

**ANALISIS KINERJA PROTOKOL ROUTING LEACH-C PADA
WIRELESS SENSOR NETWORK DENGAN MENGGUNAKAN
NS2**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer



Disusun oleh :
AMIRUL MU'MININ
NIM. 0810963030

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER**

MALANG

2015

LEMBAR PERSETUJUAN

**ANALISIS KINERJA PROTOKOL *ROUTING* LEACH-C PADA
WIRELESS SENSOR NETWORK DENGAN MENGGUNAKAN
NS2**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer



Disusun Oleh :

AMIRUL MU'MININ

NIM. 0810963030

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Kasyful Amron, ST., M.Sc
NIP. 197508032003121003

Achmad Basuki, ST., M.MG., Ph.D
NIP.197411182003121002

LEMBAR PENGESAHAN

**ANALISIS KINERJA PROTOKOL *ROUTING* LEACH-C PADA
WIRELESS SENSOR NETWORK DENGAN MENGGUNAKAN
NS2**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang komputer

Disusun Oleh :

AMIRUL MU'MININ

NIM. 0810963030

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 12 Agustus 2015

Penguji I

Sabriansyah Rizqika A, S.T., M.Eng
NIP. 198208092012121004

Penguji II

Eko Sakti P, S.kom., M.Kom
NIK. 860805 06 1 1 0252

Penguji III

Agung Setia Budi, ST, MT
NIK. 2013048704231000

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji., M.T
NIP. 196798011992031001

PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Amirul Mu'minin

NIM : 0810963030

Program Studi : Informatika / Ilmu Komputer

Judul Skripsi :

ANALISIS KINERJA PROTOKOL *ROUTING* LEACH-C PADA *WIRELESS*
SENSOR NETWORK DENGAN MENGGUNAKAN NS2

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 19 Agustus 2015

Mahasiswa,

Amirul Mu'minin

NIM. 0810963030

KATA PENGANTAR

Alhamdulillah rabbil ‘alamin. Puji syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul “ANALISIS KINERJA PROTOKOL ROUTING LEACH-C PADA WIRELESS SENSOR NETWORK DENGAN MENGGUNAKAN NS2”.

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya Malang.

Dalam pelaksanaan dan penyusunan tugas akhir ini sudah cukup banyak bantuan yang diberikan berbagai pihak, baik berupa bimbingan dan saran hingga penulisan tugas akhir ini dapat terselesaikan. Oleh karena itu, penulis dalam kesempatan ini mengucapkan terima kasih kepada :

1. Bapak Kasyful Amron, ST., M.Sc dan Bapak Achmad Basuki, ST., M.MG., Ph.D selaku dosen pembimbing 1 dan dosen pembimbing 2 yang telah memberikan pengarahan dan saran dalam penyusunan tugas akhir ini.
2. Keluarga penulis, Bapak Suyadi, Ibu Idayanah, Imam Achmad Alabidulloh, Diana Solehatul Fitri, Sri Astutik yang tidak pernah berhenti memberikan dukungan, doa dan kasih sayang kepada penulis..
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku ketua dan sekretasi Program Studi Teknik Informatika Universitas Brawijaya.
5. Bapak Achmad Ridok, Drs.,M.Kom selaku dosen Penasehat Akademik.
6. M. Ziki, Mas Alva yang telah memberikan banyak masukan dan nasehat dalam penyusunan tugas akhir ini.

7. Seluruh Dosen Fakultas Matematika dan Ilmu Pengetahuan Alam dan Program Teknologi Informasi dan Ilmu Komputer yang telah mengajarkan ilmunya kepada penulis.
8. Seluruh staf dan karyawan Program Teknologi Informasi dan Ilmu Komputer yang telah banyak membantu dalam penyusunan skripsi ini.
9. Noer Muhammad Dliyaul Haq, Dwi Prasetyo, Dito Ferrandy Raditya, Zulham Yahya, Abdul Latief Lintar, Azmi Pratama Dan Fahriadi, Razaq Auliya, Arif Nur Zahya, Muhammad Mustakim, Hernawan Adi Saputro, Andjar Marviano, Tri Wahono Sadewo, Bayu Sutawijaya, Rochmad Hidayat yang telah banyak memberikan bantuan dan pengalaman kepada penulis.
10. Semua teman-teman Ilkom 2008 yang senantiasa memberikan semangat dan doa demi terselesaikannya tugas akhir ini.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan selanjutnya. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak, baik penulis maupun pembaca

Malang, 19 Agustus 2015

Penulis

ABSTRAK

Amirul Mu'minin. 2015. Analisis Kinerja Protokol Routing LEACH-C pada Wireless Sensor Network dengan menggunakan NS2. Skripsi Program Studi Informatika/Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Pembimbing: Kasyful Amron, S.T., M.Sc. dan Achmad Basuki, ST., M.MG., Ph.D.

Wireless Sensor Network (WSN) merupakan jaringan nirkabel yang terdiri dari banyak *node* berukuran kecil yang mempunyai kemampuan sensing, komputasi dan komunikasi nirkabel. WSN banyak digunakan untuk meneliti lingkungan yang sulit dijangkau oleh manusia. Permasalahan utama pada WSN adalah energi yang terbatas sehingga penghematan energi merupakan hal yang penting. Salah satu solusi untuk mengatasi permasalahan tersebut adalah penggunaan metode *clustering*. Metode *clustering* membagi semua *node* ke dalam beberapa kelompok. Dalam satu kelompok akan dipilih sebuah *cluster head* (CH) berfungsi untuk mengumpulkan data yang didapat dari *node* pada kelompoknya dan mengirimkan data tersebut ke Base Station (BS). Salah satu protokol routing yang dirancang menggunakan metode *clustering* adalah *Low Energy Adaptive Clustering Hierarchy-centralized* (LEACH-C). Penelitian ini dilakukan dengan simulasi untuk menganalisis kinerja dari LEACH-C menggunakan program simulator Network Simulator 2 (NS2). Skenario simulasi dilakukan dengan menguji *cluster* dinamis, *cluster* statis dan *backupCH*. Pada hasil simulasi didapatkan bahwa kinerja protokol routing LEACH-C dengan penentuan jumlah *cluster* dinamis lebih baik dibandingkan dengan jumlah *cluster* statis yang diukur menggunakan parameter *packet delivery ratio*, *packet loss*, *total packet*, *lifetime* jaringan, dan konsumsi energi. Selain itu, penggunaan sistem *backupCH* pada *cluster* dinamis dapat meningkatkan *lifetime* jaringan untuk protokol routing LEACH-C.

Kata Kunci : WSN, LEACH-C

ABSTRACT

Amirul Mu'minin. 2015. Analysis of Routing protocol LEACH-C on Wireless Sensor Network used NS2. Minor Thesis Program Study of Informatics/Computer Science, Faculty of Computer Science, Brawijaya University. Supervisor : Kasyful Amron, S.T., M.Sc. dan Achmad Basuki, ST., M.MG., Ph.D.

Wireless Sensor Network (WSN) is a wireless network consist of many tiny *nodes* that have the sensing ability, computing and wireless communications. WSN generally is used to examine environment inaccessible by humans. The main problem in WSN is limited energy so that energy saving is an important task. One solution to overcome these problems is the use of clustering methods. Clustering method divides all nodes into several groups. In one group would have been a cluster head (CH) serves to collect data obtained from the nodes in the group and sends the data to the Base Station (BS). One of routing protocols are designed using clustering method is the Low Energy Adaptive Clustering Hierarchy-centralized (LEACH-C). This study was conducted with a simulation to analyze the kinerjance of LEACH-C using a simulator program Network Simulator 2 (NS2). Scenario simulations carried out by testing the dynamic cluster, static cluster and CH backup. In the simulation results showed that the kinerjance of routing protocol LEACH-C with the determination of the amount of dynamic cluster better than the measured amount of static clusters using the parameter packet delivery ratio, packet loss, total packet, the network lifetime, and energy consumption. In addition, the use of backup systems of CH on dynamic cluster can improve network lifetime for routing protocol LEACH-C.

Keywords : WSN, LEACH-C

DAFTAR ISI

KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR <i>SOURCE CODE</i>	xiv
DAFTAR LAMPIRAN	xv
BAB I	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Batasan Masalah	3
1.6. Sistematika Penulisan	4
BAB II	5
2.1 <i>Wireless Sensor Network (WSN)</i>	5
2.1.1 <i>Base Station (BS)</i>	6
2.1.2 <i>Node Sensor</i>	6
2.2 Komunikasi pada WSN	7
2.3 <i>Routing</i> pada WSN	9
2.4 <i>Clustering</i> di WSN	10
2.5 <i>Fault-tolerant WSN</i>	11
2.6 Protokol	11
2.6.1 <i>Time Division Multiple-Access (TDMA)</i>	12
2.6.2 <i>Code Division Multiple-Access (CDMA)</i>	12
2.6.3 <i>Direct Sequence Spread Spectrum (DSSS)</i>	12
2.6.4 <i>Carrier Sense Multiple Access (CSMA)</i>	13
2.7 Protokol <i>Routing</i> LEACH-C	14
2.8 Algoritma LEACH-C	15
2.8.1 <i>Fase Setup</i>	15

2.8.2	Fase Steady State.....	17
2.9	First Order Radio Model.....	19
2.10	Network Simulator 2 (NS2).....	24
BAB III.....		26
3.1.	Metode Penelitian.....	26
3.1.1.	Studi Literatur	27
3.1.2.	Perancangan Sistem	27
3.1.3.	Implementasi.....	27
3.1.4.	Pengujian dan Analisis	28
3.1.5.	Pengambilan Kesimpulan.....	32
3.2.	Perancangan.....	32
3.2.1	Perancangan Lingkungan	32
3.2.2	Perancangan Simulasi	33
3.2.3	Perancangan Protokol Routing LEACH-C.....	36
BAB IV		39
4.1	Implementasi Protokol Routing LEACH-C.....	39
4.1.1.	Implementasi Cluster Statis	39
4.1.2.	Implementasi Cluster Dinamis.....	40
4.1.3.	Implementasi Backup CH	41
4.2	Pengambilan Data.....	43
4.3	Hasil Pengolahan Data	49
4.3.1	Packet delivery ratio.....	49
4.3.1.1	Packet delivery ratio tanpa Backup CH	49
4.3.1.2	Packet delivery ratio dengan Backup CH	50
4.3.2	Packet Loss.....	51
4.3.2.1	Packet Loss tanpa Backup CH.....	51
4.3.2.2	Packet Loss dengan Backup CH.....	52
4.3.3	Lifetime Jaringan	52
4.3.3.1	Lifetime Jaringan tanpa Backup CH	52
4.3.3.2	Lifetime Jaringan dengan Backup CH	53
4.3.4	Konsumsi Energi.....	54
4.3.4.1	Konsumsi Energi tanpa Backup CH	54
4.3.4.2	Konsumsi Energi dengan Backup CH	55
4.4	Analisis Protokol Routing LEACH-C	56

