PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya dan perkenan-Nya penulis dapat menyelesaikan penulisan skripsi ini.

Karya ini tidak mungkin selesai tanpa restu dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih sedalam-dalamnya yang tidak terhingga kepada:

1. Kedua orang tua penulis, Moch Sodiq dan Ibu Rita Widayani atas pengorbanan, motivasi dan doa restunya sehingga penulis dapat menuntut ilmu sampai jenjang sarjana.
2. Bapak M. Aziz Muslim, S.T.,M.T., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya
3. Bapak Hadi Suyono, S.T., M.T.,Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
4. Bapak Ali Mustofa, S.T.,M.T. selaku Ketua Program Studi Sarjana Teknik Elektro Universitas Brawijaya
5. Ibu Ir. Nurussa'adah, M.T. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya yang selalu memberi semangat dan motivasi untuk cepat menyelesaikan skripsi.
6. Ibu Ir. Nurussa'adah, M.T. dan Bapak Eka Maulana, S.T., M.T., M.Eng. sebagai pembimbing pertama dan sebagai pembimbing kedua, ditengah kesibukan beliau berdua selalu memberikan waktu untuk diskusi dengan tulus, sabar memberikan masukan yang sungguh berharga.
7. Laboran laboraturium elektronika Bapak Mulyadi, S.T. atas semua fasilitas dan bantuan yang diberikan dalam pengerjaan skripsi ini.
8. Para Dosen Pengajar Program Studi Teknik Elektro Universitas Brawijaya, yang tidak dapat penulis sebutkan satu per satu yang telah memberikan bekal ilmu pada penulis dalam menyelesaikan studi.
9. Teman – teman seperjuangan dalam pengerjaan skripsi Cahyo, Wildan, Tata, Gloria, Rifqa, Mas Roqib atas segala dukungan dan bantuan yang telah diberikan

10. Kakak –kakak, teman – teman, dan adik – adik asisten Laboratorium Elektronika Mas Bayu, Mas Nurdin, Mas Rico, Mas Ainun, Mas Bustanul, Akbar, Sofyan , Naufal, Ghadafi atas segala bantuan dalam pengerjaan skripsi
11. Teman – Teman Divisi Mikrokontroler, terima kasih untuk pengalaman, semangat dan kerjasama yang telah terjalin selama ini
12. Teman – teman konsentrasi elektronika 2012 dan elektro 2012 atas segala dukungan dan bantuan dalam pengerjaan skripsi
13. Teman – teman kos Pak Hari, Roni, Wildan, Husni, Fathi, Syafiq, Arif, Ali terima kasih untuk keceriaan setiap hari, semangat dan bantuan selama mengerjakan skripsi ini.

Sekiranya Allah SWT membalas kebaikan semua pihak yang turut membantu skripsi ini terselesaikan. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari sempurna, namun semoga skripsi ini dapat memberikan manfaat bagi kita semua. Amin, Terima kasih.

Malang, Januari 2016

Penulis

DAFTAR ISI

DAFTAR TABEL

UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

DAFTAR LAMPIRAN

RINGKASAN

Mochamad Choiril Iman, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2016, *Rancang Bangun Transmisi Data Berbasis Wireless pada Micro Smart Grid* Dosen Pembimbing : Nurussa'adah dan Eka Maulana.

Kebutuhan energi listrik terus meningkat setiap tahun. Peningkatan kebutuhan listrik dapat disebabkan oleh penggunaan energi listrik yang tidak efisien. Pembangunan *micro smart grid* sistem merupakan upaya peningkatan efisiensi penggunaan energi listrik. Maka perlu adanya sebuah alat yang dapat memantau penggunaan tenaga listrik.

Salah satu point dari *monitoring* adalah transmisi data. Alat ini menggunakan transmisi data *wireless* yang memiliki keunggulan yaitu mengurangi tingkat kerumitan pada penggunaan kabel. Pada alat ini modul *wireless* yang digunakan adalah XBee karena memiliki jarak transmisi yang cukup jauh serta aman dari *hacking*. S*lave* yang digunakan berjumlah 3 buah yang masing – masing mengirimkan data berupa pembacaan sensor arus maupun tegangan. Setelah dikirimkan data dari *slave* kemudian diterima oleh *master* dan diteruskan ke komputer melalui kabel USB. Kemudian data hasil pembacaan sensor diubah menjadi nilai arus, tegangan, daya, energi dan kapasitas baik dari sumber, baterai maupun beban. Data tersebut kemudian ditampilkan dalam bentuk grafik maupun nilai arus, tegangan daya, energi dan kapasitas secara *real time*. Alat ini tidak hanya bekerja pada *single point* tapi juga *multipoint*.

Metode yang digunakan dalam alat ini adalah metode komunikasi *simplex*. Semakin jauh jarak transmisi data maka data yang diterima semakin berkurang. Pada transmisi data *outdoor* jarak transmisi yang persentase keberhasilan 100% mencapai jarak 80 m. Halangan yang paling mempengaruhi adalah kaca. Pada transmisi data *indoor* halangan berupa tembok sangat berpengaruh, hal ini dibuktikan persentase keberhasilan 100% hanya sampai jarak 20 m. Lebar data maksimum yang dapat ditransmisikan dengan persentase keberhasilan 100% adalah 260 *byte.*

Kata kunci: *Smart grid, Wireless,* Transmisi data, *Multipoint*

UNIVERSITAS BRAWIJAYA

SUMMARY

Mochamad Choiril Iman, Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, June 2015, Design Wireless Data Transmission Based On Micro Smart Grid, Academic Supervisor: Nurussa'adah and Eka Maulana.

The electrical energy needs keep increasing each year. The increased in demand for electricity could be caused by the inefficient use of electrical energy. The development of micro smart grid system is an effort to increase the efficiency of electrical energy usage. Hence, a device that able to monitor the use of electricity is needed.

One of the points of monitoring is the data transmission. This device used a wireless data transmission which has advantage by reducing the complexity in cables usage. This wireless module which used in this device is Xbee because of the transmission distance is quite far and safe from hacking. This device use 3 slave which each of them transmits data such as current or voltage sensor value. Once the slave's data transmitted, it received by master then forwarded to the computer through USB cable. Later the data from sensor is converted into current, voltage, power, energy, and capacity value from the source, battery and load. The data displayed in a graphs and the value of current, voltage, power, energy, and capacity in real time.

This device used simplex communication method. The farther distance of data transmission then the data received on the wane. In the outdoor data transmission, the transmission distance which has 100% success percentage reaches a distance of 80 m. The most influence obstacle is glass. In the indoor data transmission, the obstruction of walls is very influential, this was proven by the 100% success percentage is only to a distance of 20 m. The maximum data length that can be transmitted with 100% success percentage is 260 bytes.

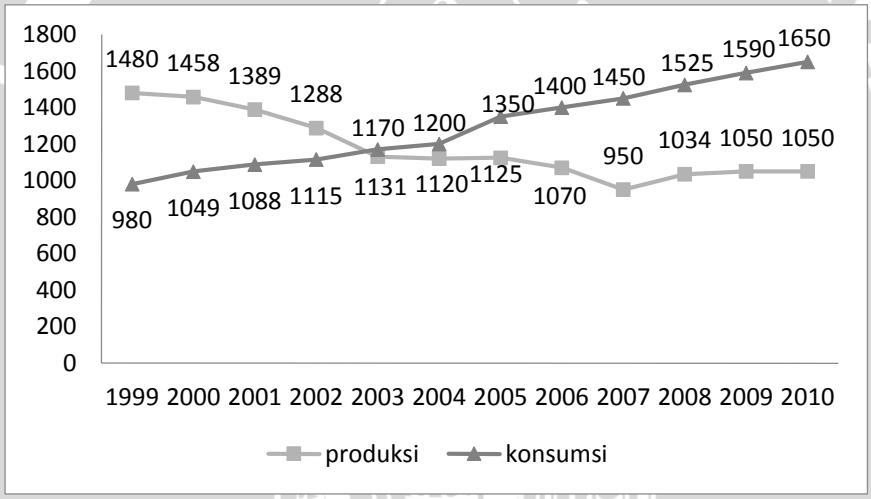
Keywords: Smart grid, Wireless, Data transmission, Multipoint

UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dewasa ini kebutuhan energi di Indonesia semakin meningkat, salah satunya adalah energi listrik. Menurut data kementrian ESDM, kebutuhan listrik di Indonesia terus meningkat hingga 8 persen per tahun. Apabila persentase tersebut diubah menjadi angka mencapai 1800MW. Akan tetapi meningkatnya kebutuhan energi listrik tidak sebanding dengan peningkatan ketersediaan energi listrik. Sejak tahun 2002 kebutuhan listrik di Indonesia sudah melebihi kapasitas pembangkit listrik ditunjukkan dalam Gambar 1. 1



Gambar 1. 1 Grafik perbandingan kebutuhan energi listrik
(Sumber: LPKEE, 2011)

Pemerintah dan Perusahaan Listrik Negara telah berusaha membangun infrastruktur listrik. Langkah yang dilakukan pemerintah ini sangat penting terutama untuk menghasilkan sumber listrik yang memadai karena setiap 1 persen pertumbuhan tentu dibutuhkan sekitar 1.5 persen pertumbuhan suplai (Budi Prasetyo, 2014). Akan tetapi sebagian besar pembangkit ini berbahan bakar fosil, padahal bahan bakar fosil fosil diperkirakan akan habis sekitar 27 tahun lagi, bahan bakar gas habis dalam kurun waktu 67 – 77 tahun lagi, dan bahan bakar padat 117 tahun lagi (Benny N Joewono, 2011).

Salah satu solusi dari masalah tersebut adalah permanenan energi menggunakan sel surya. Hal ini sangat memungkinkan mengingat Indonesia adalah negara beriklim tropis yang mendapatkan sinar matahari terus menerus sepanjang tahun. Teknologi sel surya dianggap masih terlalu mahal bagi masyarakat, karena instalasinya yang rumit dan dianggap tidak dapat memenuhi kebutuhan energi. *Smart grid* menjadi salah satu solusi permasalahan diatas. *Smart grid* adalah suatu jaringan listrik yang telah terkomputerisasi dan terautomatisasi. Di dalam *smart grid* memungkinkan pengguna untuk berinteraksi dengan jaringan sehingga memungkinkan komunikasi 2 arah antara pengguna dan produsen. Selain itu transfer energi listrik yang terjadi tidak hanya dari produsen ke pengguna namun juga sebaliknya.

Salah satu point penting dari *smart grid* adalah transmisi data. Transmisi data digunakan untuk memantau keadaan dari setiap node. Data dikirimkan dari *slave* menuju *master* agar *master* dapat memberikan keputusan. Salah satu sistem transmisi yang berkembang pesat sekarang adalah *wireless*. Dengan keuntungan berkurangnya tingkat kerumitan pada penggunaan kabel, sistem transmisi data *wireless* menjadi salah satu metode alternatif transmisi data yang cukup banyak digunakan pada saat ini. Modul *wireless* yang beredar di pasaran sangat banyak. Di antaranya adalah *bluetooth*. Akan tetapi modul tersebut mempunyai kelemahan yaitu jarak transmisi yang tidak begitu jauh. Selain itu apabila menggunakan *wifi* maka sistem rawan penyadapan. Oleh karena dalam program ini digunakan modul XBee yang memiliki jarak transmisi yang cukup jauh serta aman dari *hacking*. Berdasarkan permasalahan tersebut peneliti akan merancang sebuah sistem transmisi data berbasis *wireless* pada *micro smart grid* yang dirancang menggunakan modul XBee dan hasil monitoringnya ditampilkan pada PC secara *realtime*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penyusunan skripsi ini adalah:

1. Bagaimana merancang sistem yang dapat mentransmisikan data pada *micro smart grid* dan dapat ditampilkan pada grafik secara *realtime*?
2. Bagaimana pengaruh jarak, halangan dan format data terhadap transmisi data pada *micro smart grid*?
3. Bagaimana karakterisasi tegangan pada *slave* saat proses pengiriman data?

1.3 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal – hal yang berkaitan dengan alat akan diberikan batasan sebagai berikut:

1. Pembacaan yang dilakukan adalah besaran hasil dari unit pengukuran berupa arus (maksimal 20 A), tegangan (maksimal 24 V).
2. Jarak antara *master* dan *slave* sekitar ±50 m.
3. Jumlah *slave* yang digunakan 3 buah.
4. Mikrokontroler Arduino Mega 2560 sebagai pengolah data.
5. Modul transmisi data yang digunakan adalah XBee.

1.4 Tujuan

Tujuan dari penelitian ini adalah merancang dan membuat suatu sistem yang dapat mentransmisikan data pada *micro smart grid* serta dapat ditampilkan secara *realtime* dan mengetahui pengaruh dari jarak, halangan, dan format data terhadap transmisi data pada *micro smart grid* serta mengetahui karakterisasi tegangan pada *slave* pada saat proses pengiriman data.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori teori yang mendukung dalam perencanaan dan pembuatan sistem

BAB III Metode Penelitian dan Perancangan

Menjelaskan tentang tahapan penyelesaian skripsi

BAB IV Hasil dan Pembahasan

Memuat hasil pengujian terhadap sistem yang telah direalisasikan

BAB V Penutup

Memuat kesimpulan dan saran berdasarkan apa yang telah dicapai dalam penyelesaian skripsi.

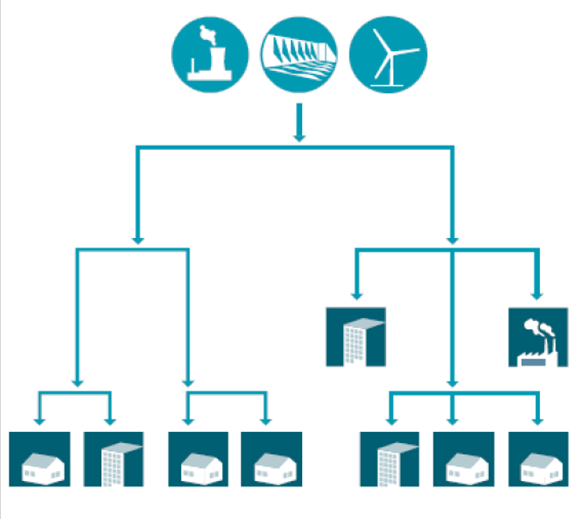
BAB II
TINJAUAN PUSTAKA

Beberapa teori pendukung yang perlu dibahas dalam pembuatan sistem ini meliputi literatur mengenai teknologi *smart grid*, *wireless*, XBee, Arduino Mega 2560, Microsoft Visual Studio.

2.1 Teknologi Smart Grid

2.1.1 Perbandingan Jaringan Listrik Tradisional dengan Smart Grid

Menurut Momoh (2012: 1), sistem jaringan listrik yang ada saat ini didesain untuk dioperasikan dengan struktur vertikal, termasuk pembangkitan, transmisi dan distribusinya yang didukung dengan kontrol dan peralatan yang menjaga keandalan, stabilitas dan efisiensi. Hal ini membuat pengguna tidak mendapatkan banyak informasi dan tidak berpartisipasi aktif dalam pendistribusian energi. Sistem jaringan listrik tradisional ditunjukkan pada Gambar 2. 1

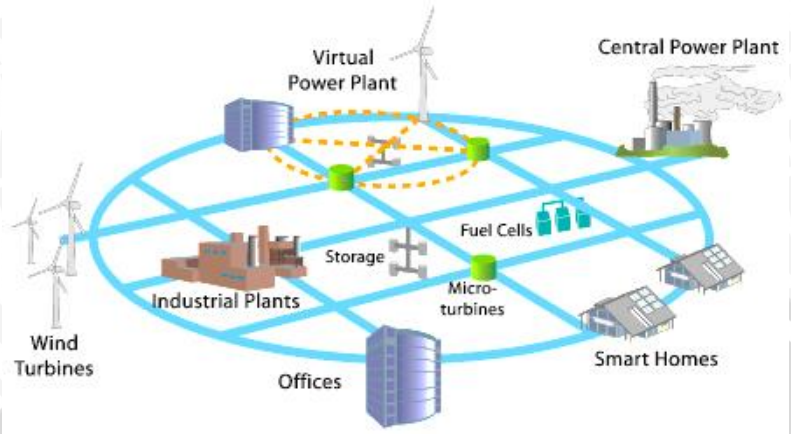


Gambar 2. 1 Jaringan listrik tradisional

Sumber: ABB (2009: 3)

Secara umum, *smart grid* merupakan generasi baru jaringan listrik yang mampu menyelesaikan berbagai permasalahan pada model jaringan listrik tradisional, seperti isu ekonomi dan emisi karbon yang merupakan masalah lingkungan (Nejad, 2013:1). Model jaringan listrik dalam *smart grid* meliputi: menangani dan mengontrol pendistribusian transfer energi antar area, mengakomodasi sumber energi terbarukan, mengatasi masalah

yang tidak terduga dan sistem komunikasi jaringan (*network*) (Momoh, 2012: 1). Model sistem *smart grid multipoint* ditunjukkan pada Gambar 2. 2



Gambar 2. 2 Sistem *Smart Grid Multipoint*

Sumber: Powegenasia (2013)

Perbandingan antara jaringan tradisional dengan *smart grid* dijelaskan dalam Tabel 2. 1

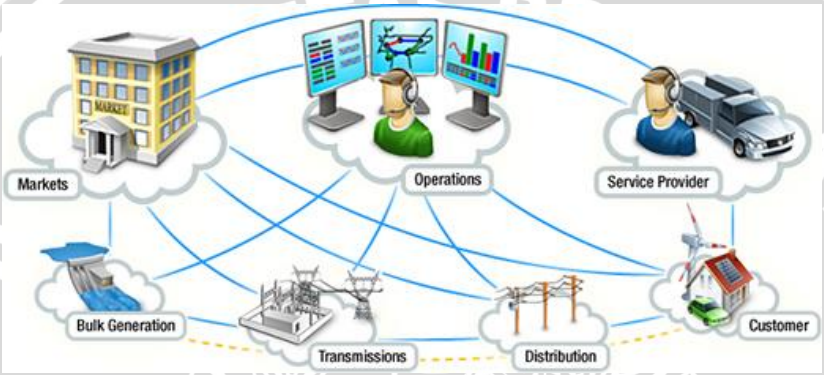
Tabel 2. 1 Perbandingan Traditional Grid dan *Smart Grid*

Traditional Grid	**Smart Grid**
- Pembangkitan energi tersentralisasi	- Pembangkitan energi terdistribusi
- Aliran daya satu arah	- Aliran daya lebih dari satu arah (*multi direction*)
- Pembangkitan mengikuti beban	- Beban mengikuti pembangkit
- Keterbatasan akses pada jaringan untuk jenis pembangkit baru	- Akses jaringan tidak terbatas dan efisien
- Operasi berdasarkan pengalaman	- Operasi berdasarkan data *real time*
- Konsumen tidak berpartisipasi aktif	- Konsumen berpartisipasi secara aktif
	- Memanfaatkan pembangkit energi terbarukan secara bergantian

