. (*Tab events object inspector*)

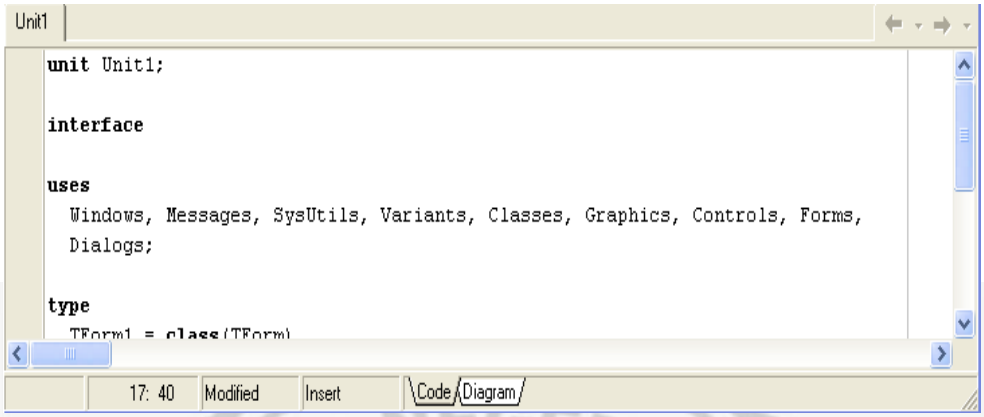
2) *TabEvent*

Bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani *event-event* (kejadian-kejadian yang berupa sebuah *procedure*) yang dapat direspon oleh sebuah komponen. Sebagai contoh jika ingin suatu kejadian akan dikerjakan saat komponen diklik, maka program dapat dituliskan pada bagian *OnClick* seperti diperlihatkan gambar 2.15b.

d. *Code Editor*

Code Editor merupakan tempat dimana kode program dituliskan. Pada bagian ini pernyataan-pernyataan dalam *Object Pascal*. Gambar 2.16 merupakan gambar *code editor* yang berisi peintah-perintah yang sama dengan bahasa pemrograman *Pascal*.

PERPUSTAKAAN UNNES



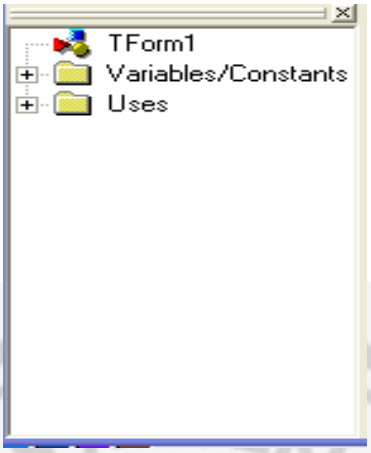
Gambar 2.16 *Code Editor*

Tittle bar yang terletak pada bagian atas jendela *code editor* menunjukkan nama file yang sedang disunting, serta pada bagian bawah terdapat tiga bagian informasi yang perlu untuk diperhatikan, yaitu:

1) Nomor baris/kolom yang terletak pada bagian paling kiri. Bagian ini berfungsi untuk menunjukkan posisi kursor di dalam jendela *code editor*.
2) *Modified* menunjukkan bahwa *file* yang sedang disunting mengalami perubahan dan perubahan tersebut belum disimpan. Teks ini akan hilang jika perubahan disimpan.
3) *Insert/Overwrite* yang terletak pada bagian paling kanan menunjukkan modus pengetikan teks pada jendela *code editor*. Insert menunjukkan bahwa modus penyisipan teks dalam kedaan aktif, sedang *Overwrite* menunjukkan bahwa modus penimpaan teks dalam keadaan aktif.

e. *Code Explorer*

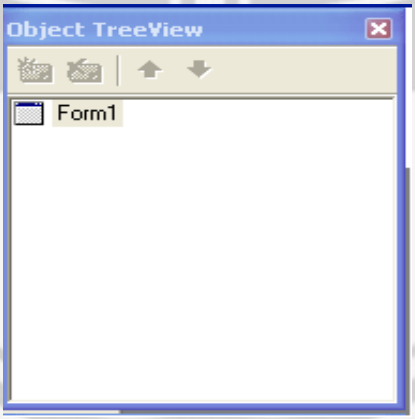
Jendela *code explorer* adalah lembar kerja baru yang terdapat di dalam Delphi 7 yang tidak ditemukan pada versi-versi sebelumnya. *Code Explorer* digunakan untuk memudahkan pemakai berpindah antar file unit yang terdapat di dalam *code explorer*,gambar 2.17.



Gambar 2.17 *Code explorer*

f. *Object Tree View*

Object tree view, seperti yang terlihat pada gambar 2.18 menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. *Object tree view* juga menampilkan hubungan logika antar komponen.



Gambar 2.18 *Object tree view*

g. Tipe *Data*

Borland Delphi 7.0 mempunyai tipe data yang dibagi dalam tujuh kelompok dasar, yaitu *integer*, *real*, *boolean*, *character*, *string*, *pointer*, dan *variant* (Wahana Komputer, 2003). Berikut ini adalah penjelasan mengenai tipe-tipe data.

1) Tipe *Integer*

Tipe *Integer fundamental* terdiri atas *Shortint, Smallint, Longint, Int64, Byte, Word,* dan *Longword.* Tabel 2.2 memperlihatkan range dan format penyimpanan masing-masing tipe data tersebut (Komputer Wahana, 2003).

Tabel 2.2 *Range* dan Format Tipe Data *Integer Fundamental*

Tipe Data	*Range*	Format
Shortint	-128…127	8 bit, bertanda
Smallint	-32768…32767	16 bit, bertanda
Longint	-2147483648…2147483647	32 bit, bertanda
Int64	-2^63…2^63-1	64 bit, bertanda
Byte	0…255	8 bit, bertanda
Word	0…65535	16 bit, bertanda
Longword	0…4294967295	32 bit, bertanda

2) Tipe *Real*

Tipe data *real* menyatakan himpunan bilangan yang dapat dinyatakan dengan notasi *floating point.* Tabel 2.3 memperlihatkan range dan format penyimpanan tipe *real fundamental* (Komputer Wahana, 2003).

Tabel 2.3 *Range* dan Format Penyimpanan tipe Data *Real*

Tipe Data	*Range*	Digit	Ukuran (*Byte*)
Real48	2.9x10^-39…1.7x10^38	11-12	6
Single	1.5x10^-45…3.4x10^38	7-8	4
Double	5.0x10^-324…1.7x10^308	15-16	8
Extended	3.6x10^-4951…1.1x10^4932	19-20	10
Comp	-2^63+1…2^63-1	19-20	8
Currency	-922337203685477.5808…	19-20	8

3) Tipe *Boolean*

Terdapat empat tipe data *Boolean* yaitu *Boolean, ByteBool, WordBool,* dan *LongBool*. Sebuah variabel *Boolean* menempati satu *byte* memori, variabel *ByteBool* juga menempati satu *byte*, variabel *WordBool* menempati dua *byte* (satu *word*), dan variabel *LongBool* menempati empat *byte* (dua *word*). Nilai *Boolean* ditunjukkan dengan *true* dan *false* (Komputer Wahana, 2003) terlihat pada tabel 2.4.

Tabel 2.4 Hubungan tipe *Boolean* dengan *ByteBool, WordBool, dan LongBool*

Boolean	***ByteBool, WordBool, LongBool***
False<*True*	*False*><*True*
Ord (*False*) = 0	*Ord* (*False*) = 0
Ord (*True*) = 1	*Ord* (*True*) >< 0
Succ (*False*) = *True*	*Succ* (*False*) = *True*
Pred (*True*) = *False*	*Pred* (*False*) = *True*

4) Tipe *Character*

Tipe *character fundamental* adalah *AnsiChar* dan *WideChar*. Nilai *AnsiChar* adalah karakter berukuran *byte* (8 bit) yang diurutkan menurut himpunan *character* lokal, yang mungkin berupa *multibyte* (Wahana Komputer, 2003).

Karakter *WideChar* menggunakan lebih dari satu *byte* untuk menyatakan setiap karakter. Dalam implementasi ini, karakter *WideChar* berukuran *word* (16 bit) yang diurutkan berdasarkan himpunan karakter *Unicode*. Karakter *Unicode* 256 pertama berkaitan dengan karakter ANSI (Wahana Komputer, 2003).

Tipe *character generic* adalah *char*, yang ekivalen dengan *AnsiChar*. Karena implementasi *char* cenderung berubah, disarankan menggunakan fungsi standar *SizeOf* daripada menggunakan konstanta yang sulit dikodekan ketika menulis program yang memerlukan penanganan karakter dengan ukuran yang berbeda-beda (Wahana Komputer, 2003).

5) Tipe *String*

Tipe *string* merepresentasikan rangkaian karakter. *AnsiString*, yang seringkali disebut dengan *long string*, adalah tipe yang dipilih untuk sebagian besar pemakaian. Tipe *string* dapat dicampur pada penerapan nilai dan ekspresi. Kompiler ini secara otomatis membentuk konversi yang diperlukan. *String* yang dilewatkan *by reference* pada fungsi atau prosedur haruslah tipe yang sesuai. *String* dapat berlaku secara eksplisit menjadi tipe *string* yang berbeda. Delphi mendukung tipe string yang didefinisikan seperti pada tabel 2.5 (Wahana Komputer, 2003).

Tabel 2.5 Macam-Macam Tipe Data *String*

Tipe Data	Panjang Maksimum	Kebutuhan Memori	Dipakai untuk
ShortString	255 karakter	2 sampai 256 byte	Kompabilitas
AnsiString	-2^31 karakter	4 byte sampai 2GB	8 bit (ANSI) *character*, DBCS ANSI, MBCS ANSI, dll
WideString	-2^30 karakter	4 byte sampai 2GB	*Unicode characters, multi user server and aplikasi multi language*

6) Tipe *Pointer*

Pointer adalah sebuah variabel yang menunjukkan sebuah alamat memori. Ketika sebuah *pointer* berisi alamat variabel lainnya, dapat dikatakan ia menunjuk ke lokasi variabel tersebut di memori atau ke data yang disimpan di sana. Contoh kasus sebuah *array* atau tipe terstruktur lainnya, *pointer* berisi alamat elemen pertama dalam struktur tersebut. Alamat tadi telah diambil, *pointer* tersebut mengandung alamat elemen pertama tersebut (Wahana Komputer, 2003).

