

REMINDER KIT SECARA WIRELESS BERBASIS MIKROKONTROLER AVR

Irwan Sanyoto, Antonius Wibowo, Lanny Agustine
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Katolik Widya Mandala Surabaya, Jalan Kalijudan 37
Surabaya

Email : irwansanyoto@gmail.com

ABSTRAK

Tujuan untuk pembuatan alat ini adalah untuk mengingatkan penggunaannya agar tidak melupakan barang-barang yang sangat penting seperti telepon genggam, kunci mobil, dan lain-lain.

Pada alat reminder kit ini terdapat dua bagian. Bagian pertama merupakan bagian slave yang terdiri dari baterai, rangkaian mikrokontroler dengan menggunakan IC ATmega8L, Modul RFM12, dan Buzzer. Pada bagian master hampir sama dengan slave namun hanya aktif saat memancarkan sinyal yang diatur setiap 3 detik sekali.

Reminder kit ini bekerja menggunakan frekuensi radio dengan menggunakan modul RFM12 yang dapat berfungsi sebagai receiver dan transmitter. Bagian slave akan diletakkan pada peralatan-peralatan yang dianggap penting misalnya dipasang pada kunci mobil, tas laptop, dan lain-lain. Bagian master akan diletakkan pada penggunaannya (dibawa oleh user) dan dapat diletakkan pada ikat pinggang atau kantong handphone. Apabila jarak master dan slave lebih dari 10 meter maka buzzer pada bagian master dan slave akan berbunyi sehingga dapat mengingatkan pengguna barang yang tertinggal tersebut.

Berdasarkan Pengujian alat untuk komunikasi antara master dan slave dapat berjalan dengan baik, namun keterbatasan sistem terletak pada kemampuan komunikasi master dan slave jika terhalang tembok, jendela, dan jarak yang mengakibatkan daya sinyal mengalami peredaman diatas 79 dBm. Dengan adanya sistem ini diharapkan kasus tertinggalnya barang dapat berkurang.

Kata kunci: Modul RFM12, mikrokontroler, transmitter, receiver, reminder kit, buzzer.

I. Pendahuluan

Kesibukan dalam pekerjaan membuat orang cenderung bergegas, sering kali banyak pula barang yang harus dibawa. Hal-hal tersebut dapat memicu kemungkinan barang yang penting seperti *handphone*, dompet, kunci dapat tertinggal sehingga perlu alat untuk pengingat agar dapat menghindari barang-barang penting tertinggal atau bahkan hilang.

Dengan itu pada skripsi ini penulis membuat alat yang berfungsi sebagai pengingat atau *reminder* secara *wireless* dengan dimensi alat yang dirancang cukup kecil sehingga efisien dan praktis untuk dipasang di berbagai barang penting. Pada alat ini terdapat 2 buah bagian yaitu bagian *master* dan bagian *slave*. Alat ini memiliki 1 *master* dan beberapa *slave* yang akan diletakkan pada alat-alat yang dianggap penting sehingga dapat mengingatkan secara otomatis apabila tertinggal. Untuk indikator *reminder* atau pengingatnya akan digunakan *buzzer* yang dipasang pada sisi *master* dan sisi *slave*. Masalah yang muncul dalam pengerjaan alat adalah:

1. Penentuan format kode (informasi) yang sesuai untuk pengenalan *device*.
2. Penentuan formasi indikasi bunyi buzzer yang mudah menarik perhatian.

3. Perancangan perangkat keras agar ukuran cukup kecil, dan sederhana dalam penggunaannya.
4. Pemrograman perangkat lunak sehingga kode dari *master* dapat dikenali oleh *slave*.
5. Perancangan rangkaian serta metode komunikasi *master* dan *slave* agar hemat energi.

Agar sistem ini lebih spesifik dan terarah, maka pembahasan masalah dalam program ini memiliki batasan-batasan sebagai berikut:

1. Pada pembuatan alat skripsi ini menggunakan 3 buah *slave* untuk dipasang pada 3 obyek dan 1 buah *master* untuk dibawa oleh pengguna dan untuk sistem minimum menggunakan ATmega8L sebagai pusat pengolahan data dan program.
2. Jarak jangkauan komunikasi maksimal antara *master* dan *slave* sejauh 10 meter.
3. Dipasangkan pada alat-alat yang seyogyanya tidak bisa jauh dari pengguna seperti *handphone*, dompet, kunci.
4. Komunikasi antara *master* dan *slave* menggunakan modul RFM12 sebagai modul *transceiverradio frequency*.
5. Alat dapat bertahan minimal 1 hari tanpa perlu mencharge baterai baik dalam saat beroperasi dan *standby*.

- Ukuran maksimal *slave* sebesar 5,5 cm x 4,5 cm x 2 cm.
- Ukuran maksimal *master* sebesar 8 cm x 5,5 cm x 3,5 cm.

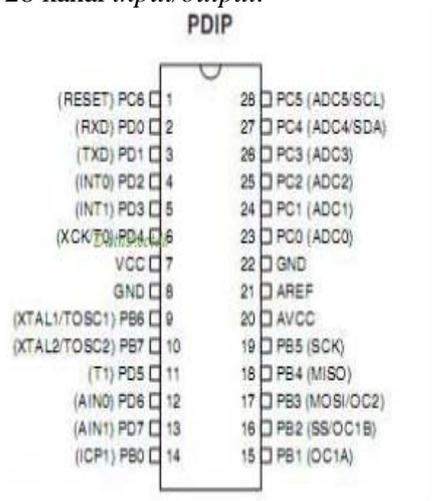
Adapun tujuan yang hendak dicapai penulis dalam pembuatan alat ini adalah: Membuat *reminder kit* yang berfungsi untuk mengingatkan penggunaanya saat ada barang-barang yang tertinggal.

II. Tinjauan Pustaka

II.1 Mikrokontroler ATmega8L^[3]

Pada skripsi ini digunakan mikrokontroler ATmega8L yang secara umum memiliki spesifikasi :

- 512 bytes *Electrical Erasable and Programmable Read Only Memory* (EEPROM)
- Mikrokontroler 8 bit dengan 8k bytes *in-system programmable flash*, dengan kemampuan 10.000 kali tulis/hapus.
- 1000 bytes internal SRAM.
- Tegangan Operasi 2,7 – 5,5 Volt.
- Oscillator* internal.
- 28 kanal *input/output*.



Gambar 1. Pin Out ATmega8L

Pada skripsi ini, fungsi dari pin yang digunakan oleh IC mikrokontroler ATmega8L pada *master* adalah sebagai berikut :

- VCC : Pin yang dihubungkan ke terminal tegangan 3,7 volt sebagai sumber tegangan untuk mikrokontroler
- GND : Pin yang dihubungkan ke *ground* sumber (VCC) sebagai sumber *ground* digital dari mikrokontroler.
- PORT B* : Terdiri dari PB0 sampai dengan PB7, merupakan 8 *bitport I/O*

dua arah yang dilengkapi dengan resistor *pull-up* disetiap pinnya.