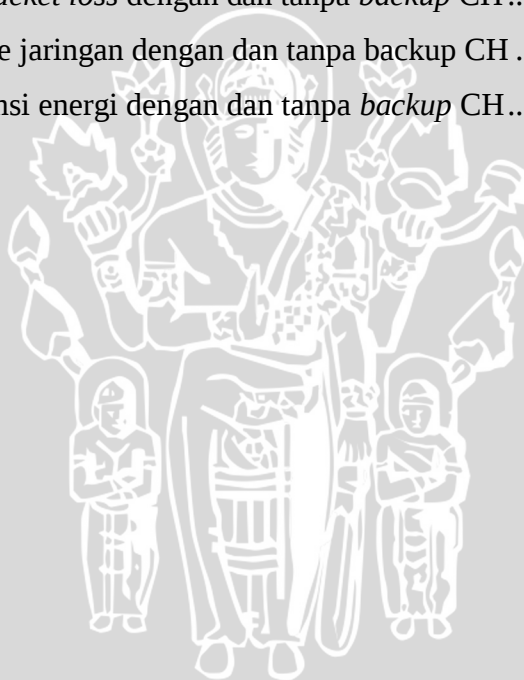
BAB V.....	61
5.1 Kesimpulan.....	61
5.2 Saran.....	61
DAFTAR PUSTAKA	63
LAMPIRAN.....	65



DAFTAR GAMBAR

Gambar 2.1 Ilustrasi skenario penggunaan WSN.....	7
Gambar 2. 2 Direct	8
Gambar 2. 3 <i>Layered / multihop</i>	8
Gambar 2. 4 Clustered.....	9
Gambar 2. 5 Taksonomi protokol routing di WSN (Al-Karaki, dkk., 2004)	10
Gambar 2. 6 Metode TDMA, FDMA dan CDMA.....	12
Gambar 2. 7 Blok diagram DSSS.	13
Gambar 2. 8 Diagram alir protokol <i>routing</i> LEACH-C	14
Gambar 2. 9 Diagram alir fase setup	17
Gambar 2. 10 Diagram alir fase Steady State.....	18
Gambar 2. 11 Pembagian fase <i>setup</i> dan fase <i>steady state</i>	18
Gambar 2. 12 Model konsumsi energi radio	19
Gambar 2. 13 Komunikasi Otcl dan C++	24
Gambar 2. 14 Struktur directory ns2	25
Gambar 3. 1 Metode penelitian	26
Gambar 3. 2 Diagram alir perancangan sistem.....	27
Gambar 3. 3 Diagram alir pengujian dan analisis sistem	28
Gambar 3. 4 Diagram alir cluster statis.....	29
Gambar 3. 5 Diagram alir cluster dinamis.....	30
Gambar 3. 6 Diagram alir tanpa sistem <i>backup</i> CH.....	31
Gambar 3. 7 Diagram alir dengan sistem <i>backup</i> CH.....	31
Gambar 3. 8 Topologi <i>node sensor</i>	33
Gambar 3. 9 Fase <i>setup</i> algoritma <i>simulated annealing</i>	37
Gambar 4. 1 File log data	44
Gambar 4. 2 File log llost	45
Gambar 4. 3 File log alive dan chmati.....	46
Gambar 4. 4 File log Chmati.txt	47
Gambar 4. 5 File log energy	48
Gambar 4. 6 Packet delivery ratio rata-rata tanpa backup CH	50
Gambar 4. 7 Packet delivery ratio rata-rata dengan backup CH	50

Gambar 4. 8 <i>Packet loss</i> rata-rata tanpa <i>backup</i> CH	51
Gambar 4. 9 <i>Packet loss</i> rata-rata dengan <i>backup</i> CH	52
Gambar 4. 10 Lifetime rata-rata tanpa <i>backup</i> CH	53
Gambar 4. 11 Lifetime jaringan rata-rata dengan <i>backup</i> CH	53
Gambar 4. 12 Konsumsi energi cluster statis tanpa <i>backup</i> CH	54
Gambar 4. 13 Konsumsi energi <i>cluster</i> dinamis tanpa <i>backup</i> CH.....	55
Gambar 4. 14 Konsumsi energi cluster statis dengan <i>backup</i> CH.....	55
Gambar 4. 15 Konsumsi energi cluster statis dengan <i>backup</i> CH.....	56
Gambar 4. 16 <i>Packet delivery ratio</i> dengan dan tanpa <i>backup</i> CH	57
Gambar 4. 17 Total <i>packet</i> sukses dengan dan tanpa <i>backup</i> CH.....	57
Gambar 4. 18 <i>Packet loss</i> dengan dan tanpa <i>backup</i> CH	58
Gambar 4. 19 Total <i>packet loss</i> dengan dan tanpa <i>backup</i> CH	58
Gambar 4. 20 Lifetime jaringan dengan dan tanpa <i>backup</i> CH	59
Gambar 4. 21 Konsumsi energi dengan dan tanpa <i>backup</i> CH.....	59



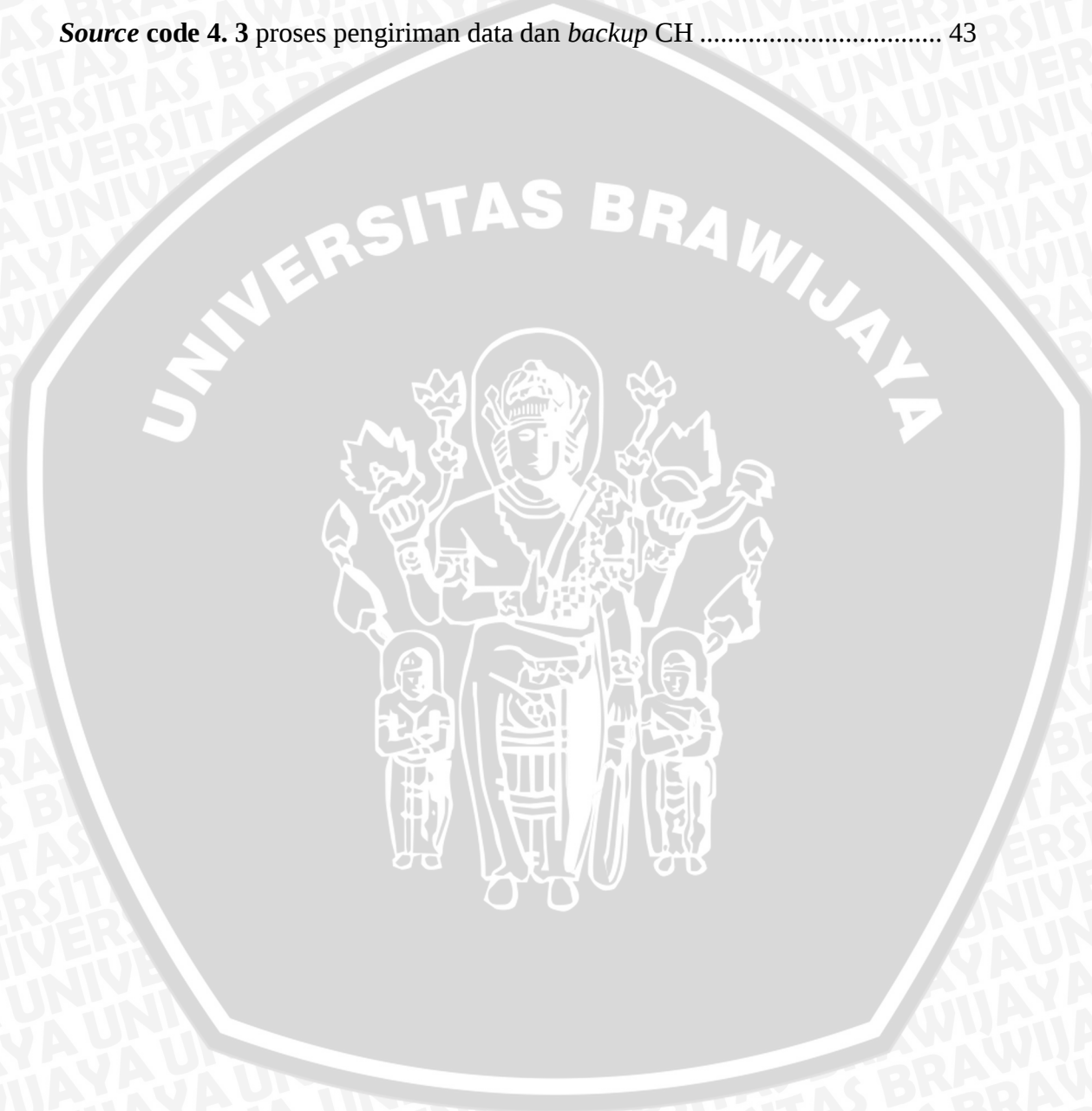
DAFTAR TABEL

Tabel 2. 1 Contoh <i>sensor</i>	5
Tabel 3. 1 Parameter simulasi.....	33



DAFTAR SOURCE CODE

Source code 3. 1 pembentukan <i>node</i> posisi <i>random</i>	36
Source code 4. 1 proses pembentukan <i>cluster</i> statis pada BS	40
Source code 4. 2 proses pembentukan <i>cluster</i> dinamis pada BS.....	41
Source code 4. 3 proses pengiriman data dan <i>backup</i> CH	43



DAFTAR LAMPIRAN

Lampiran 1 Topologi.....	65
Lampiran 2 <i>Packet delivery ratio, lifetime jaringan dan konsumsi energi</i>	68
Lampiran 3 <i>Packet loss</i>	89



BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring perkembangan jaman, kemajuan teknologi komunikasi berkembang dengan cepat termasuk teknologi nirkabel. Salah satu pemanfaatan teknologi nirkabel saat ini adalah mengamati suatu area tanpa harus berada langsung pada area penelitian, sehingga dapat mengurangi resiko saat pengamatan pada daerah yang berbahaya seperti pendeteksian tsunami, gempa bumi dan sebagainya. Teknologi ini disebut dengan *Wireless Sensor Network* (WSN).

WSN merupakan sebuah jaringan nirkabel yang terdiri dari banyak *node sensor* berukuran kecil yang secara *random* ataupun manual diletakkan pada sebuah area target (Azharuddin, dkk., 2013). *Node* pada WSN merupakan sebuah alat yang mempunyai kemampuan *sensing*, komputasi dan komunikasi nirkabel. Setiap *node* akan mengumpulkan data dari lingkungan sekitar, kemudian dikirim ke *node* lain atau langsung ke *base station* (BS) (Anastasi, dkk., 2008). Data yang dikumpulkan bisa berupa suhu, tekanan, pergerakan suatu objek, kelembaban dan sebagainya.

Salah satu permasalahan pada WSN adalah penggunaan energi. Hal ini dikarenakan catu daya yang digunakan untuk setiap *node* hanya menggunakan baterai dalam setiap proses seperti pengiriman dan penerimaan data, sehingga cadangan energi yang dimiliki sangat terbatas. Jika salah satu *node* mati maka akan mengurangi kinerja jaringan. Oleh karena itu konsumsi energi dalam setiap proses merupakan faktor paling penting untuk menentukan seberapa lama *node* dapat bertahan dan berapa banyak jumlah data yang dapat dikirimkan, sehingga energi yang digunakan harus seefisien mungkin agar dapat menghasilkan kinerja yang maksimal (Heinzelman. dkk., 2000).