Pointer dipakai untuk mengindikasikan jenis data yang disimpan pada alamat yang dikandungnya. Penggunaan umum tipe *pointer* dapat merepresentasikan sembarang data, meskipun lebih banyak tipe *pointer* dikhususkan untuk mengacu pada tipe data tertentu. *Pointer* menempati empat *byte* memori (Wahana Komputer, 2003).

7) Tipe *Variant*

Kadang-kadang diperlukan memanipulasi data yang tipenya berubah-ubah atau tidak dapat ditentukan pada saat kompilasi. Satu pilihannya adalah menggunakan variabel dan parameter bertipe *variant*, yang menyatakan nilai yang bisa berubah saat *runtime*. *Variant* menawarkan fleksibilitas lebih tetapi membutuhkan memori yang lebih besar dibanding variabel biasa, juga operasi kerjanya menjadi lebih lambat. Secara *default*, *variant* dapat berisi semua tipe data, kecuali *record, set, array statik, file, class, class reference*, dan *pointer*. *Variant* dapat beisi sembarang tipe data yang bukan tipe data terstruktur dan *pointer* (Wahana Komputer, 2003).

B. Metode atau Prosedur

Sesuai dengan tujuan perencanaan ini adalah membuat sebuah alat pengukur daya pemakaian energi listrik yang dikonversi ke dalam nominal dan dikirimkan ke komputer pusat dengan sistem telemetri. Cara yang digunakan adalah metode eksperimen laboratoris. Langkah-langkah tersebut meliputi perancangan, pembuatan, pengujian alat dan analisis kerja alat.

1. Metode laboratoris

Metode eksperimen yang digunakan dalam penelitian ini adalah eksperimen sekali tembak atau sering disebut *one shot case study*. Metode eksperimen ini mempunyai pola X,O dimana X adalah perlakuan dan O adalah tes akhir. Tabel 2.6 berikut adalah tabel bentuk metode eksperimen (Arikunto, 2002).

Tabel 2.6 Metode eksperimen

X	O
Perencanaan dan pembuatan	Tingkat keberhasilan

2. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan adalah observasi dan pengukuran. Observasi disini adalah melakukan pengamatan terhadap objek yang diuji, selanjutnya dari pengujian tersebut dilakukan pengukuran. Pengujian tersebut dimaksudkan untuk mengetahui kualitas alat yang direncanakan apakah sesuai yang direncanakan (terget) atau tidak. Apabila sudah dapat seperti target atau dapat mendekati target maka alat tersebut dapat dikatakan bagus. Target disini didasarkan perencanaan alat yang dibuat.

3. Instrumen

Instrumen adalah alat ukur yang digunakan untuk pengukuran dalam eksperimen. Alat-alat ukur yang digunakan harus mempunyai tingkat validitas yang tinggi artinya sebuah instrumen dikatakan valid apabila mampu mengukur secara tepat atau mendekati harga sesungguhnya. Selain valid, sebuah instrumen juga harus mempunyai tingkat realibilitas yang baik. Instrumen hanya dapat dipercaya bila data yang diperoleh sesuai dengan kenyataan.

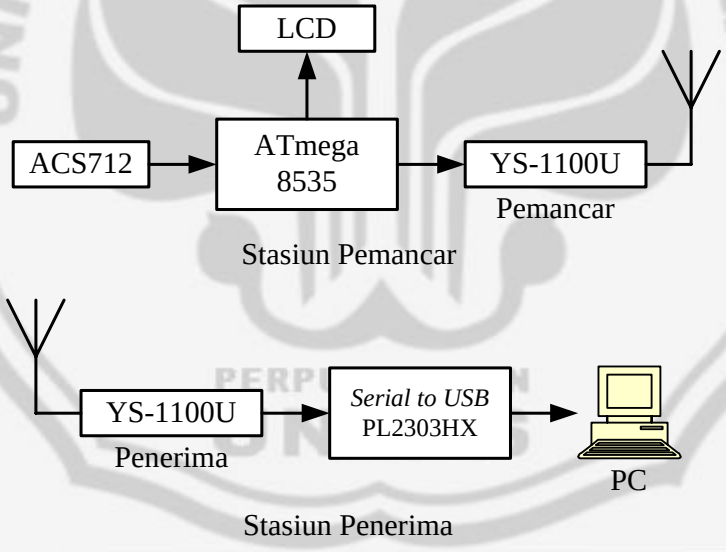
4. Teknik Analisis Data

Pengukuran unjuk kerja alat ini bertujuan untuk mengetahui kualitas kerja apakah alat ini dapat bekerja sesuai dengan harapan dalam perancangan atau tidak. Teknik analisis data disini menggunakan metode analisis diskriptif yaitu membandingkan antara perhitungan perencanaan dengan pengukuran atau pengamatan hasil eksperimen. Apabila terjadi penyimpangan, maka akan dilakukan identifikasi dari penyimpangan tersebut.

5. Prosedur

a. Diagram Blok Rangkaian

Pendukung dalam memahami cara kerja sistem, maka dibuat diagram blok perancangan yang merupakan garis besar rangkaian alat KWH-meter digital bebasis system telemetri. Gambar 2.19 menunjukkan gambar blok perancangan KWH-meter digital berbasis sistem telemetri.



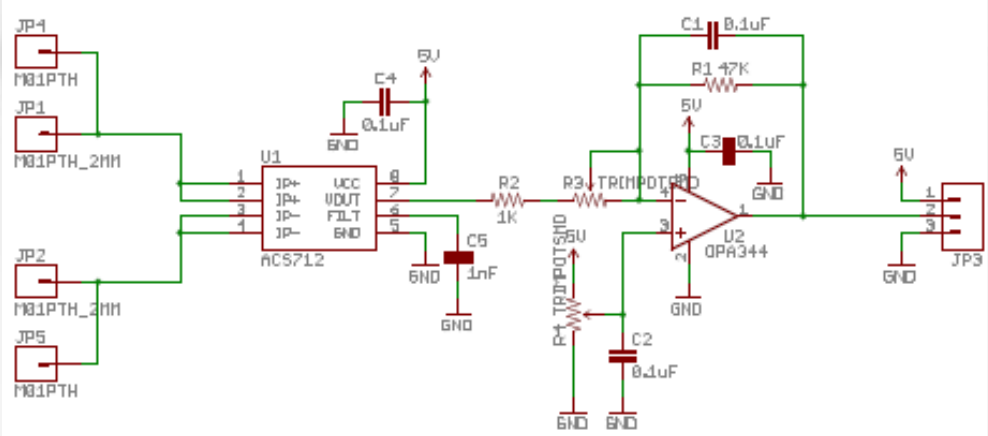
Gambar 2.19 Diagram blok rangkaian pembuatan alat

Pembuatan alat KWH-meter digital berbasis sistem telemetri ini dapat menampilkan biaya yang harus dibayarkan setiap hari maupun setiap bulan melalui LCD yang dipasang pada alat tersebut. Data yang terbaca oleh sensor akan diproses oleh mikrokontroller menjadi data digital dan diubah dalam satuan

watt dan juga dikonversi kedalam nominal rupiah yang nantinya akan ditampilkan ke LCD. Data digital yang telah diproses oleh ATmega8535 juga akan dikirimkan oleh pemancar YS-1100U melalui gelombang elektromagnet ke penerima YS-1100U pada satsiun penerima, yang kemudian akan diteruskan ke komputer melalui komunikasi serial ke USB (modul PL2303HX). Gambar rangkaian lengkap pada stasiun pemancar dan penerima dapat dilihat dalam lampiran.

b. Rangkaian IC ACS712

IC ACS712ELCTR-05B-T dapat mendeteksi arus hingga 5A dengan sensitivitas rata-rata 185mV/A (www.allegromicro.com, 2010). Prinsip kerja dari ACS712ELCTR-05B-T hampir sama dengan sensor *effect hall* lainnya yaitu dengan memanfaatkan medan magnetik di sekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Gambar 2.20 merupakan gambar rangkaian dari pembaca arus IC ACS712.



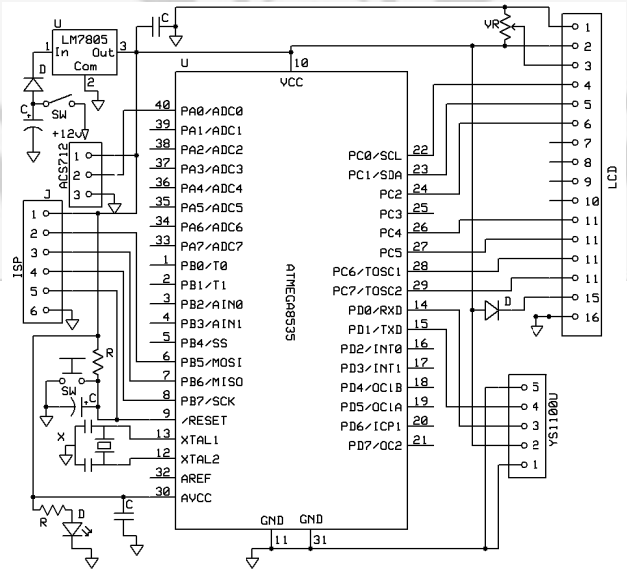
Gambar 2.20 Rangkaian pembaca arus IC ACS712

JP1 dan JP4 digabungkan dan terhubung dengan sumber tegangan AC, sedangkan JP2 dan JP5 terhubung ke beban. Rangkaian ini ditambahkan OPA344 sebagai penguat signal keluaran pembacaan IC ACS712. Fungsi VR3 dan VR4 untuk mengkalibrasi pembacaan arus IC ACS712. Keluaran pembacaan ADC IC ACS712 yang telah dikuatkan oleh OPA344 pada JP3 pin 3 terhubung dengan port A (ADC) Mikrokontroller.

c. Rangkaian Sistem Minimum Mikrokontroller

Sistem minimum mikrokontroler dalam tugas akhir ini memakai IC Mikrokontroller ATmega8535 sebagai IC utama pemroses data. Sistem mikrokontroller ATmega8535 ini terdiri dari kristal 11,0592 Mhz yang terhubung dengan pin 12 dan 13, serta kondensator bernilai 33 pf yang masing-masing kapasitor salah satu kakinya terhubung dengan *ground*. Fungsi dari rangkaian kristal ini sebagai pendukung rangkaian osilator internal. Sistem minimum ini juga dilengkapi rangkain *power on reset* supaya terjadi *reset* sistem pada saat mikrokontroller dihidupkan.