- PORT C* : Terdiri dari PC0 sampai dengan PC6, merupakan 7 *bit port* yang bisa dimanfaatkan sebagai jalur *I/O* atau dimanfaatkan fungsi khususnya. Pada pembuatan skripsi ini PINC.1 diatur agar menggunakan fungsi khususnya yaitu sebagai pembaca ADC.
- PORT D* : Terdiri dari PD0 sampai PD7, merupakan 8 *bitport* yang bisa dimanfaatkan sebagai jalur *I/O* atau dimanfaatkan fungsi khususnya. Pada pembuatan skripsi ini PIND.0 dan PIND.1 diatur agar menggunakan fungsi khususnya yaitu sebagai jalur USART.

ATmega8L memiliki tiga buah *port* yang merupakan jalur *bidirectional* dengan resistor *pull-up* internal. Bila *DDxn* diset 1 berfungsi sebagai *pin output* dan akan berfungsi sebagai *pin input* bila diset 0. Bila *PORTxndis* 1 ketika *pin* dikondisikan sebagai *pin input* maka resistor *pull-up* akan diaktifkan. Untuk mencegah hal ini maka *PORTxn* harus diset 0 atau *pin* dikondisikan sebagai *pin output*. Apabila *PORTxn* diset 1 ketika *pin* dikondisikan sebagai *pin output* maka *pin port* akan bernilai *high*. Dan apabila *PORTxndis* 0 ketika *pin* dikondisikan sebagai *pin output* maka *pin port* akan bernilai *low*. Saat mengubah kondisi port dari kondisi mengambang (*DDxn=0, PORTxn=0*) ke kondisi *output high* (*DDxn=1, PORTxn=1*) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (*DDxn=0, PORTxn=1*) atau kondisi *output low* (*DDxn=1, PORTxn=0*). Bit PUD pada register SFIOR dapat diset 1 untuk mematikan semua resistor *pull-up* dalam semua port. Masing-masing konfigurasi *port* dapat dilihat pada tabel 1.

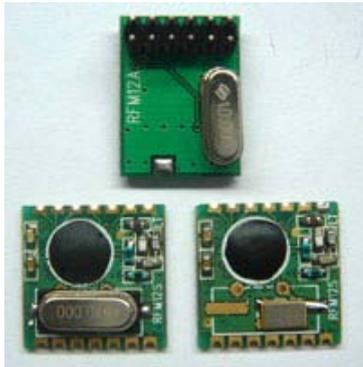
Tabel 1. Konfigurasi Port ATmega8L

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

II.2 Radio Frequency Module RFM12^[1]

Modul *Radio Frequency* merupakan salah satu modul yang digunakan untuk melakukan komunikasi sinyal digital tanpa kabel. Modul *radio frequency* RFM12^[2] merupakan salah satu

modul *Radio Frequency* yang bekerja pada band frekuensi 433 MHz. Untuk bentuk fisik modul yang digunakan dapat dilihat pada gambar 2



Gambar 2. Modul RFM12

Spesifikasi yang digunakan dari *Radio Frequency Module RFM12* antara lain:

- Tidak membutuhkan *tuning* pada frekuensi
- Menggunakan teknologi PLL resolusi tinggi (2,5Khz)
- Mempunyai *datarate* 115,2 kbps menggunakan demodulator internal dan 256 kbps menggunakan filter RC eksternal
- *Differential antenna*
- *Tuning* antena otomatis dilakukan oleh modul
- *Internal data filtering*
- 16 bit RX data FIFO
- 2 kali 8 bit TX data register
- 10 MHz crystal untuk PLL *timing*.
- *Power supply* 2.2V – 5.4 V
- *Standby Current* lebih kecil dari 3µA

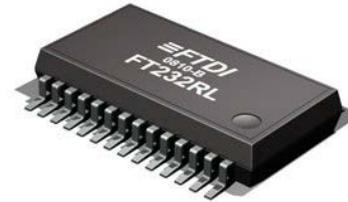
II.3 FT232 USB to UART Converter^[4]

The FT232R^[4] adalah sebuah IC yang digunakan untuk mengconvert data (address/identitas) agar dapat dibaca oleh mikrokontroler. IC FT232 memiliki beberapa fitur yaitu :

1. Tegangan kerja antara 3,3 V sampai 5,25 V
2. *Compatible* dengan USB 2.0
3. Tidak membutuhkan program tambahan.
4. Tidak membutuhkan arus yang besar.

Dalam skripsi ini FT232 digunakan sebagai media untuk memasukan *address/identitas* dari PC ke dalam *master* dan *slave*. FT232 dapat digunakan pada *Operating System/OS* Windows

98, 98SE, ME, 2000, Server 2003, XP dan Server 2008, Windows 7 32,64-bit, Windows XP dan XP 64-bit, Windows Vista and Vista 64-bit. Untuk gambar fisik dapat dilihat pada gambar 3.



Gambar 3. Bentuk Fisik FT232

III. Metode Penelitian

Pada pembuatan alat ini, terdapat dua bagian besar yang saling berhubungan, yaitu perancangan *hardware* dan *software*. Kedua bagian ini saling melengkapi untuk melaksanakan fungsi alat.

Pada perancangan *hardware*, meliputi perancangan dan pembuatan modul *interface*, yaitu membuat interface antara mikrokontroler ATmega8L dengan *Radio Frequency Module RFM12* pada bagian *master* dan *slave*. Sedangkan perancangan *software* meliputi perancangan dan pembuatan program yang akan digunakan dalam skripsi ini.

III.1 Perancangan Alat

Dalam bab ini akan dijelaskan tentang perancangan dan perencanaan alat, yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Gambar 4 merupakan diagram blok dari sistem yang dibuat:

Penjelasan dari Gambar 4 dan Gambar 5

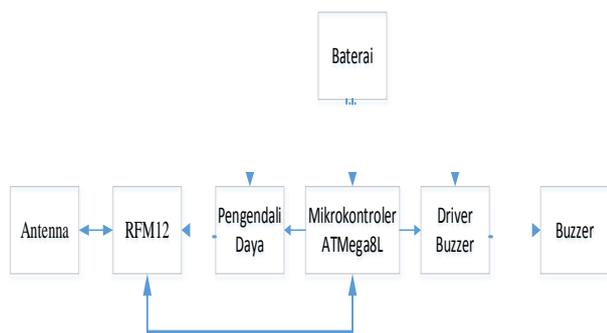
- Antena adalah bagian vital dari suatu *master* atau *slave* yang berfungsi untuk menyalurkan sinyal radio ke udara. Antena disini akan menyalurkan sinyal radio yang diterima dari udara dan diteruskan menuju RFM12.
- RFM12 adalah sebuah modul *transceiver* frekuensi radio yang menerima dan mengirimkan sinyal yang berisi data dari dan ke mikrokontroler ATmega8L.
- Pengendali daya berfungsi untuk menghidupkan atau mematikan modul RFM12. Penggunaan pengendali daya bertujuan untuk menghemat daya dimana modul RFM12 akan hidup saat akan mengirimkan sinyal dan mati setelah mengirimkan sinyal perintah dan menerima sinyal balasan dari sisi *slave*.