(ABB: 2009)

2.1.2 Manajemen dan Aplikasi *Smart Grid*

Dibutuhkan *smart management* untuk menangani sistem *smart grid* yang kompleks agar transmisi dan distribusi energi dapat dikelola dengan lebih efektif. Manajemen sistem *smart grid* yang disebut dengan *Smart Power Management System* melingkupi lima hal utama, yaitu: pemodelan sistem *smart grid,* memonitor sistem *smart grid* secara *real time,* membuat teknik pengambilan basis data (*database*) dan manajemennya, pengecekan kondisi jaringan dan pemeliharaan peralatan, serta menganalisis kesalahan jaringan *smart grid* (Tang, 2011: 1). Konsep *Smart Power Management System* untuk mengelola sebuah jaringan *smart grid* dijelaskan pada Gambar 2. 3



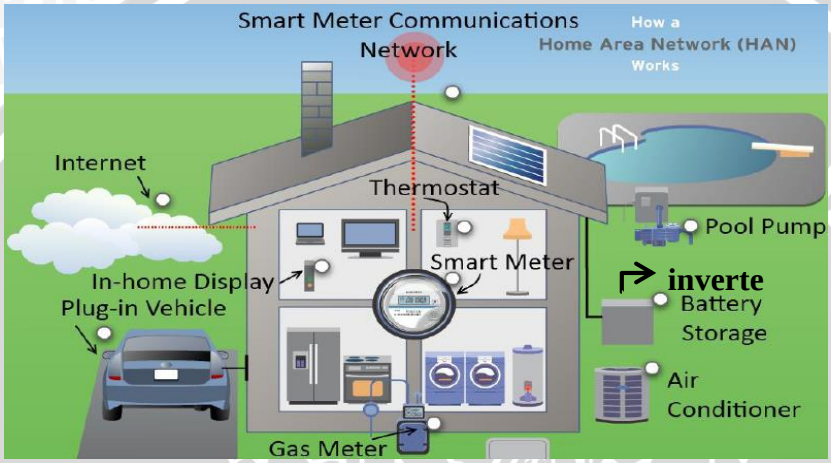
Gambar 2. 3 Konsep *Smart Power Management System*
Sumber: Hardin (2011: 1)

Berdasarkan sistem antarmukanya desain *Smart Power Management System* dibagi menjadi tiga hal

i. *Home energy management*
ii. *Building energy management*
iii. *Grid Interconnect*

Mengacu pada hal tersebut, masih banyak ruang untuk mengembangkan sistem *smart grid* menjadi lebih andal dari saat ini. Salah satu aplikasi yang masih perlu banyak pengembangan dan dekat dengan pengguna adalah HAN (*Home Area Network).* HAN merupakan infrastruktur yang dikontrol dengan teknologi tinggi sehingga memungkinkan komunikasi dua arah antara alat-alat pengguna dengan *supplier*. HAN juga memungkinkan pengguna untuk mengatur penggunaan energi alat-alat rumah tangganya dari pembangkit energi mandiri maupun *supplier* menggunakan sistem monitoring yang *real time* (Leeds, 2009: 81). Hal ini akan terealisasi dengan baik jika didukung dengan instalasi listrik yang baik pada *smart home* (Dian, 2014:1)

Secara mandiri setiap *smart home* akan memiliki sumber energi listrik yang salah satunya berasal dari panel surya dan energi listrik tersebut disimpan untuk kebutuhan pasokan listrik berbagai alat elektronik dan rumah tangganya. Melalui *smart metering* pengguna dapat secara aktif mengatur berapa banyak daya yang digunakan sendiri dan berapa banyak daya yang dibutuhkan dari *supplier* lain. Jika daya yang dimiliki dari sumber energi listriknya berlebih maka daya tersebut dapat dijual dan disalurkan kepada pengguna lain yang membutuhkan melalui jaringan *smart grid*. Aplikasi HAN (*Home Area Network*) yang terkoneksi dengan sistem *smart grid* dijelaskan pada Gambar 2. 4



Gambar 2. 4 *Home Area Network* untuk aplikasi *Smart Home*

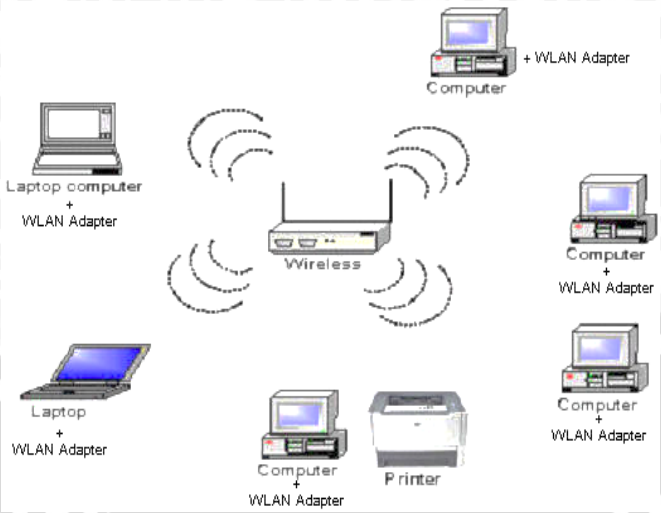
Sumber: Leeds (2009:81)

2.2 Wireless

Telekomunikasi nirkabel adalah transfer informasi antara dua atau lebih titik yang tidak terhubung oleh penghantar listrik. Jarak transmisi bisa pendek, seperti beberapa meter untuk *remote control* televisi, atau sejauh ribuan atau bahkan jutaan kilometer untuk ruang-dalam komunikasi radio. Contohnya *mobile*, dan portabel radio dua arah, telepon seluler, *personal digital assistant* (PDA), dan jaringan nirkabel. Contoh lain dari teknologi nirkabel termasuk GPS unit, pembuka pintu garasi atau pintu garasi, *wireless mouse* komputer, *keyboard* dan *headset* (audio), *headphone*, penerima radio, televisi satelit, siaran televisi tanpa kabel dan telepon.

Kelebihan utama dari komunikasi ini adalah tidak menggunakan kabel dalam komunikasinya, akan tetapi menggunakan gelombang radio. Keuntungannya adalah sifat mobilitasnya yang tinggi dan tidak tergantung kepada kabel dan koneksi tetap. Kelebihan lainnya dari komunikasi *wireless* adalah memungkinkan pengguna untuk mengakses informasi secara *realtime* selama masih dalam jangkauan komunikasi ini, sehingga

meningkatkan kualitas layanan dan produktivitasnya. Sistem komunikasi *wireless* ditunjukkan pada Gambar 2. 5



Gambar 2. 5 Gambar Jaringan *Wireless*

(Sumber: Elind, 2012)

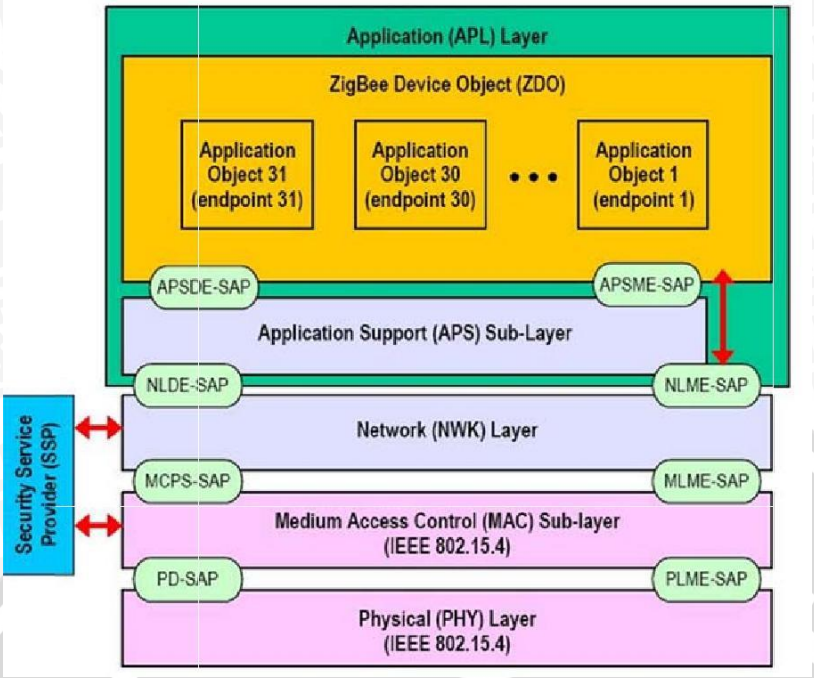
2.3 XBee

Xbee adalah sebuah spesifikasi *protocol* komunikasi radio digital berdaya rendah berdasarkan spesifikasi IEEE 802.15.4 tahun 2003 dan Zigbee Allience dengan jangkauan maksimal 120 meter. Spesifikasi IEEE 802.15.4 merupakan dasar dari ZigBee untuk lapisan bawah MAC dan PHY serta menentukan standar radio 2,4 GHz yang digunakan dunia. XBee adalah brand yang mensupport dari berbagai protokol komunikasi termasuk ZigBee 802.15.4 dan WiFi. Gambar Xbee modul diperlihatkan pada Gambar 2. 6

Gambar 2. 6 Modul XBee

(Sumber: Digi International Inc, 2015)

Standar protocol XBee sama dengan standar *Bluetooth.* Arsitektur teknologi XBee ditunjukkan pada Gambar 2. 7



Gambar 2. 7 Arsitektur teknologi XBee

(Sumber: Yuliza, 2014)

XBee banyak digunakan karena XBee mempunyai banyak keunggulan yaitu (Gilslason, 2008):

- Jangkauan 1 meter – 120 meter
- ISM (*Industrial, Scientific & Medical*) radio bands: 2.4 GHz, 868 MHz dan 915 MHz.
- Konsumsi daya rendah.
- CSMA-CA *channelacces*
- Jaringan besar (65000 node)
- Sangat aman (AES enkripsi)
- Jaringan topologi star, mesh dan saling mendukung berbagai aplikasi.
- Interopabilitas di seluruh dunia dengan produk – produk lainnya.
- Co-eksistensi dengan media nirkabel lainnya (Misalnya, WLAN, *Bluetooth*, selular).

2.4 Arduino mega 2560

Arduino Mega2560 adalah papan mikrokontroler berbasiskan ATmega2560 (datasheet ATmega2560). Arduino Mega2560 memiliki 54 pin digital *input/output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (port serial hardware), 16 MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan

menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. Arduino Mega2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega2560 adalah versi terbaru yang menggantikan versi Arduino Mega.

Dibanding versi sebelumnya Arduino Mega2560 R3 memiliki fitur-fitur baru berikut:

- pinout: Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan shield untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, *shield* akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino Due yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.
- Sirkuit RESET.
- Chip ATmega16U2 menggantikan chip ATmega8U2.

Modul Arduino Mega 2560 ditunjukkan dalam Gambar 2. 8

Gambar 2. 8 Arduino Mega 2560 R3 Bagian Depan
(Sumber: http://www.arduino.cc)

Di bawah ini spesifikasi sederhana dari Arduino Mega 2560:

Mikrokontroler	ATmega2560
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Input Voltage (limit)	6-20V
Pin Digital I/O	54 Pin (15 Pin Output PWM)
Pins Input Analog	16

Arus DC per pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	256 KB (8 KB untuk bootloader)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Arduino ATmega2560 memiliki 256 KB *flash memory* untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi pinMode() , digitalWrite() , dan digitalRead(). Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.
- Eksternal Interupsi: Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
- SPI: Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
- LED: Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).
- TWI: Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi analogReference().

Ada beberapa pin lainnya yang tersedia, antara lain:

- AREF: Referensi tegangan untuk input analog. Digunakan dengan fungsi analogReference().
- RESET: Jalur LOW ini digunakan untuk me-*reset* (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino ATmega328 menyediakan 4 hardware komunikasi serial UART TTL (5 Volt). Sebuah chip ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM *Port Virtual* (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino termasuk didalamnyaserial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

Sebuah perpustakaan SoftwareSerial memungkinkan untuk komunikasi serial pada salah satu pin digital Mega2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk perpustakaan Wire digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan perpustakaan SPI.

2.5 Microsoft Visual Studio 2012

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development*

Environment (IDE), dan dokumentasi (umumnya berupa MSDN *Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET dll.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Pada Visual Studio 2012, memiliki berbagai fitur di antaranya:.

- *Designing* dan membangun Windows Store apps.
- *Debugging, optimizing* dan *publishing*Windows Store apps.
- *Projects* dan Solusinya.
- *Window Management*.
- *Code Editing for* C++
- *Code Editing for JavaScript.*
- Visual Basic.
- Visual C#
- Visual C++
- JavaScript.
- Visual F#
- *Managing the application lifecycle.*
- *Modeling applications.*
- *Developing applications* dan berkolaborasi dengan orang lainnya dalam satu team.
- *Automating* dan *debugging* builds.
- Microsoft Test Manager.
- ASP.NET 4.5 Core Services.
- ASP.NET 4.5 Web Forms.
- *General Enhancements for Web Development*.
- *Data-Related Enhancements for Web Development*.
- *IIS Express for Web Development.*
- ASP.NET Web API.
- *LightSwitch.*
- *Data Application Development*

- *Graphics Tools*
- *Parallel Computing*
- *SharePoint Development*
- *Office Development*
- .NET Framework 4.5

BAB III
METODE PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Penyusunan proposal ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah - langkah yang perlu dilakukan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara keseluruhan perlu ditentukan terlebih dahulu sebagai acuan untuk mendapatkan sistem yang sesuai dengan keinginan dan dapat bekerja dengan efektif serta efisien. Alat yang dirancang memiliki spesifikasi sebagai berikut:

1. Modul XBee digunakan sebagai pemancar dan penerima data yang dikirimkan *slave* ke *master*.
2. Arduino Mega 2560 digunakan sebagai pengolah data dari sensor dan mengirim data ke *master*. Selain itu Arduino Mega 2560 digunakan untuk menerima data dari *slave*, kemudian meneruskan data ke PC.
3. Microsoft Visual Studio sebagai *software* penampil grafik pembacaan arduino secara *realtime*.

3.1.2 Studi Literatur

Pengumpulan data dilakukan dengan melakukan studi literatur (*library research*), penelusuran informasi digital, dan wawancara narasumber dengan sasaran tinjauan antara lain:

1. Informasi internet.
2. Pustaka-pustaka referensi.
3. Pustaka penunjang.

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung perencanaan dan perealisasian alat. Teori - teori yang dikaji adalah sebagai berikut:

1. Teori mengenai *smart grid*.
2. Teori mengenai komunikasi *wireless.*
3. Teori umum mengenai XBee.
4. Teori umum mengenai Arduino Mega 2560.
5. Teori umum mengenai Microsoft Visual Studio

3.1.3 Perancangan dan Pembuatan Alat

Dalam perancangan dan perealisasian alat membahas tentang diagram blok, perencanaan perangkat keras, dan perencanaan perangkat lunak.

A. Diagram Blok

Pembuatan diagram blok merupakan dasar dari perancangan sistem agar perancangan dan perealisasian alat berjalan secara sistematis.

B. Perancangan Perangkat Keras

Perancangan perangkat keras meliputi perencanaan rangkaian *slave* dan rangkaian *master*.

1. Rangkaian Slave

Rangkaian *slave* dirancang agar dapat membaca nilai tegangan dan arus. Selain itu rangkaian *slave* juga berfungsi untuk mengirimkan data ke *master* untuk kemudian diolah oleh komputer.

2. Rangkaian *Master*

Rangkaian *master* dirancang agar dapat menerima data dari *slave*, kemudian meneruskannya ke komputer untuk diolah dan ditampilkan secara *realtime.*

C. Perancangan Perangkat Lunak

Perencanaan dan pembuatan perangkat lunak digunakan untuk mengendalikan dan mengatur kerja alat. *Design* dan parameter yang telah dirancang kemudian diterapkan pada mikrokontroler Arduino Mega 2560 dengan menggunakan bahasa C dan program *compiler*.

3.1.4 Pengujian Alat

Untuk menganalisis kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada masing-masing blok pada

perancangan perangkat keras serta pengujian keseluruhan untuk mengetahui perangkat lunak dapat bekerja dengan baik atau tidak.

A. Pengujian Perangkat Keras

Pada bagian ini pengujian dilakukan pada masing-masing blok. Pengujian ini bertujuan untuk mengetahui apakah masing - masing blok dapat bekerja sesuai dengan fungsinya seperti yang telah direncanakan. Pengujian tersebut meliputi:

1. Pengujian Rangkaian *Slave*

Pengujian dilakukan untuk mengetahui apakah rangkaian *slave* dapat mengirimkan data ke *master*. Pengujian dilakukan dengan cara data tegangan dan arus dikimkan ke kompoter melalui kabel dan di cek melalui serial monitor.

2. Pengujian Rangkaian *Master*

Pengujian dilakukan untuk mengetahui apakah rangkaian *master* dapat menerima data dari *slave* dan meneruskannya ke komputer. Pengujian dilakukan dengan cara data dikirimkan dari *slave* ke *master*. Kemudian data yang dikirimkan dibandingkan dengan data yang diterima. Setelah itu data diteruskan ke komputer melalui kabel dan di cek menggunakan serial monitor.

3. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini dengan menyambungkan semua hardware yang dibuat berdasarkan blok diagram dan memasukkan program berupa *software* yang bekerja untuk mengendalikan hardware yang telah dibuat. Sistem bekerja dengan baik jika dapat berjalan sesuai *flowchart* yang telah direncanakan.

3.1.5 Pengambilan Kesimpulan

Kesimpulan didapat berdasarkan hasil perealisasian sistem transmisi data secara *wireless* yang telah dibuat.