Pendekatan yang dilakukan untuk menghemat energi pada WSN diantaranya dengan teknik pengolahan data hasil pembacaan *sensor* menggunakan *mining data stream*, melakukan pengelompokan data pada *node sensor* berdasarkan kedekatan data, dan algoritma pendistribusian data menggunakan *routing* protokol (Syarif. dkk., 2011). Namun penggunaan energi terbesar pada WSN adalah proses

pengiriman dan penerimaan data, maka metode paling efektif untuk meningkatkan kinerja jaringan WSN menggunakan protokol *routing* (Abad. dkk., 2011).

Salah satu protokol *routing* yang dirancang untuk menghemat energi pada teknologi WSN adalah *Low Energy Adaptive Clustering Hierarchy-centralized* (LEACH-C). Protokol *routing* LEACH-C berbasis *cluster*. Metode *cluster* disebut sebagai salah satu metode terbaik karena dapat meningkatkan umur dari jaringan (Younis, dkk., 2006) dan menghemat penggunaan energi secara keseluruhan (Azharuddin, dkk., 2013). Pada metode *clustering*, *node-node* yang tersebar dikelompokkan menjadi beberapa grup-grup kecil kemudian dipilih sebuah *node* dengan kriteria tertentu sebagai *cluster-head* (CH). CH atau *gateway* berfungsi sebagai *node* pengumpul atau data *aggregator* dari *node-node* lain dalam *cluster* tersebut yang kemudian data dikirimkan langsung ke BS. Pembentukan *cluster* pada protokol ini dibentuk secara terpusat di BS dan jumlah *cluster* ditentukan diawal.

Terdapat beberapa faktor yang mempengaruhi kinerja protokol *routing* LEACH-C seperti jumlah *cluster* tiap *round* dan *backup* CH ketika CH mati sebelum *round* selesai (Heinzelman, dkk., 2000). *Round* adalah waktu yang ditentukan untuk satu kali pembentukan semua *cluster* dan pengiriman data oleh semua *node* sampai waktu selesai.

Penelitian ini bertujuan untuk mengetahui kinerja pada protokol *routing* LEACH-C berdasarkan parameter *packet delivery ratio*, *packet loss*, *network lifetime*, dan konsumsi energi melalui sebuah simulasi WSN. Berdasarkan latar belakang tersebut maka judul tugas akhir ini yaitu “ANALISIS KINERJA PROTOKOL *ROUTING* LEACH-C PADA *WIRELESS SENSOR NETWORK* DENGAN MENGGUNAKAN SIMULATOR NS2”.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang, maka rumusan masalah dalam penelitian ini sebagai berikut :

1. Bagaimana pengaruh penentuan jumlah *cluster* yang statis dan dinamis pada setiap *round* terhadap kinerja protokol *routing* LEACH-C
2. Bagaimana pengaruh sistem *backup* CH pada *cluster* yang statis dan dinamis terhadap kinerja protokol *routing* LEACH-C

3. Bagaimana analisis kinerja protokol *routing* LEACH-C pada WSN berdasarkan parameter *packet delivery ratio*, *total packet*, *packet loss*, *network lifetime* dan konsumsi energi.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mengetahui pengaruh penentuan jumlah *cluster* yang statis dan dinamis pada setiap *round* terhadap kinerja protokol *routing* LEACH-C
2. Mengetahui pengaruh sistem *backup* CH pada *cluster* yang statis dan dinamis terhadap kinerja protokol *routing* LEACH-C
3. Mengetahui dan menganalisis kinerja protokol *routing* LEACH-C pada WSN berdasarkan parameter *packet delivery ratio*, *total packet*, *packet loss*, *network lifetime* dan konsumsi energi.

1.4. Manfaat Penelitian

Manfaat dari penulisan skripsi ini sebagai salah satu algoritma alternative untuk meningkatkan kinerja jaringan pada WSN.

1.5. Batasan Masalah

Berdasarkan masalah yang diangkat pada skripsi ini, diberikan batasan masalah sebagai berikut :

1. Penempatan *node* secara *random* dan tidak bergerak
2. Tidak memperhatikan tentang keamanan jaringan dalam WSN
3. Semua *node* hanya dapat bergabung ke satu *cluster*
4. Tidak menentukan jenis *sensor* data yang digunakan

1.6. Sistematika Penulisan

Sistematika penulisan pada skripsi ini terdiri dari beberapa bab sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan skripsi.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori yang mendukung dalam perancangan jaringan WSN.

BAB III METODE PENELITIAN DAN PERANCANGAN

Bab ini berisi metode penelitian dan perancangan sistem yang dilakukan dalam simulasi protokol *routing* LEACH-C pada WSN.

BAB IV PENGUJIAN DAN ANALISIS

Bab ini berisi pelaksanaan pengujian protokol *routing* LEACH-C pada WSN yang telah disimulasikan dan diperoleh hasil pengujian kemudian dilakukan analisis.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran pengembangan dari keseluruhan tahapan pelaksanaan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 *Wireless Sensor Network (WSN)*

Wireless Sensor Network (WSN) merupakan sebuah jaringan nirkabel yang terdiri dari banyak *node sensor* yang diletakkan secara *random* ataupun manual pada sebuah area target (Azharuddin, dkk., 2013). Tujuan utama dari WSN yaitu mengumpulkan informasi mengenai sebuah area berbahaya atau yang sulit dijangkau khususnya oleh manusia.

Beberapa penelitian mengenai WSN berfokus pada energi *aware computing* dan *distributed computing*. Teknologi WSN banyak memberikan inspirasi dalam penerapan dan penggunaan untuk segala bidang. Beberapa contoh penerapannya :

- *Monitoring lingkungan.*
- *Target tracking.*
- *Pipeline (air, minyak, gas) tracking.*
- *Monitoring pertanian.*
- *Supply chain management.*
- *Traffic management.*
- Deteksi bencana (gempa bumi, banjir, tsunami dll).

Berikut ini adalah tipe *sensor* beserta contohnya yang bisa digunakan pada WSN. Tipe *sensor* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Contoh *sensor*

Tipe <i>sensor</i>	Contoh <i>sensor</i>
<i>Temperature</i>	<i>Thermistor, thermocouple</i>
Tekanan	<i>Pressure gauge, barometer, ionization gauge</i>
Optik	<i>Photodiodes, phototransistors, infrared sensors</i>
Akustik	<i>Piezoelectric resonator, microphones</i>
Mekanik	<i>Strain gauges, tectile sensors, capacitive diaphragms</i>
Gerakan dan Getaran	<i>Accelerometers, gyroscopes, photo sensor</i>

Posisi	GPS, <i>ultrasound-based sensors</i> , <i>infrared-based sensors</i>
Kelembaban	<i>Capacitive and resistive sensors</i> , <i>hygrometers</i>
Radiasi	<i>Ionization detectors</i> , <i>Geiger-Mueller counter</i>

Setiap *node* WSN akan mengirim data *sensor* ke *base station*, dan kemudian semua data hasil *sensor* yang terkumpul akan diolah sehingga akan memberikan suatu informasi yang diinginkan. Terdapat dua komponen yang terdapat pada WSN yaitu *base station* (BS) dan *node sensor*.

2.1.1 *Base Station (BS)*

Pada WSN terdapat *sink*, disebut juga *base station* (BS), yang merupakan komponen penting di WSN. BS sangat berbeda dibandingkan dengan *node-node* yang lain di WSN, dilihat dari segi karakteristik maupun fungsinya. Karakteristik utama dari BS yaitu mempunyai energi yang jauh lebih besar dan lebih baik seperti *power grid* (Michal'ik, M., 2013) dan mempunyai area jangkauan yang luas yang dapat mencakup semua *node sensor*. Fungsi utama dari BS adalah mengumpulkan data yang didapat oleh *node sensor*. Kelebihan lain dari BS yaitu mempunyai peralatan komunikasi, pemrosesan data dan tempat penyimpanan yang lebih baik dibanding dengan *node sensor*.

2.1.2 *Node Sensor*

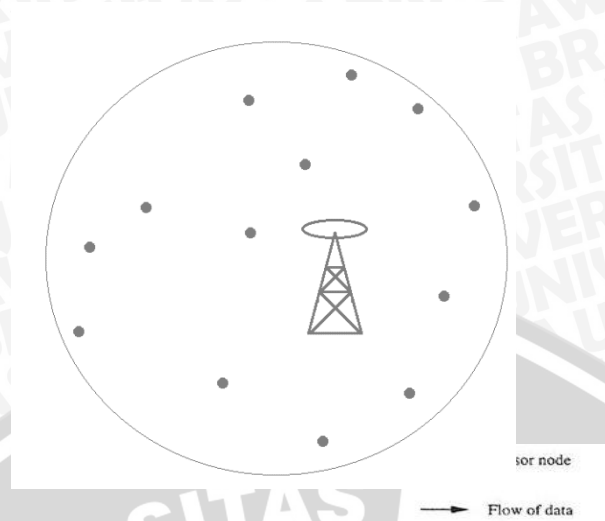
Node sensor merupakan komponen utama WSN yang bertugas untuk mengumpulkan data-data dari lingkungan sekitar misalnya suhu, tekanan udara, kelembaban udara dan beberapa parameter lingkungan lainnya. Hasil pengumpulan data maka *node sensor* perlu mengirimkan data tersebut ke *node* lain, CH ataupun langsung ke BS, oleh karena itu *node sensor* juga dilengkapi peralatan komunikasi, pemrosesan data dan penyimpanan data. *Node sensor* memiliki sumber energi yang terbatas dengan tugas-tugas yang dijelaskan sebelumnya maka penghematan energi pada *node sensor* sangat penting. Salah satu cara yang banyak dilakukan untuk menghemat energi yaitu mengaktifkan mode *sleep* pada *node sensor* apabila

2.2 Komunikasi pada WSN

The diagram illustrates a cloud-based sensor network architecture. At the center is a cloud labeled "Internet". Five functional blocks are connected to this central cloud: "Storage", "Mining", "Processing", "Analysis", and "Sensor Field 1". "Sensor Field 1" is connected to a "Base Station", which is represented by a black circle. This base station is further connected to a "Sensor", represented by a white circle. "Sensor Field 2" is also connected to the "Internet" cloud, represented by a black circle. The diagram shows a hierarchical structure where data from sensors is collected by base stations and then transmitted to the cloud for various processing tasks.

Faktor yang mempengaruhi desain WSN adalah skalabilitas, toleransi kegagalan (*fault tolerance*), area pengoperasian, topologi WSN, media transmisi, konsumsi daya, routing WSN (*Direct, Layered /multihop, Clustered*) (Abad, dkk., 2011).