- o Sistem minimum mikrokontroler ATmega8L merupakan komponen terpenting dalam alat ini, karena berfungsi sebagai pengolah data yang diterima dari RFM12. Yang kemudian memberikan respon yang selanjutnya dikirim melalui RFM12 ke *slave* atau memberikan respon untuk menyalakan *buzzer*.
- o *Buzzer* berfungsi untuk menghasilkan bunyi.
- o Baterai sebagai catu daya untuk mengaktifkan mikrokontroler, RFM12, *buzzer*.

Cara kerja dari alat yang akan dibuat adalah sebagai berikut :

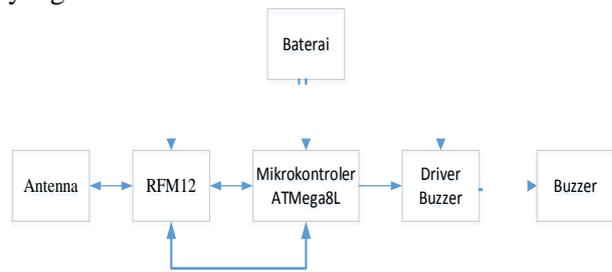
Pada bagian *master* pertama-tama yang dilakukan adalah dengan mengirimkan kode identitas *slave* secara berurutan ke *slave*. Pada bagian *master* status dari *master* akan berubah dari status kirim menjadi *standby* untuk menerima jawaban dari bagian *slave*. Apabila tidak ada jawaban dari bagian *slave* maka *buzzer* pada bagian *master* akan berbunyi. Apabila pada bagian *master* menerima jawaban dari bagian *slave* maka pada bagian *master* akan menjadi *idle* sampai pengiriman informasi berikutnya.

Bagian *slave*, *slave* akan selalu pada mode *standby* untuk menerima data dari *master* lalu kemudian sistem mikrokontroler ATmega8L akan mengecek apakah data yang diterima sesuai dengan data yang telah dikirim oleh *master*. Apabila sesuai, maka *slave* akan memberikan jawaban balik, sehingga status pada bagian *slave* akan berubah menjadi pengirim/transmitter, dan mengirimkan informasi ke *master*. Apabila bagian *slave* tidak menerima data informasi selama waktu tertentu, maka *buzzer* pada *slave* akan berbunyi yang berarti *slave* telah hilang dari jangkauan *master* atau hilang.



Gambar 4. diagram blok master

Gambar 5. merupakan diagram blok dari sistem yang dibuat:



Gambar 5. Diagram Blok Slave

Keterangan :

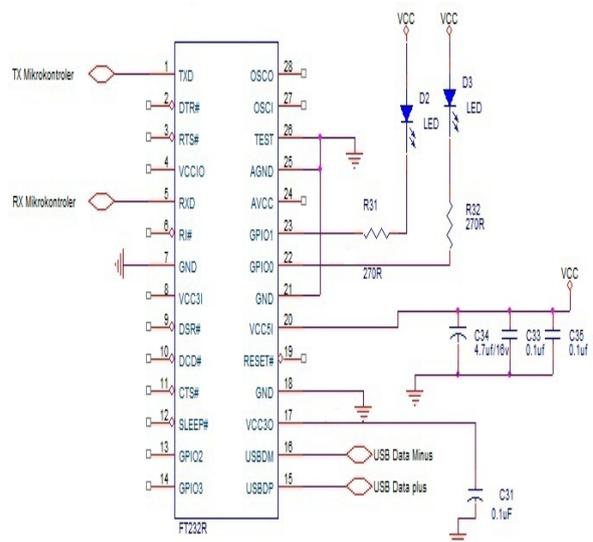
-----► Jalur daya.

—————► Jalur informasi.

III.2. Perancangan Hardware

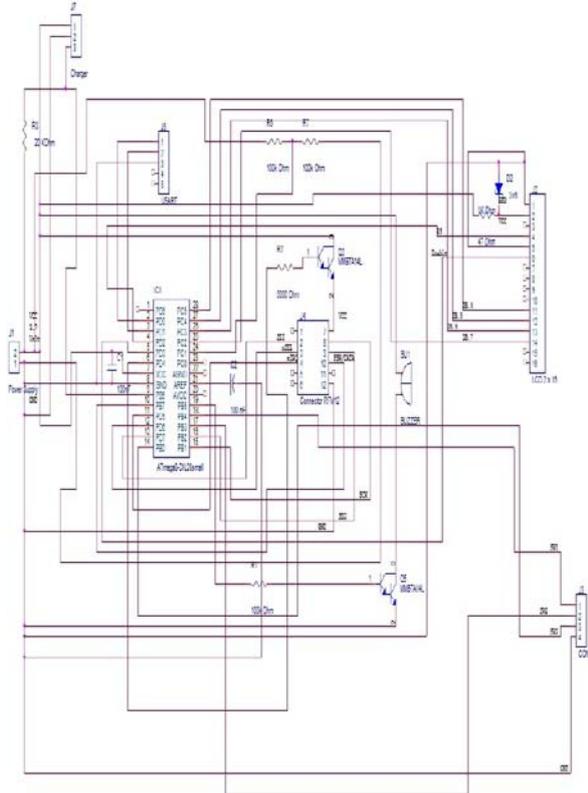
Dalam perancangan *hardware* mikrokontroler yang digunakan pada skripsi ini adalah mikrokontroler keluaran ATMEL dengan seri ATmega8L. Peran mikrokontroler pada alat ini adalah sebagai pusat pemrosesan data dan mengontrol fungsi dari rangkaian elektronik pendukung yang digunakan. Pada skripsi ini terdapat 2 bagian rangkaian yaitu rangkaian *master* dan rangkaian *slave*. Untuk gambar rangkaian pada *master* dapat dilihat pada gambar 6. Untuk rangkaian FT232R dapat dilihat pada gambar 7.