3.2 Perancangan

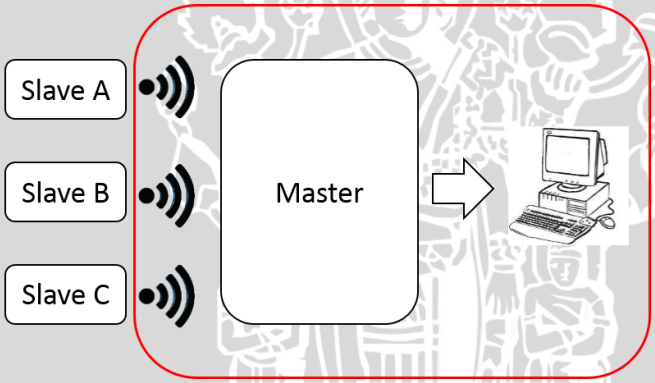
Perancangan ‘Rancang Bangun Transmisi Data Berbasis *Wireless* pada *Micro Smart Grid’* dilakukan secara bertahap sehingga memudahkan dalam analisis setiap bloknya

maupun secara keseluruhan. Adapun perencanaan untuk membangun sistem ini dibagi menjadi beberapa tahap antara lain:

1. Perencaan Sistem yang digambarkan dalam sebuah diagram blok
2. Perancangan perangkat keras yang meliputi perancangan rangkaian *slave* dan ragkaian *master*
3. Perancangan perangkat lunak (*software*) dibagi menjadi tiga bagian utama yaitu program pada mikrokontoler *slave* yang terdiri atas program utama, sub program sensor arus, sub program sensor tegangan, dan sub program pengiriman data. Program pada mikorkontroler *master* yang berisi program penerimaan data dari *slave* dan pengiriman data ke komputer. Program Desktop pada komputer yang berfungsi untuk mengolah dan menampilkan data yang dikirim oleh mikrokontroler *slave*.

3.2.1 Diagram Blok Perencanaan Sistem

Diagram blok 'Rancang Bangun Transmisi Data Berbasis *Wireless* pada *Micro Smart Grid'* ditunjukkan dalam Gambar 3.1, Gambar 3.2 dan Gambar 3.3



Gambar 3.1 Diagram blok sistem keseluruhan (*multi node*).

Gambar 3.2 Diagram blok sistem pada sebuah *slave* (*single node*).

Gambar 3.3 Diagram blok sistem pada *master*

Penjelasan mengenai diagram blok sistem yang ditunjukkan dalam Gambar 3.1 adalah sebagai berikut:

1. Dalam penelitian ini dibuat tiga buah *slave*.
2. Ketiga *slave* tersebut saling megirimkan data ke *master*.
3. Ketiga *slave* tersebut mengirimkan data ke master dengan komunikasi *wireless* menggunakan modul XBee.
4. *Master* merupakan mikrokontroler yang berfungsi untuk mengirim/meneruskan data dari *slave* menuju komputer
5. Komputer berfungsi untuk menerima data dari *master* kemudian mengolahnya dan menampilkan dalam bentuk visual secara *real time*.

Adapun penjelasan mengenai diagram blok sistem yang ditunjukkan dalam Gambar 3.2 adalah sebagai berikut:

1. Unit pengukuran merupakan rangkaian untuk membaca nilai arus, tegangan, daya, energi dan kapasitas dari *solar cell*, baterai maupun beban. Pada unit pengukuran terdiri dari rangkaian sensor arus, sensor tegangan, rangkaian pengkondisi sinyal, rangkaian LCD, rangkaian RTC, dan rangkaian memori.
2. Arduino Mega 2560 merupakan mikrokontroler *slave* yang berfungsi sebagai pembaca nilai tegangan sensor arus maupun tegangan. Selain itu Arduino Mega 2560 juga menampilkan hasil pembacaan sensor arus maupun tegangan. Mikrokontroler ini juga melakukan penyimpanan data pembacaan sensor secara berkala. Arduino Mega 2560 juga mengirimkan data hasil pembacaan sensor ke *master* secara *wireless* melalui rangkaian pemancar.
3. Modul XBee digunakan untuk mengirimkan data hasil pembacaan sensor pada *slave* ke *master* untuk kemudian diteruskan ke komputer untuk diolah dan ditampilkan pada grafik secara *real time*

Adapun penjelasan mengenai diagram blok sistem yang ditunjukkan dalam Gambar 3.3 sebagai berikut:

1. Modul XBee digunakan untuk menerima data hasil pembacaan sensor pada *slave*

2. Arduino Mega 2560 merupakan mikrokontroler *master* yang berfungsi untuk meneruskan data dari *slave* ke komputer untuk kemudian diolah oleh komputer dan ditampilkan baik dalam bentuk data maupun grafik.
3. Komputer digunakan umtuk mengolah dan menampilkan data hasil pembacaan sensor baik dalam bentuk data maupun grafik.

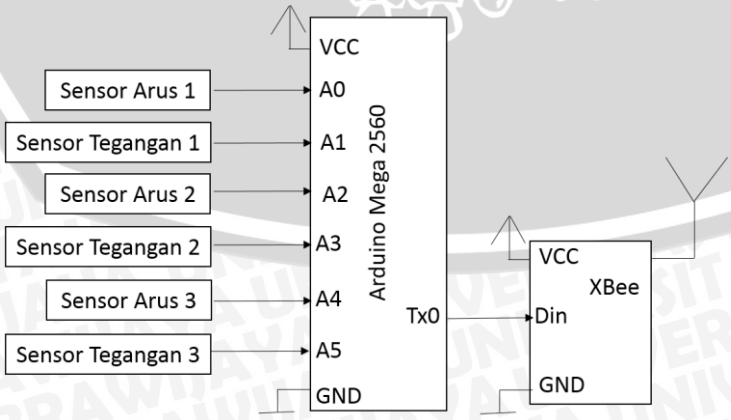
3.2.2 Prinsip Kerja Sistem

Prinsip kerja sistem berdasarkan blok diagram yang telah digambarkan dalam Gambar 3.1, Gambar 3.2 dan Gambar 3.3 adalah data pembacaan sensor arus maupun tegangan baik dari *solar cell*, baterai maupun beban bersama dengan alamat dari masing – masing *slave* dikirimkan menuju *master* melalui modul XBee. Kemudian *master* menerima data dari masing – masing *slave* dan meneruskannya ke PC. Di dalam PC data dari *slave* kemudian dipisahkan antara alamat dan data – data pembacaan sensor arus maupun tegangan. Kemudian data ditampilkan dalam program desktop berupa hasil pembacaan sensor dan grafik daya dari sumber, baterai dan beban dari masing – masing *slave*.

3.2.3 Perancangan Perangkat Keras (*Hardware*)

3.2.3.1 Rangkaian *Slave*

Rangkaian *slave* dirancang agar dapat membaca nilai tegangan dan arus dari *solar cell*, baterai dan beban. Kemudian rangkaian *slave* juga dapat mengirimkan data pembacaan sensor ke *master*. Rangkaian *slave* terdiri dari antarmuka sensor dengan mikrokontroler dan rangkaian antarmuka mikrokontrooler dengan rangkaian pemancar. Blok diagram rangkaian *slave* ditunjukkan dalam Gambar 3. 4

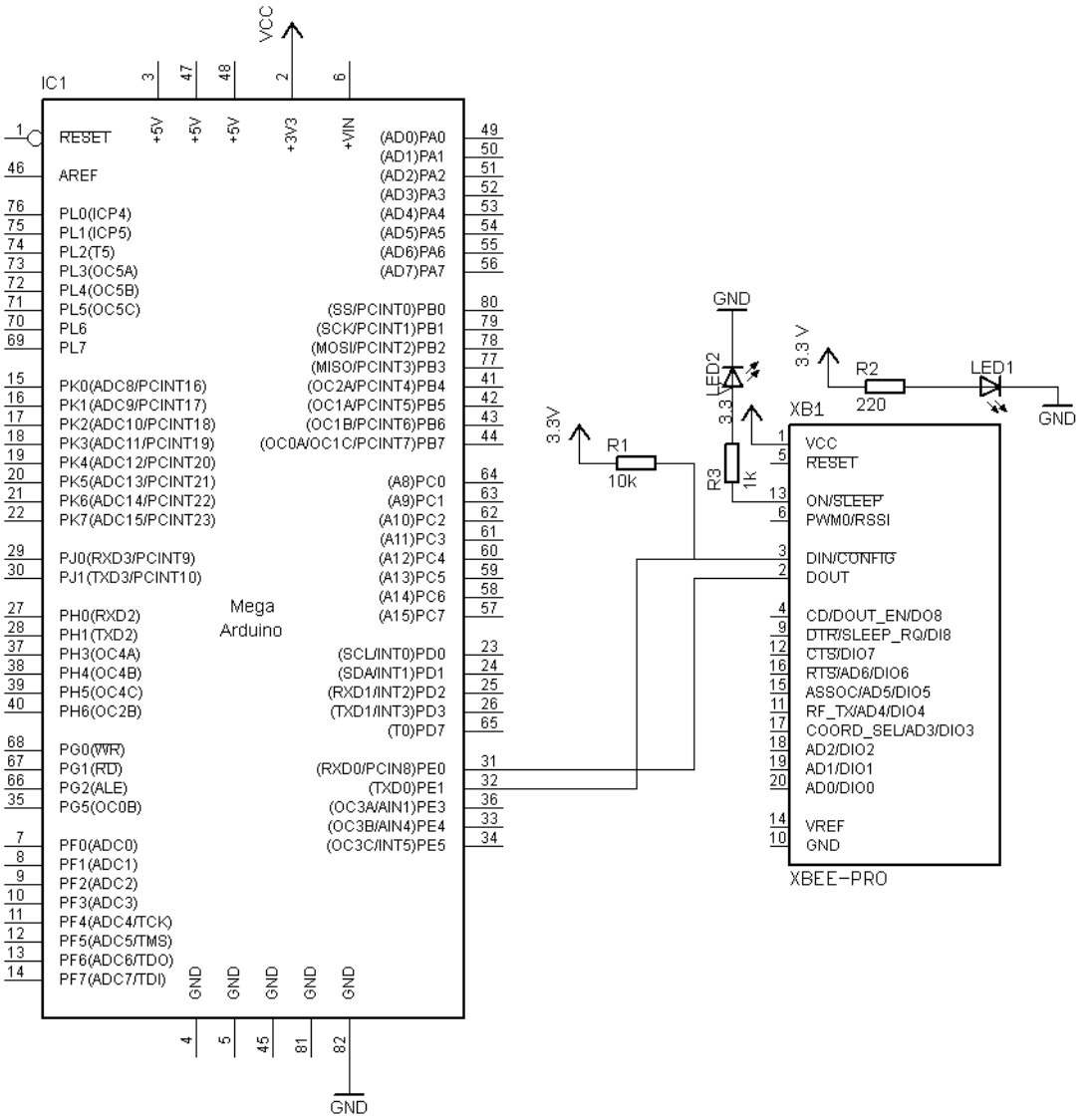


Gambar 3. 4 Blok diagram rangkaian *slave*

Pada rangkaian *slave* pin untuk membaca sensor adalah pin A0 sampai A5 karena pada pin tersebut termasuk pin ADC sesuai dengan keluaran sensor arus maupun sensor tegangan. Kemudian untuk mengirimkan data ke *master* pin Tx_0 dari Arduino Mega 2560 dihubungkan dengan pin Din XBee.

3.2.3.2 Rangkaian *Master*

Rangkaian *master* dirancang dengan tujuan agar dapat menerima data pembacaan sensor dari *slave*. Selain menerima data dari *slave* rangkaian *master* juga mengirimkan data ke komputer untuk diolah. Rangkaian *master* terdiri dari rangkaian antarmuka mikrokontroler dan rangkaian penerima data. Rangkaian *master* ditunjukkan dalam Gambar 3. 5



Gambar 3. 5 Rangkaian *master*

Pada rangkaian *master* pin Rx0 dari Arduino Mega 2560 dihubungkan ke pin Dout dari XBee untuk menerima data dari *slave*. Sedangkan pin Tx0 dari Arduino Mega 2560 dihubungkan ke pin Din XBee. Untuk catu daya yang digunakan oleh XBee adalah 3.3 V yang diambil dari pin 3.3 V Arduino. Untuk meneruskan data dari *slave* ke komputer, rangkaian master menggunakan kabel USB

3.2.4 Perancangan Perangkat Lunak

Perangkat lunak yang dirancang meliputi perangkat lunak untuk masing – masing *slave*, perangkat lunak untuk mikrokontroler *master* dan perangkat lunak pada komputer.

3.2.4.1 Perancangan Perangkat Lunak Masing – Masing *Slave*

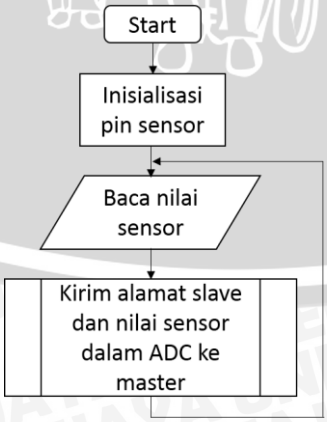
Perangkat lunak masing – masing *slave* dirancang agar dapat membaca nilai sensor, dan mengirimkan data pembacaan sensor ke *master*.

Pada perancangan perangkat lunak masing – masing *slave* terdapat program utama dan sub program pembacaan sensor dan pengiriman data ke *master*.

3.2.4.1.1 Program Utama

Tujuan program utama pada *slave* adalah untuk menjaga sistem agar bekerja secara berurutan sesuai dengan algoritma yang dirancang dan mampu menjalankan fungsinya dengan baik.

Fungsi kerja program utama antara lain mengambil data pembacaan sensor, dan mengirim data pembacaan sensor menuju *master*. Diagram alir program utama ditunjukkan dalam Gambar 3. 6

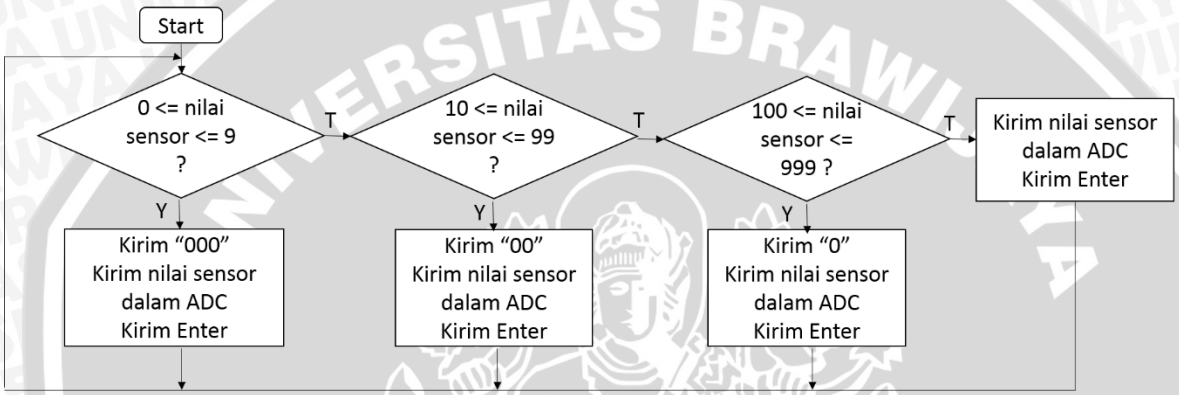


Gambar 3. 6 Diagram alir program utama pada *slave*.

Inisialisasi pin sensor berfungsi untuk mendeklarasikan pin pin yang akan digunakan untuk pembacaan sensor. Kemudian masing – masing *slave* diberikan alamat agar mudah untuk mengenal alamat *slave* dan mengolahnya di komputer. Selain mengirimkan alamat, *slave* juga mengirimkan data pembacaan sensor ke *master* untuk kemudian diolah.

3.2.4.1.2 Sub Program Kirim Data ke Master

Sub program kirim data ke master adalah fungsi yang dipanggil untuk mengirimkan data alamat dan data pembacaan sensor ke *master*. Diagram alir sub program kirim data ke *master* ditunjukkan dalam Gambar 3. 7



Gambar 3. 7 Diagram alir sub program kirim data ke *master*

Pada diagram alir sub program kirim data ke *master* data pembacaan sensor diberi alamat terlebih dahulu agar dapat dikenali dari *slave* mana program berasal. Kemudian data pembacaan dikondisikan agar jumlah data dari setiap sensor adalah 4 *byte* untuk memudahkan dalam proses *parsing* data. Format data yang dikirim ditunjukkan pada Gambar 3. 8

Header	I_s	V_s	I_b	V_b	I_l	V_l	Enter

Gambar 3. 8 Format data

Keterangan:

I_s : data arus sumber (4 *byte*)

V_s : data tegangan sumber (4 *byte*)

I_b : data arus baterai (4 *byte*)

V_b : data tegangan baterai (4 *byte*)

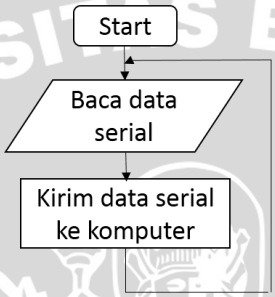
I_l : data arus beban (4 *byte*)

V_l : data tegangan beban (4 *byte*)

Format data yang dikirim diawali dengan *header* atau alamat dari *slave* dan diakhiri dengan *enter*. Setelan dikondisikan data pembacaan sensornya, data kemudian dikirim ke *master* untuk kemudian diolah.

3.2.4.2 Perancangan Perangkat Lunak Master

Perangkat lunak *master* dirancang agar dapat menerima data dari *slave* lalu meneruskan data tersebut ke komputer. Diagram alir program utama pada *master* diperlihatkan dalam Gambar 3. 9



Gambar 3. 9 Diagram alir program utama pada *master*.

Pada Gambar 3.9 ditunjukkan fungsi dari master yaitu menerima data dari *slave* dan melanjutkannya ke komputer. Data diterima oleh *master* dari *slave* melalui rangkaian penerima. Kemudian dari *master* data diteruskan ke komputer melalui kabel USB untuk diolah.

3.2.4.3 Perancangan Perangkat Lunak Desktop

Perancangan perangkat lunak pada komputer bertujuan agar data dari sistem dapat diolah, dipantau dan ditampilkan dalam grafik secara *realtime*.

Pada perancangan perangkat lunak *desktop* meliputi konversi ADC, perancangan *layout* tampilan, program utama, sub program kelompokkan data.

3.2.4.3.1 Konversi Nilai ADC

Konversi nilai ADC digunakan untuk mengubah nilai ADC menjadi nilai arus dan tegangan. Hal ini dikarenakan data yang dikirim oleh *slave* masih berupa nilai ADC dan data yang ditampilkan pada komputer adalah data nilai tegangan dan arus.

