JCONES Vol. 5, No. 1 (2016) 43-60

Journal of Control and Network Systems

Situs Jurnal : http://jurnal.stikom.edu/index.php/jcone

PENENTUAN RUTE TERBAIK PADA WSN (*WIRELESS SENSOR NETWORK*) BERDASARKAN PARAMETER RSSI (*RECEIVED SIGNAL STRENGTH INDICATOR*)

Rizky Aderusman[1)] **Jusak**[2)] **Yosefine Triwidyastuti**[3)]
Program Studi/Jurusan Sistem Komputer
Fakultas Teknologi dan Informatika
Institut Bisnis dan Informatika Stikom Surabaya
Jl. Raya Kedung Baruk 98 Surabaya, 60298
Email: 1)r.aderusman@gmail.com, 2)jusak@stikom.edu, 3)yosefine@stikom.edu

Abstract: *Wireless sensor network (WSN) is a wireless network consisting of a collection of nodes that are scattered in a particular area. Each node has the ability to collect data and communicate with other nodes. However many studies showed that the WSN technology bears some drawbacks. One of the weakness is in terms of efficiency of energy utilization. Ineffectiveness of energy utilization in the WSN mainly caused by long and complex algorithms for route/path determination*

In this work determination of the best route is implemented by comparing parameters, So nearly value of RSSI (Received Signal Strength Indicator) for the selection of the route, by adapting mechanism in routing protocol OSPF (Open Shortest Path First). For testing the routing protocol, we used temperature data acquired from LM35 sensors. Sensors are planned and then implemented on all nodes WSN.

By implementing these schemes for routing determination, the WSN based on parameter RSSI/LOW and the exercise of routing protocol adopted by OSPF works. So the testing will be carried out by 30 times, 15 times has been tried at indoor and another 15 has been tried at outdoor. Within result of 27 times can find the best route based by lower RSSI parameter at that time. That research had been tried in distance between 1 – 40 meters at indoor condition and distance between 1 - 100meters at outdoor condition. But another 3 trials can't find the best route, 1 trial at indoor condition and 2 trials at outdoor condition

Keywords: *WSN, Node, Best Determination These, Parameter, RSSI*

Wireless sensor network (WSN) adalah suatu jaringan nirkabel yang terdiri dari kumpulan *node* yang tersebar di suatu area tertentu. WSN dapat terdiri dari ratusan hingga ribuan *node* yang terpasang secara penuh dalam area geografis yang luas. Tiap *node* memiliki kemampuan untuk mengumpulkan data dan berkomunikasi dengan *node* lainnya, yang kemudian menentukan jalurnya pada tiap-tiap *node*nya melalui transmisi radio secara intensif. WSN menggunakan *sensor* yang bisa dimanfaatkan untuk memonitor fisik atau kondisi suatu lingkungan sekitar, seperti suhu, suara, getaran, gelombang elektromagnetik, tekanan, gerakan, dll. Akan tetapi dengan kemampuan WSN yang sudah hebat tersebut, WSN masih memiliki beberapa kelemahan salahsatunya yaitu, dalam hal penggunaan energi yang tidak efektif (AL-Karaki & Kamal, 2004).

Ketidakefektifan pemanfaatan energi pada WSN disebabkan oleh beberapa faktor. Salahsatunya adalah penggunaan rute yang tidak tepat dalam hal pengiriman atau penerimaan datanya, yang dimaksud penggunaan rute yang tidak tepat yaitu suatu proses pengiriman dan penerimaan data yang dilakukan secara umum tanpa menerapkan beberapa hal yang lain, dan mungkin salahsatunya adalah dalam hal pemilihan kekuatan sinyal pada setiap *node*.

Dalam pemilihan kekuatan sinyal, sinyal yang terkuatlah yang akan dijadikan acuan untuk pemilihan rute terbaik. Dengan kata lain pada saat proses pengiriman atau penerimaan data yang

akan menggunakan rute terbaik, secara tidak langsung akan berdampak pada penghematan energi. Oleh karena itu kekuatan sinyal pada tiap *node* dalam pengiriman atau penerimaan data sangatlah berpengaruh.

Berdasarkan permasalahan tersebut, diperlukan suatu penelitian mengenai penentuan rute terbaik pada saat pengiriman atau penerimaan data. Pada penelitian ini penulis akan menggunakan parameter RSSI (*Received Signal Strength Indicator*) yang pada saat itu berada kondisi LOW. Dimana sinyal yang mendekati nol yang digunakan untuk mempertimbangkan kekuatan sinyal pada masing – masing *node* yang digunakan. Jadi dengan didapatkannya rute terbaik pada saat pengiriman maupun penerimaan data, maka diharapkan penggunaan energi pada WSN dapat digunakan seefisien mungkin.

METODE PENELITIAN

Metode penelitian yang digunakan pada tugas akhir ini menggunakan rekayasa perangkat lunak pada perangkat *node* WSN dengan beberapa tahapan-tahapan yang dijelaskan pada Gambar 1

Gambar 1 Metode Penelitian

1. Penentuan Kriteria Sistem

Pada tahap ini penulis menggunakan RSSI sebagai parameternya. Selanjutnya untuk menemukan rute terbaik berdasarkan parameter RSSI, nilai RSSI akan dibandingkan terlebih dahulu lalu dipilih nilai RSSI yang pada saat itu berada pada kondisi *LOW* (mendekati nol).

2. Desain Perangkat dan algoritma

Pada tahap ini akan menentukan algoritma perbandingan rute. Jadi calon rute terbaik akan dipilih dan dibandingkan dengan calon rute yang lain melalui proses perbandingan RSSI yang pada saat itu berada pada kondisi *LOW*.

3. Perancangan Sistem

Pada tahap ini akan membuat perancangan sistem dimana didalamnya berisi berisi proses cara kerja sistem yang meliputi proses sebelum *convergen* dan sesudah *convergen*. Kemudian proses perancangan *hardware* maupun perancangan *software.* Untuk perancangan *hardware* meliputi pembuatan rangkaian *push button*, desain topologi, sampai menentukan *node* WSN, dll. Untuk perancangan dari segi *software* meliputi pembuatan *flowchart*, setting xbee pada *software* XCTU, sampai *compile* program arduino pada *software* arduino IDE.

4. Pengujian Perangkat

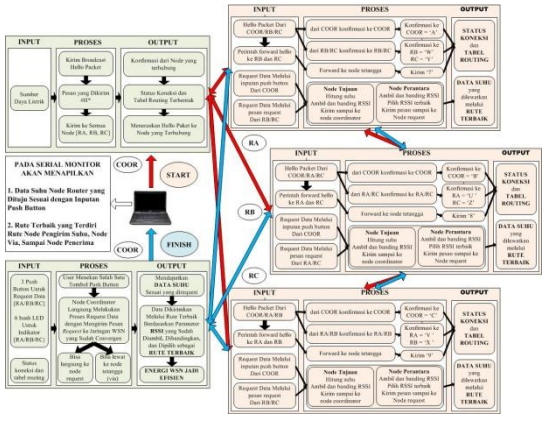
Pada tahap ini akan membahas prosedur pengujian mulai dari pengujian rute terbaik, pengujian tempat, sampai pengujian jarak, dll.

5. Kesimpulan

Setelah semua tahapan-tahapan pada metode penelitian sudah diterapkan, maka diperoleh kesimputan. Kesimpulan yang didapatkan merupakan kesimpulan yang terjadi pada saat melakukan pengujian.

1. Blok Diagram

Pada Gambar 2 menjelaskan blok diagram sistem



Gambar 2 Blok diagram sistem

Untuk lebih jelasnya Gambar 2 akan di tampilkan pada lampiran 1. Dari blok diagram sistem diatas terbagi atas cara kerja 4 *node* yaitu node *coordinator*, *node router* A (RA), *node router* B (RB), dan *node router* C (RC). Pada masing-masing *node* memiliki 2 proses, yaitu proses pengenalan dan proses permintaan data. Untuk tiap prosesnya memiliki 3 bagian yaitu bagian *input*, bagian proses, dan bagian *output*. Berikut adalah penjelasan ketiga bagian tersebut.

1 Proses pengenalan *node* (sebelum convergen)

a) Bagian *input*

Pada bagian *input* pertama-tama *node* RA, RB, dan RC tidak akan melakukan proses apapun apabila *node coordinator* tidak mengirimkan pesan ke salah satu *node* ataupun *node coordinator* dalam hal ini mati (tidak ada sumber daya). Ketika *node coordinator* memiliki sumber daya maka dengan sendirinya *node* RA, RB, dan RC akan siap-siap menerima paket dari *node coordinator*. Paket tersebut dinamakan sebagai *hello packet*.

b) Bagian proses

Pada bagian proses, dimana *node coordinator* sudah aktif dan akan mengirimkan *hello packet* secara *broadcast* ke *node* lain yang pada saat itu aktif. Dalam hal ini *node* yang aktif adalah *node* RA, RB, dan RC. Pesan *hello* yang dikirimkan *node coordinator* ke *node* RA, RB, dan RC adalah **#H***. apabila RA, RB, dan RC sudah mendapatkan paket tadi kini RA, RB dan RC wajib mengkonfirmasi ke *node coordinator* kalau paket yang dikirimkan oleh *node coordinator* sudah terkirim.