Sistem minimum tersebut menunjukkan bahwa konektor LCD terdapat pada *port* C mikrokontroller.Konektor ISP pada *port* 6 (MOSI), 7 (MISO), 8 (SCK), 9 (RESET), 10 (VCC), dan 11 (GROUND). *Port* 30 dan *port* 31 berfungsi sebagai *port* yang menghubungkan dengan rangkaian *transmitter* dan *receiverRadio*, *port* 14 dan 15 merupakan bagian yang digunakan sebagai sistem komunikasi serial ke USB. Sistem minimum ini diberi IC regulator LM7805, karena konsumsi tegangan IC mikrokontroller sebesar 5V yang didapat dari sumber tegangan DC 12V diberi regulator LM7805 menjadi 5V. Gambar rangkaian sistem minimum dari Mikrokontroller ATmega8535 ini dapat dilihat pada 2.21 berikut.



Gambar 2.21 Rangkaian sistem minimum mikrokontroller

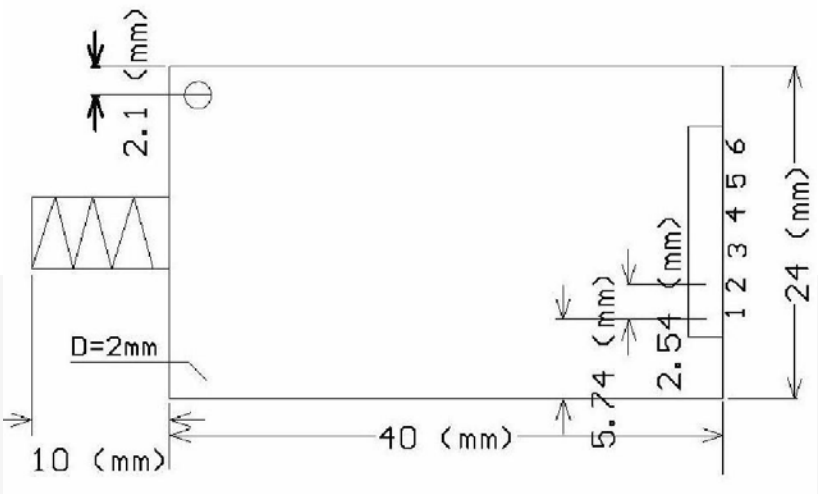
d. Pemancar dan Penerima

Rangkaian pemancar akan mengirimkan data yang diberikan dari mikrokontroller ATmega8535 melalui port komunikasi serial yaitu P30 dan P31 yang sebelumnya data digital tersebut diubah oleh modem. Pemancar yang digunakan pada sistem pelanggan adalah pemancar YS-1100U yang mempunyai 8 kanal dengan frekuensi yang berbeda. Frekuensi yang digunakan pada sistem ini adalah 433 Mhz dengan jarak jangkauan komunikasi sekitar <300 meter pada *baudrate* 9600 bps dan maksimum <500 meter dengan *baudrate* 1200 bps dengan memakai antena 2 meter di tempat terbuka (www.yishi.net.cn, 2011).

Gambar 2.22 YS-1100U *RF Data Transceiver* (www.yishi.net.cn, 2011)

Hasil pengubahan data digital menjadi sinyal frekuensi oleh rangkaian modulator FSK *(Frekuency Shift Keying)* kemudian diteruskan ke rangkaian selanjutnya yaitu rangkaian pemancar (*transmitter*). Rangkaian pemancar pada tugas akhir ini menggunakan pemancar YS-1100U yang di dalamnya terdapat modulator yang berfungsi untuk menumpangkan sinyal pembawa dengan sinyal informasi yang akan dikirim, supaya data dapat dipancarkan.

Pemancar YS-1100U menerima data dari mikrokontroller yang berupa sinyal informasi, kemudian sinyal frekuensi ini dimodulasi dengan frekuensi yang dibangkitkan oleh osilator. Sinyal informasi tersebut kemudian ditumpangkan pada frekuensi pembawa untuk dikirimkan ke antena untuk dipancarkan.



Gambar 2.23 Definisi pin YS-1100U *RF Data Transceiver*
(www.yishi.net.cn, 2011)

Tabel 2.7 Pin-Pin YS-1100U *RF Data Transceiver* (www.yishi.net.cn, 2011)

Pin	Nama Pin	Fungsi	Level	Terhubung dengan terminal
1	GND	Ground		Ground
2	Vcc	Tegangan Input	+3.3 – 5.5 V	
3	RXD/TTL	Input Serial Data	TTL	TxD
4	TXD/TTL	Output Serial Data	TTL	RxD
5	DGND	Digital Grounding		
6	NC	Pengujian dalam pabarik		

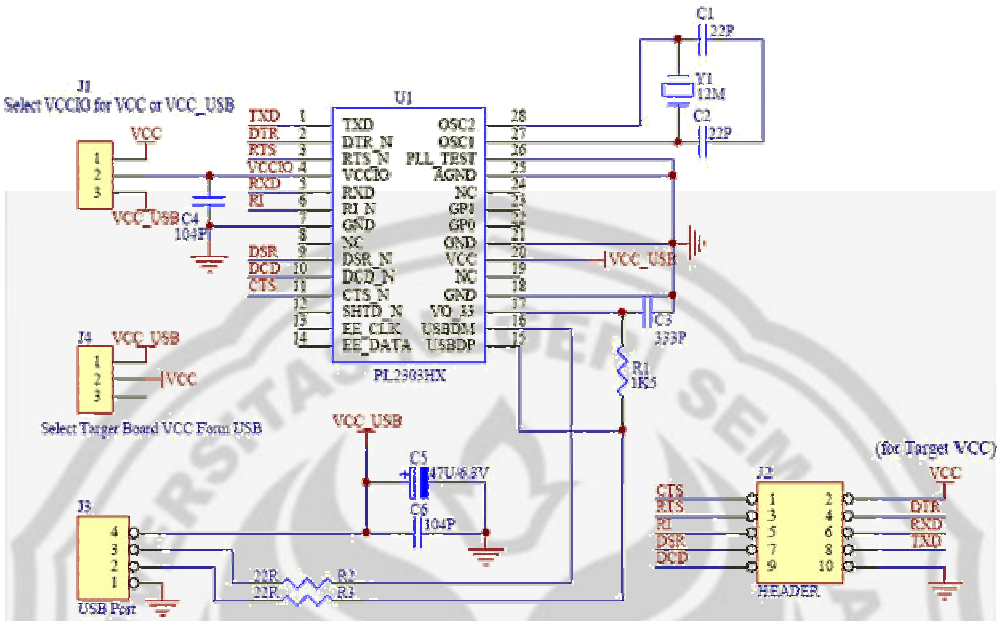
Penerima sinyal yang dikirim dari *transmitter* adalah sebuah rangkaian *receiver* untuk menerima data yang dikirimkan. Rangkaian penerima yang digunakan pada sistem ini sama dengan pemancarnya, yaitu YS-1100U, ini dikarenakan YS-1100U dapat juga digunakan sebagai pengirim sekaligus penerima.

e. Penampil *Liquid Crystal Display* (LCD)

Pembuatan alat ini ini penulis menggunakan M1632 LCD Modul sebagai penampil data pada stasiun pengirim. Bagian ini terdiri dari dua bagian utama, yang pertama adalah panel LCD sebagai media penampil informasi, yang terdiri dari 2 baris yang tiap barisnya bisa menampung 16 huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi M1632 dengan mikrokontroler yang memakai tampilan LCD itu, sehingga pemakaian M1632 menjadi sederhana, sistem lain cukup mengirimkan kode-kode ASCII dari informasi yang ingin ditampilkan. Dalam pengiriman data M1632 terdiri dari dua mode yaitu mode 8 bit dan mode 4 bit. Penulis menggunakan mode 4 bit untuk menghemat penggunaan port paralel dan terutama untuk penyederhanaan dalam pembuatan perangkatnya.

Gambar 2.24 Hubungan port C ATmega8535 dengan LCD

f. Rangkaian *Serial to USB* PL2303HX



Gambar 2.25 Rangkaian *Serial to USB* PL2303HX

Rangkaian pada gambar 2.25 tersebut, Ground dan Vcc terhubung jadi satu dengan USB dan modul RF *transceiver* YS1100U. Pin 1 (TX) IC PL2303HX terhubung dengan pin 3 (RX) YS1100U, sedangkan pin 5 (RX) IC PL2303HX terhubung dengan pin 4 (TX) YS1100U. Power supplay (Vcc dan Ground) bersumber dari PC (komputer) melalui port USB.

g. Perancangan Perangkat Lunak Mikrokontroler

Perangkat lunak yang digunakan dalam mikrokontroler ATmega8535 dibuat menggunakan bahasa pemrograman C. Perangkat lunak tersebut berfungsi sebagai protokol dalam proses pengiriman data. Hal-hal yang diatur pada protokol pengiriman data tersebut antara lain *baudrate* atau kecepatan pengiriman data, mode pengiriman data, jumlah *bit* per *byte* data serta aturan verifikasi dan koreksi kesalahan.