ADC yang digunakan adalah ADC 10 bit sehingga bit ADC maksimal yang kita gunakan adalah:

$$ADC = 2^n \quad (4-1)$$

Sehingga bit ADC maksimal yang kita gunakan adalah:

$$ADC = 2^{10}$$

$$ADC = 1024$$

Rumus untuk merubah bit ADC menjadi nilai tegangan ditunjukkan dalam Persamaan (4 - 2)

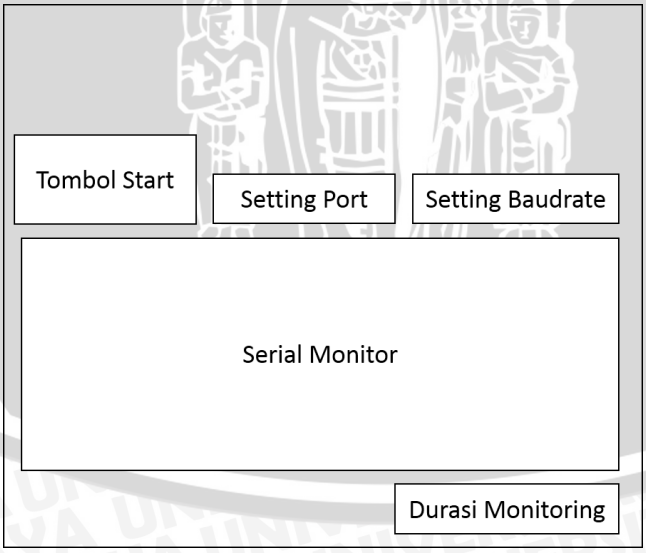
$$Vin = \frac{ADC}{1024}(V_{REF}) \quad (4-2)$$

Untuk range arus yang diukur pada penelitian ini adalah 0 – 20 A. Sedangkan range tegangan pada penelitian ini adalah 0 – 24 volt

3.2.4.3.2 Perancangan *Layout* Tampilan

Layout tampilan dirancang untuk mempermudah dalam melakukan pengawasan pada sistem dan mempermudah melakukan kontrol pada sistem. Pada perancangan layout tampilan dibuat 4 tab untuk memudahkan dalam pemantauan data arus, tegangan, daya, energi dan kapasitas dari masing – masing *slave*.

Pada layout tampilan tab pertama menujukkan pengaturan *port* dan *baudrate* dari program *desktop*. Selain itu juga terdapat *serial monitor* untuk memudahkan dalam melihat *slave* mana saja yang mengirimkan data. Gambar *layout* tampilan tab pertama yang direncanakan ditunjukkan dalam Gambar 3. 10

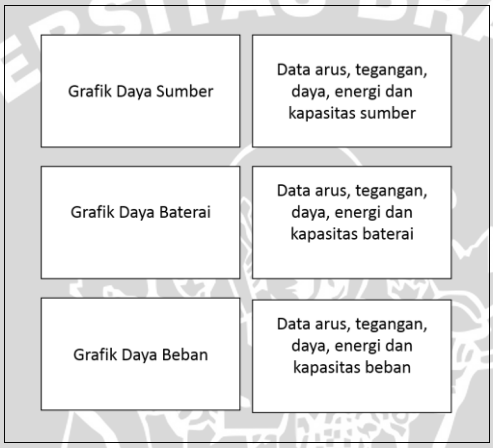


Gambar 3. 10 Rencana *layout* tampilan pada tab pertama *dekstop*.

Pada perencaanaan *layout* tampilan tab pertama terdapat beberapa panel yang memiliki fungsi – fungsi tertentu. Tombol *start* digunakan untuk memulai program *desktop*. Selain

memulai ketika program dijalankan, tombol *start* akan berubah menjadi tombol *pause* yang berfungsi untuk jeda program *desktop*. Kemudian tombol *setting port* digunakan untuk mengatur *port* mana yang akan digunakan. Begitu pula dengan *setting baudrate* juga digunakan untuk mengatur *baudrate* yang kita gunakan. Sedangkan *serial monitor* digunakan untuk melihat penerimaan data serial komputer.

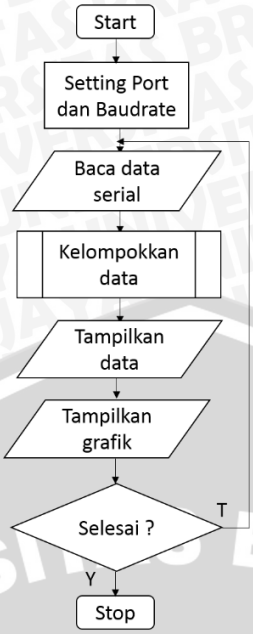
Pada *layout* tampilan tab kedua sampai ke empat memuat data pembacaan arus, tegangan, daya, energi dan kapasitas dari masing – masing *slave* baik dari sumber, baterai maupun beban. Selain itu pada *layout* tampilan tab kedua juga terdapat grafik daya dari masing – masing *slave* baik dari sumber, baterai maupun beban. Gambar layout tampilan tab kedua sampai ke empat ditunjukkan dalam Gambar 3. 11



Gambar 3. 11 *Layout* tampilan tab kedua sampai ke empat

3.2.4.3.3 Perancangan Program Utama

Program utama pada komputer dirancang agar komputer dapat membaca data yang dari *master*, menampilkan data arus, tegangan, daya, energi dan kapasitas. Selain itu juga grafik daya dari sumber, baterai dan beban dari masing – masing *slave*. Diagram alir program utama ditunjukkan dalam Gambar 3. 12



Gambar 3. 12 Diagram alir program utama komputer

Pada awalnya kita lakukan pengaturan *port* dan *baudrate* terlebih dahulu. Kemudian data dari *master* dibaca oleh komputer. Setelah itu paket data yang diterima dari *master* dikelompokkan sesuai ketentuan untuk kemudian di tampilkan baik dalam bentuk data maupun bentuk grafik.

3.2.4.3.4 Sub Program Kelompokkan Data

Sub program kelompokkan data digunakan untuk memisahkan paket data yang dikirim oleh *slave* dan mengelompokkan sesuai yang ditentukan kemudian data yang sudah dikelompokkan ditampilkan dalam bentuk data maupun grafik. Format data yang dikirim oleh *slave* ditunjukkan dalam Gambar 3. 13

Header	I_s	V_s	I_b	V_b	I_l	V_l	Enter

Gambar 3. 13 Format data

Keterangan:

I_s : data arus sumber (4 *byte*)

V_s : data tegangan sumber (4 *byte*)

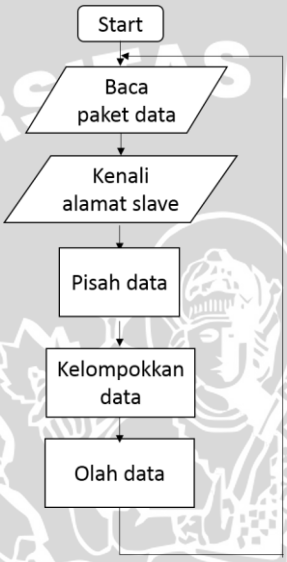
I_b : data arus baterai (4 *byte*)

V_b : data tegangan baterai (4 *byte*)

I_l : data arus beban (4 *byte*)

V_l : data tegangan beban (4 *byte*)

Format data yang dikirim master berjumlah 25 *byte* dengan rincian 1 *byte* untuk *header* dan data arus maupun tegangan sumber, baterai, dan beban masing – masing 4 *byte*. Data kemudian dipilah – pilah sesuai dengan ketentuan agar mudah ditampilkan dalam bentuk data maupun dalam bentuk grafik. Diagram alir sub program pemisah data ditunjukkan dalam Gambar 3. 14



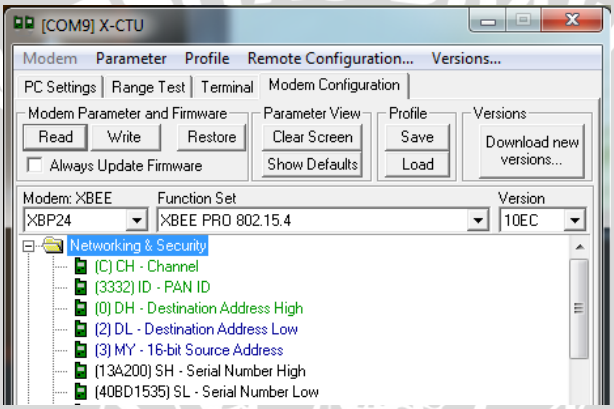
Gambar 3. 14 Diagram alir sub program pemisah data

Setelah komputer menerima data dari master kemudian paket data dikenali dari *header* paket data. Apabila *header* dari paket data yang diterima adalah "A" maka data yang diterima adalah data dari *slave* A. Begitu pula ketika data yang diterima adalah "B" maka data yang diterima adalah data dari *slave* B. Setelah dikenali *header* selanjutnya adalah mengelompokkan datanya dengan rincian data ke 2 - 5 adalah arus sumber, data ke 6 - 9 adalah tegangan sumber, data ke 10 – 13 adalah arus baterai, data ke 14 – 17 adalah tegangan baterai, data ke 18 – 21 adalah arus beban dan data ke 22 – 25 adalah tegangan beban.

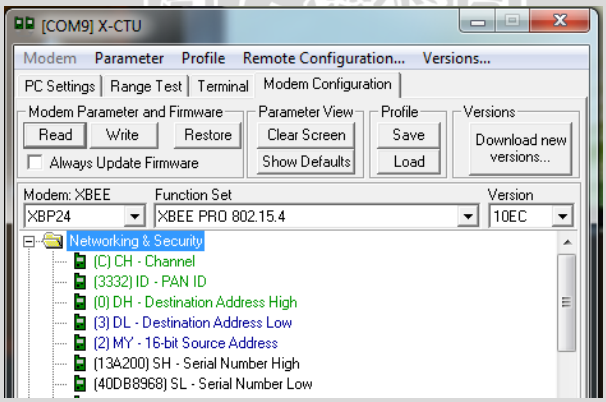
Setelah data diolah data ditampilkan pada layar *desktop*. Data ditampilkan dalam 2 bentuk yaitu dalam bentuk data dan dalam bentuk grafik. Data yang ditampilkan pada grafik adalah data daya baik daya dari sumber, baterai maupun dari beban. Selain ditampilkan pada grafik parameter daya juga ditampilkan dalam bentuk data. Selain daya parameter lain seperti arus tegangan, energi dan kapasitas baik dari sumber, baterai maupun beban daitampilkan dalam bentuk data secara *real time*.

3.2.5 Pengaturan Konfigurasi XBee

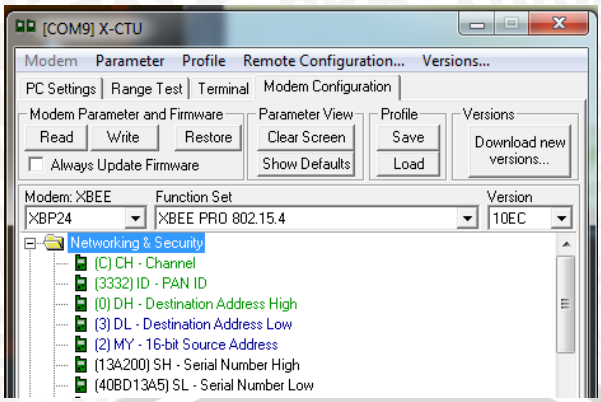
Sebelum melakukan transmisi *wireless* Modul XBee terlebih dahulu di atur XBee mana yang berperan sebagai *master* dan yang berperan sebagai *slave*. Untuk mengkonfigurasi XBee digunakan *software* X-CTU. Agar dapat menentukan XBee mana saja yang dapat berkomunikasi maka parameter XBee yang harus diatur adalah PAN ID (*Personal Area Network*). Agar dapat berkomunikasi PAN ID dalam satu jaringan harus sama. Kemudian XBee yang berperan sebagai *master* diatur DL (Destination Address Low), dan MY (16 bit Source Address) berbeda dengan XBee yang berperan sebagai *slave*. Konfigurasi XBee ditunjukkan pada Gambar 3.15, Gambar 3.16, Gambar 3.17 dan Gambar 3.18.



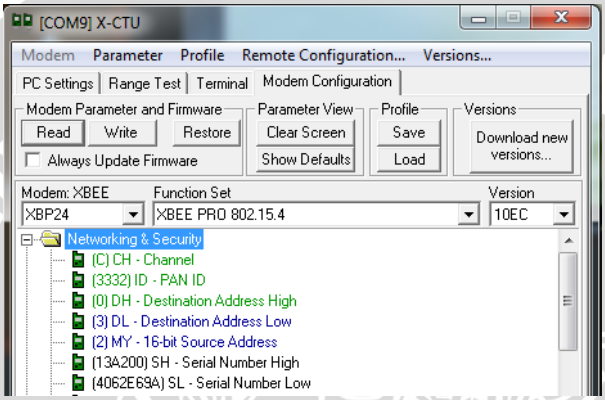
Gambar 3.15 Konfigurasi XBee *master* pada X-CTU



Gambar 3.16 Konfigurasi XBee *slave* A pada X-CTU



Gambar 3.17 Konfigurasi XBee *slave* B pada X-CTU



Gambar 3.18 Konfigurasi XBee *slave* C pada X-CTU

Pada gambar terlihat bahwa:

DL slave A = DL slave B = DL slave C

MY slave A = MY slave B = MY slave C

DL master = MY slave A/B/C

MY master = DL slave A/B/C

PAN ID master = PAN ID A/B/C

Dalam gambar dapat disimpulkan bahwa sebuah jaringan yang dibentuk oleh XBee memiliki PAN ID yang sama sedangkan DL dan MY dari *master* berlawanan dengan *slave*. XBee sendiri memiliki *serial number low* yang berbeda beda. Hal ini agar data yang dikirimkan oleh XBee tidak saling bertabrakan pada saat masing – masing slave mengirimkan data secara serempak.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengujian Jarak dan Halangan Transmisi Data

4.1.1 Tujuan

Tujuan pengujian jarak dan halangan transmisi data adalah untuk mengetahui pengaruh jarak dan halangan terhadap pengiriman data.

4.1.2 Alat yang Digunakan

Alat yang digunakan dalam pengujian jarak dan halangan transmisi data antara lain:

1. 2 buah laptop
2. 2 buah modul XBee
3. 2 buah Arduino Mega 2560
4. Kabel jumper secukupnya
5. Kotak kaca
6. Kotak kayu
7. Kertas karton
8. Meteran gulung

4.1.3 Prosedur Pengujian

Pengujian dilakukan 2 tempat yaitu *outdoor* dan *indoor*. Pengujian *outdoor* dilakukan pada tanggal 7 Januari 2016 pukul 20.00 WIB dengan kondisi setelah hujan di Jalan antara Fakultas Teknik dan Fakultas Ilmu Sosial dan Ilmu Politik Universitas Brawijaya. Sedangkan pengujian *indoor* dilakukan pada 2 Januari 2016 di Lorong Gedung C Teknik Elektro Universitas Brawijaya pukul 10.00 WIB. Pengujian dilakukan dengan mengukur jarak antara *slave* dan *master* pada saat *slave* mengirimkan data. Format data yang dikirimkan slave ditunjukkan dalam Gambar 4. 1

Header	I_s	V_s	I_b	V_b	I_l	V_l	Enter

Gambar 4. 1 Format data

Keterangan:

I_s : data arus sumber (4 *byte*)

V_s : data tegangan sumber (4 *byte*)

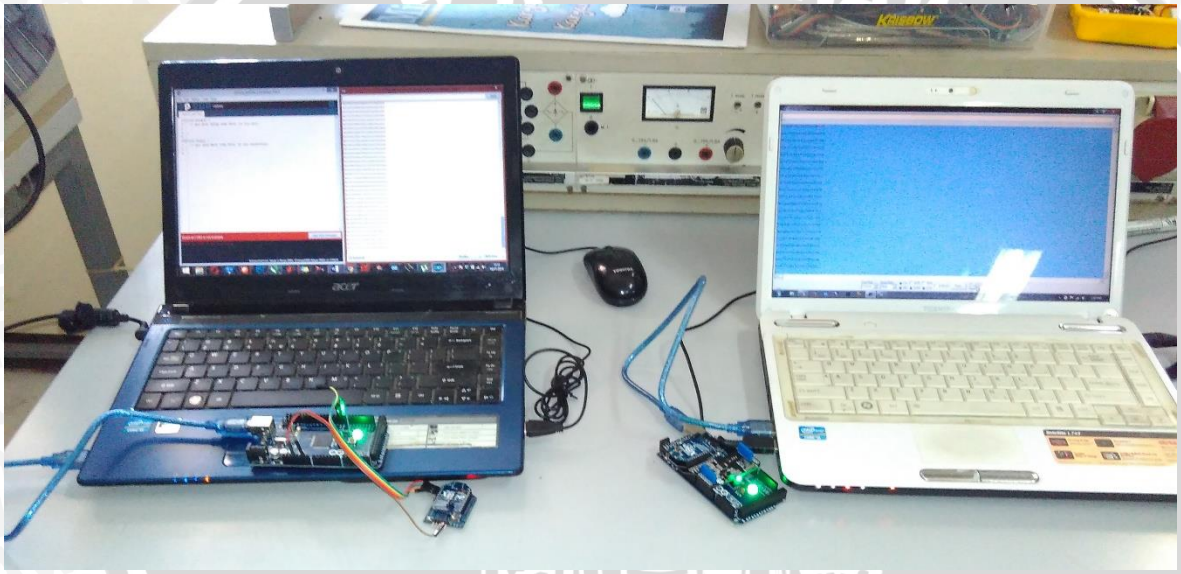
I_b : data arus baterai (4 *byte*)

V_b : data tegangan baterai (4 *byte*)

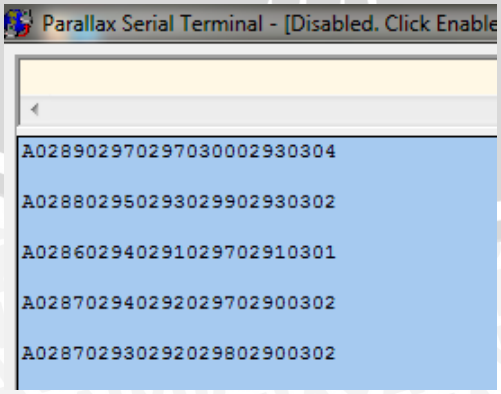
I_l : data arus beban (4 *byte*)

V_l : data tegangan beban (4 *byte*)

Selain jarak pada master juga diberikan halangan berupa kaca, kayu dan kertas karton. Proses pengujian ditunjukkan dalam Gambar 4. 2 dan Gambar 4. 3



Gambar 4. 2 *Master* dan *slave* yang digunakan pada pengujian jarak dan halangan pada transmisi data



Gambar 4. 3 Serial monitor pada *master*

Prosedur pengujian jarak dan halangan pada transmisi data *outdoor* adalah sebagai berikut:

1. Set DL dan MY dari XBee *master* dan XBee *slave*. DL dan MY dari *slave* dan *master* saling berkebalikan.
2. Rangkai rangkaian *slave*. Hubungkan pin Din XBee ke pin Tx_0 Arduino Mega 2560 dan pin Dout XBee ke pin Rx_0 Arduino Mega 2560
3. Hubungkan Arduino Mega 2560 dengan komputer melaui kabel USB
4. Buka *serial monitor* yang ada di komputer
5. Lakukan langkah 1 – 4 untuk rangkaian *master*
6. Kirim paket data dari *slave* ke master
7. Perhatikan *serial monitor* pada rangkaian *master*
8. Bandingkan data yang dikirim dan data yang diterima
9. Catat hasilnya
10. Jauhkan master dari *slave* secara bertahap
11. Kirim lagi paket data yang sama ke *master*
12. Bandingkan lagi data yang dikirim dan data yang diterima dan catat hasilnya
13. Berikan halangan pada rangkaian master dengan menutup rangkaian master dengan kaca
14. Ulangi langkah ke 6 – 12 dan catat hasilnya
15. Kemudian halangan diubah menjadi kayu dan kertas karton