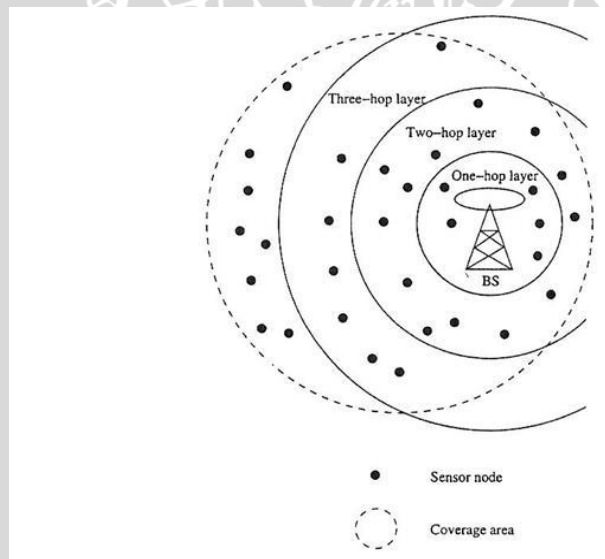
- Terdapat *base station*.
- Dalam pengiriman data dikirim secara langsung ke BS.



Gambar 2. 2 Direct

Layered / multihop:

- adanya lapisan-lapisan dari *sensor* di sekitar BS.
- lapisan i : semua *node* sejauh i -hop dari BS.
- semakin dekat jarak dengan BS, semakin rendah daya yang dibutuhkan.

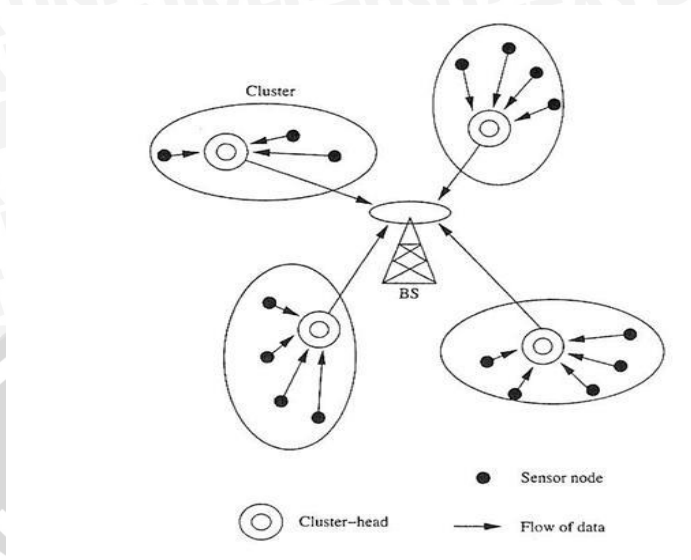


Gambar 2. 3 Layered / multihop

Clustered :

- mengelompokkan *node sensor* menjadi *cluster*.
- setiap *cluster* diatur oleh *cluster-head* (CH).
- hanya *cluster head* yang mengirimkan data ke BS.

- terkadang terdapat data *fusion*.
- adanya *self organizing* atau terpusat.

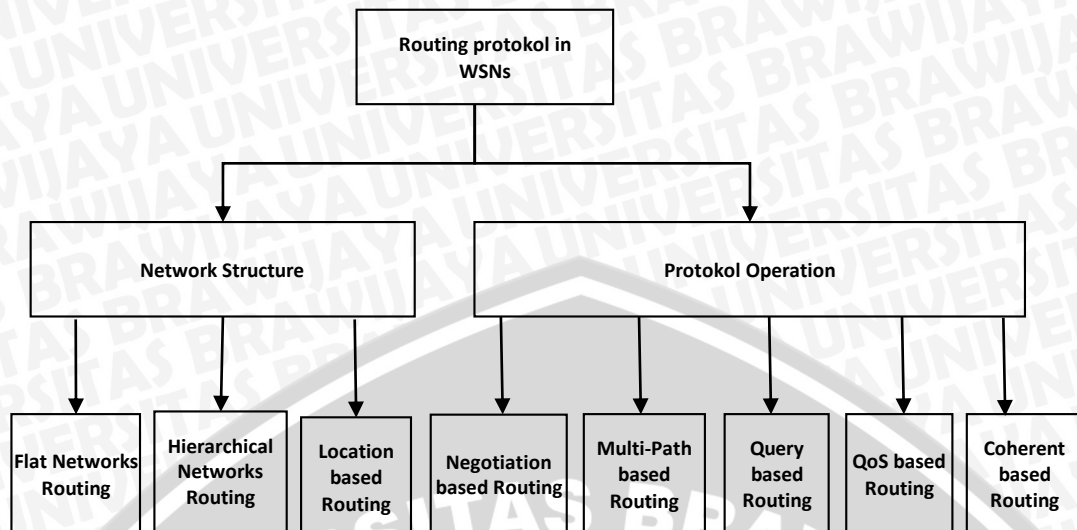


Gambar 2.4 Clustered

Data fusion adalah proses integrasi dari banyak data dan pengetahuan yang merupakan representasi dari sejumlah sensor yang tersebar pada suatu lokasi

2.3 Routing pada WSN

Alur data yang terjadi pada WSN berawal dari *node sensor* yang melakukan pengumpulan data kemudian mengirimkan ke *node* lain atau langsung ke BS. *Node sensor* memiliki energi yang terbatas oleh karena itu *routing* atau pemilihan rute pada WSN merupakan hal yang penting. Dapat dilihat taksonomi dari protokol *routing* di WSN pada gambar 2.5.



Gambar 2. 5 Taksonomi protokol *routing* di WSN (Al-Karaki, dkk., 2004)

Secara umum, *routing* di dalam WSN dapat dibagi menjadi tiga yaitu *flat based routing*, *hierarchical based routing* dan *location based routing*. Pada *flat based routing* semua memiliki fungsi dan tugas yang sama. Di dalam *hierarchical based routing* terdapat dua peran *node* yaitu sebagai CH dan *cluster-member* (CM). *Location based routing* menggunakan posisi *node* untuk mengatur pengiriman data di jaringan (Al-Karaki, dkk., 2004).

2.4 Clustering di WSN

Clustering di WSN adalah salah satu metode yang cara kerjanya membagi *node-node* ke dalam kelompok dan memilih *node* terbaik untuk menjadi koordinator. Dengan menggunakan metode *clustering* ini terbukti dapat meningkatkan umur dari jaringan dan dapat menghemat energi. Selain itu, dengan *clustering* aggregasi data dapat dilakukan dengan lebih mudah dan efisien.

Tantangan utama dari *clustering* di WSN adalah pemilihan CH. Pemilihan CH ini banyak dijadikan fokus penelitian karena pemilihan CH merupakan tahap terpenting pada metode *clustering*. Permasalahan yang ada di dalam pemilihan CH yang banyak dijadikan topik penelitian antara lain: pemilihan jumlah CH pada sebuah jaringan, pemilihan lokasi atau penempatan CH dan cara melakukan pemilihan CH.

2.5 *Fault-tolerant* WSN

Fault-tolerant WSN adalah WSN yang memiliki kemampuan untuk mengatasi permasalahan yang terjadi. Permasalahan yang ada pada WSN umumnya terjadi karena ketidakmampuan sebuah *node* untuk menjalankan fungsinya. Berdasarkan lama dari *fault*, terdapat tiga jenis *fault* yaitu: *transient* atau sementara, *intermittent* atau berselang dan *permanent* atau tetap (Afsar, dkk., 2014).

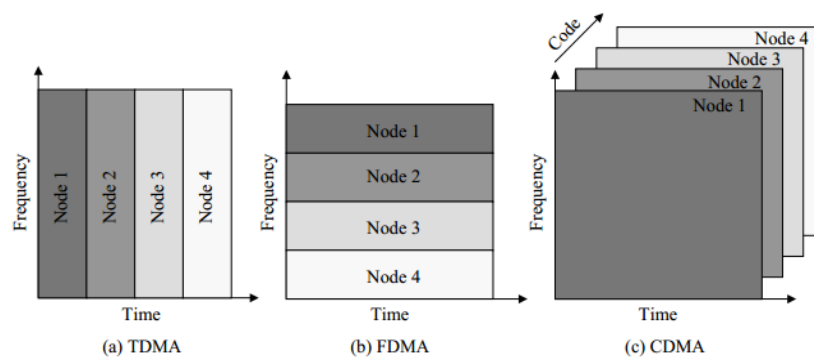
Metode *fault-tolerant* dapat dibagi menjadi beberapa kelompok yaitu (Paradis, dkk., 2007):

1. *Fault prevention*, mencegah atau menghindari terjadinya *fault*.
2. *Fault detection*, cara untuk mengetahui apakah terjadi *fault* di jaringan dengan menggunakan parameter-parameter yang berbeda.
3. *Fault isolation*, cara untuk menghubungkan berbagai indikasi *fault* yang terjadi dan mengusulkan berbagai hipotesis *fault*.
4. *Fault identification*, untuk mentes setiap hipotesis yang diusulkan untuk memastikan lokasi *fault* dan mengenali *fault* tersebut.
5. *Fault recovery*, cara untuk mengatasi *fault* yang terjadi.

Pada WSN berbasis *clustering* ada dua permasalahan yang sering dibahas yaitu tidak berfungsinya CH dan masalah *node* yang tidak dapat bergabung ke dalam sebuah kelompok.

2.6 Protokol

LEACH-C menggunakan *Media Access Control* (MAC) yaitu *MacSensor*. Protokol ini adalah kombinasi dari *Carrier-Sense Multiple Access* (CSMA), *Time-Division Multiple Access* (TDMA), *Code-Division Multiple Access* (CDMA) dan *Direct-Sequence Spread Spectrum* (DS-SS). Aplikasi ini menentukan MAC protokol yang digunakan untuk mengirim setiap pesan berdasarkan kendala protokol *routing*. Jika pesan berisi data untuk dikirim ke *cluster-head*, maka pengirimannya menggunakan slot TDMA dengan DS-SS kode ke *cluster-head*. Menggunakan pendekatan hal tersebut, protokol MAC dipilih sedemikian rupa untuk mengurangi disipasi energi dan meminimalkan terjadinya tabrakan (misalnya menggunakan DS-SS diatas TDMA atau menggunakan CSMA untuk mengurangi jumlah tabrakan).



Gambar 2. 6 Metode TDMA, FDMA dan CDMA

2.6.1 Time Division Multiple-Access (TDMA)

Time Division Multiple-Access (TDMA) adalah metode akses kanal untuk jaringan medium bersama. Hal ini memungkinkan beberapa pengguna untuk berbagi kanal frekuensi yang sama dengan membagi sinyal dalam slot waktu yang berbeda (Giannattasio. Dkk, 2009). TDMA dapat menghemat energi yaitu dengan cara membiarkan *Node* tetap dalam status *sleep* sampai waktu dimana *Node* harus mengirimkan data (Heinzelman, dkk., 2002). Selain itu juga TDMA mengurangi penggunaan energi dengan menghilangkan *collision* dan menghindari *idle listening*.