Seperti yang terlihat pada tabel 1 di bawah ini.

Tabel 1 Kode Pesan *Broadcast Hello Packet*

Node		**Status Pesan**			
Dari	Ke	Hello Packet	Konfirmasi	*Forward*	Status koneksi
Coordinator	RA	#H*	A	@1	C.RA1
	RB	#H*	B	@2	C.RB1
	RC	#H*	C	@3	C.RC1
RA	RB	7	U	-	A.B1
	RC	7	V		A.C1
RB	RA	8	W	-	B.A1
	RC	8	X		B.C1
RC	RA	9	Y	-	C.A1
	RB	9	Z		C.B1

c) Bagian *output*

Pada bagian *output*, kini semua *node* memiliki informasi status koneksi yang meberitahukan *node* mana saja yang terhubung dengannya. Sehingga node sudah saling mengenal dengan *node* tetangganya bisa juga dikatakan hubungan antar *node* menjadi *convergen*. Jadi apabila nanti ada *node* yang ingin berkomunikasi dengan *node* lain, *node* tersebut akan melihat *table routing* terlebih dahulu sebelum melakukan proses komunikasi ke *node* yang lain. Pada blok diagram diatas apabila semua *node* melakukan proses *broadcast*, terima, maupun *forward hello packet* dengan baik maka dapat dikatakan bahwa hubungan antara keempat *node* yakni *node coordinator*, *node* RA, *node* RB, dan *node* RC saling mengenal (*convergen*) dengan informasi *node* tetangga yang dimiliki.

2 Proses permintaan data (sesudah *convergen*)

Setelah melakukan proses diatas, kini hubungan antar *node* 1 dengan node yang lain bisa dikatakan sudah *convergen*. Jadi langkah selanjutnya adalah user bisa meminta data suhu pada *node* yang diinginkan, berikut adalah 3 bagian pada proses permintaan data.

a) Bagian *input*

Pada bagian *input*, terdapat 3 buah *push button* dan 6 buah LED yang berada pada *node coordinator*. Rangkaian *push button* dan LED yang berada di *node coordinator* ini tugasnya hanyalah membantu *node coordinator* dalam proses pertukaran informasi, baik itu informasi tentang *routing* maupun informasi tentang permintaan atau penerimaan data. Sedangkan pada *node* RA, RB, dan RC *input*nya hanyalah kode *request* data, bisa kode dari *node coordinator*nya langsung bisa juga kode dari *node* tetangga. Biasanya hal ini terjadi apabila *node request* tidak mengenal dengan *node coordinator* (tidak konek). Sebelum masuk proses ini, sebelumnya semua *node* melakukan proses *routing* terlebih dahulu dengan mengirimkan beberapa paket termasuk *hello packet* sampai akhirnya jaringan menjadi *convergen*. Sehingga *node* yang berhasil membangun hubungan dengan tetangganya maka *indicator* pada *node* tersebut akan menyala, begitu juga yang terjadi pada *node coordinator*.

Apabila semua *node* dalam hal ini *node* RA, RB, dan RC berhasil membangun hubungan dengan *node coordinator* maka *indicator* LED untuk hubungan *node coordinator* dengan *node* RA, RB, dan RC akan menyala. Apabila yang berhasil berhubungan hanya beberapa *node*, maka bisa dikatakan hanya beberapa *node* itu saja yang

indikatornya menyala pada *node coordinator*. Oleh karena itu pada penelitian ini user bisa meminta data dengan 2 cara, pertama meminta data pada *node request* secara langsung (terjadi apabila hubungan antara *node coordinator* dan *node request connect*). Dan yang kedua meminta data pada *node request* secara tidak langsung dengan meminta bantuan *node via* (terjadi apabila hubungan antara *node coordinator* dan *node* request tidak *connect* sedangkan *node via connect* dengan *node coordinator* dan juga *node request*). Permintaan data sendiri dilakukan dengan cara menekan tombol *push button* yang ada pada *node coordinator*.

b) Bagian Proses

Pada bagian ini proses permintaan data bisa dilakukan dengan melihat 2 kondisi. Kondisi pertama kondisi hubungan *node coordinator* dengan *node request* adalah *connected*. Dan kondisi yang kedua kondisi *node coordinator* dengan *node request disconnected*. Pada penelitian ini penulis mengasumsikan *node* RA, RB dan RC sudah *convergen* dengan *node coordinator* dan berada pada kondisi *connected*. Jadi user bisa menekan salah satu *push button* yang tersedia, jika ingin mengetahui data suhu dari RA *user* bisa menekan tombol RA, hal yang sama juga terjadi pada RB dan RC. Misalkan yang ditekan *user* adalah tombol RC maka tugas *coordinator* selanjutnya adalah mengirimkan kode permintaan data kepada *node* RC, kemudian *node* coordinator menunggu hasilnya. Untuk kode permintaan data setiap *node* berbeda-beda hal ini dilakukan untuk mengurangi adanya *packet* yang bertabrakan dan juga *packet loss*. seperti yang terlihat pada tabel 2 di bawah ini.

Tabel 2 Kode Permintaan Data Suhu

Node		**Kode Pesan**	
Dari	**Ke**	***Connected***	***Disconnected***
Coordinator	RA	*conn.a#	*diss.a#
	RB	*conn.b#	*diss.b#
	RC	*conn.c#	*diss.c#

Apabila kode sudah dikirimkan kini tugas *node* yang dimintai data suhu untuk menjalankan tugasnya. Dalam hal ini *node* RC adalah menghitung data suhu, menghitung parameter RSSI semua *node* yang sudah dikenal, kemudian membandingan parameter RSSI, dan yang terakhir membungkus data dan parameter menjadi 1 *field* data.

Perlu diingat parameter yang digunakan adalah parameter RSSI suatu parameter untuk mengukur sinyal *node* mana yang terkuat, dalam penelitian ini penulis menggunakan pin 6 pada xbee untuk perhitungan RSSI. Parameter inilah yang nantinya akan dijadikan sebagai penentuan rute terbaik, rute dimana data akan dilewatkan dari *source node* ke *destination node* (proses *request*) kemudian dari *destination node* ke *source node* (proses *reply*). Apabila pada saat pemilihan rute terdapat beberapa calon rute yang dihasilkan maka disinilah tugas parameter RSSI untuk memilih rute yang benar-benar terbaik dengan cara membandingkan beberapa calon rute. Seperti yang terlihat pada tabel 3 di bawah ini.

Tabel 3 Perbandingan Calon Rute Terbaik

Node		**Koneksi**	**Perbandingan RSSI**	**Kirim Ke**
Source	***Destination***			
C	RC	Hanya C	Tidak ada perbandingan	C
		Hanya RA	Tidak ada perbandingan	RA
		Hanya RB	Tidak ada perbandingan	RB
		C & RA	C > RA	C
		C & RA	C < RA	RA
		C & RB	C > RB	C
		C & RB	C < RB	RB
		RA & RB	RA > RB	RA
		RA & RB	RA < RB	RB
		C & RA & RB	C = RA = RB	Pertama Koneksi
C	RA	Hanya C	Tidak ada perbandingan	C
		Hanya RB	Tidak ada perbandingan	RB
		Hanya RC	Tidak ada perbandingan	RC
		C & RB	C > RB	RB
		C & RB	C < RB	RB
		C & RC	C > RC	RC
		C & RC	C < RC	RC
		RB & RC	RB > RC	RB
		RB & RC	RB < RC	RC
		C & RA & RB	C = RA = RB	Pertama Koneksi
C	RB	Hanya C	Tidak ada perbandingan	C

	Hanya RA	Tidak ada perbandingan	RA
	Hanya RC	Tidak ada perbandingan	RC
	C & RA	C > RA	C
	C & RA	C < RA	RA
	C & RC	C > RC	C
	C & RC	C < RC	RC
	RA & RC	RA > RC	RA
	RA & RC	RA < RC	RC
	C & RA & RB	C = RA = RB	Pertama Koneksi

c) Bagian *output*

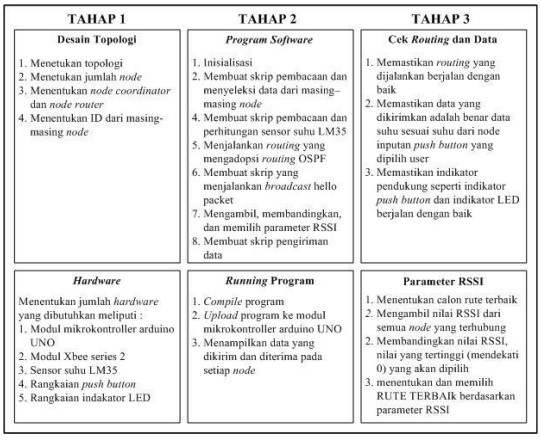
Pada bagian *output* dari sisi *node* RC, kini *node* RC selaku *node* yang di*request* oleh *user* akan mengirimkan data suhu ke *node coordinator*. Kemudian data suhu akan dikirimkan melewati rute terbaik berdasarkan parameter RSSI yang sudah diambil, dibandingkan, dan juga ditetapkan sebagai rute terbaik.