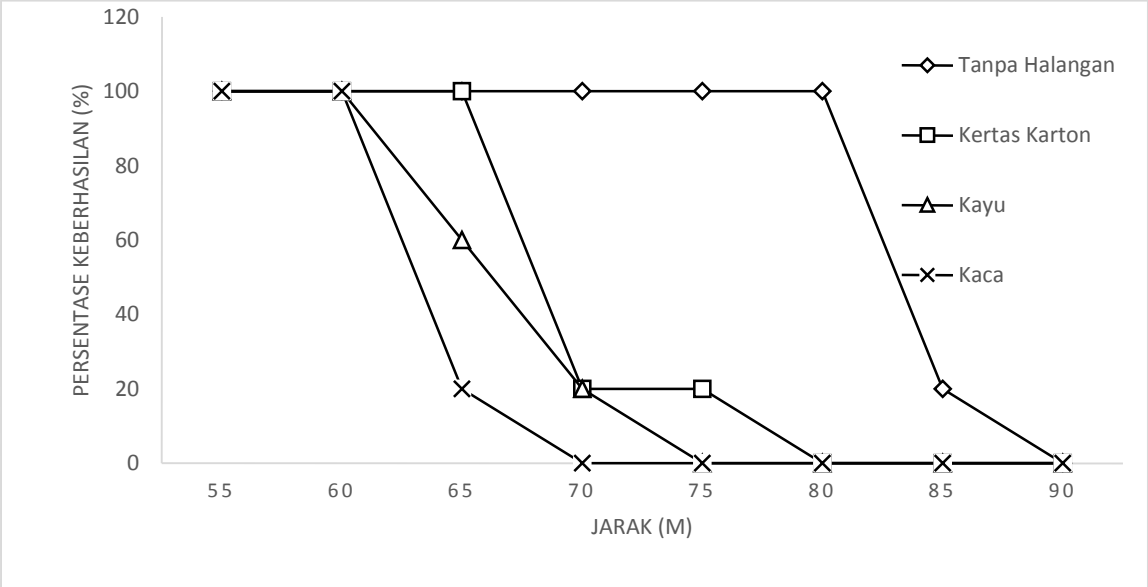
Selain pengujian pada *outdoor* pengujian juga dilakukan *indoor*. Prosedur pengujian jarak dan halangan pada transmisi data *indoor* hampir sama dengan *outdoor* hanya berbeda tempat. Selain itu halangan yang diberikan berupa tembok.

4.1.4 Hasil Pengujian dan Analisis

Pengujian jarak dan halangan pada transmisi data dilakukan pada range jarak 55 m – 90 m. Pengujian ini mengabaikan *noise* lain seperti cuaca, kelembapan, suhu dan *noise* lain yang dapat mengganggu proses transmisi data. Hasil pengujian jarak dan halangan pada transmisi data *outdoor* ditunjukkan dalam Tabel 4. 1 dan Gambar 4. 4

Tabel 4. 1 Hasil pengujian dan halangan pada transmisi data *outdoor*

	Jumlah Data Benar yang Diterima (*byte*)				Persentase Keberhasilan (%)			
Jarak (m)	Tanpa Halangan	Kertas Karton	Kayu	Kaca	Tanpa Halangan	Kertas Karton	Kayu	Kaca
55	130	130	130	130	100	100	100	100
60	130	130	130	130	100	100	100	100
65	130	130	78	26	100	100	60	20
70	130	26	26	0	100	20	20	0
75	130	26	0	0	100	20	0	0
80	130	0	0	0	100	0	0	0
85	26	0	0	0	20	0	0	0
90	0	0	0	0	0	0	0	0

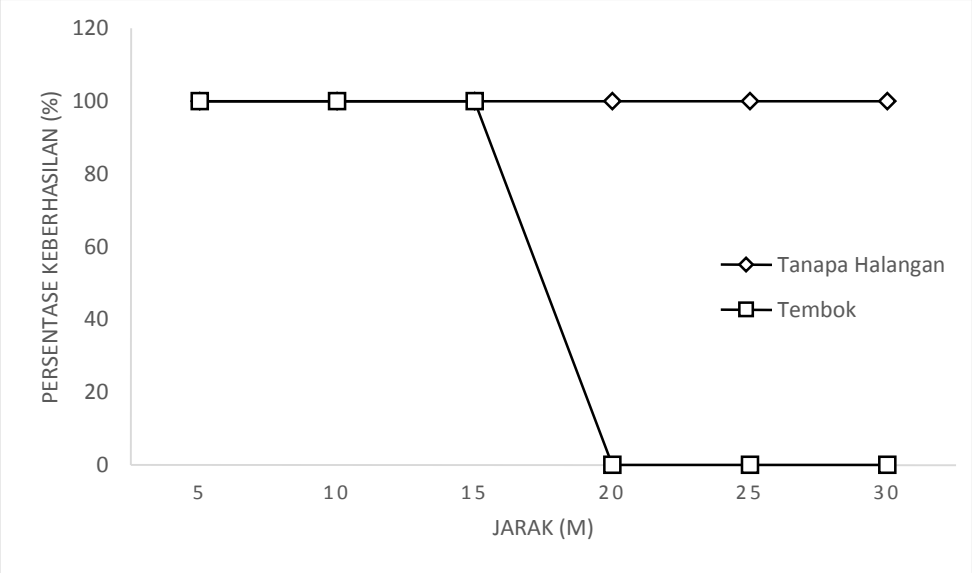


Gambar 4. 4 Grafik pengujian jarak dan halangan pada transmisi data *outdoor*

Untuk pengujian transmisi data *indoor* range jarak yang digunakan adalah 0 – 30 m. Untuk halangan yang diberikan pada pengujian jarak dan halangan transmisi data *indoor* adalah tembok. Sementara *noise* lain seperti suhu dan kelembapan dapat diabaikan. Data hasil pengujian jarak dan transmisi data *indoor* ditunjukkan dalam Tabel 4. 2 dan Gambar 4. 5

Tabel 4. 2 Hasil pengujian jarak dan halangan pada transmisi data *indoor*

	Jumlah Data Benar yang Diterima (*byte*)		Persentase Keberhasilan (%)	
Jarak (m)	Tanpa Halangan	Tembok	Tanpa Halangan	Tembok
5	130	130	100	100
10	130	130	100	100
15	130	130	100	100
20	130	0	100	0
25	130	0	100	0
30	130	0	100	0



Gambar 4. 5 Grafik pengujian jarak dan transmisi data *indoor*

Grafik diatas menunjukkan hubungan jarak dengan persentase keberhasilan pengiriman. Untuk mencari persentase keberhasilan pengiriman data digunakan persamaan (5 -1)

$$\%\ Keberhasilan = \frac{N}{N\ max}\ x\ 100\% \qquad (5-1)$$

Dimana N adalah jumlah data diterima yang sesuai dengan data yang dikirim, dan N_{max} adalah jumlah data yang dikirimkan.

Pada pengujian jarak dan halangan pada transmisi data *outdoor* dapat diketahui jarak trasmisi data yang persentase keberhasilan pengiriman data 100% saat tanpa halangan mencapai 80 m. Untuk halangan yang diberikan halangan yang paling berpengaruh adalah kaca. Pada grafik terlihat persentase keberhasilan pengirman data 100% berada sampai jarak

60 m. Hal ini dikarenakan kaca memiliki daya pantul yang baik sehingga transmisi data terganggu.

Kemudian pada pengujian jarak dan halangan pada transmisi data *indoor* proses pengiriman data berjalan relatif lancar. Hal ini dikarenakan jarak transmisi pada pengujian jarak dan halangan pada transmisi data *indoor* lebih dekat dibandingkan dengan pengujian saat *outdoor*. Halangan berupa tembok sangat berpengaruh pada transmisi data *indoor*. Dapat dilihat pada grafik persentase keberhasilan 100% hanya sampai pada jarak 15 m. Artinya halangan berupa tembok sangat mempengaruhi proses transmisi data.

4.2 Pengujian Format Data pada Transmisi Data

4.2.1 Tujuan

Tujuan pengujian format data pada transmisi data adalah untuk mengetahui pengaruh format data pada pengiriman data.

4.2.2 Alat yang Digunakan

Alat yang digunakan dalam pengujian format data pada transmisi data adalah sebagai berikut:

1. 2 buah laptop
2. 2 buah modul XBee
3. 2 buah Arduino Mega 2560
4. Kabel jumper secukupnya
5. Kotak kaca
6. Kotak kayu
7. Kertas karton
8. Meteran gulung

4.2.3 Prosedur Pengujian

Pengujian dilakukan pada tanggal 7 Januari 2016 di Jalan antara Fakultas Teknik dan Fakultas Ilmu Sosial dan Ilmu Politik Universitas Brawijaya pukul 20.00 WIB dengan kondisi setelah hujan. Pengujian dilakukan dengan membuat jarak anatara *slave* dan *master* tetap yaitu 75 m, sedangkan format data yang dikirim diubah ubah.

Prosedur pengujian rangkaian format data pada transmisi data adalah sebagai berikut:

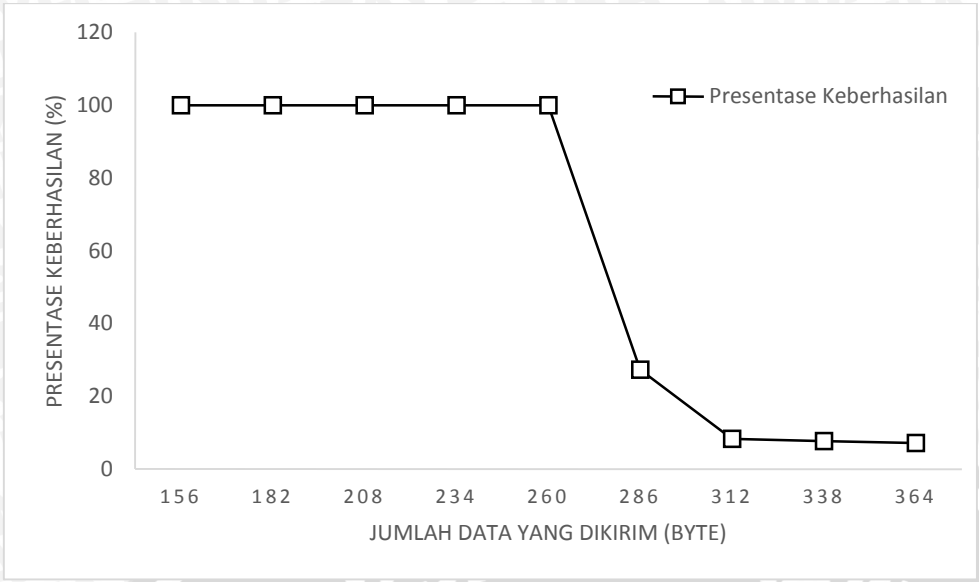
1. Set DL dan MY dari XBee *master* dan XBee *slave*. DL dan MY dari *slave* dan *master* saling berkebalikan.
2. Rangkai rangkaian *slave.* Hubungkan pin Din XBee ke pin Tx_0 Arduino Mega 2560 dan pin Dout XBee ke pin Rx_0 Arduino Mega 2560
3. Hubungkan Arduino Mega 2560 dengan komputer melaui kabel USB
4. Buka *serial monitor* yang ada di komputer
5. Lakukan langkah 1 – 4 untuk rangkaian *master*
6. Atur jarak transmisi data sejauh 75 m
7. Kirim data sejumlah 156 *byte* dari *slave* ke master
8. Perhatikan *serial monitor* pada rangkaian *master*
9. Bandingkan data yang dikirim dan data yang diterima
10. Catat hasilnya
11. Ubah format data yang dikirimkan secara berkala dan catat hasilnya

4.2.4 Hasil Pengujian dan Analisis

Pengujian format data pada transmisi data dilakukan pada jarak 75 m. Format data yang dikirim adalah karakter abjad A sampai Z diulang sebanyak 6 kali dan ditambah secara berkala. Pengujian ini mengabaikan *noise* lain seperti cuaca, kelembapan, suhu dan *noise* lain yang dapat mengganggu proses transmisi data. Hasil pengujian format data pada transmisi data ditunjukkan dalam Tabel 4. 3 dan Gambar 4. 6

Tabel 4. 3 Hasil pengujian format data pada transmisi data

Data Yang Dikirim (*byte*)	Data Benar yang Diterima (*byte*)	Persentase Keberhasilan (%)
156	156	100
182	182	100
208	208	100
234	234	100
260	260	100
286	78	27.27272727
312	26	8.333333333
338	26	7.692307692
364	26	7.142857143



Gambar 4. 6 Grafik pengujian format data pada transmisi data

Pada pengujian format data pada transmisi data dapat diketahui jumlah data yang dikirimkan persentase keberhasilannya 100% mencapai 260 *byte*. Kemudian dari grafik diatas dapat disimpulkan bahwa semakin banyak jumlah data yang dikirim maka persentase keberhasilan semakin menurun dan pada jumlah data tertentu akan konstan.

4.3 Pengujian Tegangan pada Proses Transmisi Data

4.3.1 Tujuan

Tujuan pengujian tegangan pada proses transmisi data adalah untuk mengetahui tegangan rangkaian pemancar dan penerima pada saat mengirim atau menerima data

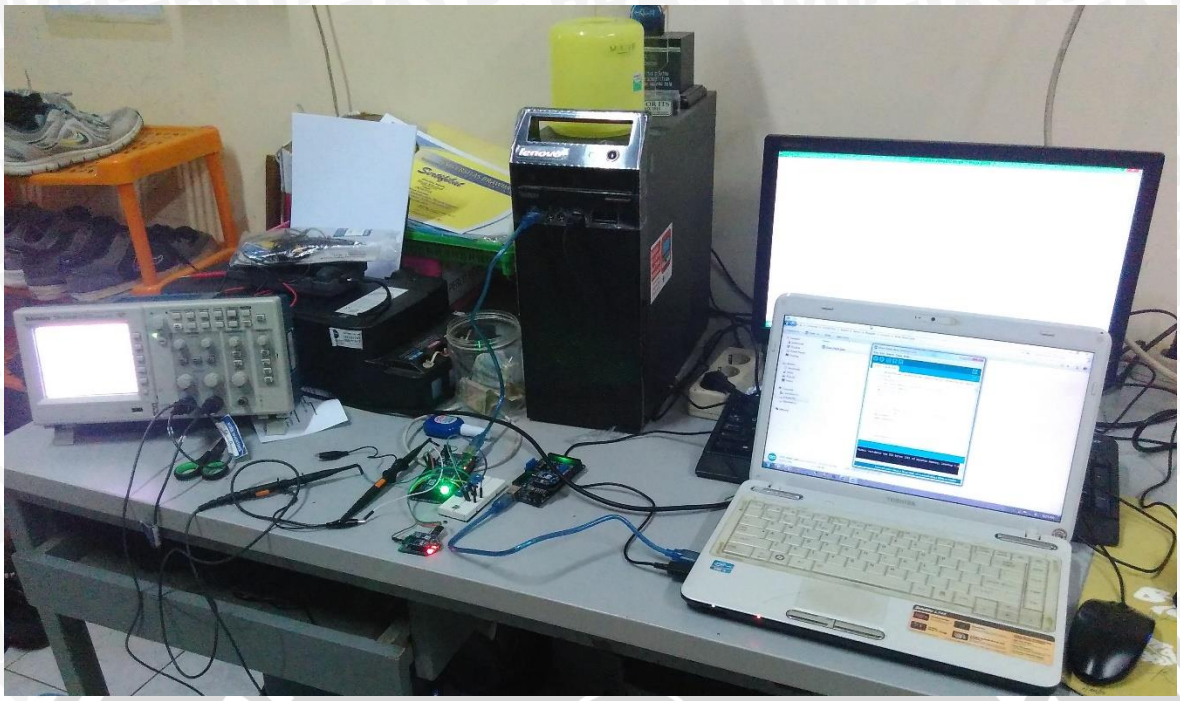
4.3.2 Alat yang Digunakan

Alat yang digunakan pada pengujian tegangan pada proses transmisi data antara lain:

1. 2 buah laptop
2. 2 buah modul XBee
3. 2 buah Arduino Mega 2560
4. Kabel jumper secukupnya
5. 1 buah *osciloscop*

4.3.3 Prosedur Pengujian

Prose pengujian tegangan pada proses transmisi data ditunjukkan dalam Gambar 4. 7



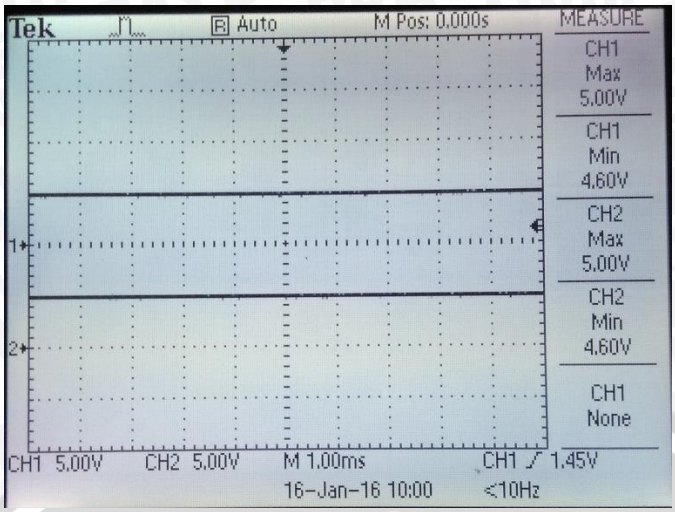
Gambar 4. 7 Proses pengujian tegangan pada proses transmisi data

Prosedur pengujian tegangan pada proses transmisi data adalah sebagai berikut:

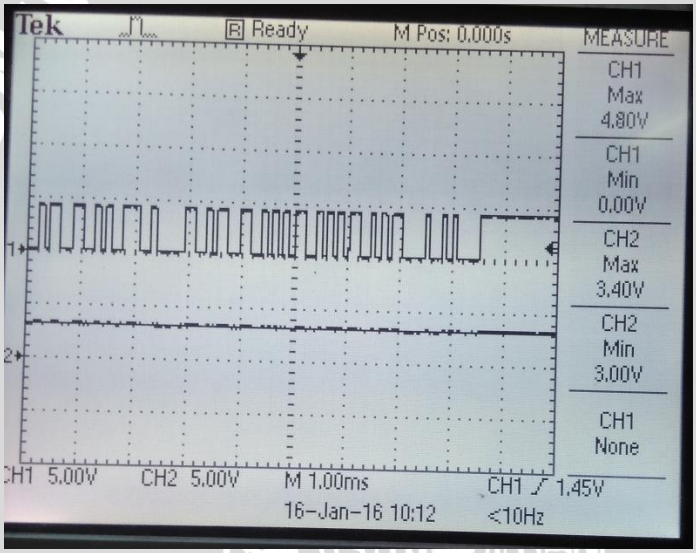
1. Set DL dan MY dari XBee *master* dan XBee *slave*. DL dan MY dari *slave* dan *master* saling berkebalikan.
2. Rangkai rangkaian *slave*. Hubungkan pin Din XBee ke pin Tx_0 Arduino Mega 2560 dan pin Dout XBee ke pin Rx_0 Arduino Mega 2560
3. Lakukan langkah 1 dan 2 untuk rangkaian *master*
4. Hubungkan Tx_0 Arduino Mega 2560 ke channel 1 osciloscope dan Rx_0 ke channel 2 osciloscope
5. Catat dan amati di oscilloscope
6. Kirim paket data ke *master*
7. Catat dan amati di osciloscope

4.3.4 Hasil Pengujian dan Analisis

Pada pengujian tegangan pada proses transmisi data dibandingkan tegangan dari *slave* saat *slave* mengirimkan data dan tidak. Channel 1 pada osciloscope adalah pin Tx_0 Arduino Mega 2560 dan channel 2 osciloscope adalah pin Rx_0 Arduino Mega 2560. Hasil pengujian tegangan pada proses transmisi data ditunjukkan dalam Gambar 4. 8 dan Gambar 4. 9



Gambar 4. 8 Grafik ketika slave tidak menigrimkan data



Gambar 4. 9 Grafik ketika slsve mengirimkan data

Pada gambar 4.8 dapat dilihat ketika *slave* mengirimkan data tegangan dari Tx_0 berkisar antara 4.6 – 5 Volt. Begitu juga dengan pin Rx_0 tegangan saat tidak menerima data berkisar antara 4.6 V – 5 Volt. Berarti dapat dikatakan ketika slave tidak menerima maupun mengirim data maka pin Tx_0 maupun Rx_0 berlogika tinggi.