2.6.2 Code Division Multiple-Access (CDMA)

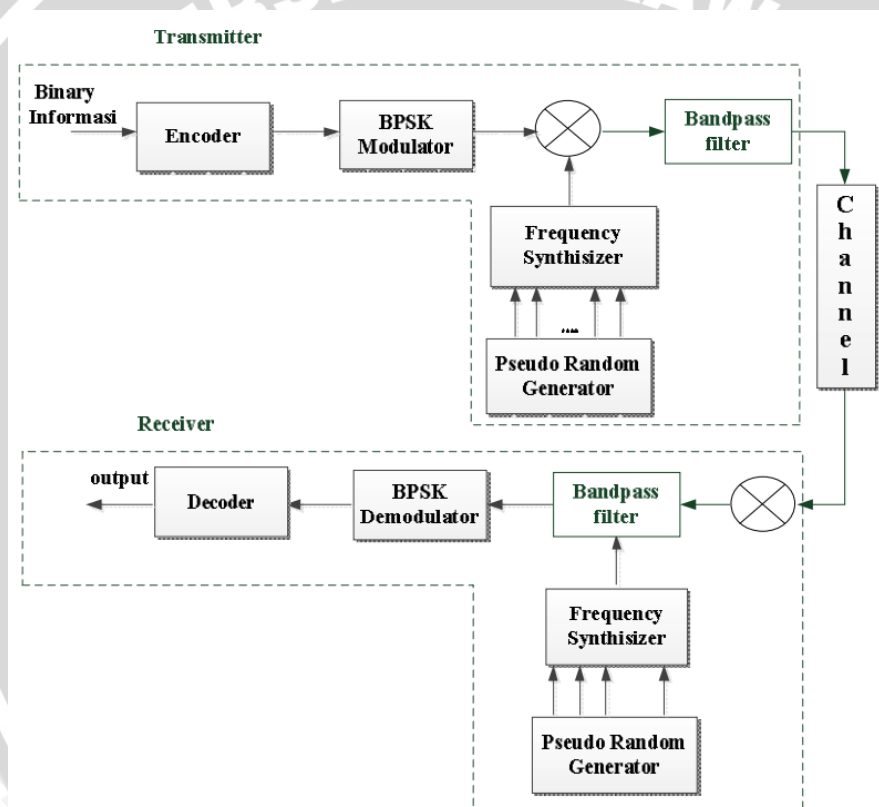
Code Division Multiple Access (CDMA) adalah sebuah bentuk pemultipleksan (bukan sebuah skema pemodulasian) dan sebuah metode akses secara bersama yang membagi *kanal* tidak berdasarkan waktu (seperti pada TDMA) atau frekuensi (seperti pada FDMA), namun dengan cara mengkodekan data dengan sebuah kode khusus yang diasosiasikan dengan tiap *kanal* yang ada dan menggunakan sifat-sifat *interferensi* konstruktif dari kode-kode khusus itu untuk melakukan pemultipleksan. Metode pengiriman ini digunakan untuk CH ketika mengirim data ke BS.

2.6.3 Direct Sequence Spread Spectrum (DSSS)

DSSS dikenal juga sebagai *Direct Sequence Code Division Multiple Access (DS-CDMA)*, DSSS merupakan salah satu cara menyebarkan modulasi sinyal digital di udara. Rentetan informasi dikirim dengan membagi sekecil mungkin

sinyal, lalu ditumpangkan pada *kanal* frekuensi yang ada di dalam *spektrum* tertentu.

DSSS merupakan jenis *spread spectrum* yang paling dikenal dan paling banyak digunakan, karena sistem ini dikenal paling mudah diimplementasikan. Sebagian besar peralatan atau piranti nirkabel yang ada di pasaran sekarang ini menggunakan teknologi DSSS. DSSS merupakan suatu metode pengiriman data dimana sistem pengirim dan penerima berada dalam set frekuensi yang lebarnya adalah 22Mhz. Saluran yang lebar ini memungkinkan piranti untuk memancarkan lebih banyak informasi pada data rate yang lebih tinggi. Dimana *blok diagram* sistem DSSS pada sisi *transmitter* dan *receiver* dapat dilihat pada gambar 2.7.



Gambar 2. 7 Blok diagram DSSS.

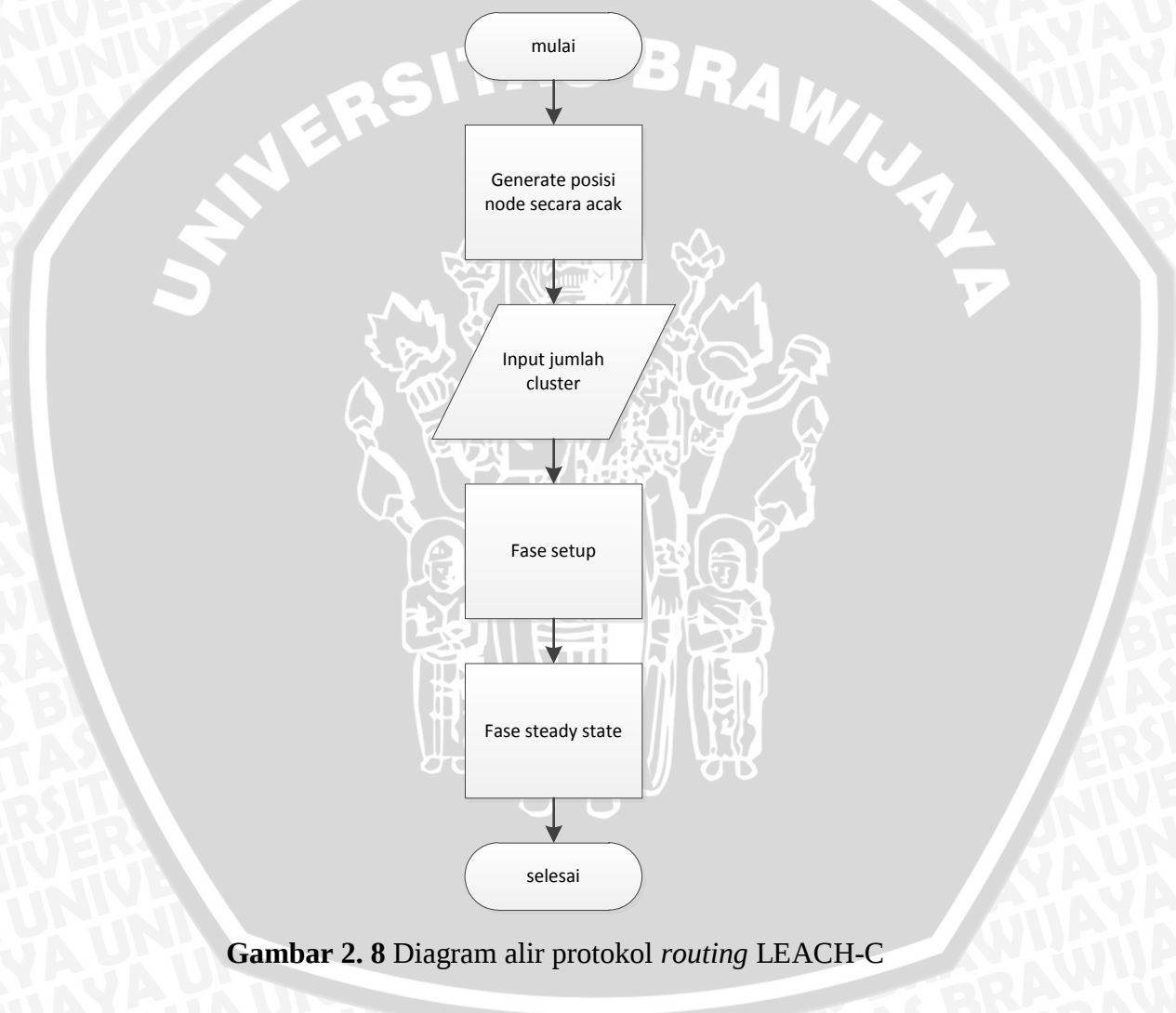
2.6.4 Carrier Sense Multiple Access (CSMA)

Carrier Sense Multiple Access (CSMA) adalah protokol *Media Access Control (MAC)* dimana *node* menverifikasi adanya lalu lintas lainnya sebelum transmisi pada media transmisi bersama, seperti bus listrik, atau sebuah *band* dari

spectrum elektromagnetik. Metode pengiriman ini digunakan untuk mengirim informasi pada fase *setup* di LEACH-C.

2.7 Protokol Routing LEACH-C

Protokol *routing* LEACH-C merupakan protokol *routing* yang memiliki kelebihan pada fase *setup*, pada fase ini CH dan *cluster* ditentukan oleh BS menggunakan algoritma pemilihan *cluster* yang optimal seperti *simulated annealing*. Diagram alir protokol *routing* LEACH-C dapat dilihat pada gambar 2.8.



Gambar 2. 8 Diagram alir protokol *routing* LEACH-C

Ketika LEACH-C menghasilkan *cluster* yang lebih baik, terdapat juga kerugian pada algoritma *centralized cluster* dari pada *distributed algorithm*. Protokol ini memerlukan informasi awal yaitu posisi, dimana posisi ini biasanya didapat dari *Global Positioning System* (GPS) atau alat lain untuk menentukan posisi. LEACH-

C untuk fase *setup* membutuhkan energi yang lebih besar dari *distributed algorithm* karena setiap *Node* mengirimkan informasi lokasi dan juga sisa energi ke BS menggunakan *Carrier Sense Multiple Access* (CSMA) secara langsung pada setiap putaran awal. Konsumsi energi awal pada LEACH-C akan selalu tetap karena pengirim dan penerima pada setiap putaran sama.

Walaupun terjadi peningkatan konsumsi energi pada fase *setup* LEACH-C, tapi secara keseluruhan LEACH-C lebih baik. Diantara fase *setup* dan fase *steady state*, fase *steady state* lebih banyak membutuhkan energi untuk transmisi data. Sehingga dengan terbentuknya *cluster* yang lebih optimal pada fase *setup* dengan *centralized algorithm* maka energi konsumsi yang dibutuhkan untuk fase *steady state* lebih optimal.

2.8 Algoritma LEACH-C

Algoritma LEACH-C dibagi menjadi dua fase yaitu fase *setup* dan fase *steady state*. Proses algoritma LEACH-C dijelaskan sebagai berikut.

2.8.1 Fase Setup

Fase *setup* pada protokol *routing* LEACH-C, masing-masing *Node* mengirimkan informasi mengenai lokasi dan tingkat energi ke *Base Station* (BS). Untuk menentukan *cluster* yang optimal, BS perlu memastikan bahwa beban energi didistribusikan secara merata diantara semua *node*. Hal pertama yang dilakukan untuk memastikan energi didistribusikan secara merata yaitu dengan menghitung energi rata-rata dari keseluruhan *node* pada simulasi dan jika *node* mempunyai energi dibawah energi rata-rata, maka *node* tersebut tidak memiliki peluang untuk menjadi *cluster head* untuk putaran saat ini.