Selanjutnya dari sisi *node coordinator* yaitu selaku *node request*, akan menampilkan beberapa informasi antara lain :

- Data suhu *node router* yang dituju sesuai dengan inputan *push button*
- Rute terbaik yang terdiri dari rute *node* pengirim suhu (d*estination node*), *node via*, dan juga *node* penerima (*source node*).

2. Perancangan Sistem

Gambar 3 menjelaskan perancangan sistem yang dibuat pada judul tugas akhir "Penentuan Rute Terbaik Pada WSN (*Wireless Sensor Network*) Berdasarkan Parameter RSSI (*Received Signal Strength Indicator*)".



Gambar 3 Bagan Proses Perancangan Sistem

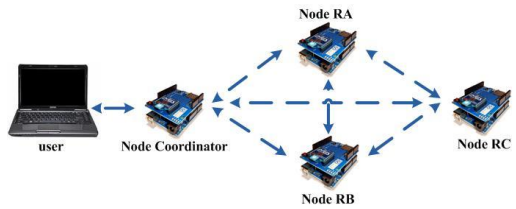
Pada bagan ini. proses sistem dibagi ke dalam tiga tahap. Tahap pertama yang dilakukan adalah menyiapkan *hardware* yang dibutuhkan dalam sistem serta membuat desain topologinya. Selanjutnya tahap kedua yang dilakukan adalah membuat skrip pada *software* arduino IDE dan meng-*compile* program, yang selanjutnya di *upload* ke dalam modul mikrokontroller arduino uno. Kemudian yang terakhir tahap ketiga yang dilakukan adalah memastikan *indicator* LED dan *push button* berjalan dengan baik, kemudian memastikan juga kevalidan data suhu yang akan dikirimkan, dan yang terakhir memastikan proses berjalannya data sesuai dengan *routing* yang ditentukan. Selanjutnya menentukan calon rute yang kemungkinan dilewati data, kemudian mengambil nilai RSSI dan membandingkannya. RSSI yang terbesar (mendekati 0) yang akan dipilih sebagai rute terbaik.

3. *Desain Topologi*

Pada perancangan sistem ini menggunakan model topologi *point to multipoint*. Model topologi ini memungkinkan *node* untuk berkomunikasi dengan 2 *node* atau lebih asallkan tetap dalam 1 jaringan. Pada saat pengujian nanti akan menggunkan 2 macam topologi.

1. Topologi A

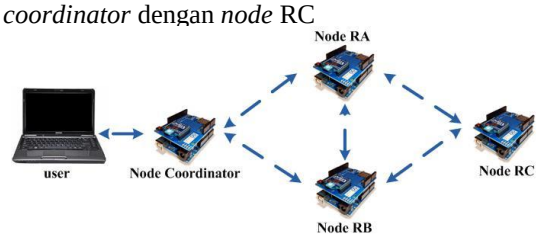
Percobaan pertama menggunakan topologi A seperti Gambar 4. Karakteristik topologi ini yaitu semua *node* saling terhubung.



Gambar 4 Topologi A

2. Topologi B

Percobaan kedua menggunakan topologi B seperti Gambar 5. Karakteristik topologi ini yaitu semua *node* saling terhubung kecuali *node coordinator* dengan *node* RC

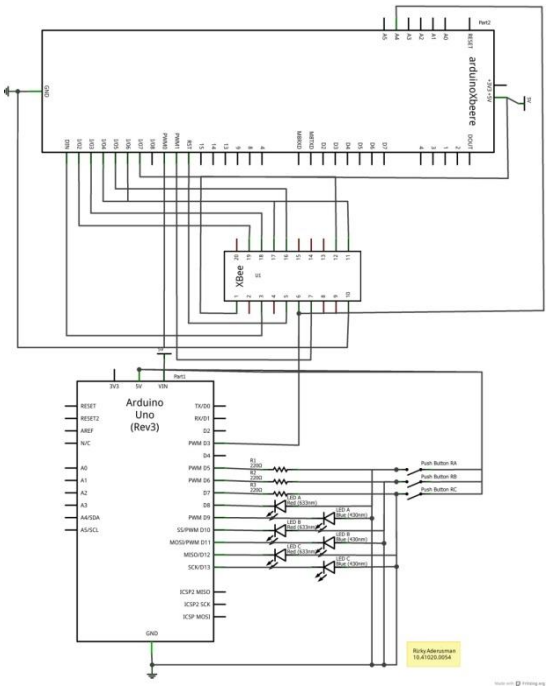


Gambar 5 Topologi B

4. *Hardware*

Pada penelitian ini sebelum membuat perangkat keras penulis membuat skematik (rangkaian simulasi) terlebih dahulu yang bertujuan untuk membantu agar nantiya tidak ada kesalahan pada rangkaian perangkat keras yang sesungguhnya. Untuk *node coordinator* bisa dilihat pada Gambar 6 dan untuk *node router* pada Gambar 7.

1. *Node coordinator*

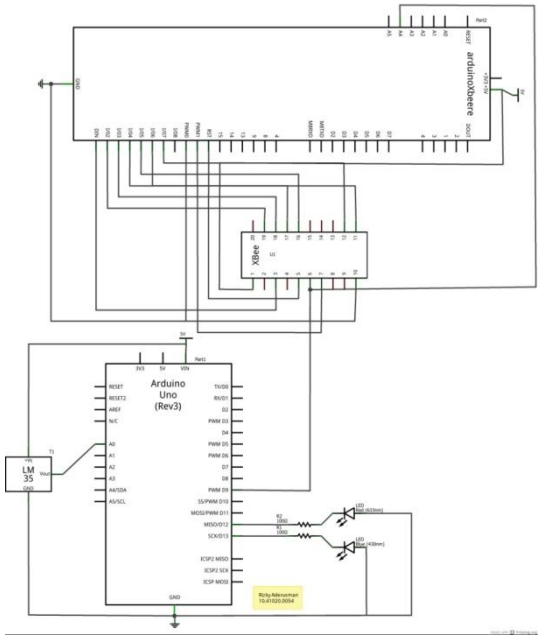


Gambar 6 Skematik *node coordinator*

Keterangan :

- Pin 3 PWM0/RSSI mengambil nilai RSSI
- Pin 5 pbA, mengaktifkan *push button* RA
- Pin 6 pbB, mengaktifkan *push button* RB
- Pin 7 pbC, mengaktifkan *push button* RC
- Pin 8 ledA_merah, menyalakan led A merah
- Pin 9 ledA_biru, menyalakan led A biru
- Pin 10 ledB_merah menyalakan led B merah
- Pin 11 ledB_biru menyalakan led B biru
- Pin 12 ledC_merah menyalakan led C merah
- Pin 13 ledC_biru menyalakan led C biru
- VCC sumber tegangan (5v – 9v)
- GND *ground*

2. *Node router* (RA, RB, RC)



Gambar 7 Skematik *node router* (RA, RB, RC)

Keterangan :

- Pin 9 PWM0/RSSI, mengambil nilai RSSI
- Pin 12 mengaktifkan led warna merah
- Pin 13 mengaktifkan led warna biru
- Pin A1 mengambil data suhu sensor LM35
- VCC sumber tegangan (5v – 9v)
- GND *ground*

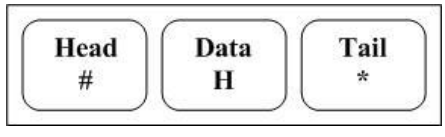
5. *Format Penulisan Pesan*

Setiap data yang dikirimkan dari *node* satu ke *node* yang lain pastilah memiliki pesan-pesan didalamnya yang dibungkus menjadi satu kesatuan pesan. Dalam penelitian ini format penulisan pesan dibagi menjadi 4 bagian yaitu :

a) Format penulisan *broadcast hello packet*
b) Format penulisan data suhu
c) Format penulisan parameter RSSI
d) Format penulisan pesan *routing*

Berikut adalah penjelasan dari masing-masing format penulisan pesan.