Pada gambar 7 pin 15 dan 16 dihubungkan dengan USB yang nantinya akan berkomunikasi dengan PC dan pin 1 dan 5 yang merupakan pin TX dan RX yang akan terhubung dengan ATmega 8L.



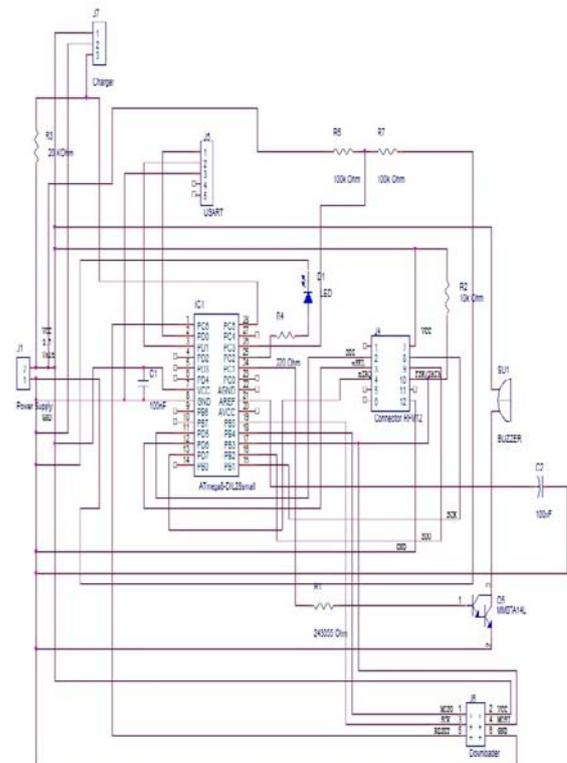
Gambar 7 gambar rangkaian FT232R

Pada gambar 7 pin 15 dan 16 dihubungkan dengan USB yang nantinya akan berkomunikasi dengan PC dan pin 1 dan 5 yang merupakan pin TX dan RX yang akan terhubung dengan ATmega 8L.



Gambar 6.Gambar Rangkaian *master*

Port yang digunakan pada pembuatan alat ini adalah Port B, C, dan D, dimana port B (pin 1, 2, 7), *Port D* (pin 5, 6, 7) digunakan untuk komunikasi antara modul RFM12 dengan mikrokontroler dan port C (pin 2,3,4,5) digunakan sebagai jalur komunikasi antara LCD dengan mikrokontroler, port D digunakan sebagai jalur USART dan jalur I/O yang berhubungan dengan tombol. Sedangkan untuk gambar rangkaian *slave* dapat dilihat pada gambar 7.



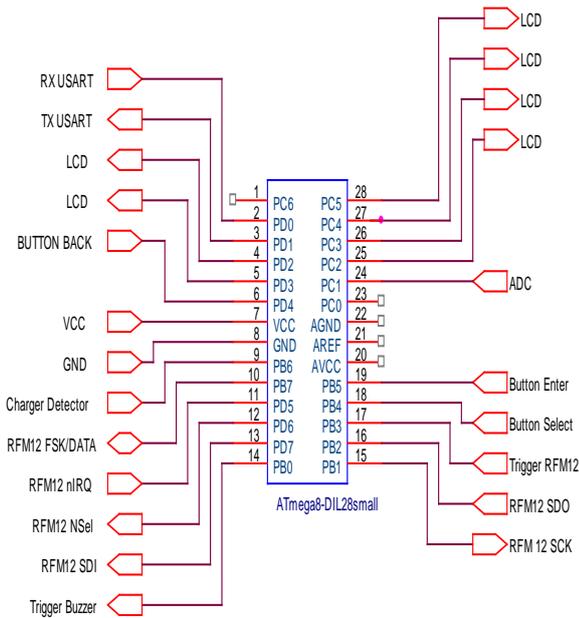
Gambar 7.Rangkaian Pada *Master* Mikrokontroler ATmega8L dan interfacenya

III.3 Perancangan Interface Hardware

Dalam perancangan Interface Hardware ada yang saling menghubungkan-hubungkan antara modul dan ada yang menghubungkan antara modul dengan mikrokontroler ATmega8L.

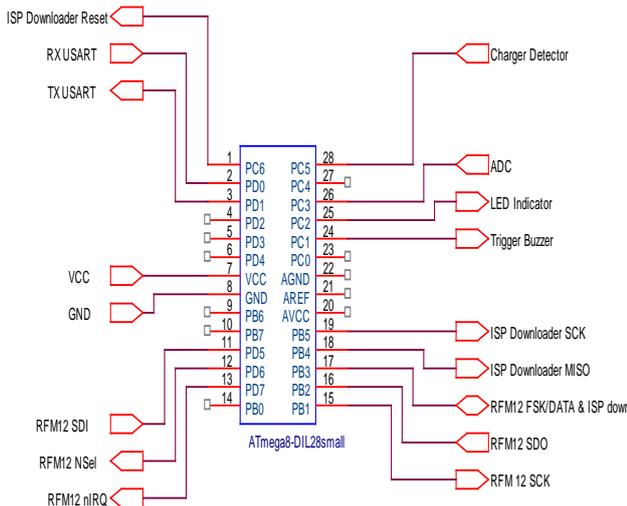
Pada Gambar 8 Menampilkan interface keseluruhan mikrokontroler ATmega8L pada *master*.

Pada Gambar 10 Menampilkan interface pada mikrokontroler ATmega8L dengan *LCD* yang berguna untuk menampilkan perintah dalam bentuk teks. *Port* yang digunakan untuk menghubungkan mikrokontroler ATmega8535 dengan *LCD* dapat dilihat pada Tabel 2.



Gambar 8. Interface Mikrokontroler ATmega8L pada Master

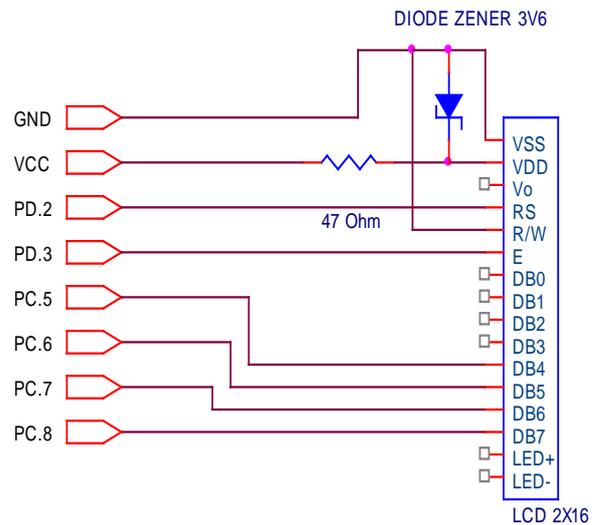
Pada Gambar 8 Menampilkan interface keseluruhan mikrokontroler ATmega8L pada slave.