Untuk mendapatkan energi rata-rata pada tiap *round* dapat dilihat pada persamaan 2.1.

$$avg_energy = \frac{\sum(energi_tiap_node)}{total_node} \quad (2.1)$$

Untuk menemukan *cluster* yang optimal pada LEACH-C tidak berhenti pada tahap seleksi *node* berdasarkan sisa energi. Pada tahap selanjutnya BS akan

memilih secara *random node* CH berdasarkan jumlah *cluster* yang diinginkan (C). kemudian menghitung jarak *node* CH dengan *nonCH* ($f(C)$) untuk menentukan nilai *cost* sebagai nilai *global optimum* awal. Selanjutnya BS menentukan CH baru secara *random* dari *node* yang sudah diseleksi berdasarkan energi rata-rata sehingga menemukan (C'). Dari CH yang baru kemudian menghitung kembali jarak CH dan *nonCH* sehingga mendapatkan ($f(C')$) untuk dibandingkan dengan nilai *global optimum* awal, jika nilai *cost* CH baru lebih rendah maka nilai tersebut digunakan untuk mengganti nilai *global optimum* awal. Jika nilai *cost* lebih tinggi atau sama dengan dari *global optimum* maka proses dilanjutkan dengan membandingkan nilai *probabilitas random* (0 sampai 1) dengan *probabilitas* P_k . jika nilai *random* lebih kecil atau sama dengan *probabilitas* P_k , maka nilai *cost* tersebut dijadikan sebagai *global optimum* begitu seterusnya hingga jumlah maksimal iterasi dan tingkat pembaharuan *global optimum* terpenuhi. Sehingga kumpulan CH yang menjadi nilai *global optimum* akhir digunakan untuk membentuk *cluster* yang optimal.

Probabilitas P_k dapat dilihat pada persamaan 2.2.

$$P_k \begin{cases} e^{-(f(C')-f(C))/\alpha_k} & : f(C') \geq f(C) \\ 1 & : f(C') < f(C) \end{cases} \quad (2.2)$$

Keterangan :

α_k adalah parameter jumlah iterasi

$f(C)$ adalah jarak kuadrat antara CH dan *nonCH* yang sudah tertentu (*cost*)

$f(C')$ adalah jarak kuadrat antara CH dan *nonCH* yang baru

Sedangkan untuk perhitungan jarak kuadrat antara CH dan *nonCH* dapat dilihat pada persamaan 2.3.

$$f(C) = f(C') = \sum_{i=1}^N \min_{c \in C} d^2(i, c) \quad (2.3)$$

Keterangan :

$d(i, c)$ adalah jarak antara *node* i dan *node* c .

Sedangkan nilai jarak (d) didapat dari persamaan 2.4.

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 \quad (2.4)$$

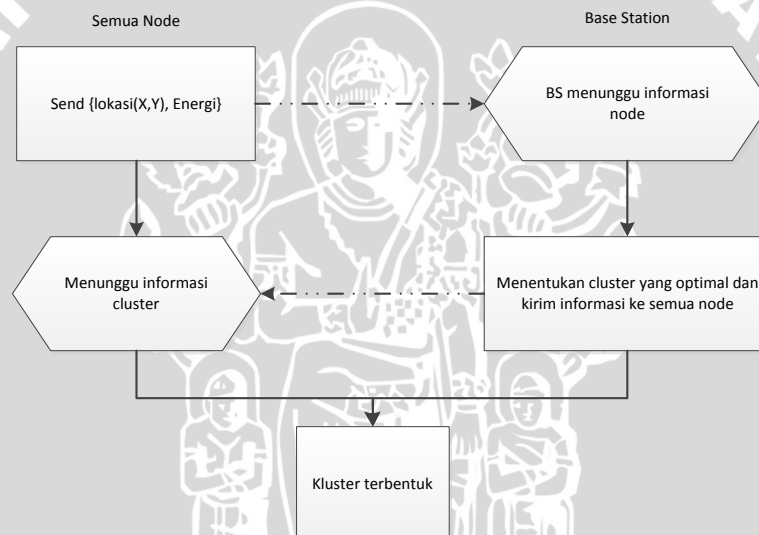
Untuk persamaan iterasi (α_k) dapat dihitung dengan persamaan berikut.

$$\alpha_k = 1000e^{\frac{k}{20}} \quad (2.5)$$

Keterangan :

k adalah nilai iterasi

Ketika *cluster-head* dan *cluster* yang optimal didapat, BS akan mengirimkan informasi *cluster-head identification* (CH ID) untuk setiap *node*. Jika *node* ID sesuai dengan CH ID, maka *node* tersebut bertindak sebagai CH dan menentukan *Time Division Multiple-Access* (TDMA) slot waktu untuk transmisi data setiap member ke CH. Diagram alir untuk fase *setup* dapat dilihat pada gambar 2.9.

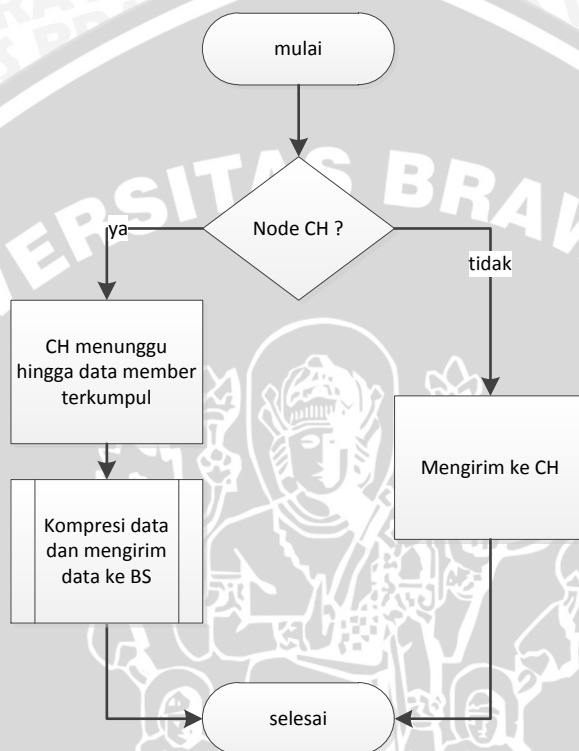


Gambar 2.9 Diagram alir fase *setup*

2.8.2 Fase *Steady State*

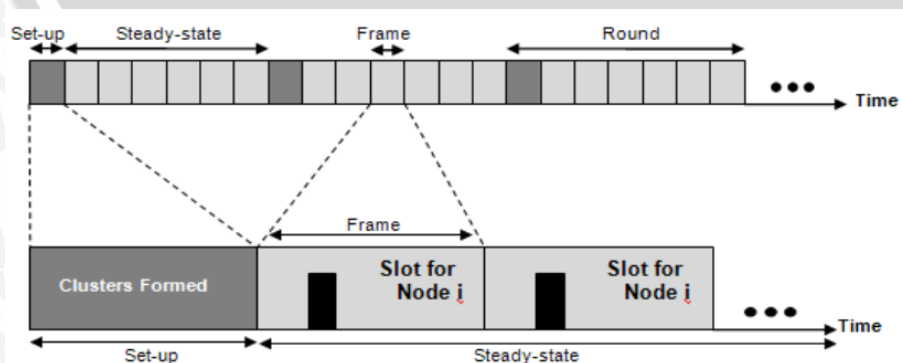
Pada fase *steady state*, semua *node* member mengirim data sekali ke CH untuk tiap *frame* selama pembagian slot waktu transmisi. Durasi waktu transmisi data untuk setiap member ukurannya tetap, sehingga lama waktu CH untuk mengirim data ke BS bergantung pada jumlah *node* member dalam *cluster*. Semua *node* member mengirimkan data ke CH berdasarkan TDMA *schedule* yang sudah ditentukan. CH menerima data dari setiap anggota *cluster* dan melakukan kompresi data kemudian mengirimkan data tersebut ke BS.

Proses transmisi data menggunakan DSSS *spreading code* untuk meminimalisasi interferensi antar *cluster* dalam jaringan. Proses fase *steady state* menghabiskan waktu lebih lama dibandingkan proses fase *setup*, karena transfer data terjadi melalui transmisi radio secara intensif. Sedangkan *setup* hanya pemilihan CH dan pembentukan *cluster*. Diagram alir proses fase *steady state* dapat dilihat pada gambar berikut.



Gambar 2. 10 Diagram alir fase *steady state*

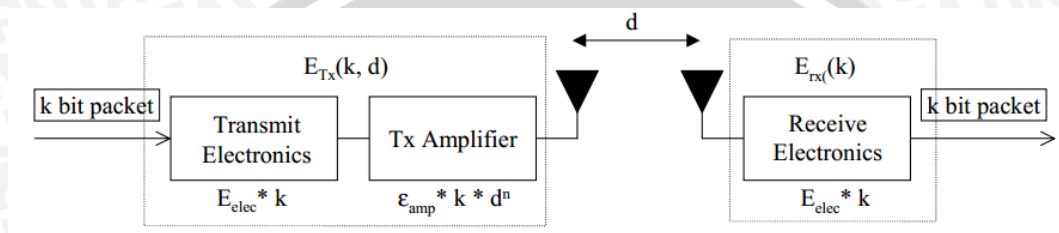
Pembagian fase *setup* dan fase *steady state* terhadap waktu dapat dilihat pada gambar berikut.



Gambar 2. 11 Pembagian fase *setup* dan fase *steady state*

2.9 First Order Radio Model

Pada simulasi LEACH-C ini menggunakan model konsumsi energi radio. Propagasi gelombang elektromagnetik dapat dimodelkan berdasarkan jarak *transmitter* dan *receiver*. Konsumsi energy untuk *transmitter* dipengaruhi oleh besarnya energi radio elektronik dan juga energi *amplifier* sedangkan konsumsi energi untuk *receiver* hanya bergantung pada energi radio elektronik. Model konsumsi energi radio dapat dilihat pada gambar 2.12.



Gambar 2. 12 Model konsumsi energi radio

Selain itu konsumsi energi untuk mentransmisikan *packet* data bergantung pada jarak antara *transmitter* dan *receiver*, sehingga untuk mentransmisikan l -bit data dengan jarak d dapat dilihat dari persamaan 2.6 (Heinzelman, dkk., 2000).

$$E_{Tx}(l, d) = E_{tx-elec}(l) + E_{Tx-amp}(l, d) \quad (2.6)$$

Dan untuk menerima data dapat dilihat dari persamaan 2.7.

$$\begin{aligned} E_{Rx}(l) &= E_{Rx-elec}(l) \\ E_{Rx}(l) &= lE_{elec} \end{aligned} \quad (2.7)$$

E_{elec} adalah energi yang digunakan untuk pengoperasian *circuit* pada perangkat radio dari *node sensor*, sedangkan E_{tx-amp} adalah energi untuk memperkuat sinyal pengirim agar sinyal yang sampai pada penerima masih memenuhi nilai minimal yang menjadi batas penerimaan sinyal informasi pada *receiver* atau disebut sensitivitas penerimaan.