a) Format penulisan *broadcast hello packet*



Gambar 8 Format *Broadcast Hello Packet*

Keterangan:
Head : *Header* adalah pesan pembuka, untuk proses *broadcast hello packet header* diwakili dengan karakter '#'
Data : Data adalah kode pesan yang dikirim, untuk proses *broadcast hello packet* data diwakili dengan karakter 'H'
Tail : *Tailer* adalah pesan penutup, untuk proses *broadcast hello packet tailer* diwakili dengan karakter '*'

b) Format penulisan data suhu

Gambar 9 Format Penulisan data suhu

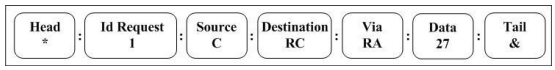
Keterangan:
Head : *Header* adalah pesan pembuka, untuk format pesan data suhu *header* diwakili dengan karakter '@'
Suhu : Suhu adalah nilai yang diperoleh dengan sensor LM35, nilai ini bisa berubah-ubah tergantung kondisi lingkungan. Sebagai contoh untuk nilai suhu diwakili dengan nilai '29'
Tail : *Tailer* adalah pesan penutup, untuk format pesan data suhu *tailer* diwakili dengan karakter '%'

c) Format penulisan parameter RSSI

Gambar 10 Format Penulisan Parameter RSSI

Keterangan:
Head : *Header* adalah pesan pembuka, untuk format pesan parameter RSSI *header* diwakili dengan karakter '@'
RSSI : RSSI adalah parameter yang bisa mengambil kekuatan sinyal. Pada penelitian ini sinyal yang diambil adalah sinyal dengan kondisi *LOW*. Sebagai contoh untuk nilai parameter RSSI diwakili dengan angka '3'
Tail : *Tailer* adalah pesan penutup, untuk format pesan parameter RSSI *tailer* diwakili dengan karakter '#'

d) Format penulisan pesan *routing*



Gambar 11 Format Penulisan Pesan *Routing*

Keterangan:
Head : *Header* adalah pesan pembuka, untuk format pesan *routing header* diwakili dengan karakter '*'
Id Request : *Id* R*equest* yang mengartikan permintaan data yang ke berapa sebagai contoh permintaan yang pertama, diwakili dengan nilai '1'
Source : *Source* sebagai *node* awal (peminta data suhu) yang pada penelitian ini hanya diwakili oleh 1 *node* saja yakni *node coordinator*, untuk *source* diwakili nama *node* misalkan 'C'
Destination: *Destination* sebagai *node* tujuan, dimana *node* tersebut bergantung pada posisi *push button* yang di*request* oleh user. Sebagai contoh *destination* adalah *node* RC jadi diwakili dengan karakter 'RC'
Via : Via adalah node penghubung antara *source node* dan *destination* node. Sebagai contoh via melewati node RA jadi untuk pesannya diwakili dengan karakter 'RA'
Data : Data adalah nilai yang diperoleh dengan sensor LM35, nilai ini bisa berubah-ubah tergantung kondisi lingkungan node yang pada saat itu di*request*. Sebagai contoh untuk data diwakili dengan nilai '27'
Tail : *Tailer* adalah pesan penutup, untuk format pesan *routing tailer* diwakili dengan karakter '&'.

Untuk menghindari penumpukkan data pada saat proses *request* atau *reply* maka dibuatlah pemisah pesan yang diwakili dengan karakter ':'. Jadi dengan begitu kemungkinan salah masuk data akan lebih kecil.

HASIL DAN PEMBAHASAN

Pengujian Sistem

Pada penelitian ini pengujian yang dilakukan merupakan pengujian terhadap perangkat keras (*hardware*) dan juga perangkat lunak (*software*) secara keseluruhan yang bertujuan untuk mengetahui apakah sistem yang dibuat sudah berjalan sesuai dengan apa yang diharapkan. Dalam hal kategori tempat pengujian dilakukan di 2 kategori tempat, yakni tempat tertutup (*indoor*) dan tempat terbuka (*outdoor*). Untuk tiap tempatnya pengujian yang dilakukan yakni meliputi pengujian proses *broadcast* hello *packet*,

pengujian proses *forward* hello *packet*, pengujian rute, pengujian jarak.

1. Kategori Tempat

Dalam hal kategori tempat pengujian dilakukan di 2 kategori tempat, yakni tempat tertutup (*indoor*) dan tempat terbuka (*outdoor*)

a) Tempat Tertutup (*indoor*)

Kategori tempat yang pertama yang akan diuji adalah tempat tertutup (*indoor*). Tempat yang dipilih berada pada suatu ruangan didalam area bergedung dengan luas kurang lebih 7 meter tiap ruangannya. Seperti yang terlihat pada Gambar 12

Gambar 12 Tempat tertutup (*indoor*)

b) Tempat Terbuka (*outdoor*)

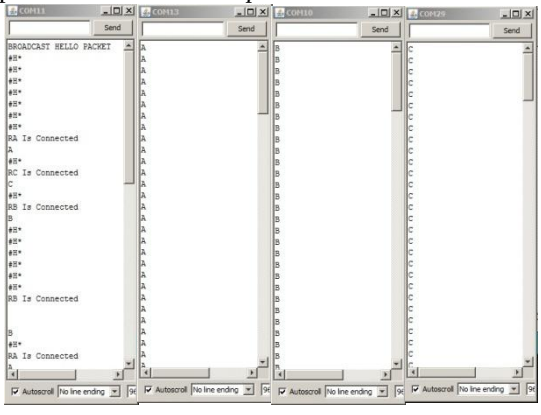
Kategori tempat yang kedua yang akan diuji adalah tempat terbuka (*outdoor*). Tempat yang dipilih berada pada suatu lahan kosong (lapangan) dengan luas area kurang lebih 150 meter. Seperti yang terlihat pada Gambar 13

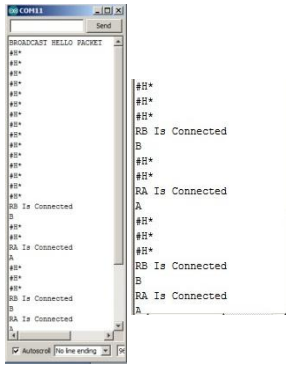
Gambar 13 Tempat terbuka (*outdoor*)

2. Hasil Pengujian

a) Proses *broadcast* hello *packet*

pada Gambar 19 menjelaskan hasil pengujian proses *broadcast* hello *packet*



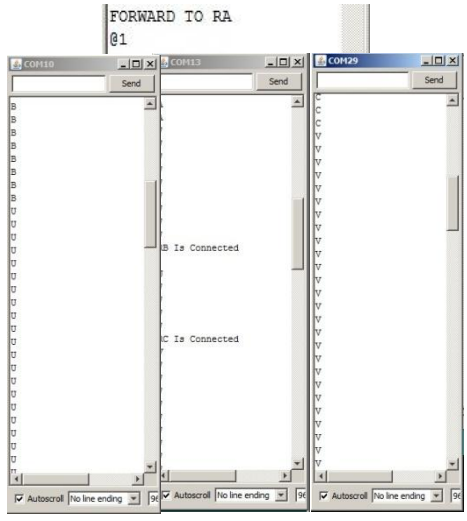


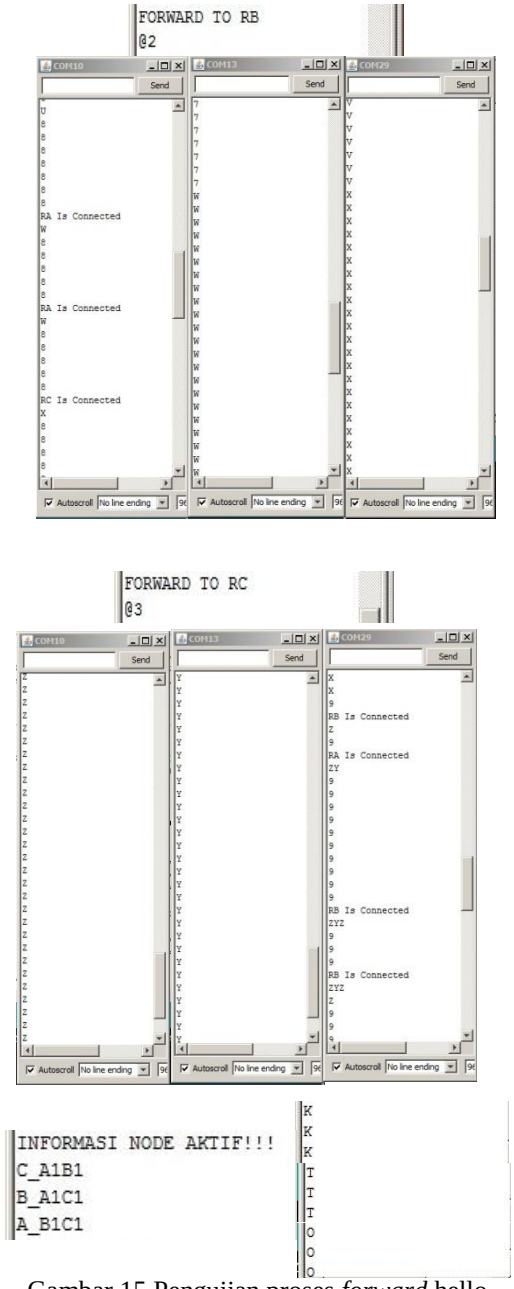
Gambar 14 Pengujian proses *broadcast* hello *packet* (*indoor*, atas) dan (*outdoor*, bawah)

Proses *broadcast* hello *packet* dilakukan oleh *node coordinator* dengan mengirimkan pesan **#H*** dan ditampilkan pada *serial monitor* yang ada pada *software* arduino IDE. Mula-mula hello *packet* akan dikirimkan secara *broadcast* ke semua *node* yang terhubung dengan *node coordinator*, apabila *node* RA menerima pesan hello maka *node* RA akan mengirimkan karakter 'A' sebagai pesan konfirmasi. Apabila *node* RB menerima pesan hello maka *node* RB akan mengirimkan karakter 'B' sebagai pesan konfirmasi. Dan apabila *node* RC juga menerima pesan hello maka *node* RC akan mengirimkan karakter 'C' sebagai pesan konfirmasi. Lalu pada waktu yang bersamaan *node coordinator* akan melihat pesan dari *node* mana saja yang masuk, sehingga mengetahui *node coordinator connected* dengan *node* mana saja.

b) Proses *forward hello packet*

pada Gambar 15 menjelaskan hasil pengujian proses *forward* hello *packet*





Gambar 15 Pengujian proses *forward* hello

Pada *node coordinator* akan mengetahui *node* mana saja yang terhubung dengan *node coordinator*. Informasi ini bisa digunakan untuk membantu apabila *node coordinator* ingin meminta data ke *node request* yang statusnya tidak terhubung dengannya, maka *node coordinator* akan meminta bantuan ke *node* jembatan, *node* yang menghubungkan antara *node coordinator* dengan *node request*.