Gambar 9. Interface Mikrokontroler ATmega8L pada Slave

Tabel 2 Port Yang Digunakan Pada Mikrokontroler ATmega8L Oleh LCD

Port yang digunakan	Fungsi
PC.5	DB.4 pada LCD
PC.6	DB.5 pada LCD
PC.7	DB.6 pada LCD
PC.8	DB.7 pada LCD
PD.3	EN pada LCD
PD.2	RS pada LCD

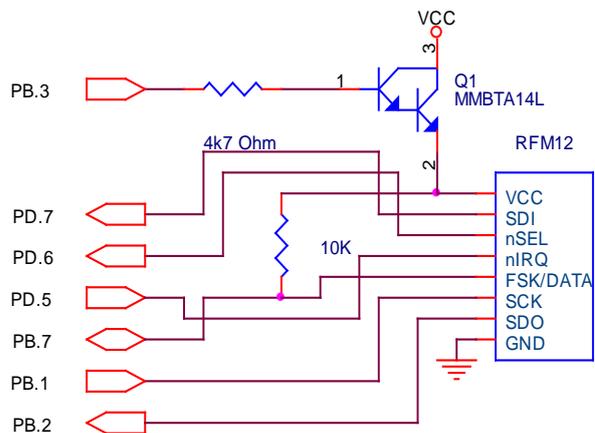


Gambar 10 Interface Receiver Mikrokontroler ATmega8535 Dengan LCD

Pada Gambar 8 Menampilkan interface pada mikrokontroler ATmega8L dengan radio frequency module RFM12 yang berguna untuk mengirimkan perintah dan data secara nirkable. Port yang digunakan untuk menghubungkan mikrokontroler ATmega8L dengan RFM12 dapat dilihat pada tabel 3

Tabel 3Port Yang Digunakan Pada Mikrokontroler ATmega8L Oleh RFM12

Port yang digunakan	Fungsi
PB.3	Pemicu RFM12
PD.7	SDI
PD.6	nSEL
PD.5	nIRQ
PB.7	FSK/DATA
PB.1	SCK
PB.2	SDO`



Gambar 11 Rangkaian Mikrokontroler ATmega8L Dengan Radio Frequency Module RFM12

Pada Gambar 13 Menampilkan rangkaian pada *transmitter* mikrokontroler ATmega324 dengan *LED* yang berguna untuk Indikator warna *LED* pada sensor suara (*EasyVR-Shield*). *Port* yang digunakan untuk menghubungkan mikrokontroler ATmega324 dengan *LED* dan penjelasannya dapat dilihat pada Tabel 8.

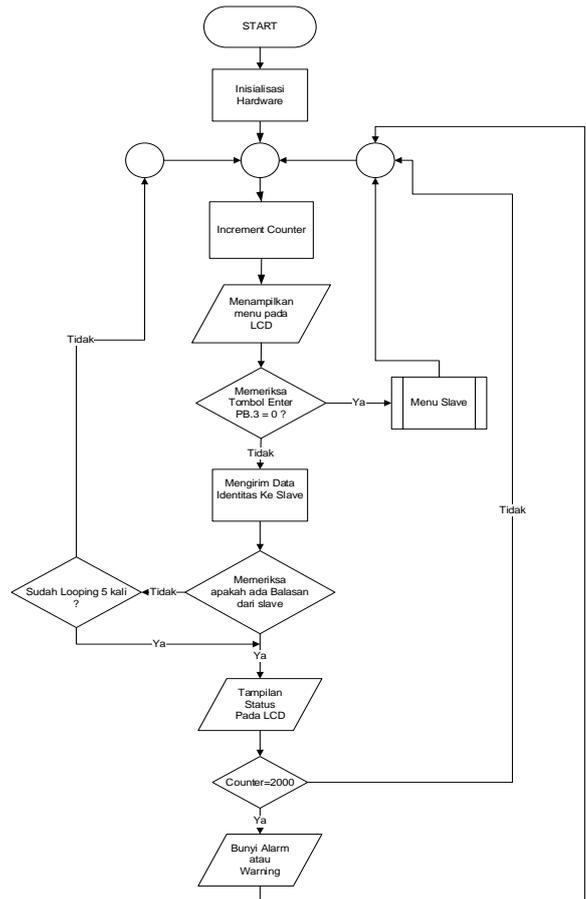
III.4 Perancangan Software

Dalam perancangan perangkat lunak digunakan 2 macam *software* bantuyaitu *CodeVisionAVR Evaluation V2.05.0* dan *Microsoft Visual C#* sebagai compiler. *CodeVision AVR* digunakan untuk melakukan pemograman pada mikrokontroler ATmega8L sementara *Microsoft Visual C#* digunakan untuk melakukan pembuatan program yang akan digunakan pada *PC*.

Pada skripsi ini mikrokontroler ATmega8L digunakan pada kedua sisi alat (*master* dan *slave*) untuk mengontrol proses komunikasi nirkabel, tampilan di LCD, penekanan tombol, dan komunikasi serial dengan PC lewat sarana komunikasi *USART*. Ada beberapa bagian dalam perancangan *software* pada mikrokontroler ATmega8L yakni, jika *master* dan *slave* lebih dari 10 meter maka *buzzer* akan berbunyi, fungsi untuk mematikan *slave*, fungsi untuk mengaktifkan *slave*, dan fungsi untuk mencari *slave*.

Pada perancangan *software* ini terbagi 2 bagian yaitu bagian *master* yang bertujuan agar mikrokontroler ATmega8L pada *master* dapat melakukan tugasnya dengan maksimal. Tugas dari Mikrokontroler pada bagian *master* adalah untuk menampilkan data pada LCD dan untuk mengontrol RFM12 agar dapat mengirimkan dan menerima data dari *slave*. *Flowchart* mikrokontroler bagian *master* dapat dilihat pada gambar 12.

Sedangkan pada *slave* perancangan *software* ini bertujuan agar mikrokontroler ATmega8L pada *slave* dapat melakukan tugasnya dengan maksimal. Tugas dari Mikrokontroler pada bagian *slave* adalah untuk menerima data dari bagian *master* dan untuk mengontrol RFM12 agar dapat menerima dan mengirimkan data dari *master*. *Flowchart* mikrokontroler bagian *slave* dapat dilihat pada gambar 13.



Gambar 12 Flowchart Program Master

Berikut ini penjelasan singkat mengenai *flowchar* gambar 12.