Adapun pada grafik 4.9 adalah ketika slave mengirimkan data. Pada pin Tx_0 data yang didapat berkisar 0 Volt – 4.8 Volt. Grafik yang terlihat pada gambar 4.9 adalah paket data yang dikirim oleh *slave*. Untuk channel 2 tegangan berkisar antara 3 – 3.4 volt

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian tiap bagian dan keseluruhan sistem yang telah dilaksanakan didapatkan kesimpulan sebagai berikut:

1. Perancangan sistem transmisi data berbasis *wireless* pada *micro smart grid* terdiri dari 3 *slave*. Data dari pembacaan sensor kemudian ditransmisikan secara *wireless* melalui modul XBee. Setelah itu ditangkap dan diteruskan oleh master melalui USB ke komputer untuk diolah dan ditampilkan pada grafik secara *real time.* Adapun metode komunikasi pada penelitian ini adalah metode komunikasi *simplex* dimana data hanya dikirim dari *slave* ke *master* dan tidak sebaliknya. Data yang ditrasmisikan sebanyak 25 *byte* dengan rincian I *byte header*, 4 *byte* data arus sumber, 4 *byte* data tegangan sumber, 4 *byte* data arus baterai, 4 *byte* data tegangan baterai, 4 *byte* data arus beban, 4 *byte* data tegangan beban
2. Semakin jauh jarak transmisi data maka data yang diterima semakin berkurang. Pada transmisi data *outdoor* jarak transmisi yang persentase keberhasilan 100% mencapai jarak 80 m. Halangan yang paling mempengaruhi adalah kaca. Saat transmisi data *indoor* halangan berupa tembok sangat berpengaruh, hal ini dibuktikan persentase keberhasilan 100% hanya sampai jarak 20 m. Lebar data maksimum yang dapat ditransmisikan dengan persentase keberhasilan 100% adalah 260 *byte.*
3. Tegangan pin Tx_0 *slave* pada proses pengiriman data berkisar 0 – 4.8 volt bergantung data yang dikirimkan. Sedangkan ketika *slave* tidak mengirimkan data tegangannya 5 Volt

5.2 Saran

Saran-saran dalam pengimplementasian maupun peningkatan unjuk kerja sistem ini dapat diuraikan sebagai berikut:

1. Sistem pengiriman *wireless* yang digunakan dapat menggunakan media lain seperti cahaya dan bunyi.
2. *Slave* yang digunakan diperbanyak untuk sistem yang lebih besar

DAFTAR PUSTAKA

ABB. 2009. "When Grids Get Smart". Germany: Power Technologies

Atmel.2014.Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V.USA: Atmel.

Digi International Inc.2011. XBee® & XBee-PRO® ZB.USA: Digi International Inc

Kementrian ESDM.2013. Pemakaian Listrik Semester 1/2013, Tumbuh Tinggi Di Sektor Produktif Dan Rendah Di Kelompok Konsumtif. (online), http: //www.esdm.go.id/berita/listrik/39-listrik/6373-pemakaian-listrik%09semester-12013-tumbuh-tinggi-di-sektor-produktif-dan-rendah-di%09kelompok-konsumtif.html. Diakses tanggal 2 Juni 2015

Momoh, James A. 2012. SMART GRID: Fundamentals of Design and Analysis. Institute of Electrical and Electronics Engineering (IEEE) Press. United States of America: John Wiley & Sons, Inc.

Nejad, Mohsen Fadee., Amin Mohammad Saberian, dkk. 2013. Application of Smart Power Grid in Developing Countries. Power Engineering and Optimization Conference (PEOCO): 427-431. Langkawi: IEEE

Satya, Bayu N T S.2015.*Rancang Bangun Monitoring dan Switching Untuk Mengetahui Arus dan Tegangan pada Microgrid.*Malang: Perpustakaan Ruang Baca Teknik Elektro Universitas Brawijaya

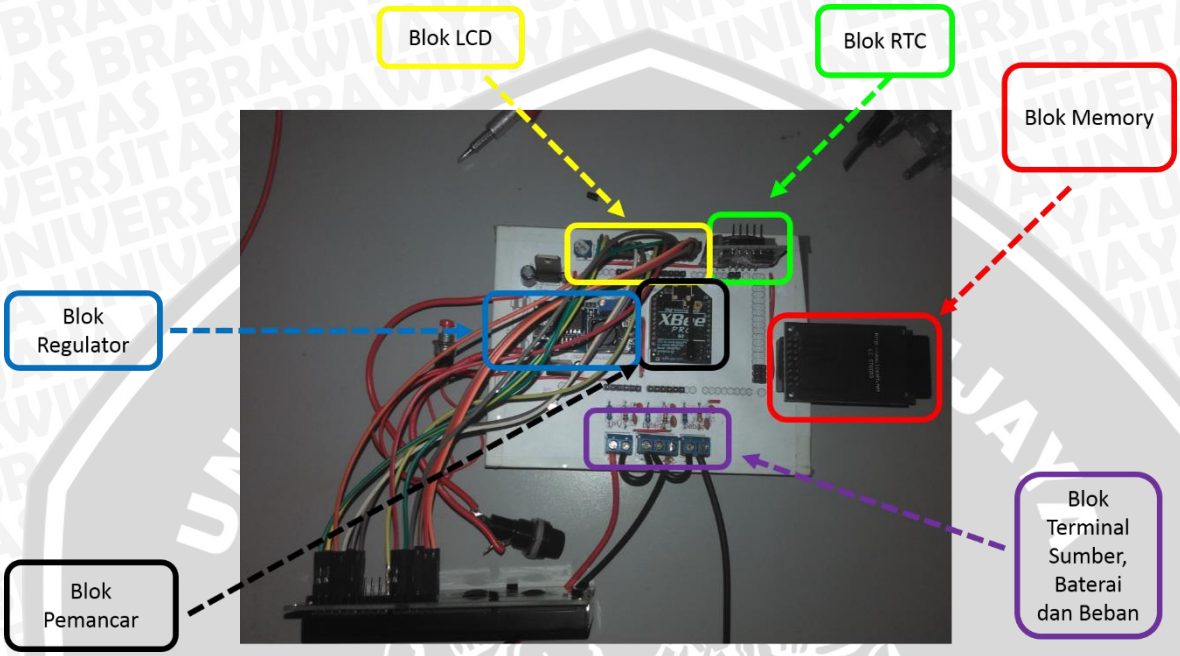
Tang, Grace Q. 2011. Smart Grid Management & Visualization: Smart Power Management System. Emerging Technologies for a Smarter World (CEWIT): 1-6. New York: IEEE.

Widaringtyas, Dian S.2014. Inverter15V DC-220V AC Berbasis Tenaga Surya Untuk Aplikasi Single Point Smart Grid. Malang: Perpustakaan Ruang Baca Teknik Elektro Universitas Brawijaya.

Yuliza. 2014. Komunikasi Antar Robot Menggunakan RFXbee dan Arduino Microcontroller.

LAMPIRAN

Lampiran 1. Foto Alat



Gambar 1 Gambar setiap slave tampak dalam

Lampiran 2. Dokumentasi Kegiatan

2.1 Solar Cell yang dipasang di Gedung A Teknik Elektro Universitas Brawijaya

Gambar 2 Solar Cell yang dipasang di Gedung A Teknik Elektro Universitas Brawijaya

2.2 Unit pengukuran yang terpasang di Laboratorium Elektronika Proses Teknik Elektro Universitas Brawijaya

Gambar 3 Unit pengukuran yang terpasang di Laboratorium Elektronika Proses Teknik Elektro Universitas Brawijaya

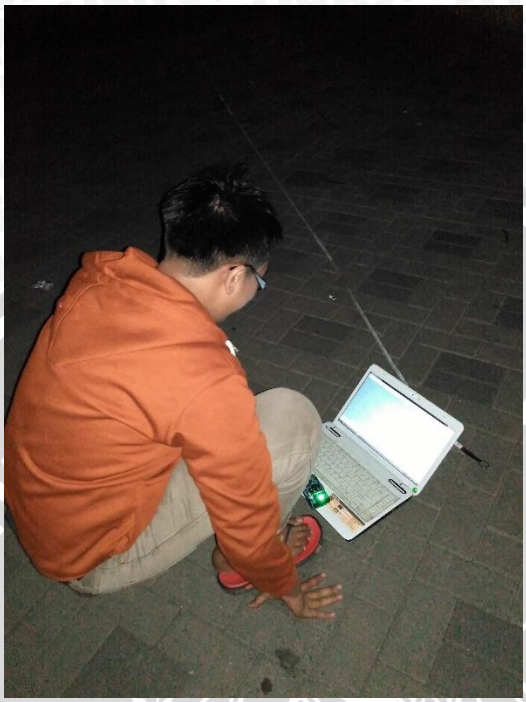
2.3 Setting XBee

Gambar 4 Setting XBee

2.4 Pengujian jarak, halangan dan format data pada transmisi data

Gambar 5 Pengujian jarak, halangan dan format data pada transmisi data

2.5 Proses pengujian transmisi data *outdoor*



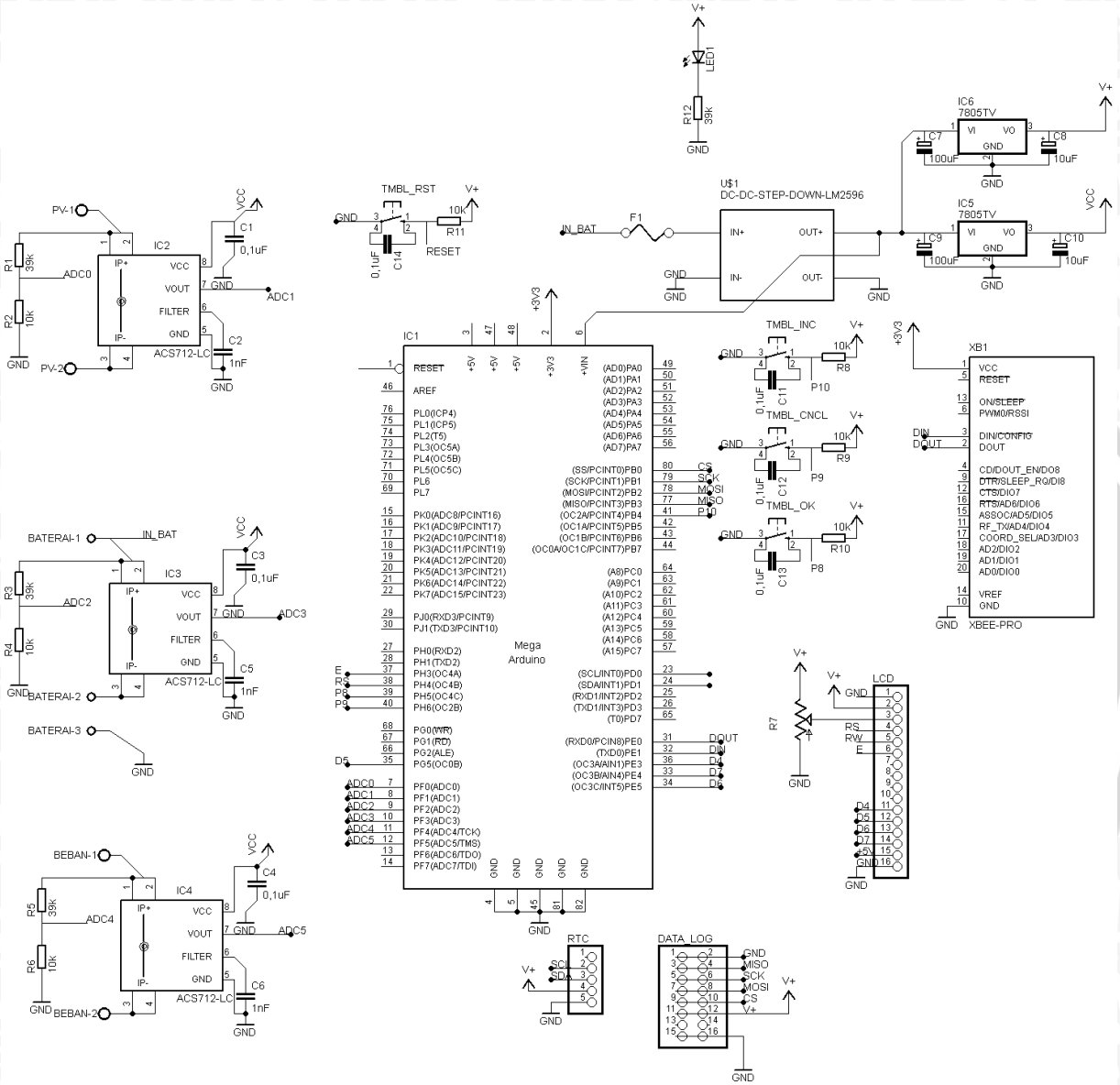
Gambar 6 Proses pengujian transmisi data *outdoor*