Nilai E_{elec} dan E_{tx-amp} bergantung pada nilai beberapa parameter seperti, panjang gelombang pengirim (λ), penguatan (G), bi rate(R_b), tinggi antenna (h), dan sensitivitas penerima. Setiap parameter dijelaskan sebagai berikut,

1. Panjang Gelombang Pengirim (λ)

Panjang gelombang didefinisikan sebagai jarak yang ditempuh oleh satu bukit dan lembah dari gelombang elektronik. Dan dapat dihitung menggunakan persamaan 2.8.

$$\lambda = \frac{c}{f} \quad (2.8)$$

Dengan f adalah gelombang frekuensi yang digunakan dan c adalah cepat rambat gelombang radiasi elektromagnetik, jika dalam ruang hampa maka kecepatan yang digunakan adalah kecepatan cahaya.

2. Bit rate (R_b)

Bit rate didefinisikan sebagai laju pengiriman setiap detik dengan satuan *bit* per detik atau disebut dengan *bandwidth* jaringan.

3. Penguatan (G)

Merupakan perbandingan daya pada antenna dimana nilai penguatan ini dipengaruhi oleh penguatan daya pada transmitter dan juga *receiver*. Satuan untuk penguatan daya ini adalah *decibel* dB.

4. Tinggi Antena (h)

Tinggi antena didefinisikan sebagai jarak yang diambil dari dasar (tanah) ke antena. Pengaruh ketinggian antena ini digunakan untuk menghitung jarak *free space* atau jarak pantul gelombang yang dikirim ke *destination*.

5. Sentivitas Penerima ($P_{r-Threshold}$)

Antena pada sisi penerima memiliki kemampuan untuk menangkap sinyal yang berada dalam frekuensi kerjanya. Beberapa dari sinyal itu akan didefenisikan sebagai derau. Sinyal dianggap sebagai informasi jika memiliki daya sama atau lebih besar dari nilai ambang batas penerimaan dari *receiver* atau dikenal dengan istilah sensitivitas penerima.

6. Energi Elektronik (E_{elec})

Dilihat dari gambar 2.12 energi elektroniks (E_{elec}) adalah bagian dari digital *coding*, *modulation* dan *filtering* signal sebelum dikirim ke tranmisi *amplifier*. Konsumsi energy per bit pada elektronik *transceiver* ditentukan dengan

$$E_{elec} = 50 \text{ nJ/bit} .$$

7. Amplify Energi (E_{amp})

Gelombang elektromagnetik dapat dipropagasikan dengan 2 cara yaitu jika jarak antara *transmitter* dan *receiver* lebih kecil dari pada jarak *crossover* maka model propagasi menggunakan *free space* (d^2). Ketika jarak antara *transmitter* dan *receiver* lebih besar dari pada jarak *crossover* maka menggunakan model propagasi *two-ray ground* (d^4) dimana sinyal dipantulkan untuk menjangkau *receiver*. Jadi sinyal dikirim langsung atau dipantulkan berdasarkan pada jarak, karena dalam simulasi ini berada pada ruang tanpa penghalang.

Untuk nilai transmit *amplifier* energi pada model propagasi *free space* (d^2) dapat dilihat dari persamaan 2.9.

$$\mathcal{E}_{friss_amp} = \frac{P_{R-Threshold} \cdot (4\pi)^2}{R_b \cdot G_t \cdot G_r \cdot \lambda^2} \quad (2.9)$$

Keterangan:

$\mathcal{E}_{friss_amp}$ adalah energi *amplifier* dalam J/bit/m².

$P_{R-Threshold}$ adalah minimum *power* penerimaan *packet* (W).

R_b adalah *bandwidth* dalam bit/detik.

G_t adalah antena pengirim *gain*.

G_r adalah antena penerima *gain*.

λ adalah panjang gelombang sinyal *carrier* dalam meter.

Sedangkan untuk nilai transmit *amplifier* energi pada model propagasi *two-ray ground* (d^4) dapat dilihat dari persamaan 2.10.

$$\mathcal{E}_{two_ray_amp} = \frac{P_{R-Threshold}}{R_b \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2} \quad (2.10)$$

Keterangan:

$\mathcal{E}_{two_ray_amp}$ adalah energi *two-ray amplifier* dalam J/bit/m⁴.

$P_{R-Threshold}$ adalah minimum *power* penerimaan *packet* (W).

R_b adalah *bandwidth* dalam bit/detik.

G_t adalah antena pengirim *gain*.

G_r adalah antena penerima *gain*.

h_r adalah tinggi antena penerima dalam meter.

h_t adalah tinggi antena pengirim dalam meter.

Untuk membedakan penggunaan kedua energi diatas maka harus menentukan parameter jarak $d_{crossover}$ dimana nilai jarak tersebut dapat dihitung dari persamaan 2.11.

$$(d_{crossover})^2 = \frac{16 \cdot \pi^2 \cdot L \cdot h_r^2 \cdot h_t^2}{\lambda^2} \quad (2.11)$$

atau

$$d_{crossover} = \sqrt{\frac{\mathcal{E}_{friss_amp}}{\mathcal{E}_{two_ray_amp}}} \quad (2.12)$$

Keterangan:

$d_{crossover}$ adalah jarak *crossover* dalam meter.

L adalah sistem *loss factor* ($L \leq 1$).

h_r adalah tinggi antena penerima dalam meter.

h_t adalah tinggi antena pengirim dalam meter.

λ adalah panjang gelombang sinyal *carrier* dalam meter.

Jika jarak pengirim dan penerima lebih kecil dari nilai $d_{crossover}$ maka sinyal dipropagasikan *free space*. Jika jarak pengirim dan penerima lebih besar dari nilai $d_{crossover}$ maka sinyal dipropagasikan *two-ray ground* / dipantulkan.

Sehingga untuk mentransmisikan l -bit data dengan jarak d dapat dilihat dari persamaan 2.12.

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\mathcal{E}_{friss_amp}d^2 & d < d_{crossover} \\ lE_{elec} + l\mathcal{E}_{two_ray_amp}d^4 & : d \geq d_{crossover} \end{cases} \quad (2.13)$$

Dilihat dari persamaan model propagasi *free space* dan *two-ray ground* dipengaruhi oleh nilai sentivitas penerima ($P_{r-Threshold}$) maka *power* yang dibutuhkan agar data sampai ke penerima dan bisa dibaca harus sama atau lebih dari ($P_{r-Threshold}$). Jika R_b adalah radio bit rate, *power* transmit P_t sama dengan transmit energi per bit $E_{tx-amp}(l, d)$ maka terbentuk persamaan 2.13.

$$P_t = E_{Tx-amp}(l, d)R_b \quad (2.14)$$

Ketika nilai $E_{Tx-amp}(l,d)$ dimasukkan pada persamaan di atas maka akan terbentuk persamaan 2.14.

$$P_t = \begin{cases} \varepsilon_{friss_amp} \cdot bw \cdot d^2 & , d < d_{crossover} \\ \varepsilon_{two_ray_amp} \cdot bw \cdot d^4 & , d \geq d_{crossover} \end{cases} \quad (2.15)$$

Keterangan:

P_t adalah daya untuk transmisi dalam Watt.

ε_{friss_amp} adalah energi *amplifier* dalam J/bit/m².

$\varepsilon_{two_ray_amp}$ adalah energi *two-ray amplifier* dalam J/bit/m⁴.

bw adalah *bandwidth* dalam bit/detik.

d adalah jarak antara *transmitter* dan *receiver* dalam meter.

$d_{crossover}$ adalah jarak *crossover* dalam meter.

Jika menggunakan *channel* model yang bergantung pada ketinggian antenna, panjang gelombang, penguatan maka didapat persamaan 2.14

$$P_r = \begin{cases} \frac{\varepsilon_{friss-amp} R_b G_t G_r \lambda^2}{(4\pi)^2} & d < d_{crossover} \\ \varepsilon_{two-ray-amp} R_b G_t G_r h_t^2 h_r^2 & : d \geq d_{crossover} \end{cases} \quad (2.16)$$

Keterangan:

P_r adalah daya untuk *receive* dalam Watt.

ε_{friss_amp} adalah energi *amplifier* dalam J/bit/m².

$\varepsilon_{two_ray_amp}$ adalah energi *two-ray amplifier* dalam J/bit/m⁴.

G_t adalah antenna pengirim gain.

G_r adalah antenna penerima gain.

h_r adalah tinggi antenna penerima dalam meter.

h_t adalah tinggi antenna pengirim dalam meter.

λ adalah panjang gelombang sinyal *carrier* dalam meter.

8. Energi Beamforming

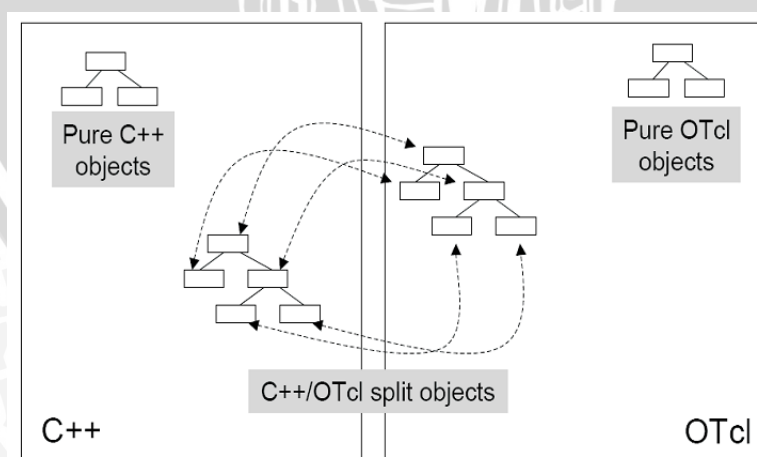
Tugas dari CH adalah menampung semua data dari member kemudian dikirimkan ke BS. Tetapi setiap data dari member tersebut dikumpulkan terlebih dahulu kemudian dilakukan proses agregasi data, dimana proses agregasi juga

membutuhkan energi. Energi untuk melakukan proses agregasi data disebut sebagai energi *beamforming* dengan satuan 5nJ/bit/signal.

2.10 Network Simulator 2 (NS2)

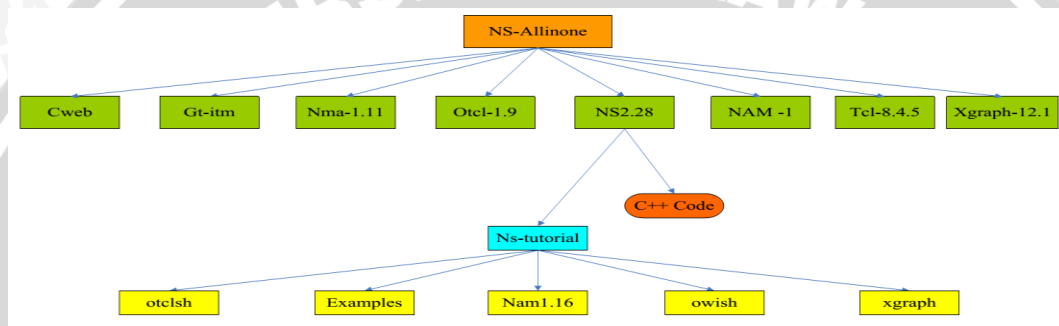
NS2 adalah alat simulasi jaringan yang bersifat *open source* yang banyak digunakan dalam mempelajari struktur dinamika dari jaringan komunikasi. Simulasi dari jaringan nirkabel dan protokol (seperti algoritma *routing* TCP dan UDP) dapat diselesaikan dengan baik menggunakan simulator ini. Beberapa keuntungan menggunakan *network simulator* sebagai perangkat lunak simulasi adalah *network simulator* dilengkapi dengan tool validasi, *network simulator* bersifat *open source* di bawah GPL (*Gnu Public License*) dan dapat digunakan pada sistem operasi Windows ataupun Linux. NS dibangun sebagai varian dari *REAL Network Simulator* pada tahun 1989 di UCB (*university of California berkeley*). Dari awal tim ini membangun sebuah perangkat lunak simulasi jaringan internet untuk kepentingan riset interaksi antar protokol dalam konteks pengembangan protokol internet saat ini dan masa yang akan datang.