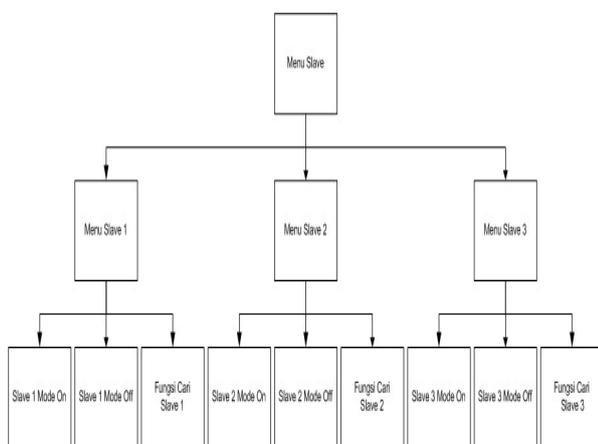
Pada awal mula inisialisasi *hardware* dan port mikrokontroler ATmega8L diinisialisasi sebagai *input*, *output*, atau sebagai pembaca ADC. Kemudian mikrokontroler akan mengatur RFM12 agar bersiap menerima data dari *master*. Kemudian mikrokontroler akan memeriksa apakah data (identitas) yang diterima sesuai dengan data yang dimiliki tiap *slave*. Apabila data yang diterima sesuai dengan data (identitas) maka mikrokontroler akan mengatur RFM12 agar mengirimkan data balasan ke *master* sebagai tanda bahwa *device* dapat berkomunikasi. Apabila *slave* tidak menerima data maka pada saat *counter* mencapai 70 maka *buzzer* akan berbunyi sebagai tanda bahwa *device* tidak dapat berkomunikasi atau hilang. Apabila *counter* belum mencapai 70 maka mikrokontroler akan kembali mengatur RFM12 agar bersiap menerima data.

Setelah menerima data maka mikrokontroler akan memeriksa apakah data tersebut memiliki perintah untuk menyalakan *buzzer* pada *slave* jika tidak maka mikrokontroler akan mengatur port pemicu *buzzer* agar aktif. Apabila perintah

tidak sesuai maka mikrokontroler akan memeriksa apakah data yang diterima memiliki perintah untuk mematikan *buzzer* atau tidak. Apabila data yang diterima memiliki perintah untuk mematikan *buzzer* maka mikrokontroler akan mengatur agar port pemicu *buzzer* tidak diaktifkan dan kemudian memberikan balasan kepada *master*. Apabila data yang diterima tidak sesuai maka mikrokontroler akan memeriksa lagi apakah data yang diterima memiliki perintah untuk mencari *slave* apabila data sesuai maka mikrokontroler akan mengatur agar *buzzer* langsung berbunyi dan kemudian akan member balasan ke bagian *master*. Apabila data yang tidak memiliki perintah yang sesuai maka mikrokontroler akan memeriksa nilai *counter* apakah telah mencapai 70, apabila *counter* masih belum mencapai 70 maka mikrokontroler akan kembali mengatur RFM12 agar bersiap menerima data dari *master*. Apabila *counter* telah mencapai 70 maka mikrokontroler akan membunyikan *buzzer* sebagai tanda *device* tidak dapat berkomunikasi atau belum menerima data yang sesuai.

III.5 Menu-menu pada master untuk mengatur slave.

Pada *master* terdapat menu-menu yang digunakan untuk mengatur *slave* misalnya fungsi untuk menghidupkan *slave*. Fungsi ini bisa diakses dengan menekan tombol tau *push button* yang tersedia. Terdapat 3 jenis tombol yang tersedia yaitu tombol “Back”, “Select”, dan “Enter” untuk lebih jelasnya bagan menu dapat dilihat pada gambar 13.



Gambar 13. Gambar bagan menu pada *master* untuk mengontrol *slave*

Berikut ini penjelasan singkat mengenai *flowchar* gambar 13.

Apabila tombol *enter* ditekan, saat pendeteksian tombol *enter* pada *Flowchart master*, maka akan masuk ke menu-menu untuk mengendalikan *slave*. Untuk mengoperasikan menu *slave* tersebut terdapat 3 buah tombol navigasi yaitu : “Enter”, “Select”, dan “Back”. Tombol “Enter” digunakan untuk masuk kedalam menu atau menjalankan menu yang telah dipilih. Tombol “select” digunakan untuk navigasi menu seperti pada gambar 3.13. Tombol “Back” digunakan untuk kembali ke menu awal atau ke menu sebelumnya. Dalam tiap menu *slave* juga terdapat submenu. Sebagai contoh, apabila diinginkan masuk ke menu untuk mematikan *slave 2* maka saat tampilan menu *slave* tekan *enter* dan kemudian akan muncul tampilan menu *slave 1*. Kemudian tekan tombol *select*, maka akan beralih masuk ke submenu untuk *slave 2* yang terdiri dari 3 pilihan. Kemudian tekan *enter* sehingga akan muncul tampilan *slave 2 mode on*. Untuk navigasi ke sub menu *off* maka tekan tombol *select* sehingga menu beralih ke *slave 2 mode off*. Lalu tekan tombol *enter* untuk menjalankan fungsi *mode off* tersebut.

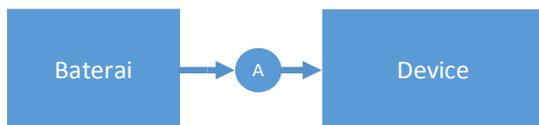
Pada awal mula program, masing-masing pin port mikrokontroler ATmega8L yang digunakan diinisialisasi baik sebagai *input*, *output*, atau sebagai jalur pembacaan ADC. Kemudian mikrokontroler akan memberikan perintah ke LCD agar menampilkan menu pada LCD. Kemudian mikrokontroler akan memulai perhitungan *counter*, dimana nilai awal dari *counter* adalah 0. Setelah itu mikrokontroler akan memerintahkan LCD agar menampilkan tampilan-tampilan pada LCD (judul skripsi, nama penulis dan lain-lain). Lalu mikrokontroler memeriksa apakah tombol *enter* ditekan atau tidak, apabila tombol *enter* ditekan maka mikrokontroler akan memberikan perintah LCD agar menampilkan menu-menu yang digunakan untuk mengontrol *slave*. Menu-menu yang ditampilkan untuk mengontrol *slave* akan dijelaskan pada subbab selanjutnya. Apabila tombol *enter* tidak ditekan maka mikrokontroler akan memerintahkan RFM12 agar mengirimkan data identitas yang berupa 7 Bytes data yang terdiri dari 6 bytes identitas dan 1 bytes untuk perintah. Kemudian setelah mengirim data identitas mikrokontroler akan memerintahkan RFM12 agar bersiap menerima data.