2.6 Pengujian tegangan pada proses transmisi data

Gambar 7 Pengujian tegangan pada proses transmisi data

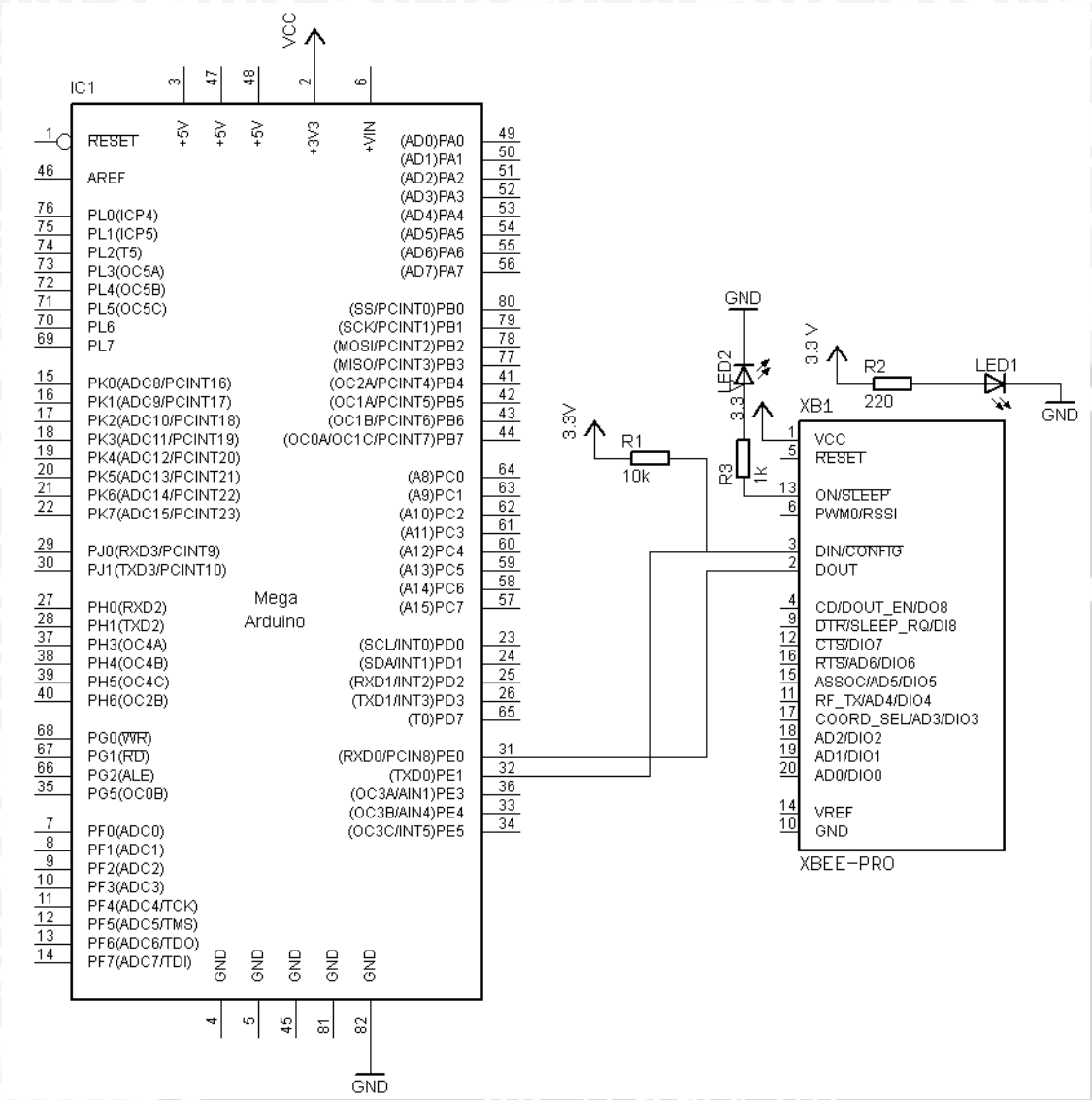
Lampiran 3. Skematik Rangkaian

3.1 Skematik rangkaian *slave*



Gambar 8 Skematik rangkaian *slave*

3.2 Skematik rangkaian *master*



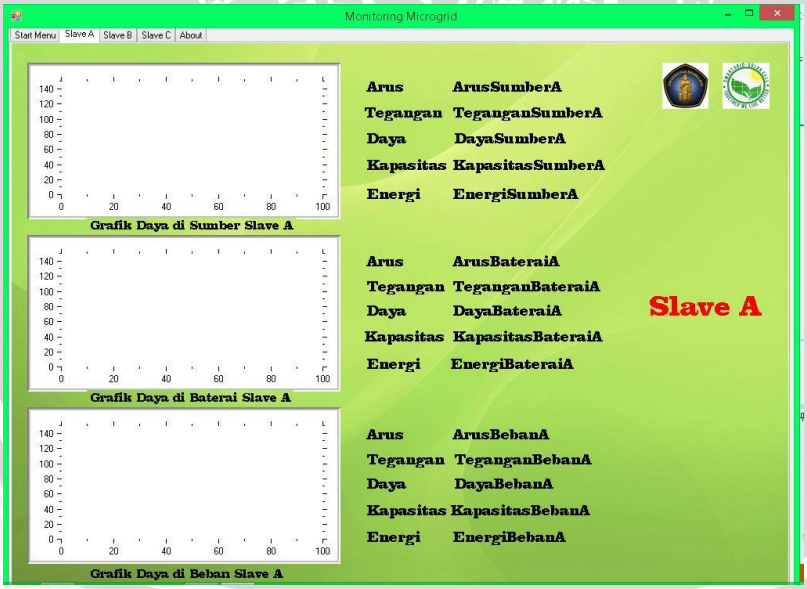
Gambar 9 Skematik rangkaian *master*

Lampiran 4. Tampilan Program Dekstop

4.1 Tampilan tab pertama program desktop sebelum aktif

Gambar 10 Tampilan tab pertama program desktop sebelum aktif

4.2 Tampilan tab ke 2 program desktop sebelum aktif

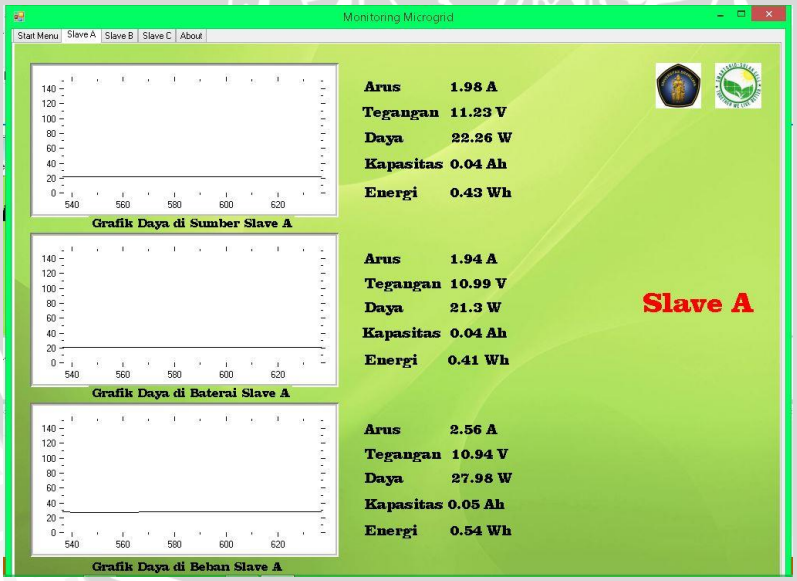


Gambar 11 Tampilan tab ke 2 program desktop sebelum aktif

4.3 Tampilan tab pertama program desktop setelah aktif

Gambar 12 Tampilan tab pertama program desktop setelah aktif

4.4 Tampilan tab ke 2 program desktop setelah aktif



Gambar 13 Tampilan tab ke 2 program desktop setelah aktif

Overview

XBee Product Family

The XBee family of embedded RF modules provides OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/Mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the XBee can substitute one XBee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-tomarket.

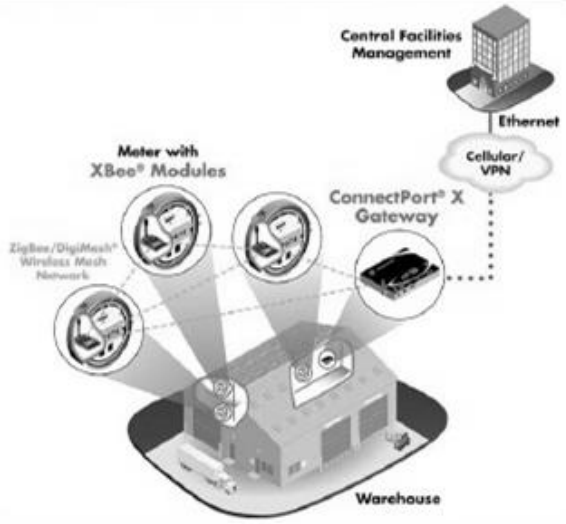
Why XBee Multipoint RF Modules?

XBee multipoint RF modules are ideal for applications requiring low latency and predictable communication timing. Providing quick, robust communication in point-to-point, peer-to-peer, and multipoint/star configurations, XBee multipoint products enable robust end-point connectivity with ease. Whether deployed as a pure cable replacement for simple serial communication, or as part of a more complex hub-and-spoke network of sensors, XBee multipoint RF modules maximize wireless performance and ease of development.

Drop-in Networking End-Point Connectivity

XBee OEM RF modules are part of Digi's Drop-in Networking family of end-to-end connectivity solutions. By seamlessly interfacing with compatible gateways, device adapters and extenders, XBee embedded RF modules provide developers with true beyond-the-horizon connectivity.

Application Highlight



Features/Benefits

- 802.15.4/Multipoint network topologies
- 2.4 GHz for worldwide deployment
- 900 MHz for long-range deployment
- Fully interoperable with other other Digi networking products, including gateways, device adapters and extenders
- Common XBee footprint for a variety of RF modules
- Low-power sleep modes
- Multiple antenna options
- Industrial temperature rating (-40° C to 85° C)
- Low-power and long-range variants available

Lampiran 5. Datasheet

6.1 XBee Modul

Platform	XBee® 802.15.4 (Series 1)	XBee-PRO® 802.15.4 (Series 1)	XBee-PRO® XSC
Performance			
RF Data Rate	250 kbps	250 kbps	10 kbps / 9.6 kbps
Indor/Urban Range	100 ft (30 m)	300 ft (100 m)	Up to 1200 ft (370 m)
Outdoor/RF Line-of-Sight Range	300 ft (100 m)	1 mi (1.6 km)	Up to 6 mi (9.6 km)
Transmit Power	1 mW (+0 dBm)	60 mW (+18 dBm)*	100 mW (+20 dBm)
Receiver Sensitivity (1% PER)	-92 dBm	-100 dBm	-106 dBm
Features			
Serial Data Interface	3.3V CMOS UART	3.3V CMOS UART	3.3V CMOS UART (5V Tolerant)
Configuration Method	API or AT Commands, local or over-the-air	API or AT Commands, local or over-the-air	AT Commands
Frequency Band	2.4 GHz	2.4 GHz	902 MHz to 928 MHz
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)	DSSS (Direct Sequence Spread Spectrum)	FHSS (Frequency Hopping Spread Spectrum)
Serial Data Rate	1200 bps - 250 kbps	1200 bps - 250 kbps	1200 bps - 57.6 kbps
ADC Inputs	(6) 10-bit ADC inputs	(6) 10-bit ADC inputs	None
Digital I/O	8	8	None
Antenna Options	Chip, Wire Whip, U.FL, & RPSMA	Chip, Wire Whip, U.FL, & RPSMA	Wire Whip, U.FL, RPSMA
Networking & Security			
Encryption	128-bit AES	128-bit AES	No
Reliable Packet Delivery	Retries/Acknowledgments	Retries/Acknowledgments	Retries/Acknowledgements
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 Channels	PAN ID, 64-bit IEEE MAC, 12 Channels	PAN ID, 32-bit Address, 7 Channels
Power Requirements			
Supply Voltage	2.8 - 3.4VDC	2.8 - 3.4VDC	3.0 - 3.6VDC
Transmit Current	45 mA @ 3.3VDC	215 mA @ 3.3VDC	265 mA typical
Receive Current	50 mA @ 3.3VDC	55 mA @ 3.3VDC	65 mA typical
Power-Down Current	<10 uA @ 25° C	<10 uA @ 25° C	45 uA pin Sleep
Regulatory Approvals			
FCC (USA)	OUR-XBEE	OUR-XBEEPRO	MCQ-XBEEXSC
IC (Canada)	4214A-XBEE	4214A-XBEEPRO	1846A-XBEEXSC
ETSI (Europe)	Yes	Yes* Max TX 10 mW	No
C-TICK Australia	Yes	Yes	No
Telec (Japan)	Yes	Yes*	No

6.2 Arduno Mega 2560

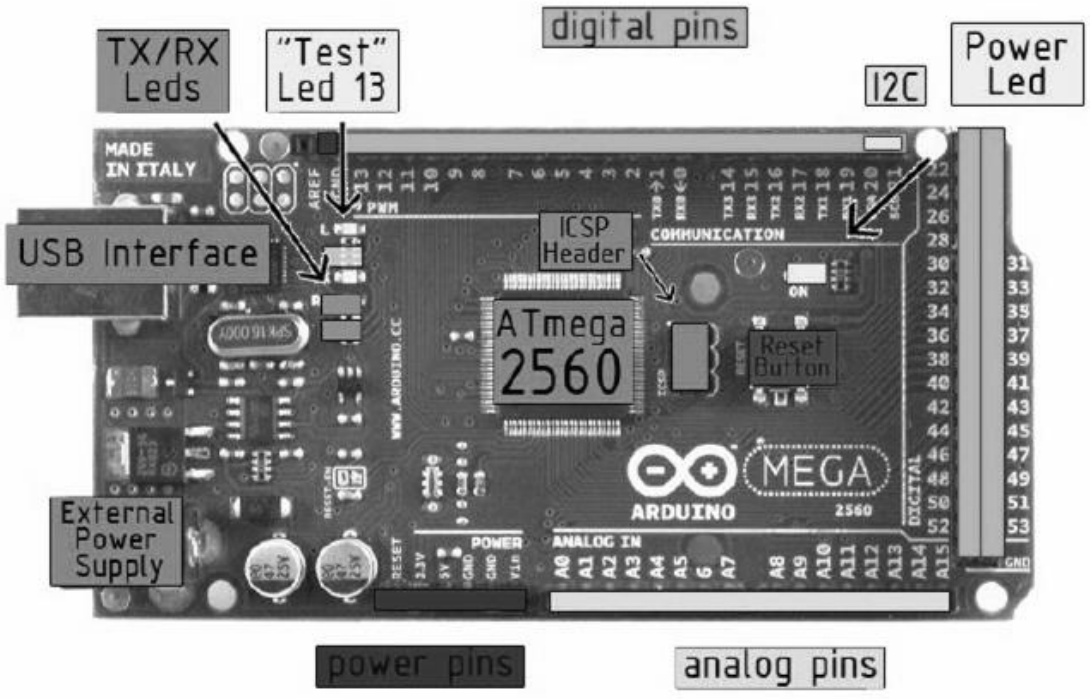
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Lampiran 6. Listing Program

6.1 Listing Program Slave A

```
int
sensorValue0,sensorValue1,sensorValue
2,sensorValue3,sensorValue4,sensorValu
e5;
String A;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  A = String ("A");
}
void loop() {
  // put your main code here, to run
repeatedly:
  Serial.print(A);
  bacaSensor();
  tampilanData();
  Serial.println();
  delay(1000);

}
void bacaSensor(void)
{
  sensorValue0 =analogRead(A0);
  sensorValue1 =analogRead(A1);
  sensorValue2 =analogRead(A2);
  sensorValue3 =analogRead(A3);
  sensorValue4 =analogRead(A4);
  sensorValue5 =analogRead(A5);
  }
  void tampilanData(void)
  {
    if(sensorValue0 >= 0 &&
sensorValue0 <= 9)
    {
      Serial.print("000");
      Serial.print(sensorValue0);
      }
    else if(sensorValue0 >= 10 &&
sensorValue0 <= 99)
    {
      Serial.print("00");
      Serial.print(sensorValue0);
      }
    else if(sensorValue0 >= 100 &&
sensorValue0 <= 999)
    {
      Serial.print("0");
      Serial.print(sensorValue0);
      }
    else
    {
      Serial.print(sensorValue0);
      }
     if(sensorValue1 >= 0 &&
sensorValue1 <= 9)
    {
      Serial.print("000");
      Serial.print(sensorValue1);
      }
     else if(sensorValue1 >= 10 &&
sensorValue1 <= 99)
    {
      Serial.print("00");
      Serial.print(sensorValue1);
      }
     else if(sensorValue1 >= 100 &&
sensorValue1 <= 999)
    {
      Serial.print("0");
      Serial.print(sensorValue1);
      }
    else
    {
      Serial.print(sensorValue1);
      }
     if(sensorValue2 >= 0 &&
sensorValue2 <= 9)
    {
      Serial.print("000");
      Serial.print(sensorValue2);
      }
     else if(sensorValue2 >= 10 &&
sensorValue2 <= 99)
    {
      Serial.print("00");
      Serial.print(sensorValue2);
      }
    else if(sensorValue2 >= 100 &&
sensorValue2 <= 999)
```

```
  {
   Serial.print("0");
   Serial.print(sensorValue2);
   }else
  {
   Serial.print(sensorValue2);
   }
    if(sensorValue3 >= 0 &&
sensorValue3 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue3);
   }
  else if(sensorValue3 >= 10 &&
sensorValue3 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue3);
   }
  else if(sensorValue3 >= 100 &&
sensorValue3 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue3);
   }
   else
  {
   Serial.print(sensorValue3);
   }
    if(sensorValue4 >= 0 &&
sensorValue4 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue4);
   }
  else if(sensorValue4 >= 10 &&
sensorValue4 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue4);
   }
  else if(sensorValue4 >= 100 &&
sensorValue4 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue4);
   }
   else
  {
   Serial.print(sensorValue4);
   }
    if(sensorValue5 >= 0 &&
sensorValue5 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue5);
   }
  else if(sensorValue5 >= 10 &&
sensorValue5 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue5);
   }
  else if(sensorValue5 >= 100 &&
sensorValue5 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue5);
   }
   else
  {
   Serial.print(sensorValue5);
   }
  }
```

6.2 Listing Program Slave B

```
Int
sensorValue0,sensorValue1,sensorValue
2,sensorValue3,sensorValue4,sensorValu
e5;
String B;
void setup() {
 // put your setup code here, to run once:
 Serial.begin(9600);
 B = String ("B");
}
void loop() {
 // put your main code here, to run
repeatedly:
 Serial.print(B);
 bacaSensor();
 tampilanData();
 Serial.println();
 delay(1000);

}
void bacaSensor(void)
{
 sensorValue0 =analogRead(A0);
 sensorValue1 =analogRead(A1);
 sensorValue2 =analogRead(A2);
```

```
 sensorValue3 =analogRead(A3);
 sensorValue4 =analogRead(A4);
 sensorValue5 =analogRead(A5);
 }
 void tampilanData(void)
 {
  if(sensorValue0 >= 0 &&
sensorValue0 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue0);
   }
  else if(sensorValue0 >= 10 &&
sensorValue0 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue0);
   }
  else if(sensorValue0 >= 100 &&
sensorValue0 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue0);
   }
  else
  {
   Serial.print(sensorValue0);
   }
   if(sensorValue1 >= 0 &&
sensorValue1 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue1);
   }
  else if(sensorValue1 >= 10 &&
sensorValue1 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue1);
   }
  else if(sensorValue1 >= 100 &&
sensorValue1 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue1);
   }
  else
  {
   Serial.print(sensorValue1);
   }
   if(sensorValue2 >= 0 &&
sensorValue2 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue2);
   }
  else if(sensorValue2 >= 10 &&
sensorValue2 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue2);
   }
  else if(sensorValue2 >= 100 &&
sensorValue2 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue2);
   }
  else
  {
   Serial.print(sensorValue2);
   }
   if(sensorValue3 >= 0 &&
sensorValue3 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue3);
   }
  else if(sensorValue3 >= 10 &&
sensorValue3 <= 99)
  {
   Serial.print("00");
   Serial.print(sensorValue3);
   }
  else if(sensorValue3 >= 100 &&
sensorValue3 <= 999)
  {
   Serial.print("0");
   Serial.print(sensorValue3);
   }
  else
  {
   Serial.print(sensorValue3);
   }
   if(sensorValue4 >= 0 &&
sensorValue4 <= 9)
  {
   Serial.print("000");
   Serial.print(sensorValue4);
   }
```

```
   else if(sensorValue4 >= 10 &&
sensorValue4 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue4);
    }
   else if(sensorValue4 >= 100 &&
sensorValue4 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue4);
    }
    else
   {
    Serial.print(sensorValue4);
    }
     if(sensorValue5 >= 0 &&
sensorValue5 <= 9)
   {
    Serial.print("000");
    Serial.print(sensorValue5);
    }
   else if(sensorValue5 >= 10 &&
sensorValue5 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue5);
    }
   else if(sensorValue5 >= 100 &&
sensorValue5 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue5);
    }
    else
   {
    Serial.print(sensorValue5);
    }
   }
```