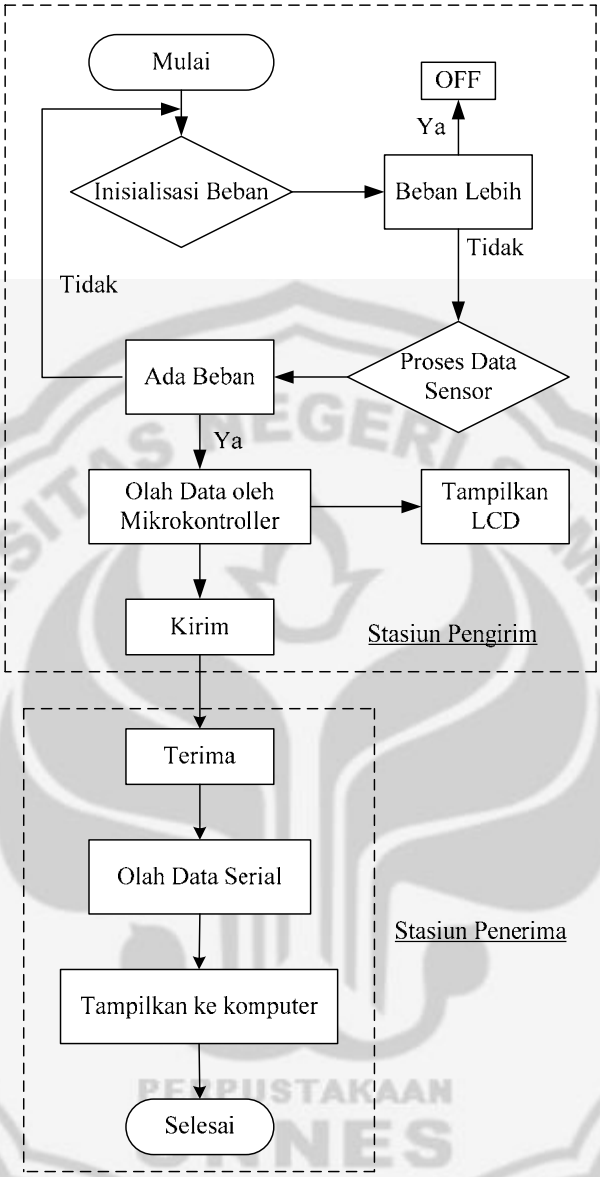
Sistem ini diset untuk melakukan komunikasi serial delapan bit dengan *baudrate* sebesar 1200 bps. Kedua buah mikrokontroler harus memiliki kecepatan

pengiriman data yang sama. Pengaturan *baudrate* ini dilakukan dengan perangkat lunak melalui pengaturan kerja *timer* 1 dan pengesetan Rs dan E pada LCD.

Gambar 2.26 merupakan *flowchart* program utama. Pertama kali program akan menginisialisasi apakah beban sudah sesuai kapasitas yang di anjurkan, lebih kecil atau malah terlalu besar. Jika beban terlalu besar, maka sistem secara otomatis akan mati, dan jika tidak ada beban maka sistem tidak akan melakukan penghitungan/pengolahan data dari sensor.

Data yang diambil telah diatur *timer*, *counter*, dan *serial port* untuk komunikasi data, selanjutnya data akan diambil dari data *port counter*, sehingga didapat perhitungan *counter*. Apakah waktunya sudah 1 detik, jika ” Ya ” maka langsung tertampil di LCD, selain itu data juga akan langsung dikirim ke stasiun penerima melalui YS-1100U *RF Data Transceiver* sebagai pemancar RF dan juga akan diterima oleh YS-1100U *RF Data Transceiver* yang telah dipasang pada stasiun penerima. Data terus akan diulang untuk data-data berikutnya. Pertama kali program akan menginisialisasi *timer* dan *serial port* untuk komunikasi data. Stasiun penerima pertama kali akan menerima data dari stasiun pengirim, setelah itu data diteruskan ke *port* serial komputer melalui RS232, sehingga dapat diakses dengan komputer dan disimpan melalui *dekstop*. Gambar 2.26 berikut menunjukkan diagram alir *(flow chart)* dari perangkat lunak yang dibuat dan untuk *listing program* AVR pada mikrokontroller serta *listing program code editor* Delphi 7.0 dapat dilihat pada lampiran.

PERPUSTAKAAN
UNNES



Gambar 2.26 *Flowchart Program*

C. Pengujian Alat

Pengujian ini dilakukan untuk mengetahui kehandalan dari alat KWH-meter Digital Telemetri; seperti pengiriman data, respon, tingkat kesalahan dan keakurasian penghitungan, serta jangkauan pengiriman data. Pertama untuk mengetahui kehandalan dalam keakurasian dan tingkat kesalahan penghitungan daya yang terpakai serta kesetabilan pembacaan KWH-meter Digial dibandingkan

dengan KWH-meter PLN, maka alat KWH-meter digital telemetri ini akan dilakukan pengujian dengan menghubungkan pararel dengan KWH-meter PLN. Pengujian dilakukan pada pelanggan dengan daya 1300 per KWH. Pengujian dilakukan selama beberapa jam dengan beban yang relatif tetap dan mencatat hasil data yang terbaca oleh alat KWH-meter digital dan KWH-meter PLN per satu satuan waktu seperti yang tertera pada tabel 2.8 dan pada gambar 2.27. Data yang diperoleh diperbandingkan dan dihitung tingkat keakurasian atau tingkat kesalahan alat dibandingkan dengan KWH-meter PLN, serta kestabilan pembacaan alat dibandingkan dengan KWH-meter PLN. Berikut tabel 2.8 adalah data hasil pengujian tingkat keakurasian dan tingkat kesalahan serta kesetabilan pembacaan alat KWH-meter Digital Telemetri dibandingkan dengan KWH-meter milik PLN.

Tabel 2.8 Data pengujian alat pertama

No	Hari, Tanggal	Jam	Pembacaan Alat				Selisih (x dan y)
			KWH-meter Digital		KWH-meter PLN		
			Daya/kWh	Selisih (x)	Daya/kWh	Selisih (y)	
1.	Senin, 4 Juli 2011	00.00	0,468	0,117	25862,9	0,1	0,017
		01.00	0,585		25863,0		
		01.00	0,585	0,117	25863,0	0,1	0,017
		02.00	0,702		25863,1		
		02.00	0,702	0,117	25863,1	0,1	0,017
		03.00	0,816		25863,2		
		03.00	0,816	0,117	25863,2	0,1	0,017
		04.00	0,933		25863,3		
		04.00	0,933	0,117	25863,3	0,1	0,017
		05.00	1,050		25863,4		
2.	Selasa, 5 Juli 2011	00.00	7,496	0,409	25869,8	0,4	0,09
		01.00	7,905		25870,2		
		01.00	7,905	0,409	25870,2	0,4	0,009
		02.00	8,314		25870,6		
		02.00	8,314	0,409	25870,6	0,4	0,009
		03.00	8,723		25871,0		
		03.00	8,723	0,409	25871,0	0,4	0,009
		04.00	9,132		25871,4		
		04.00	9,132	0,409	25871,4	0,4	0,009
		05.00	9,541		25871,8		

Pengukuran pada pelanggan dengan daya 1300 per KWH

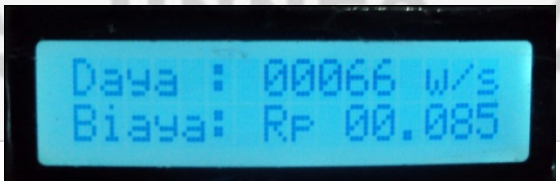
Gambar 2.27 Data Pengujian Alat yang Dipararel dengan KWH-meter PLN

Pengujian alat yang kedua dilakukan untuk mengetahui bagaimanakah respon sensor terhadap beban, kehandalan alat dalam melakukan pengiriman dan penerimaan data setiap satuan waktu pada jarak yang berbeda. Hasil dari pengujian alat tersebut diperlihatkan pada tabel 2.8 dan gambar 2.28 & 2.29.

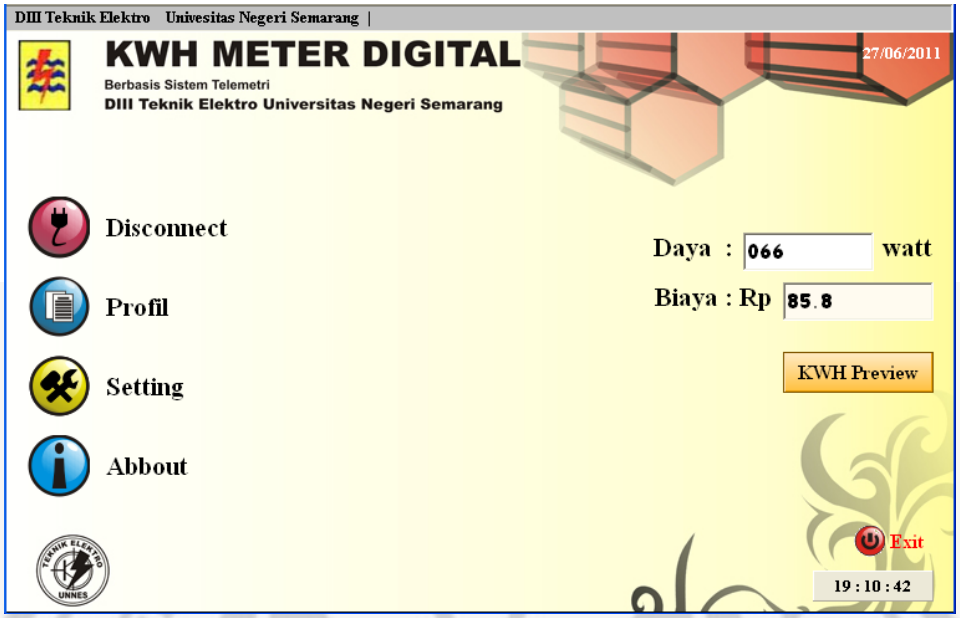
Tabel 2.9 Data pengujian alat kedua

No.	Hari/tanggal	Jam	Jarak	Pembacaan Alat			
				Stasiun Pengirim (tampilan LCD)		Stasiun Penerima (Tampilan Komputer)	
				Daya	Biaya	Daya	Biaya
1.	Kamis, 23 Juni 2011	19.06	1 m	00000	Rp 00000	00	Rp 0
		19.10	5 m	00066	Rp 00085	066	Rp 85,8
		19.30	10 m	00241	Rp 00313	0241	Rp 313,3
		20.00	30 m	00352	Rp 00457	0352	Rp 457,6
		21.00	50 m	00763	Rp 00991	0763	Rp 991,9
		22.00	100 m	01036	Rp 01346	01036	Rp 1346,8
		23.00	200 m	01325	Rp 01722	0	Rp 0

Pengukuran pada pelanggan dengan daya 1300 per KWH



Gambar 2.28 Tampilan LCD Stasiun Pengirim



Gambar 2.29 Tampilan Pada Komputer di Stasiun Penerima

D. Pembahasan

Berdasarkan data-data yang telah diambil dalam pengujian pertama, dapat diketahui bahwa alat akan membaca besar daya beban yang terpasang pada alat dari menit pertama sampai satu jam. Pengujian dilakukan dua kali dalam kurun waktu dua hari. Awal pengujian hari Senin 4 Juli 2011 yaitu pada jam 00.00, LCD alat menunjuk angka 0,468 sedangkan KWH-meter PLN menunjuk pada angka 25862,9 sekian. Jam selanjutnya yaitu pukul 01.00, LCD alat menunjuk pada angka 0585 w/s atau 0,585 kWh sedangkan KWH-meter PLN berada pada angka 25863,0 kWh sekian.