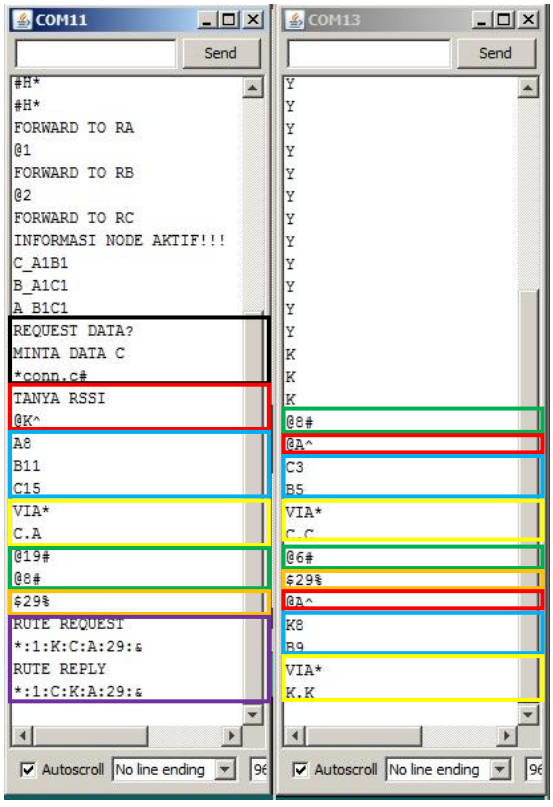
proses perintah *forward* hello *packet*, proses ini dilakukan untuk membantu agar semua *node* saling mengenal dengan *node* tetangganya masing-masing. Seperti yang terjadi pada *node* RA, *node coordinator* akan memerintahkan *node* RA untuk *forward* hello dengan cara kirim karakter '!' apabila *node* RA menerima karakter tersebut maka dengan sendirinya *node* RA langsung mem*forward* hello *packet* dengan cara kirim karakter 7. Dan apabila ada konfirmasi dari *node* yang lain tinggal mengecek saja karakter yang masuk apakah 'U' atau 'V'. Hal yang sama juga terjadi pada *node* RB dan juga *node*. *Node coordinator* akan memiliki informasi kondisi semua *node* mulai dari yang *node* yang *connected* sampai *node* yang *disconnected*, sehingga hubungan tiap *node* saling mengenal.

c) Proses pemilihan rute

Langkah pertama yakni mengaktifkan semua *node* dan pastikan sudah melakukan proses *broadcast* dan *forward* hello *packet* dengan baik. Sebagai contoh penulis akan mensimulasikan proses penentuan rute diatas yang ada pada Gambar 4.5. mula-mula *node coordinator* akan meminta data suhu pada *node* RC karena user telah menekan tombol RC. berhubung *node coordinator* sudah mengenal *node* RC jadi pesan yang dikirimkan untuk *request* RC adalah ***conn.c#,** selanjutnya *node coordinator* akan meminta data RSSI ke *node* tetangga yang menuju ke *node* RC dengan mengirimkan pesan **@K^.** langkah ini dilakukan untuk melihat ada berapa calon rute untuk menuju *node* RC. kalau sudah dikirim *node coordinator* akan menunggu balasan RSSI dari *node* yang menerima pesan tadi. jika ada data masuk *node* RA, *node* RB dan *node* RC akan melihat data yang masuk memerintahkan untuk apa. Apabila data yang masuk memerintahkan untuk menghitung nilai RSSI dari *node* RA atau *node* RB, atau *node* RC menuju ke *node coordinator*. masa *nodc* RA, RB dan *node* RC akan mengirimkan nilai RSSInya ke *node coordinator*.

Selanjutnya *node coordinator* akan membandingkan nilai RSSI yang diterima tadi kemudian dipilih nilai RSSI mana yang mendekati nol itulah yang akan dipilih. Setelah proses membandingkan nilai RSSI untuk pemilihan calon rute selesai maka nantinya akan didapatkan 1 rute terbaik dengan ketentuan nilai RSSInya lebih baik daripada calon rute yang lain. *Node coordinator* telah memilih *node* RA sebagai rute *request*nya, kemudian *node coordinator* akan meminta bantuan *node* RA untuk meneruskan proses *request* tadi dengan mengirimkan pesan **VIA*C.A**

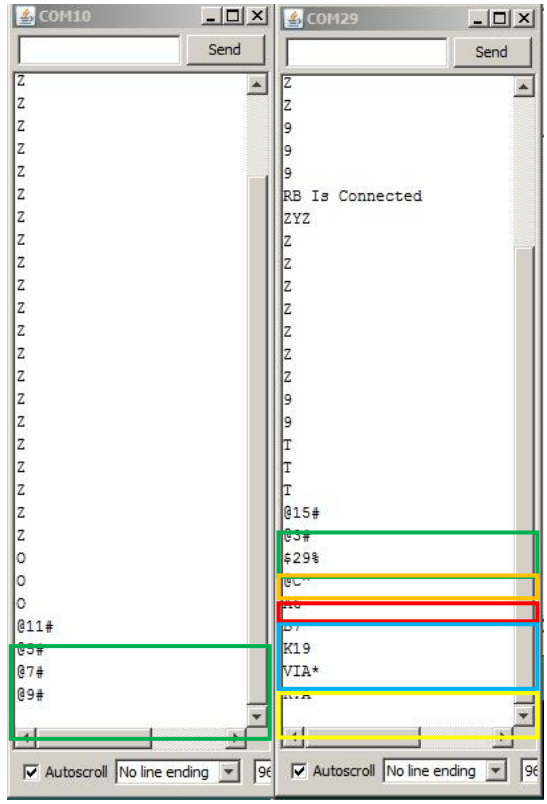
Lalu *node* RA akan mengirimkan permintaan RSSI ke *node* tetangga dalam hal ini *node* RB dan *node* RC dengan mengirimkan pesan **@A*** tadi **.** Lalu *node* RB dan *node* RC akan menghitung nilai RSSInya sekaligus mengirimkanya ke *node* RA tadi, kemudian *node* RA membandingkanya dan lalu milih *node* RC. apabila RC adalah *node request* maka kewajiban RC adalah merekam data suhu disekitarnya, dan akhirnya mendapatkan data suhu 29 °C, lalu mengirimkan *node* RC meminta nilai RSSI *node* sekitar untuk mengirimkan data suhunya. Pesan yang dikirimkan adalah ***C^**. kemudian yang terakhir melakukan proses *reply* data yang prosesnya hampir sama dengan proses *request* tadi, yakni melakukan proses minta nilai RSSI ke *node* yang terhubung, lalu menerima dan membandingkanya untuk pemilihan rutenya sampai proses pengiriman datanya tadi**.** untuk lebih jelasnya bisa melihat Gambar 21 dibawah ini yang menerangkan proses pemilhan rute



Gambar 16 Pengujian proses pemilihan rute *outdoor*

Keterangan

: *Request* data suhu
: Meminta RSSI *node* tetangga
: Menerima RSSI *node* tetangga
: Banding RSSI dan pilih rute terbaik
: Mengirim RSSI ke *node* tetangga
: Mendapatkan data suhu
: Rute terbaik yang dilewatkan



Gambar 17 Pengujian proses pemilihan rute *outdoor*

Keterangan

: *Request* data suhu
: Meminta RSSI *node* tetangga
: Menerima RSSI *node* tetangga
: Banding RSSI dan pilih rute terbaik
: Mengirim RSSI ke *node* tetangga
: Mendapatkan data suhu
: Rute terbaik yang dilewatkan

d) Proses pengujian jarak

Dari beberapa kali pengujian yang dilakukan, baik itu di tempat tertutup (*indoor*) maupun di tempat terbuka (*outdoor*). Maka didapatkanlah hasil pengiriman dan penerimaan datanya adalah sebagai berikut. Tabel 4 adalah keterangan untuk pengujian di tempat tertutup (*indoor*). Dan Tabel 5 adalah pengujian di tempat terbuka (*outdoor*).