Kemudian mikrokontroler akan memeriksa data balasan yang diterima yang merupakan 3 bytes yang berisi 1 data bytes pengenal, 1 bytes data berisi nilai kapasitas baterai *slave* dan 1 bytes data status *slave*. Apabila dalam proses mengirim dan menerima data terjadi kegagalan maka mikrokontroler akan memeriksa proses mengirim dan menerima ini telah dilakukan selama 5 kali, apabila proses tersebut belum dilakukan selama 5 kali maka mikrokontroler akan kembali mengulang dari menambahkan *counter*, dan apabila proses mengirim dan menerima data telah dilakukan 5 kali maka mikrokontroler akan melanjutkan dengan menampilkan status-status seperti kapasitas baterai, tegangan baterai. Setelah menampilkan status-status tersebut mikrokontroler akan mengecek apakah *counter* telah mencapai 2000 yang berarti waktu telah mencapai 1 menit. Setelah itu apabila belum mencapai 2000 maka *counter* maka mikrokontroler akan menambahkan *counter*. Apabila *counter* telah mencapai 2000 maka mikrokontroler akan membunyikan *buzzer* sebagai *warning* atau peringatan apabila terdapat *slave* yang tidak dapat berkomunikasi atau hilang. Apabila tidak ada *slave* yang hilang atau tidak dapat berkomunikasi maka mikrokontroler akan mereset *counter* menjadi 0.

IV. Hasil Penelitian

IV.1 Pengukuran Konsumsi Daya pada Slave

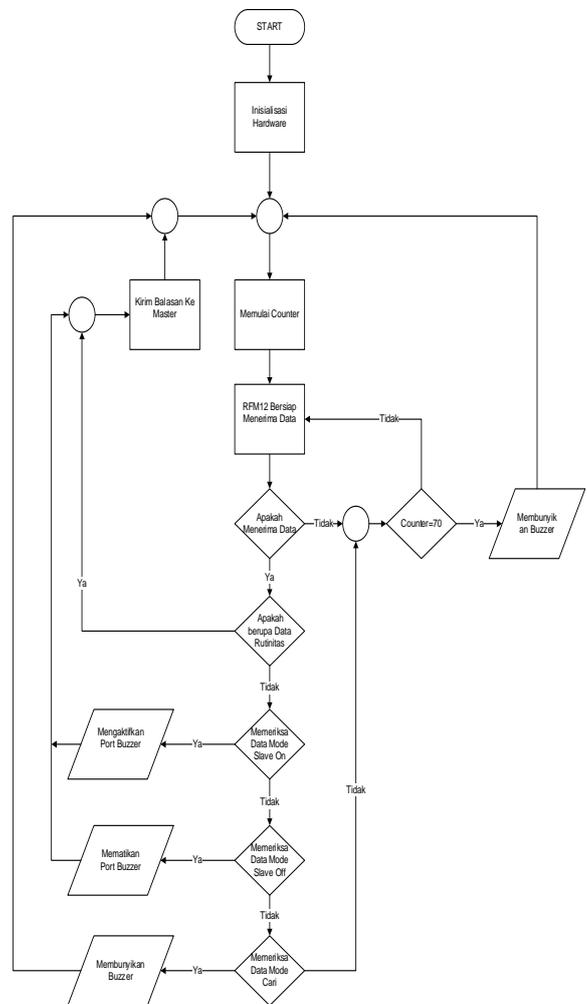
Pengukuran konsumsi daya dilakukan untuk mengetahui besar daya yang dikonsumsi oleh alat tersebut. Metode yang digunakan dapat dilihat pada Gambar 15 dimana dipasangkan ampere meter pada jalur daya baterai menuju alat. Alat ukur yang digunakan adalah digital multimeter SANWA CD800a.



Gambar 15. Metode Pengukuran Konsumsi Daya

Hasil pengukuran daya yang dikonsumsi alat secara keseluruhan dapat dilihat pada table 4.1. Pada saat mikrokontroler aktif, konsumsi arus yang dibutuhkan adalah 8,18mA sedangkan pada saat *slave* mengirimkan data balasan ke *master* arus yang dibutuhkan sebesar 22,16 mA. Sedangkan saat *slave* menyalakan *buzzer*

sebagai tanda peringatan arus yang dibutuhkan sebesar 25,58 mA. Apabila pada saat *slave* menerima data dan menghidupkan LED maka arus 10,05 mA. Untuk lebih jelas dapat dilihat pada tabel 4.



Gambar 14 Flowchart Program Slave

Berikut ini penjelasan singkat mengenai *flowchart* pada Gambar 14.

Tabel 4. Hasil pengukuran arus pada *master*

Kondisi	arus (saat tegangan 4V)
<i>Master</i> keadaan idle + LCD	51,26 mA
<i>Master Mengirim data</i> + LCD	45,98 mA
<i>Master</i> membunyikan <i>buzzer</i> + LCD	71,76 mA
Tombol navigasi ditekan + LCD	58,64 mA

Hasil pengukuran daya yang dikonsumsi *slave* secara keseluruhan dapat dilihat pada table 5. Pada saat mikrokontroler aktif, konsumsi arus yang dibutuhkan adalah 8,18mA sedangkan pada saat *slave* mengirimkan data balasan ke *master* arus yang dibutuhkan sebesar 22,16 mA. Sedangkan saat *slave* menyalakan buzzer sebagai tanda peringatan arus yang dibutuhkan sebesar 25,58 mA. Apabila pada saat *slave* menerima data dan menghidupkan LED maka arus 10,05 mA.

Tabel 5 Hasil pengukuran arus pada *slave*

Kondisi	arus saat 4V
Mikrokontroler aktif	8,18 mA
<i>slave standby</i>	14,45 mA
<i>slave</i> menerima data + LED menyala	17,116 mA
<i>slave</i> mengirimkan balasan	22,126 mA
<i>slave</i> memberikan peringatan <i>buzzer</i>	25,568 mA

IV.2 Pengujian tampilan kapasitas baterai pada *master*

Tampilan kapasitas baterai pada LCD bertujuan agar pengguna atau *user* dapat mengetahui kapasitas baterai dari *master* dan *slave*. Oleh karena itu tampilan baterai harus dibandingkan dengan alat ukur yang sesuai. Alat ukur yang digunakan sebagai pembanding adalah multimeter SANWA CD800a dimana multimeter tersebut digunakan sebagai pengukur tegangan. Untuk hasil tampilan dan hasil pengukuran dapat dilihat pada tabel 6

Tabel 6 Tabel Perbandingan antara tampilan pada LCD dan Multimeter

Kapasitas (%)	Tampilan Tegangan pada LCD (Volt)	Tegangan pada Multimeter (Volt)
0%	3.68 V	3.7 V
10%	3.74 V	3.75 V
25%	3.81 V	3.8 V
40%	3.88 V	3.85 V
50%	3.92 V	3.9 V
60%	3.96 V	3.95 V
75%	4.03 V	4 V
85%	4.07 V	4.05 V
95%	4.1 V	4.1 V
100%	4.14 V	4.15 V