Pengujian hari pertama memakai beban yang relatif lebih kecil, seperti lampu, obat anti nyamuk elektrik dan kipas angin. Pengujian hari kedua beban ditambah yang lebih besar, seperti penghangat nasi, komputer dan kulkas. Ini dapat terlihat dimana perubahan pada hari pertama relatif kecil, hanya 0,1 kWh perjamnya. Hari kedua perubahannya agak besar, sekitar 0,4 kWh perjamnya.

Perubahan pembacaaan alat per satuan waktu (jam) untuk beban yang sama dan tetap bisa dikatakan setabil, ini bisa terlihat dari data tabel 2.8 dimana perubahan angka setiap jamnya selalu sama. Terlihat dari awal pengujian jam

00.00 sampai jam 01.00 terjadi perubahan sebesar 0,117 kWh dan seterusnya perubahan setiap jamnya juga sama. Begitu juga untuk pengujian hari kedua Selasa 5 Juli 2011, awal pengujian pada jam 00.00 alat menunjuk angka 7496 w/s, sedangkan pada jam 01.00 berubah menjadi 7905 w/s dan jam 02.00 berubah lagi menjadi 8314 setiap jamnya terjadi perubahan sebesar 409 w/s.

Pembacaan alat dengan KWH-meter PLN ada sedikit selisih, tetapi selisih disini relatif kecil. Terlihat pada pengujian hari pertama Senin 4 Juli 2011, teradi selisih sebesar 0,017 kWh dan pada pengujian hari kedua Selasa 5 Juli 2011 terjadi selisih sebesar 0,009 kWh. Selisih disini kemungkinan karena pembacaan mata manusia pada KWH-meter PLN yang tidak bisa selalu tepat dan KWH-meter PLN hanya menampilkan data per kWh, sehingga satu angka dibelakang koma sering tidak bisa terbaca secara tepat, apalagi disini alat menampilkan sampai 3 angka dibelakang koma (per watt) sedangkan KWH-meter PLN tidak. Selisih kemungkinan juga bisa disebabkan karena alat KWH-meter ini perangkatnya berupa rangkaian elektronik dengan olah data dan tampilan digital, sedangkan KWH-meter PLN rangkaiannya berupa kumparan dan gear mekanis dengan tampilan roller angka mekanis. Tampak seperti gambar 2.30 berikut.

Gambar 2.30 Perbedaan tampilan KWH-meter Digital dengan KWH-meter PLN

Data hasil pengujian respon sensor terhadap beban dan kehandalan alat dalam melakukan pengiriman serta penerimaan data setiap satuan waktu pada jarak yang berbeda dapat dilihat pada tabel 2.9 dan gambar pengujian alat pada gambar 2.29 & 2.30. Pengiriman data dari stasiun pengirim ke stasiun penerima memakai modul radio YS-1020UB dengan frekuensi 433MHz dan *baudrate* 9600 bps. Data pada tabel 2.9 menunjukkan bahwa data pada stasiun pengirim dan stasiun penerima adalah hampir sama pada setiap satuan waktu yang hampir

bersamaan dari jarak yang berbeda (1 m – 100 m). Perbedaan tampilan pada LCD dan komputer, dikarenakan pada LCD hanya bilangan asli yang ditampilkan dan pada komputer angka real. Pengiriman juga hanya daya saja yang dikirimkan, bukan kedua-duanya (daya dan biaya). Ini berarti respon alat pada stasiun penerima dapat bekerja secara maksimal.

Beradasarkan data tabel pengujian (tabel 2.9), pengiriman data dari stasiun pengirim ke stasiun penerima pada jarak 200 m tidak dapat terbaca (data 0 pada stasiun penerima) atau data *error*, padahal pada dasar teori telah dijelaskan bahwa alat tersebut mampu melakukan pengiriman data sampai 800m pada *baudrate* 9600 dengan antena 2m di tempat terbuka. Ini dikarenakan pengujian dilakukan di dalam ruangan dengan memakai antena biasa bawaan dari modul tersebut dan sumber daya yang dipakai juga kecil *(notebook)*, sehingga memungkinkan hanya mampu mencapai jarak <200m. Selain itu, mungkin faktor cuaca dan kondisi iklim pada saat pengujian juga bisa berpengaruh terhadap pengiriman data. Berikut gambar 2.31 tampak alat penerima dan pengirim (KWH-meter Digital Telemtri) pada stasiun penerima dan pengirim beserta antena dan kelengkapan.

Gambar 2.31 Alat penerima dan pengirim (KWH-meter Digital Telemetri)

BAB III
PENUTUP

A. Kesimpulan

Berdasarkan pengujian alat dan analisis pembahasan yang telah dilakukan, maka dapat diambil beberapa kesimpulan antara lain :

1. Pembacaan alat setabil terhadap beban yang sama dalam beberapa kurun waktu.
2. Tingkat kesalahan pembacaan alat relatif kecil, dan kekaurasian penghitungan alat cukup tinggi dibandingkan dengan KWH-meter PLN. Kesalahan dimungkinkan karena pembacaan mata manusia pada KWH-meter PLN yang tampilannya per kWh bukan per *watt*.
3. Tingkat respon dan kehandalan pengiriman data adalah baik, sama dengan data yang dikirimkan tidak ada error data dalam setiap detik.
4. Jangkauan pengiriman data telemetri kurang dari 200 m dengan *baudrate* 9600 frekuensi 433 MHz dan memakai antena bawaan dari modul RF *Transceiver* YS-1100U di dalam ruangan atau tempat tertutup.

B. Saran

Melihat kinerja alat pada saat pengujian dan telah dipaparkan analisisnya pada pembahasan, berikut beberapa saran yang mungkin bisa meningkatkan kinerja alat kedepan.

- Pengujian alat lebih baik dilakukan pada beberapa pelanggan dengan tarif daya berbda, agar dapat membandingkan tingkat keakurasian pembacaan alat.
- Alat perlu dikembangkan lagi untuk beberapa pelanggan (multistation) dengan tarif daya yang berbeda-beda.
- Sistem tampilan di komputer bisa ditambahkan sistem web, agar bisa diakses di mana saja lewat internet.

PERPUSTAKAAN UNNES

DAFTAR PUSTAKA

Acroname. 2011. *Datasheet Serial to USB PL2303HX*. Diakses pada tanggal 28 Juli 2011 dari http://www.acroname.com.

Allegro. 2011. *Datasheet ACS712*. Diakses pada tanggal 27 Maret 2011 dari http://www.allegromicro.com.

Andi. 2003. *Panduan Praktis Pemrograman Borland Delphi 7.0*. Semarang: Wahana Komputer.

Anonim. 2002. *ComPort Library*. Diakses pada tanggal 3 Mei 2010 dari http://sourceforge.net.

Arifianto, B. 2009. *Modul Training Microcontroller for Beginner*. Diakses pada tanggal 5 Mei 2010 dari http://www.max-tron.com.

Arikunto, S. 2002. *Prosedur Penelitian*. Jakarta: PT. Renika Cipta.

Bejo, A. 2001. *C&AVR Bahasa C Mikokontroler ATmega8535*. Yogyakarta: Graha Ilmu.

_____.2008. *C&AVR Rahasia Kemudahan Bahasa C dalam Mikokontroler ATmega8535*. Yogyakarta: Graha Ilmu.

Fraden, Jacob. 2005. *Handbook of Modern Sensor* . Thermoscan, Inc : San Diego, California.

Kadir, A. 2006. *Dasar Aplikasi Database MySQL-Delphi*. Yogyakarta: Andi.

Sapiie, Nishino. 1994. *Pengukuran dan Alat-alat Listrik*. Jakarta: PT. Pradnya Paramita.

Suharsimi, A. 2002. *Prosedur Penelitian*. Jakarta: PT. Renika Cipta.

Vision, C. 2010. *ATmega8535 Minimum System Modul*. Diakses pada tanggal 28 Juni 2010 dari http://www.google.com/creative_vision.html

Wardhana, L. 2006. *Belajar Sendiri Mikrokontroller AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi Offset.

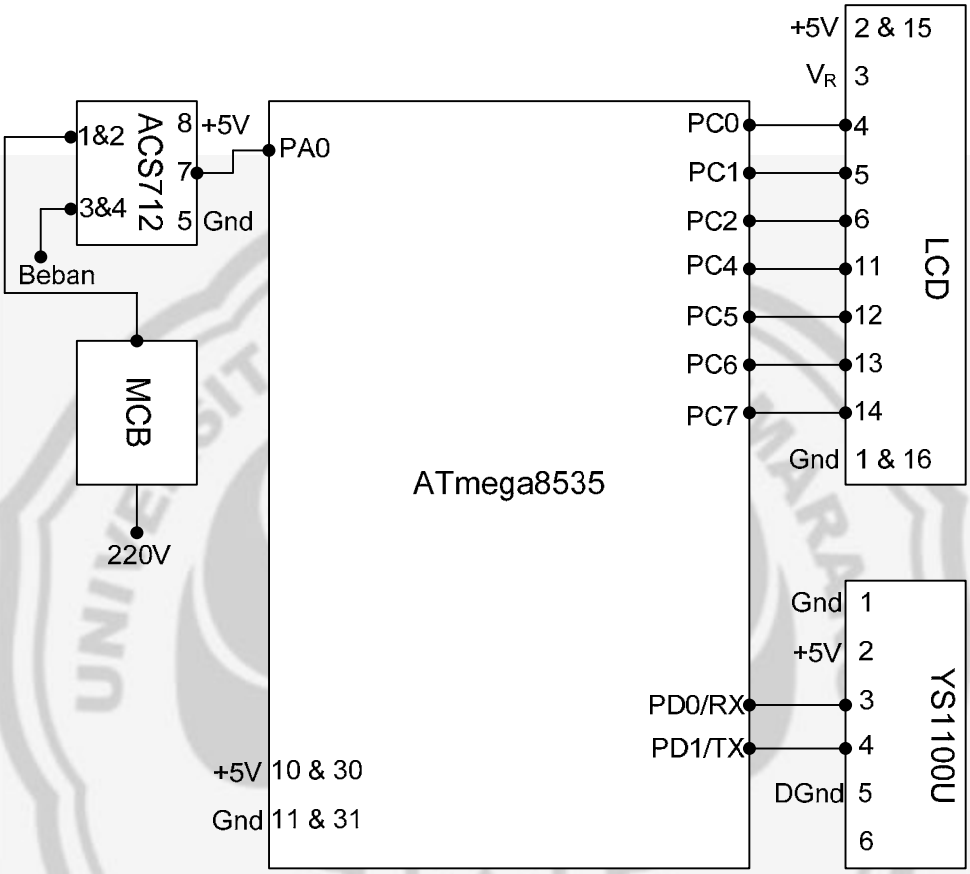
Yishi. 2011. *YS-1100U RF Data Transceiver*. Diakses pada tanggal 22 Mei 2011 dari http://www.yishi.net.cn.