Tabel 4 Hasil Pengujian Jarak Pengiriman dan Penerimaan data di tempat tertutup (*indoor*)

No.	Jarak (meter)	Keterangan
1.	10	Ok
2.	20	Ok
3.	30	Ok
4.	40	Ok
5.	41	Gagal
6.	42	Gagal
7.	43	Gagal
8.	44	Gagal
9.	45	Gagal
10.	46	Gagal
11.	47	Gagal
12.	48	Gagal
13.	48	Gagal
14.	50	Gagal

Tabel 5 Hasil Pengujian Jarak Pengiriman dan Penerimaan data di tempat terbuka (*outdoor*)

No.	Jarak (meter)	Keterangan
1.	10	Ok
2.	20	Ok
3.	30	Ok
4.	40	Ok
5.	50	Ok
6.	60	Ok
7.	70	Ok
8.	80	Ok
9.	90	Ok
10.	100	Ok
11.	101	Gagal
12.	102	Gagal
13.	103	Gagal
14.	104	Gagal
15.	105	Gagal
16.	106	Gagal
17.	107	Gagal
18.	108	Gagal
19.	109	Gagal
20.	110	Gagal

Pada area *indoor* dengan jarak 1 - 40 meter pengiriman dan penerimaan data antar *node* dapat berkomunikasi dengan baik, akan tetapi pada jarak 41 meter dan selebihnya antar *node* tidak dapat berkomunikasi. Itu berarti komunikasi antar *node* tersebut terputus, sehingga *node* tidak bisa mengirim atau menerima data dari *node* lain.

Sedangkan untuk area *outdoor* dengan jarak 1 – 100 meter *node* dapat berkomunikasi dengan baik. Setelah jarak melebihi 100 meter komunikasi terputus. Dengan demikian hasil yang didapat sama dengan spesifikasi pada *datasheet* xbee S2 yang mengatakan kalau xbee S2 bisa berkomunikasi dengan jarak 1-40 meter untuk area *indoor* dan jarak 1-100 meter untuk area *outdoor.*

PENUTUP

Kesimpulan

Kesimpulan dari tugas akhir dengan judul "PENENTUAN RUTE TERBAIK PADA WSN (*WIRELESS SENSOR NETWORK*) BERDASARKAN PARAMETER RSSI (*RECEIVED SIGNAL STRENGTH INDICATOR*)" adalah sebagai berikut :

1. Sistem WSN dalam penelitan penentuan rute terbaik ini menggunakan 4 buah *node* yang terdiri dari 1 buah *node coordinator*, dan 3 buah *node router* yang ketiganya diberi ID *node* RA, *node* RB, dan *node* RC. *node coordinator* dan *node router* terdiri dari modul mikrokontroller arduino sebagai otak dari keseluruhan sistem untuk menjalankan algoritma sesuai dengan kebutuhan mulai dari algoritma *broadcast* hello *packet*, lalu algoritma *forward* hello *packet*, kemudian algoritma permintaan dan perhitungan nilai RSSI maupun data suhu, sampai algoritma perbandingan rute pada proses *request* maupun *reply* data. Yang kemudian data yang dihasilkan dikirimkan secara nirkabel dengan menggunakan modul xbee series 2 beserta shieldnya.
2. Sistem melakukan aktifitas *request* dan *reply* data apabila ada permintaan dari user, hingga sampai proses pemilihan rutenya melalui komunikasi nirkabel dan didapatkan hasil pengujian sebagai berikut:
   a) Parameter yang diambil benar-benar parameter RSSI, dengan ketentuan nilai RSSI berada pada kondisi *LOW* yang dijadikan sebagai acuan untuk memilih 1 rute terbaik. Jadi semakin kecil nilai RSSI yang dimiliki maka semakin besar pula kemungkinan untuk dijadikan sebagai rute terbaik
   b) Apabila memiliki lebih dari 1 rute terbaik, maka rute yang diprioritaskan menjadi rute terbaik adalah rute yang pertama kali terhubung/terkoneksi dengan *node coordinator.*

c) Semakin jauh posisi dan jarak antar *node* maka semakin besar pula kemungkinan data akan hilang, oleh sebab itu apabila melakukan pengiriman maupun penerimaan data dengan kondisi seperti itu diusahakan agar aktifitas komunikasi pada trafik yang lain diperkecil.
d) Hasil pengujian dalam hal pengiriman dan penerimaan data secara nirkabel menggunakan Xbee *series* 2 memiliki jangkauan jarak pengiriman dan penerimaan data antara 1 – 40 meter untuk kondisi didalam ruangan *(indoor*). Sedangkan untuk kondisi *outdoor* jangkauan jaraknya hanya sampai 1 – 100 meter saja .
e) Berdasarkan hasil pengujian yang dilakukan sebanyak 30 kali percobaan dengan rincian 15 percobaan pada kondisi *indoor* dan 15 percobaan pada kondisi *outdoor*. Dengan hasil 27 percobaan dapat menemukan rute terbaik berdasarkan parameter RSSI yang terendah saat itu dengan rincian jarak antara 1 – 40 meter untuk kondisi *indoor* dan 1 – 100 meter untuk kondis*i outdoor*. Kemudian 3 percobaan sisanya tidak bisa menemukan rute terbaik dengan rincian 1 percobaan pada kondisi *indoor* dan 2 percobaan pada kondisi *outdoor*

Saran

Agar diperoleh hasil yang lebih baik dan sebagai pengembangan pada penelitian berikutnya sebaiknya mempertimbangkan hal berikut ini:

1. Penentuan rute terbaik adalah salah satu cara untuk mengatasi kelemahan WSN dalam hal efisiensi energinya. Mungkin kedepannya bisa menerapkan cara-cara yang lain agar energi WSN tetap terjaga, salahsatunya adalah menerapkan metode *sleep node* untuk menghemat dayanya, atau juga bisa memprioritaskan bandwidth terbesar atau delay terkecil dalam pengiriman datanya. Dan Masih banyak cara – cara yang lain untuk diterapkan kedepannya
2. Untuk kevalidan hasil terhadap pemilihan rute terbaik disarankan untuk menambahkan parameter-parameter yang lain selain parameter RSSI, misalnya parameter *hop* atau *cost*. Hal ini dilakukan untuk membantu melakukan perbandingan apabila memiliki parameter RSSI yang sama
3. Jika ingin mendapatkan jangkauan jarak komunikasi secara nirkabel yang lebih luas, maka bisa menggunakan modul Xbee Pro *series* 2 disetiap *node*nya

DAFTAR PUSTAKA

AL-Karaki, J. N., & Kamal, A. E. (2004). *Routing Techniques In Wireless Sensor Network : A Survey*, 8-11.

Ali, S.J dan Syed, Roy, P., 2011. *Energy Saving Methods in Wireless Sensor Networks.* Halmstad University. pp 12-28.

Chinh, H.D dan Tan, Y.K., 2011. *Smart Wireless Sensor Network.* INTECHWEB.ORG. pp 44-49.

Faludi, Robert. 2011. *Building Wireless Sensor Network.* O'Reilly Media Inc. pp 161-177.

Fitri, Riri. 2008. *High Analisis Kinerja Protokol Routing OSPF*. University Indonesia, Indonesia. pp 7-18.

Jamal, N.A dan Kemal, E.A., 2004. *Routing Techniques In Wireless Sensor Networks : A Survey.* The Hashemite University, Jordania. pp 6-28

Lewis F.L. 2004. *Wireless Sensor Network.* University of Texas at Arlington. pp 1-18.