IV.3 Pengujian keberhasilan fungsi on/off *slave* dari *master*

Pengujian keberhasilan mematikan atau menghidupkan *slave* dilakukan dengan cara mengaktifkan fungsi on/off *slave* yang berada pada *master*. Jarak Pengujian yang dilakukan adalah 6 meter. Pada pengujian ini *master* akan mengirimkan perintah ke *slave* agar menyalakan atau mematikan *buzzer*. Apabila *master* mendapat balasan dari *slave* maka pada LCD keluar tampilan bahwa *slave* telah merespon Untuk hasil pengujian dapat dilihat pada tabel 7

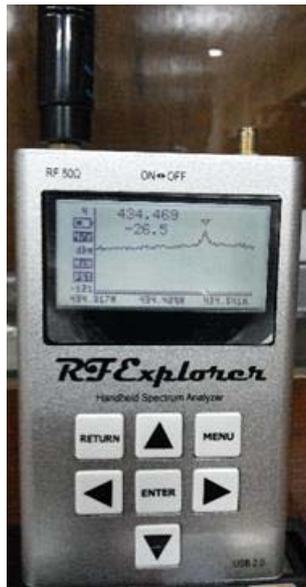
Tabel 7 Hasil Pengujian fungsi on/off

Hasil pengiriman perintah On	Hasil Pengiriman perintah Off
Berhasil	Berhasil
Berhasil	Berhasil
Gagal	Berhasil
Berhasil	Gagal
Berhasil	Berhasil
Berhasil	Gagal
Berhasil	Berhasil
Berhasil	Berhasil
Gagal	Berhasil
Gagal	Berhasil
Berhasil	Berhasil
Gagal	Gagal

IV.4 Pengujian Daya Pancar RFM12

Pengujian Jarak pancar RFM12 diukur oleh *spectrum analyzer.spectrum analyzer* yang digunakan dalam pengukuran adalah *RF Explorer Handled Spectrum Analyzer*. Bentuk fisik alat ukur dapat dilihat pada gambar 16

Pengukuran dilakukan secara *line of sight* yang dilakukan di lapangan upacara Universitas Katolik Widya Mandala Kalijudan, dan didalam ruangan lab.mikroprocessor. Untuk pengukuran pada lab mikroprocessor dilakukan pada titik-titik seperti pada gambar 17



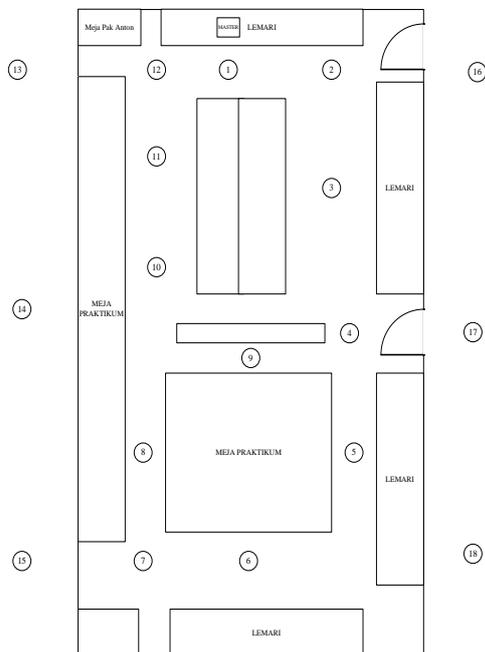
Gambar 16 RF Explorer Handled Spectrum Analyzer

5	434.467	-76,0	Berhasil
6	434.561	-75,5	Berhasil
7	434.557	-80,0	Berhasil
8	434.755	-77,0	Berhasil
9	434.467	-79.5	Berhasil
10	434.791	-71.5	Berhasil
11	434.771	-67	Berhasil
12	434.713	-66.5	Berhasil
13	434.47	-82.5	Gagal
14	434.469	-90.5	Gagal
15	434.469	-85,0	Gagal
16	434.469	-69,5	Berhasil
17	434.47	-76,0	Berhasil
18	434.47	-78,0	Gagal

Hasil Pengukuran secara *line of sight* di lapangan upacara Universitas katolik Widya Mandala Kalijudan dapat dilihat pada tabel 4.6. Metode yang dilakukan hamper sama dengan pengukuran pada ruangan yaitu dengan mengatur agar *master* mengirimkan data secara terus menerus.

Tabel 4.6 Hasil Pengukuran secara *line of sight*

Jarak	Frekuensi (MHz)	Power (dBm)	Komunikasi dengan Slave
1m	434.553	-31	berhasil
2m	434.535	-40.5	berhasil
3m	434.521	-45	berhasil
4m	434.413	-49.5	berhasil
5m	434.561	-53.5	berhasil
6m	434.623	-59	berhasil
7m	434.623	-61.5	berhasil
8m	434.795	-67.5	berhasil
9m	434.895	-71.5	berhasil
10m	434.647	-75	berhasil
11m	434.468	-78.5	berhasil
12m	434.229	-81	gagal
13m	434.303	-83.5	gagal



Gambar.17 Lokasi Pengambilan Sampel Pengukuran.

Tabel 4.5 Hasil Pengukuran Pada Ruangan Mikroprocessor

No.	Frekuensi (MHz)	Power (dBm)	Komunikasi dengan Slave
1	434.469	-54,5	Berhasil
2	434.469	-65,0	Berhasil
3	434.469	-64,0	Berhasil
4	434.469	-71,0	Berhasil

Kesimpulan

Dari hasil perancangan, pembuatan, pengujian, dan pengukuran yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan menggunakan data sebanyak 6 bytes identitas dan 1 bytes perintah, komunikasi antara *master* dan *slave* dapat berjalan dengan baik.
2. Rangkaian *battery charger* yang menggunakan IC HT3786D bekerja dengan baik sehingga mampu melakukan proses *recharge* baterai dan dapat mendeteksi level kapasitas baterai.
3. Berdasarkan Pengujian alat untuk komunikasi antara *master* dan *slave* dapat berjalan dengan baik, namun keterbatasan sistem terletak pada kemampuan komunikasi *master* dan *slave* jika terhalang tembok, jendela, dan jarak yang mengakibatkan daya sinyal mengalami peredaman diatas 79 dBm.

Daftar pustaka

- [1]. ---, RFM12 Universal ISM Band FSK Transceiver, HOPERF, China, 2006.
- [2].Atmel, ATMega8L datasheet, Atmel Corporation, USA, 2006.
- [3].---, FT232R USB UART IC, Future Technology Devices International Limited, Maret 2012.