Zhanggischan, Zuhal. 2004. *Prinsip Dasar Elektroteknik*. Jakarta: PT. Gramedia Pustaka Utama.

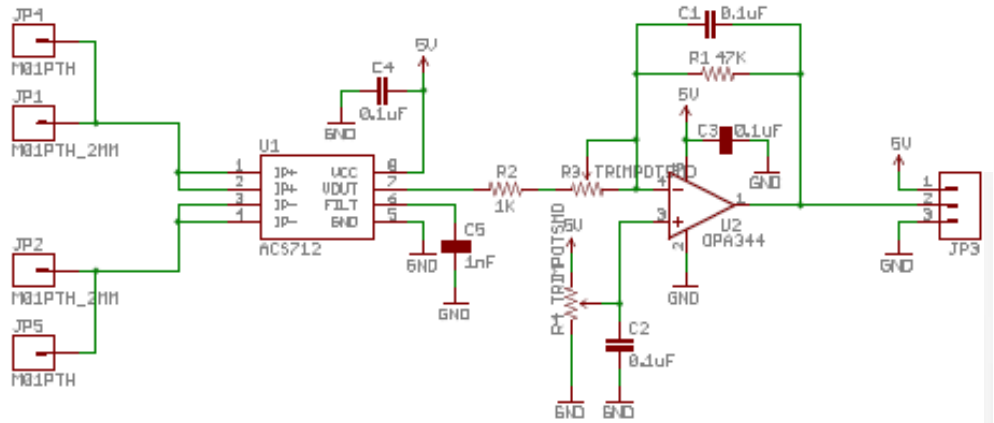
LAMPIRAN

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

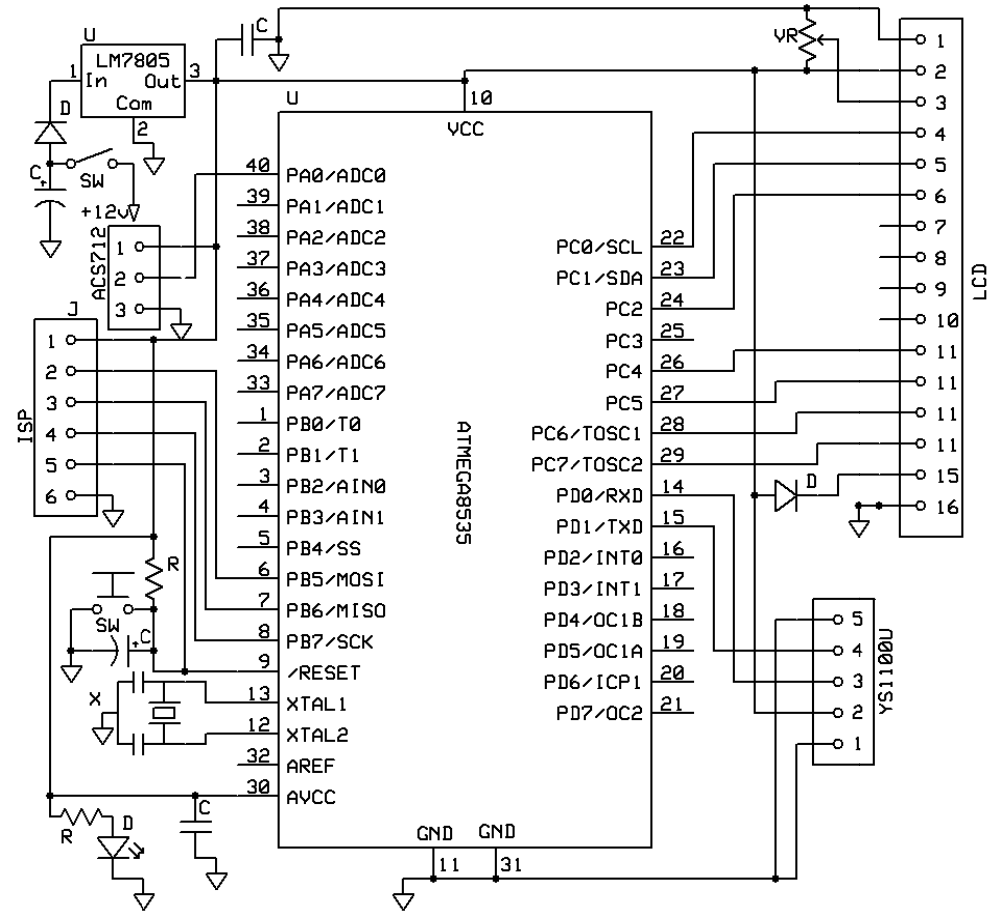
RANGKAIAN BLOK STASIUN PEMANCAR



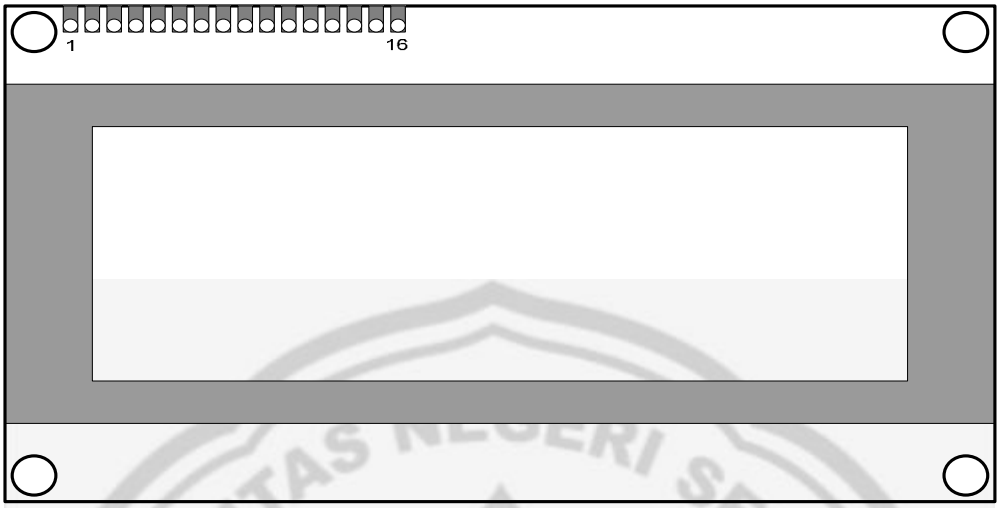
RANGKAIAN STASIUN PEMANCAR



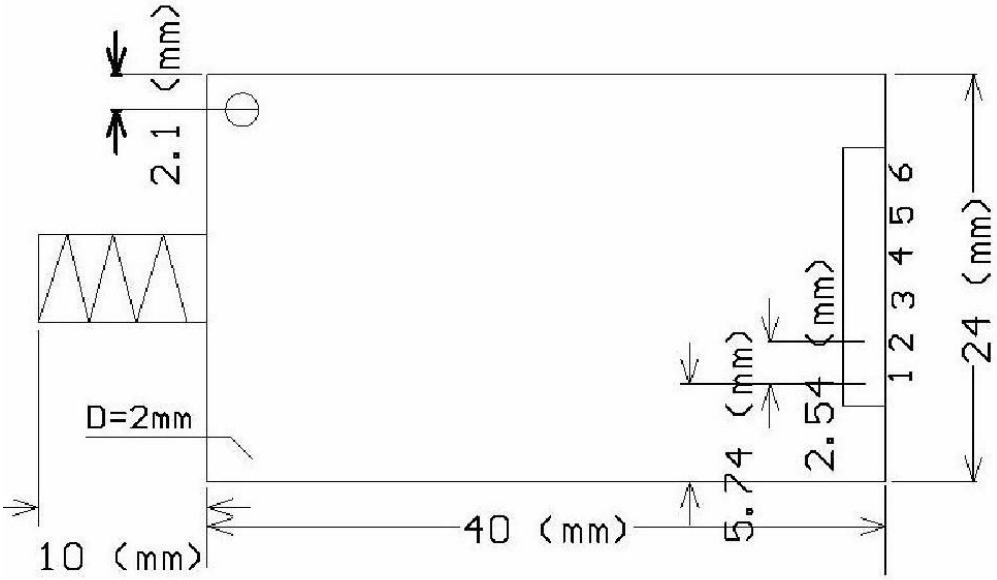
Rangkaian pembaca arus ACS712



Rangkaian mikrokontroller ATmega8535

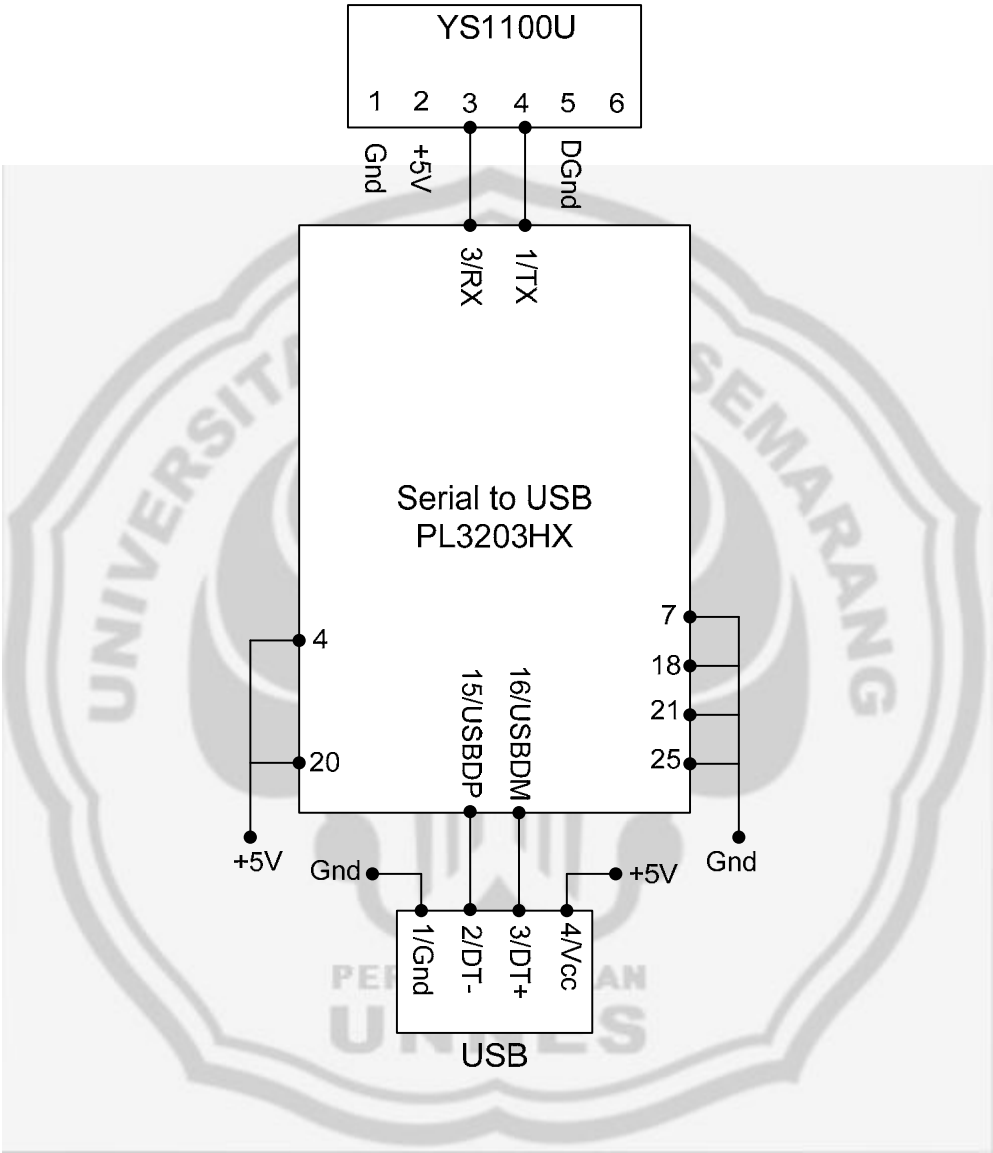


Penampang panel LCD (M1632)

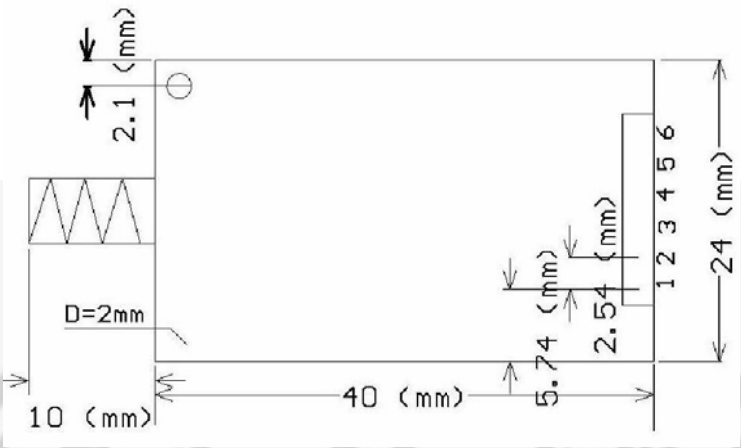


Penampang pin YS-1100U RF Data Transceiver

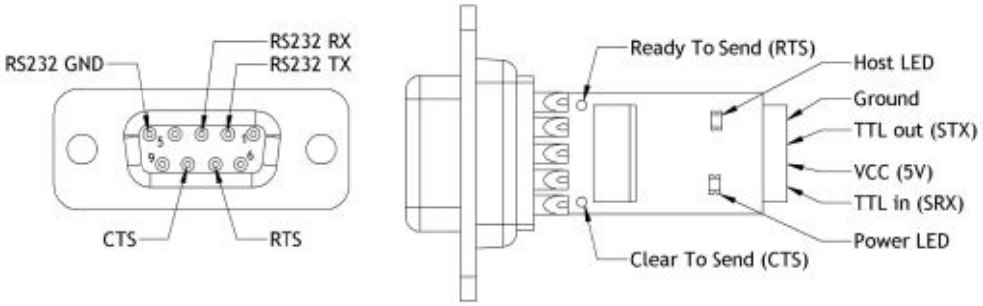
RANGKAIAN BLOK STASIUN PENERIMA

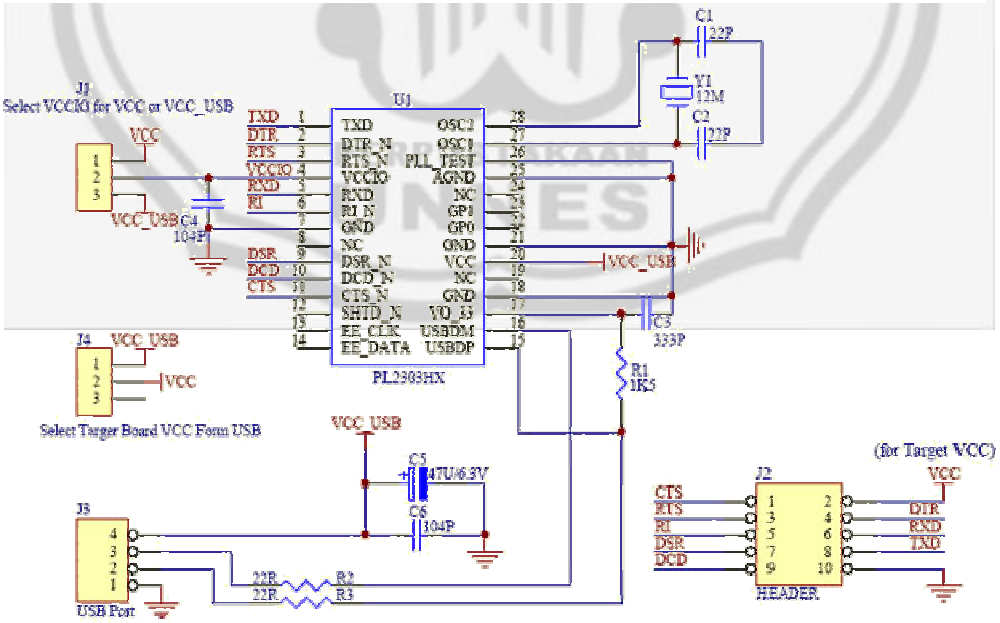


RANGKAIAN STASIUN PENERIMA



Penampang pin YS-1100U RF Data Transceiver





Modul Serial to USB PL2303HX

LISTING PROGRAM

1. Program Utama AVR Mikrokontroller ATmega8535

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.8 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 1/4/2011
Author  :
Company :
Comments:


Chip type           : ATmega8535
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 128
*****************************************************/

#include <mega8535.h>
#include <delay.h>
// Standard Input/Output functions
#include <stdio.h>

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

#define FIRST_ADC_INPUT 0
#define LAST_ADC_INPUT 0

unsigned int adc_data[LAST_ADC_INPUT-
FIRST_ADC_INPUT+1];
#define ADC_VREF_TYPE 0x40

// ADC interrupt service routine
// with auto input scanning
```

```
interrupt [ADC_INT] void adc_isr(void)
{
register static unsigned char input_index=0;
// Read the AD conversion result
adc_data[input_index]=ADCW;
// Select next ADC input
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