6.3 Listing Program Slave C

```
int
sensorValue0,sensorValue1,sensorValue
2,sensorValue3,sensorValue4,sensorValu
e5;
String C;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  C = String ("C");
}
void loop() {
  // put your main code here, to run
repeatedly:
  Serial.print(C);
  bacaSensor();
  tampilanData();
  Serial.println();
  delay(1000);

}
void bacaSensor(void)
{
  sensorValue0 =analogRead(A0);
  sensorValue1 =analogRead(A1);
  sensorValue2 =analogRead(A2);
  sensorValue3 =analogRead(A3);
  sensorValue4 =analogRead(A4);
  sensorValue5 =analogRead(A5);
  }
  void tampilanData(void)
  {
    if(sensorValue0 >= 0 &&
sensorValue0 <= 9)
    {
     Serial.print("000");
     Serial.print(sensorValue0);
     }
    else if(sensorValue0 >= 10 &&
sensorValue0 <= 99)
    {
     Serial.print("00");
     Serial.print(sensorValue0);
     }
    else if(sensorValue0 >= 100 &&
sensorValue0 <= 999)
    {
     Serial.print("0");
     Serial.print(sensorValue0);
     }
    else
   {
     Serial.print(sensorValue0);
     }
     if(sensorValue1 >= 0 &&
sensorValue1 <= 9)
    {
     Serial.print("000");
     Serial.print(sensorValue1);
     }
```

```
    else if(sensorValue1 >= 10 &&
sensorValue1 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue1);
    }
    else if(sensorValue1 >= 100 &&
sensorValue1 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue1);
    }
    else
   {
    Serial.print(sensorValue1);
    }
   if(sensorValue2 >= 0 &&
sensorValue2 <= 9)
   {
    Serial.print("000");
    Serial.print(sensorValue2);
    }
    else if(sensorValue2 >= 10 &&
sensorValue2 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue2);
    }
   else if(sensorValue2 >= 100 &&
sensorValue2 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue2);
    }
    else
   {
    Serial.print(sensorValue2);
    }
     if(sensorValue3 >= 0 &&
sensorValue3 <= 9)
   {
    Serial.print("000");
    Serial.print(sensorValue3);
    }
   else if(sensorValue3 >= 10 &&
sensorValue3 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue3);
    }
   else if(sensorValue3 >= 100 &&
sensorValue3 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue3);
    }
    else
   {
    Serial.print(sensorValue3);
    }
     if(sensorValue4 >= 0 &&
sensorValue4 <= 9)
   {
    Serial.print("000");
    Serial.print(sensorValue4);
    }
   else if(sensorValue4 >= 10 &&
sensorValue4 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue4);
    }
   else if(sensorValue4 >= 100 &&
sensorValue4 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue4);
    }
    else
   {
    Serial.print(sensorValue4);
    }
     if(sensorValue5 >= 0 &&
sensorValue5 <= 9)
   {
    Serial.print("000");
    Serial.print(sensorValue5);
    }
   else if(sensorValue5 >= 10 &&
sensorValue5 <= 99)
   {
    Serial.print("00");
    Serial.print(sensorValue5);
    }
   else if(sensorValue5 >= 100 &&
sensorValue5 <= 999)
   {
    Serial.print("0");
    Serial.print(sensorValue5);
    }
```

```
   else
   {
    Serial.print(sensorValue5);
    }
  }
```

6.4 Listing Program Master

```
String Terima = "";
void setup() {
 Serial.begin(9600);
}
void loop()
{
 if(Serial.available()) {
 while (Serial.available() > 0) {
char inChar =(char)Serial.read();
  Terima+=inChar;
 }
 Serial.print(Terima);
    Terima ="";
 }}
```

6.5 Listing Program Dekstop

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Dekstop
{
    public partial class Form1 : Form
    {
Double a;
double ArusSumber_A, ArusBaterai_A,
ArusBeban_A, TeganganSumber_A,
TeganganBaterai_A, TeganganBeban_A;
double DayaSumber_A, DayaBaterai_A,
DayaBeban_A;
double KapasitasSumber_A,
KapasitasBaterai_A,KapasitasBeban_A;
double EnergiSumber_A,
EnergiBaterai_A, EnergiBeban_A;
double ArusSumber_B, ArusBaterai_B,
ArusBeban_B, TeganganSumber_B,
TeganganBaterai_B, TeganganBeban_B;
double DayaSumber_B, DayaBaterai_B,
DayaBeban_B;
double KapasitasSumber_B,
KapasitasBaterai_B,KapasitasBeban_B;
double EnergiSumber_B,
EnergiBaterai_B, EnergiBeban_B;
double ArusSumber_C, ArusBaterai_C,
ArusBeban_C, TeganganSumber_C,
TeganganBaterai_C, TeganganBeban_C;
double DayaSumber_C, DayaBaterai_C,
DayaBeban_C;
double KapasitasSumber_C,
KapasitasBaterai_C,KapasitasBeban_C;
double EnergiSumber_C,
EnergiBaterai_C, EnergiBeban_C;
int detik, menit , jam, waktu;
string RXstring, Slave;

public Form1()
        {
InitializeComponent();
        }
private void Form1_Load(object sender,
EventArgs e)
        {
            string[] NamaPort =
System.IO.Ports.SerialPort.GetPortName
s();
for (int i = 0; i <= NamaPort.Length -
1; i++)
            {

DaftarPort.Items.Add(NamaPort[i]);
            }
        }
//<<< Bagian Penampil Grafik >>>>>>>>>
        private void
timer1_Tick(object sender, EventArgs
e)
        {
            a++;
// Slave A
GrafikSumberA.AddXY(0, a,
Convert.ToDouble(DayaSumber_A));
GrafikBateraiA.AddXY(0, a,
Convert.ToDouble(DayaBaterai_A));
GrafikBebanA.AddXY(0, a,
Convert.ToDouble(DayaBeban_A));
// Slave B
GrafikSumberB.AddXY(0, a,
Convert.ToDouble(DayaSumber_B));
GrafikBateraiB.AddXY(0, a,
Convert.ToDouble(DayaBaterai_B));
GrafikBebanB.AddXY(0, a,
Convert.ToDouble(DayaBeban_B));
// Slave C
GrafikSumberC.AddXY(0, a,
Convert.ToDouble(DayaSumber_C));
GrafikBateraiC.AddXY(0, a,
Convert.ToDouble(DayaBaterai_C));
GrafikBebanC.AddXY(0, a,
Convert.ToDouble(DayaBeban_C));
        }
private void
serialPort1_DataReceived(object
sender,
System.IO.Ports.SerialDataReceivedEven
tArgs e)
        {
RXstring = serialPort1.ReadLine();
//Terima paket data yang lengkap
```

```
Slave = RXstring.Substring (0, 1);
//Membaca nama slave (Huruf pertama
pada paket data)

//<<< Bagian Pemisah Data
>>>>>>>>>>>>>>>
if (Slave == "A")
{   //Sumber A
ArusSumber_A =
((Convert.ToDouble(RXstring.Substring(
5, 4)) * 0.0049) - 2.55)/0.01;//ambil
Data dari bit pertama sebanyak 4 bit
TeganganSumber_A =
(Convert.ToDouble(RXstring.Substring(1
, 4)) * 0.02238) + 0.1125;// cari Vin
dengan rumus ADC 10 bit dan
pengkondisian tegangan dari 0 sampai
25 ke 0 sampai 5 V
DayaSumber_A = ArusSumber_A *
TeganganSumber_A;
EnergiSumber_A = ((DayaSumber_A *
waktu)/3600);
KapasitasSumber_A = ((ArusSumber_A *
waktu) / 3600);
//Baterai A
ArusBaterai_A =
((Convert.ToDouble(RXstring.Substring(
13, 4)) * 0.0049) - 2.55)/0.01;// cari
Vin dengan rumus ADC 10 bit 0 sampai 5
V
TeganganBaterai_A =
(Convert.ToDouble(RXstring.Substring(9
, 4)) * 0.02238) + 0.1125;
DayaBaterai_A = ArusBaterai_A *
TeganganBaterai_A;
EnergiBaterai_A = ((DayaBaterai_A *
waktu)/3600);
KapasitasBaterai_A = ((ArusBaterai_A *
waktu) / 3600);
//Beban A
ArusBeban_A =
((Convert.ToDouble(RXstring.Substring(
21, 4)) * 0.0049) - 2.55)/0.01;
TeganganBeban_A =
(Convert.ToDouble(RXstring.Substring(1
7, 4)) * 0.02238) + 0.1125;
DayaBeban_A = ArusBeban_A *
TeganganBeban_A;
EnergiBeban_A = ((DayaBeban_A *
waktu)/3600);
KapasitasBeban_A = ((ArusBeban_A *
waktu) / 3600);
            }
            if (Slave == "B")
            {   //Sumber B
ArusSumber_B =
((Convert.ToDouble(RXstring.Substring(
5, 4)) * 0.0049) - 2.55)/0.01
TeganganSumber_B =
(Convert.ToDouble(RXstring.Substring(1
, 4)) * 0.02238) + 0.1125;
DayaSumber_B = ArusSumber_B *
TeganganSumber_B;
EnergiSumber_B = ((DayaSumber_B *
waktu)/3600);
KapasitasSumber_B = ((ArusSumber_B *
waktu) / 3600);
//Baterai B
ArusBaterai_B =
((Convert.ToDouble(RXstring.Substring(
13, 4)) * 0.0049) - 2.55)/0.01;
TeganganBaterai_B =
(Convert.ToDouble(RXstring.Substring(9
, 4)) * 0.02238) + 0.1125;
DayaBaterai_B = ArusBaterai_B *
TeganganBaterai_B;
EnergiBaterai_B = ((DayaBaterai_B *
waktu) / 3600);
KapasitasBaterai_B = ((ArusBaterai_B *
waktu) / 3600);
//Beban B
ArusBeban_B =
((Convert.ToDouble(RXstring.Substring(
21, 4)) * 0.0049) - 2.55)/0.01;
TeganganBeban_B =
(Convert.ToDouble(RXstring.Substring(1
7, 4)) * 0.02238) + 0.1125;
DayaBeban_B = ArusBeban_B *
TeganganBeban_B;
EnergiBeban_B = ((DayaBeban_B * waktu)
/ 3600);
KapasitasBeban_B = ((ArusBeban_B *
waktu) / 3600);
            }
            if (Slave == "C")
            {   //Sumber C
ArusSumber_C =
((Convert.ToDouble(RXstring.Substring(
5, 4)) * 0.0049) - 2.55)/0.01;
TeganganSumber_C =
(Convert.ToDouble(RXstring.Substring(1
, 4)) * 0.02238) + 0.1125;
DayaSumber_C = ArusSumber_C *
TeganganSumber_C;
EnergiSumber_C = ((DayaSumber_C *
waktu) / 3600);
KapasitasSumber_C = ((ArusSumber_C *
waktu) / 3600);
//Baterai C
ArusBaterai_C =
((Convert.ToDouble(RXstring.Substring(
13, 4)) * 0.0049) - 2.55)/0.01;
TeganganBaterai_C =
(Convert.ToDouble(RXstring.Substring(9
, 4)) * 0.02238) + 0.1125;
DayaBaterai_C = ArusBaterai_C *
TeganganBaterai_C;
EnergiBaterai_C = ((DayaBaterai_C *
waktu) / 3600);
KapasitasBaterai_C = ((ArusBaterai_C *
waktu) / 3600);
//Beban C
```

```
ArusBeban_C =
((Convert.ToDouble(RXstring.Substring(
21, 4)) * 0.0049) - 2.55)/0.01;
TeganganBeban_C =
(Convert.ToDouble(RXstring.Substring(1
7, 4)) * 0.02238) + 0.1125;
DayaBeban_C = ArusBeban_C *
TeganganBeban_C;
EnergiBeban_C = ((DayaBeban_C * waktu)
/ 3600);
KapasitasBeban_C = ((ArusBeban_C *
waktu) / 3600);
            }
//<<<<<<< END >>>>>>>>>>>
this.Invoke(new
EventHandler(tampil_kata));
        }
private void tampil_kata(object
sender, EventArgs e)
        {
richTextBox1.AppendText(RXstring);
//<<< Bagian Menampilkan Data
>>>>>>>>>>>
//<<<<<<< Slave A >>>>>>
// Sumber A
ArusSumberA.Text =
(Math.Round(ArusSumber_A,
2)).ToString() + " A";//tampilkan data
hasil pembacaan dan di convert dua
digit di belakang koma
TeganganSumberA.Text =
(Math.Round(TeganganSumber_A,
2)).ToString() + " V";
DayaSumberA.Text =
(Math.Round(DayaSumber_A,
2)).ToString() + " W";
EnergiSumberA.Text =
(Math.Round(EnergiSumber_A,
2)).ToString() + " Wh";
KapasitasSumberA.Text =
(Math.Round(KapasitasSumber_A,
2)).ToString() + " Ah";
// Baterai A
ArusBateraiA.Text =
(Math.Round(ArusBaterai_A,
2)).ToString() + " A";
TeganganBateraiA.Text =
(Math.Round(TeganganBaterai_A,
2)).ToString() + " V";
DayaBateraiA.Text =
(Math.Round(DayaBaterai_A,
2)).ToString() + " W";
EnergiBateraiA.Text =
(Math.Round(EnergiBaterai_A ,
2)).ToString() + " Wh";
KapasitasBateraiA.Text =
(Math.Round(KapasitasBaterai_A,
2)).ToString() + " Ah";
// Beban A
ArusBebanA.Text =
(Math.Round(ArusBeban_A,
2)).ToString() + " A";
TeganganBebanA.Text =
(Math.Round(TeganganBeban_A,
2)).ToString() + " V";
DayaBebanA.Text =
(Math.Round(DayaBeban_A,
2)).ToString() + " W";
EnergiBebanA.Text =
(Math.Round(EnergiBeban_A,
2)).ToString() + " Wh";
KapasitasBebanA.Text =
(Math.Round(KapasitasBeban_A,
2)).ToString() + " Ah";
//<<<<<<<<< Slave B >>>>>>>
// Sumber B
ArusSumberB.Text =
(Math.Round(ArusSumber_B,
2)).ToString() + " A";
TeganganSumberB.Text =
(Math.Round(TeganganSumber_B,
2)).ToString() + " V";
DayaSumberB.Text =
(Math.Round(DayaSumber_B,
2)).ToString() + " W";
EnergiSumberB.Text =
(Math.Round(EnergiSumber_B,
2)).ToString() + " Wh";
KapasitasSumberB.Text =
(Math.Round(KapasitasSumber_B,
2)).ToString() + " Ah";
// Baterai B
ArusBateraiB.Text =
(Math.Round(ArusBaterai_B,
2)).ToString() + " A";
TeganganBateraiB.Text =
(Math.Round(TeganganBaterai_B,
2)).ToString() + " V";
DayaBateraiB.Text =
(Math.Round(DayaBaterai_B,
2)).ToString() + " W";
EnergiBateraiB.Text =
(Math.Round(EnergiBaterai_B,
2)).ToString() + " Wh";
KapasitasBateraiB.Text =
(Math.Round(KapasitasBaterai_B,
2)).ToString() + " Ah";
// Beban B
ArusBebanB.Text =
(Math.Round(ArusBeban_B,
2)).ToString() + " A";
TeganganBebanB.Text =
(Math.Round(TeganganBeban_B,
2)).ToString() + " V";
DayaBebanB.Text =
(Math.Round(DayaBeban_B,
2)).ToString() + " W";
EnergiBebanB.Text =
(Math.Round(EnergiBeban_B,
2)).ToString() + " Wh";
KapasitasBebanB.Text =
(Math.Round(KapasitasBeban_B,
2)).ToString() + " Ah";
//<<<<<<<<<< Slave C >>>>>>>
```

```
// Sumber C
ArusSumberC.Text =
(Math.Round(ArusSumber_C,
2)).ToString() + " A";
TeganganSumberC.Text =
(Math.Round(TeganganSumber_C,
2)).ToString() + " V";
DayaSumberC.Text =
(Math.Round(DayaSumber_C,
2)).ToString() + " W";
EnergiSumberC.Text =
(Math.Round(EnergiSumber_C,
2)).ToString() + " Wh";
KapasitasSumberC.Text =
(Math.Round(KapasitasSumber_C,
2)).ToString() + " Ah";
// Baterai C
ArusBateraiC.Text =
(Math.Round(ArusBaterai_C,
2)).ToString() + " A";
TeganganBateraiC.Text =
(Math.Round(TeganganBaterai_C,
2)).ToString() + " V";
DayaBateraiC.Text =
(Math.Round(DayaBaterai_C,
2)).ToString() + " W";
EnergiBateraiC.Text =
(Math.Round(EnergiBaterai_C,
2)).ToString() + " Wh";
KapasitasBateraiC.Text =
(Math.Round(KapasitasBaterai_C,
2)).ToString() + " Ah";
// beban C
ArusBebanC.Text =
(Math.Round(ArusBeban_C,
2)).ToString() + " A";
TeganganBebanC.Text =
(Math.Round(TeganganBeban_C,
2)).ToString() + " V";
DayaBebanC.Text =
(Math.Round(DayaBeban_C,
2)).ToString() + " W";
EnergiBebanC.Text =
(Math.Round(EnergiBeban_C,
2)).ToString() + " Wh";
KapasitasBebanC.Text =
(Math.Round(KapasitasBeban_C,
2)).ToString() + " Ah";
//<<<<<<<<<< END >>>>>>>>>
        }
        private void
richTextBox1_KeyPress(object sender,
KeyPressEventArgs e)
        {
            if (!serialPort1.IsOpen)
return;
char[] buff = new char[1];
buff[0] = e.KeyChar;
serialPort1.Read(buff, 0, 1);//membaca
data yang dikirim dari serial
            e.Handled = true;
        }
        private void
Form1_FormClosing(object sender,
FormClosingEventArgs e)
        {
            if (serialPort1.IsOpen)
serialPort1.Close();
        }
        private void
btnStart_Click_1(object sender,
EventArgs e)
        {
            btnStart.BackColor =
Color.Green;
            timer1.Start();
            timer2.Start();
            if (!serialPort1.IsOpen)
                try
                {
serialPort1.PortName =
Convert.ToString(DaftarPort.SelectedIt
em);
serialPort1.BaudRate =
Convert.ToInt32(DaftarBaudrate.Selecte
dItem);

serialPort1.Open();
                    btnStart.Text =
"Stop";
                    btnStart.BackColor
= Color.Red;

richTextBox1.ReadOnly = false;
                }
                catch
                {
MessageBox.Show("Erorr, Pastikan
serial port yang digunakan benar dan
telah terhubung!!!");
                }
            else
            {
                serialPort1.Close();
                btnStart.Text =
"Start";
                btnStart.BackColor =
Color.White;
                timer1.Stop();
                timer2.Stop();
            }

        }
//<<<<<<<<<<<<<Penampil Waktu>>>>>
        private void
timer2_Tick(object sender, EventArgs
e)
        {
            waktu = waktu + 1;
            detik = detik + 1;
Penghitung.Text = jam.ToString() + "
Jam" + " " + menit.ToString() + "
Menit" + " " + detik.ToString() + "
Detik";
```

```
            if (detik == 60)
            {
                detik = 0;
                menit = menit + 1;
                detik = detik + 1;
Penghitung.Text = jam.ToString() + "
Jam" + " " + menit.ToString() + "
Menit" + " " + detik.ToString() + "
Detik";
               if (menit == 60)
                {
                    detik = 0;
                    menit = 0;
                    jam = jam + 1;
                    menit = menit + 1;
                    detik = detik + 1;
Penghitung.Text = jam.ToString() + "
Jam" + " " + menit.ToString() + "
Menit" + " " + detik.ToString() + "
Metik";
                }
            }
        }
    }
}
```