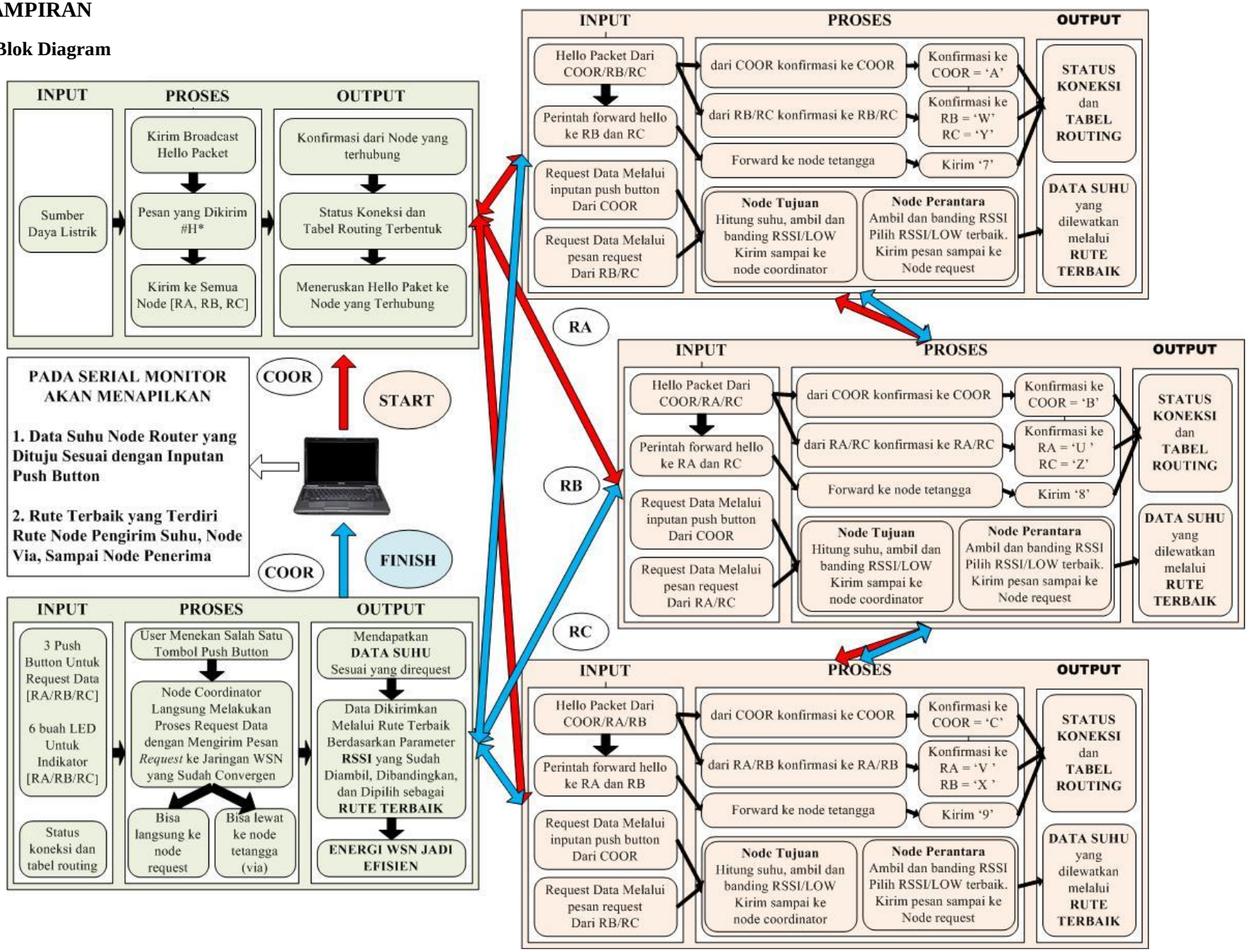
Nikolaos A. Pantazis, Stevanos A. Nikolidakis. (2013). *Energy-Efficient Routing Protocols in Wireless Sensor Networks A Survey*, 2-3.

Poellabauer, Christian. 2010. *Fundamentals of Wireless Sensor Networks.* Wiley Series on Wireless Communication and Mobile Computing. pp 150-176.

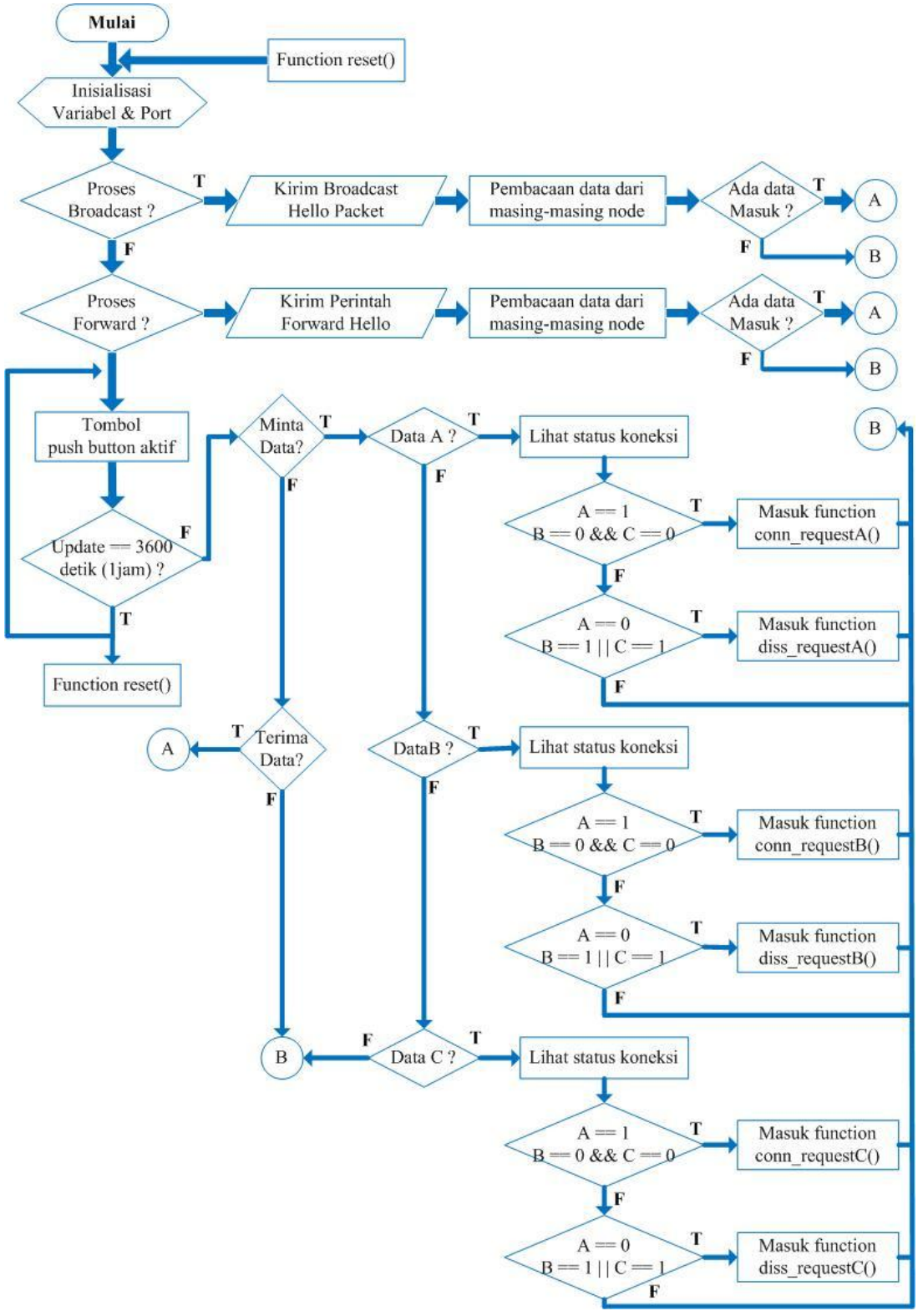
Syafril, Deny. 2013. *Penghematan Daya Pada Sensor Node Menggunakan Metode Pengaturan Waktu Kirim Data.* pp 1-11.