   input_index=0;
ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE &
0xff))+input_index;
// Delay needed for the stabilization of the ADC input
voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
}

// Standard Input/Output functions
#include <stdio.h>
#include "Setting.c"
#include "Deklar.c"
#include "tes.c"



// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer 1 value
TCNT1H=0xD5;
TCNT1L=0xF5;
// Place your code here
if (disTgl==0)

{BacaArus();   }
}


// Declare your global variables here

void main(void)
{
Deklarasi();

// I2C Bus initialization
i2c_init();
```

```
// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);

// LCD module initialization
lcd_init(16);
```

```
//
// h=21; m=09; s=53;
// rtc_set_time(h,m,s);
//
//
//
// h=10; m=07; s=11;
// rtc_set_date(h,m,s);
//


// Global enable interrupts
#asm("sei")
KWHmode=1;
disTgl=0;

PORTC.3=1;

lcd_clear();
lcd_gotoxy(4,0);
lcd_putsf("KWH Meter");
lcd_gotoxy(5,1);
lcd_putsf("By Huda");
delay_ms(1000);

HargaTemp=Harga;
BiayaTemp=Biaya;
MeterTemp=Meter;
HargaTemp=1300;
BiayaTemp=0;
MeterTemp=0;

while (1)
      {
      // Place your code here
      //simulasi Arus
```

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN UNNES

```
        //Arus=1;
        if(OK==0)
         {
          Status=1;
          Menu();
         }


        if(Naik==0)
         {delay_ms(300);
          disTgl=1;
          Display_TGLJAM();
         }
        else if(Naik==1)
         {
         disTgl=0;
         // Mengirim data Meter ke port serial
         printf("%i",Meter);   //printf("%i\n") dihapus
         //Hex2Dec_4byte(Meter);
         //putchar(0x30+PuluhRibu);
         //putchar(0x30+Ribuan);
         //putchar(0x30+Ratusan);
         //putchar(0x30+Puluhan);
         //putchar(0x30+Satuan);