LAMPIRAN

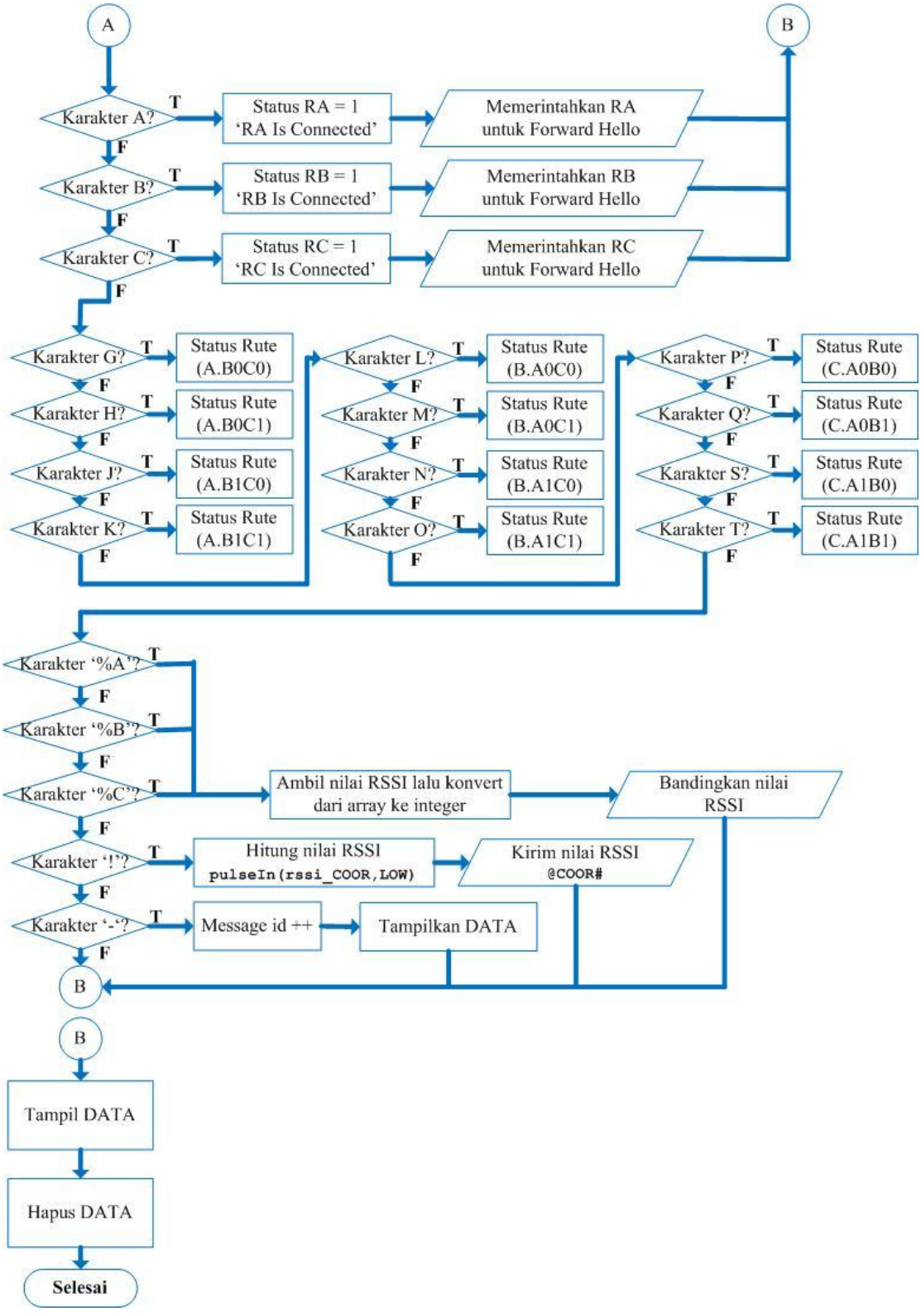
1. Blok Diagram



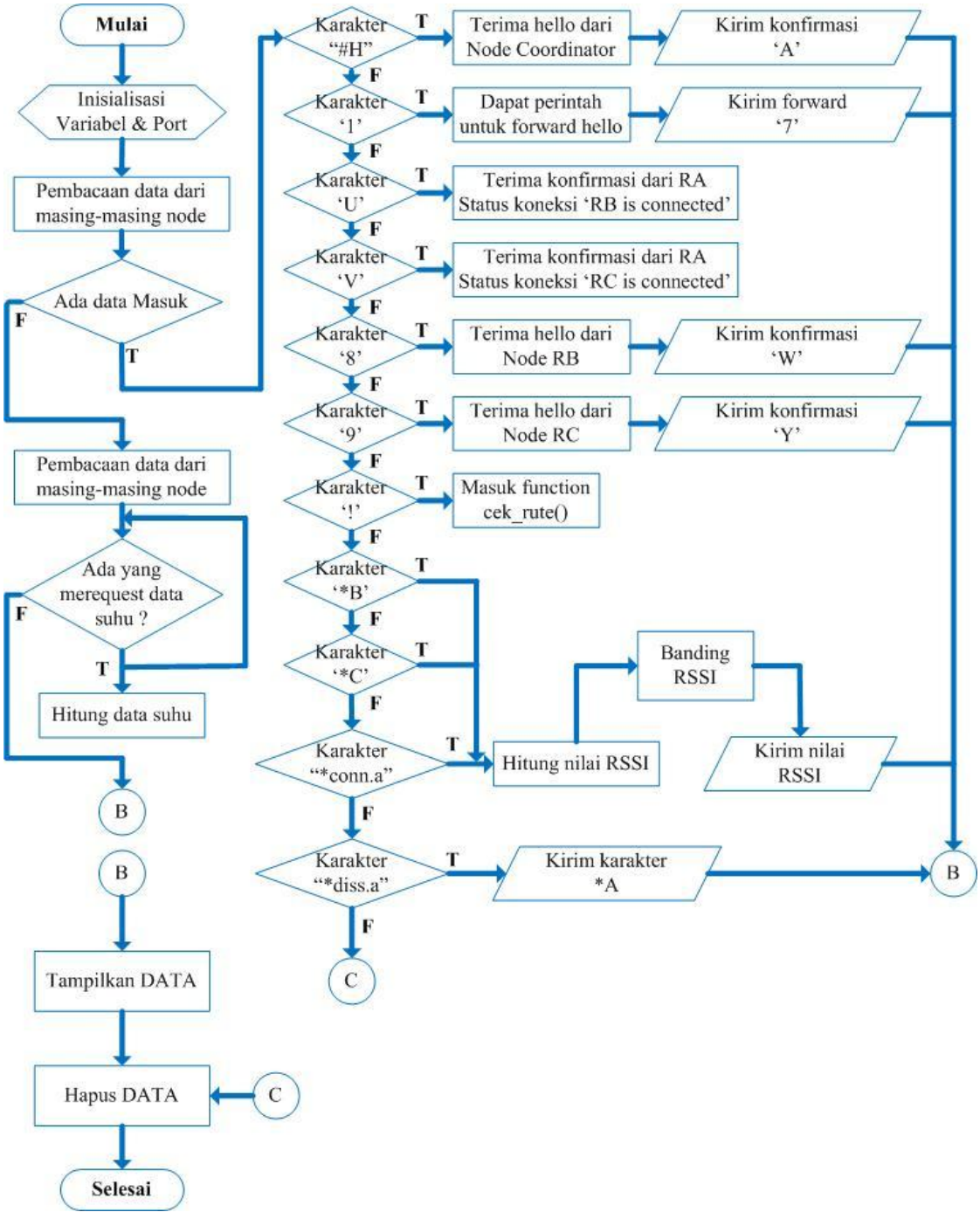
2. *Flowchart Node Coordinator*



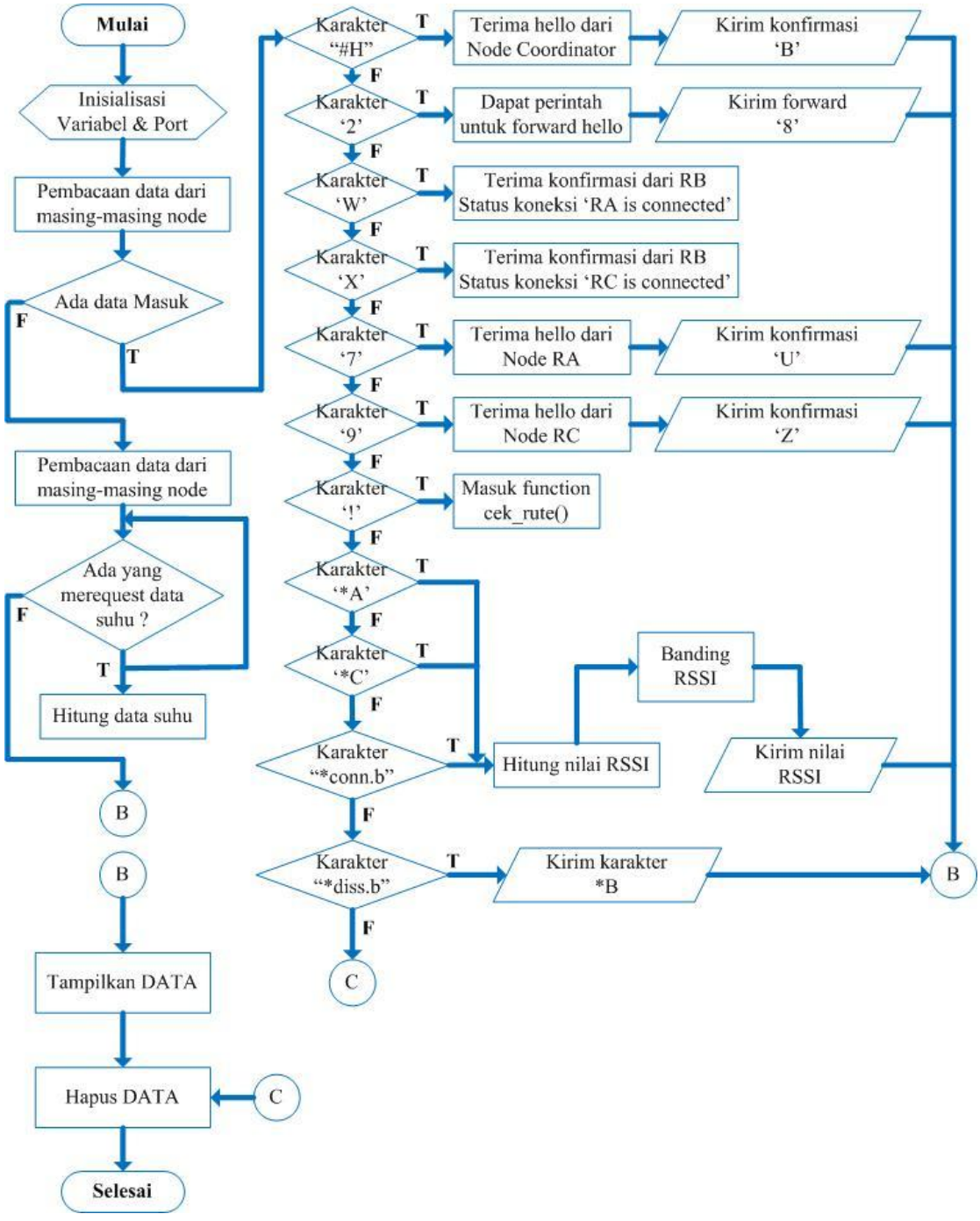
***Flowchart Node Coordinator* (Lanjutan)**



3. *Flowchart Node* RA



4. *Flowchart Node* RB



5. *Flowchart Node* RC