        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putsf("Daya : ");
        Hex2Dec_4byte(Meter);
        lcd_putchar(0x30+PuluhRibu);
        lcd_putchar(0x30+Ribuan);
        lcd_putchar(0x30+Ratusan);
        lcd_putchar(0x30+Puluhan);
        lcd_putchar(0x30+Satuan);

        if (KWHmode==0)
        {
        lcd_putsf(" kWh");
        lcd_gotoxy(0,1);
        lcd_putsf("Biaya: ");
        lcd_putsf("Rp ");

BiayaTemp=(Meter/1000)*(HargaTemp/3600);//(Meter*HargaT
emp)/3600;
        Biaya=BiayaTemp;
        Hex2Dec_4byte(BiayaTemp);
        lcd_putchar(0x30+PuluhRibu);
```

```
        lcd_putchar(0x30+Ribuan);
        lcd_putsf(".");
        lcd_putchar(0x30+Ratusan);
        lcd_putchar(0x30+Puluhan);
        lcd_putchar(0x30+Satuan);
        }

        else if (KWHmode==1)
        {
        lcd_putsf(" w/s");
        lcd_gotoxy(0,1);
        lcd_putsf("Biaya: ");
        lcd_putsf("Rp ");
        BiayaTemp=Meter*HargaTemp/1000;
        Biaya=BiayaTemp;
        Hex2Dec_4byte(BiayaTemp);
        lcd_putchar(0x30+PuluhRibu);
        lcd_putchar(0x30+Ribuan);
        lcd_putsf(".");
        lcd_putchar(0x30+Ratusan);
        lcd_putchar(0x30+Puluhan);
        lcd_putchar(0x30+Satuan);
        }
        /*
        lcd_gotoxy(0,1);
          Hex2Dec_4byte(DataADC);
          lcd_putchar(0x30+PuluhRibu);
          lcd_putchar(0x30+Ribuan);
          lcd_putchar(0x30+Ratusan);
          lcd_putchar(0x30+Puluhan);
          lcd_putchar(0x30+Satuan);
          lcd_putsf(" ");
          Hex2Dec_4byte(KWH);
          lcd_putchar(0x30+PuluhRibu);
          lcd_putchar(0x30+Ribuan);
          lcd_putchar(0x30+Ratusan);
          lcd_putchar(0x30+Puluhan);
          lcd_putchar(0x30+Satuan);
          */
         }
        delay_ms(200);
        };
}
```

2. **Program Utama Delphi 7.0**
   a. **Menu Awal**

```
unit KWH2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, Buttons, CPort,
StdCtrls, XiButton;

type
  TForm2 = class(TForm)
    Image1: TImage;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    ComPort1: TComPort;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Timer1: TTimer;
    Panel1: TPanel;
    Timer2: TTimer;
    Label5: TLabel;
    SpeedButton6: TSpeedButton;
    Edit2: TEdit;
    Edit3: TEdit;
    Label6: TLabel;
    Label7: TLabel;
    Timer3: TTimer;
    XiButton1: TXiButton;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
```

```
    procedure Button1Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure XiButton1Click(Sender: TObject);
    procedure Timer3Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  I: integer;
  N , A: string;

implementation

uses KWH3, KWH4, KWH5, KWH6;

{$R *.dfm}

procedure TForm2.SpeedButton1Click(Sender:
TObject);
begin
Application.Terminate;
end;

procedure TForm2.SpeedButton4Click(Sender:
TObject);
begin
Comport1.ShowSetupDialog;
end;

procedure TForm2.SpeedButton3Click(Sender:
TObject);
begin
Form2.Hide;
Form3.Show;
end;

procedure TForm2.SpeedButton5Click(Sender:
TObject);
begin
Form2.Hide;
Form4.Show;
end;
```

```
procedure TForm2.Timer1Timer(Sender: TObject);
begin
I := I + 1;
 Edit1.Text:=
 '
'+
 '
'+
 '
'+
 '
'+
 ' TUGAS AKHIR  Diploma III  ' +
 '| KWH-meter Digital Berbasis Sistem Telemetri
dengan Tampilan dalam Rupiah |'+
 '   Program & Design by : Nur Huda
5350308003   |' +
 '   DIII Teknik Elektro    Univesitas Negeri
Semarang  |';
 N := COPY(Edit1.Text,I,LENGTH(Edit1.Text));
 A := N + COPY(Edit1.Text,1,I);
 Edit1.Text := A;
 IF I = 1 + LENGTH(Edit1.Text) THEN I := 1;
end;

procedure TForm2.Timer2Timer(Sender: TObject);
begin
panel1.Caption :=formatDateTime('hh : nn :
ss',Time);
label5.Caption
:=formatDateTime('dd/mm/yyyy',Date);
Form6.Label10.Caption:=FormatDateTime('dd/mm/yyyy
',Date);
end;

procedure TForm2.SpeedButton6Click(Sender:
TObject);
begin
Comport1.Open;
Form3.Timer1.Enabled:=true;
SpeedButton2.Show;
SpeedButton6.Hide;
label1.Caption:='Disconnect';
end;

procedure TForm2.SpeedButton2Click(Sender:
TObject);
```

```
begin
Comport1.Close;
Form3.Timer1.Enabled:=false;
SpeedButton2.Hide;
SpeedButton6.Show;
label1.Caption:='Connect';
end;

procedure TForm2.SpeedButton7Click(Sender:
TObject);
begin
Form5.Show;
Form2.Hide;
end;

procedure TForm2.Button1Click(Sender: TObject);
begin
Form6.Show;
Form2.Hide;
end;

procedure TForm2.SpeedButton8Click(Sender:
TObject);
begin
Form6.Show;
Form2.Hide;
end;

procedure TForm2.FormShow(Sender: TObject);
begin
Timer3.Enabled:=true;
end;

procedure TForm2.XiButton1Click(Sender: TObject);
begin
Form6.Show;
Form2.Hide;
end;

procedure TForm2.Timer3Timer(Sender: TObject);
begin
Edit2.Text:=Form3.Edit5.Text;
Edit3.Text:=Form3.Edit6.Text;
end;

end.
```

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES

b. Edit Profil

```
unit KWH3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, Buttons, StdCtrls, Grids, DBGrids,
DBClient, Provider, DB, ADODB,
  ExtCtrls, RpRave, RpDefine, RpCon, RpConDS,
jpeg;

type
  TForm3 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Edit5: TEdit;
    Label6: TLabel;
    Edit6: TEdit;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Edit8: TEdit;
    Label10: TLabel;
    Edit9: TEdit;
    Label11: TLabel;
    Button3: TButton;
    SpeedButton1: TSpeedButton;
    DBGrid1: TDBGrid;
    SpeedButton2: TSpeedButton;
    Timer1: TTimer;
    RvDataSetConnection1: TRvDataSetConnection;
    RvProject1: TRvProject;
    Image1: TImage;
    Button4: TButton;
    ADOConnection1: TADOConnection;
    ADOTable1: TADOTable;
    ClientDataSet1: TClientDataSet;
```

```
    DataSetProvider1: TDataSetProvider;
    DataSource1: TDataSource;
    ComboBox2: TComboBox;
    Timer2: TTimer;
    Label12: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  petik:char= '''';

var
  Form3: TForm3;
  baca,SQL,hasil: string;

implementation

uses KWH2, KWH6, KWH5;

{$R *.dfm}

procedure TForm3.Button1Click(Sender: TObject);
begin
Form3.Hide;
Form2.Show;
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
Form3.Hide;
Form5.Show;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
```

```
if dbgrid1.Visible = true then begin
dbgrid1.Visible := false end else
dbgrid1.Visible := true;
ClientDataSet1.Active:=True;
ClientDataSet1.Refresh;
end;

procedure TForm3.SpeedButton2Click(Sender:
TObject);
begin
if MessageDlg('Semua data akan dihapus?',
    mtconfirmation,
    [mBYes,mBNo], 0) = mRYes then
begin
SQL:= 'DELETE FROM Table1';
ADOTable1.Active:=False;
ClientDataSet1.Active:=False;
ADOConnection1.Execute(SQL);
ADOTable1.Active:=True;
ClientDataSet1.Active:=True;
Edit1.Text:=' ';
Edit2.Text:=' ';
Edit3.Text:=' ';
Combobox2.Text:=' ';
Edit5.Text:=' ';
Edit6.Text:=' ';
Edit8.Text:=' ';
Edit9.Text:=' ';
end;
end;

procedure TForm3.Timer1Timer(Sender: TObject);
var
d,s,b,t,p : real;
begin
Form2.Comport1.ReadStr(baca,2);
Edit5.Text:='0'+baca;

  d:=StrToFloat(Combobox2.Text);
  s:=StrToFloat(Edit5.Text);
  b:=d*s/1000;
  p:=b*9/100;
  t:=b+p;
  Edit6.Text:=FloatToStr(b);
  Edit8.Text:=FloatToStr(p);
  Edit9.Text:=FloatToStr(t);
```

```
Form2.ComPort1.Connected:= false; //refresh perlu
di edit2
Form2.ComPort1.Connected:= true;  //refresh perlu
di edit2

ClientDataSet1.Active:=true;
ClientDataSet1.Refresh;
ClientDataSet1.Last;
end;

procedure TForm3.SpeedButton1Click(Sender:
TObject);
begin
RvProject1.Execute;
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
if MessageDlg('Data akan disimpan?',
    mtconfirmation,
    [mBYes,mBNo], 0) = mRYes then
begin
hasil:=
'INSERT INTO
Table1(Bulan,ID,Nama,UPJ,Tarif,Meter,Biaya,PPJ,To
tal) values('+
petik + formatDateTime('mm / yyyy',date) + PETIK
+ ',' +
petik + Edit1.Text + petik + ',' +
petik + Edit2.Text + petik + ',' +
petik + Edit3.Text + petik + ',' +
petik + Combobox2.Text + petik + ',' +
petik + Edit5.Text + petik + ',' +
petik + Edit6.Text + petik + ',' +
petik + Edit8.Text + petik + ',' +
petik + Edit9.Text + petik + ')';
ADOConnection1.Execute(hasil);
end;
end;

procedure TForm3.FormShow(Sender: TObject);
begin
ClientDataSet1.Active:=True;
ClientDataSet1.Refresh;
end;
```

```
procedure TForm3.Timer2Timer(Sender: TObject);
begin
Label12.Caption :=FormatDateTime(' mm /
yyyy',Date);
end;

end.
```

UNIVERSITAS NEGERI SEMARANG
PERPUSTAKAAN
UNNES