NS dibangun dengan menggunakan 2 bahasa pemrograman, yaitu Otcl/Tcl dan C++. Tcl/Otcl digunakan pada script simulasi yang ditulis oleh NS *user* dan pada *library* sebagai *simulator object*, otcl juga berperan sebagai interpreter. Sedangkan C++ digunakan untuk library yang berisi *event scheduler*, protokol dan *network component* yang diimplementasikan pada simulasi oleh user. Komunikasi antara Otcl/Tcl dan C++ dapat dilihat dari gambar 2.13.



Gambar 2. 13 Komunikasi Otcl dan C++

Bahasa C++ digunakan pada library karena mampu mendukung *runtime* simulasi yang cepat, meskipun simulasi melibatkan jumlah *packet* dan sumber data yang besar. Sedangkan bahasa Tcl memberikan respon *runtime* yang lebih lambat dari C++, tetapi jika terdapat kesalahan syntax dan perubahan script berlangsung cepat dan interaktif. *User* dapat mengetahui letak kesalahannya yang dijelaskan pada *console*, sehingga *user* dapat memperbaiki dengan cepat. Terdapat 3 bagian dalam membangun simulasi di NS2, yaitu program utama, program membuat *node* dan program untuk sending data. Struktur directory pada NS2 dapat dilihat pada gambar 2.14.



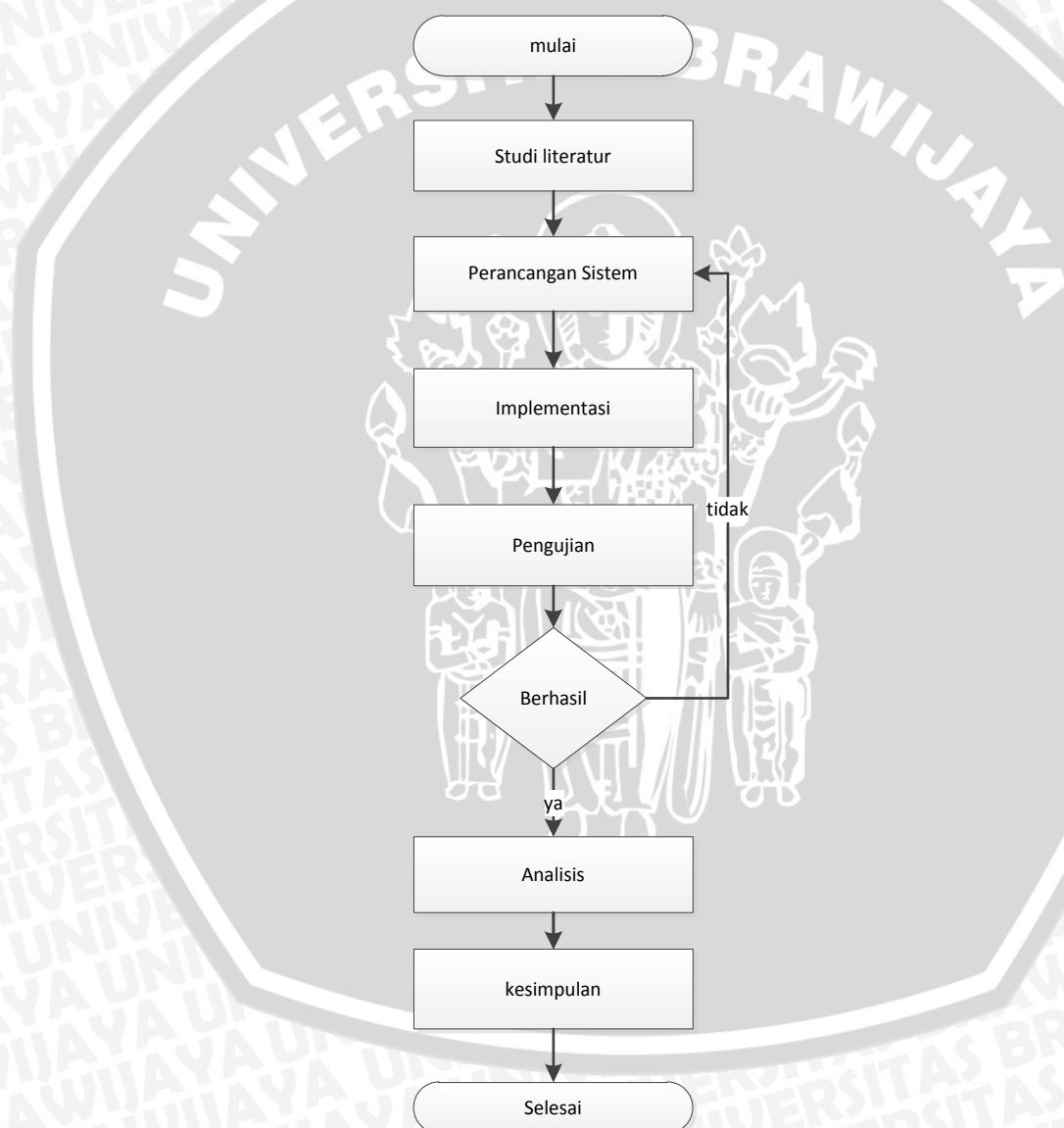
Gambar 2. 14 Struktur directory NS2

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1. Metode Penelitian

Bagian metodologi menjelaskan tahapan-tahapan yang digunakan dalam penelitian. Tahapan-tahapan penelitian digambarkan dalam bentuk diagram alir seperti pada gambar 3.1.



Gambar 3. 1 Metode penelitian

Sumber : [Perancangan]

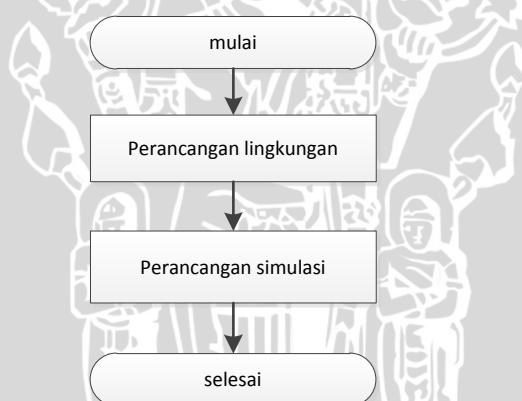
3.1.1. Studi Literatur

Studi literatur dilakukan untuk mempelajari dan memahami konsep yang terkait dengan teori-teori yang mendukung dalam perancangan dan simulasi protokol *routing* LEACH-C pada WSN. Sumber bacaan yang digunakan berupa *text book*, *paper*, dan sumber dari internet. Teori-teori yang digunakan diantaranya sebagai berikut :

- a. WSN
- b. LEACH-C
- c. NS2

3.1.2. Perancangan Sistem

Perancangan sistem simulasi WSN dengan menggunakan protokol *routing* LEACH-C diuji dengan 3 skenario yaitu *cluster* statis, *cluster* dinamis dan *backup* CH yang akan dilakukan menggunakan simulator NS2 diagram alir perancangan sistem ditunjukkan pada gambar 3.2.



Gambar 3. 2 Diagram alir perancangan sistem

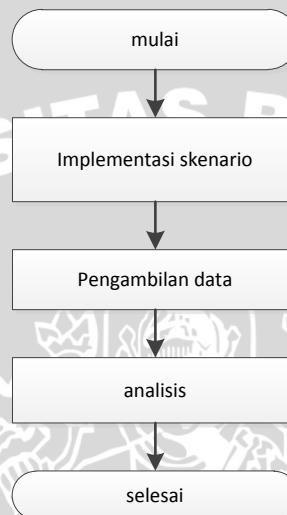
Sumber : [Perancangan]

3.1.3. Implementasi

Untuk tahap selanjutnya adalah implementasi dalam bentuk simulasi yang terdiri dari perancangan sistem dan skenario yang telah dibentuk. Simulasi dilakukan dengan menggunakan program simulator NS2.

3.1.4. Pengujian dan Analisis

Pada tahap ini, simulasi dilakukan dalam beberapa skenario dan hasilnya diolah sehingga dapat dianalisis. Pengujian ini dilakukan untuk melihat bagaimana kinerja dari protokol *routing* LEACH-C dengan perbedaan pembentukan *cluster* dengan ada atau tidak adanya *backup* CH. Tujuan utama adalah melihat bagaimana kinerja protokol pada masing-masing skenario pengujian. Tahapan pengujian dan analisis digambarkan dalam bentuk diagram alir seperti pada gambar 3.3.



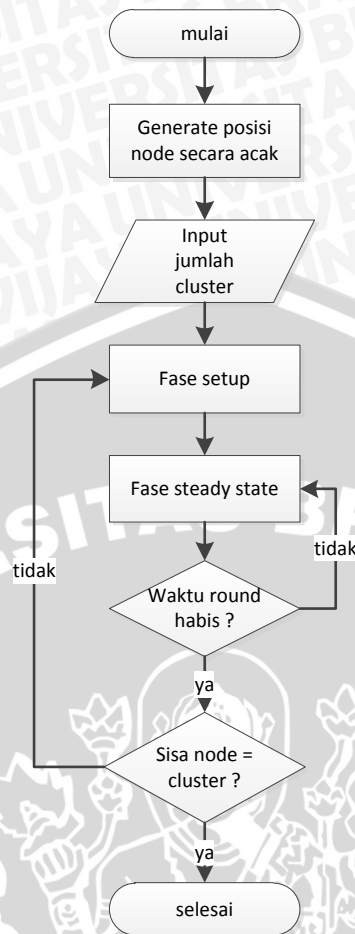
Gambar 3. 3 Diagram alir pengujian dan analisis sistem

Sumber : [Perancangan]

Pengujian dilakukan dengan mensimulasikan WSN menggunakan protokol *routing* LEACH-C pada area sensor 100x100 meter dengan jumlah 100 *node* sensor yang disusun secara *random* diujikan berdasarkan 3 skenario sebagai berikut :

1. *Cluster* Statis

Tahap pengujian pada skenario ini dengan cara mensimulasikan protokol *routing* LEACH-C, dimana jumlah *cluster* yang digunakan akan selalu sama untuk setiap *round*. Diagram alir simulasi protokol pada skenario ini dapat dilihat dari gambar 3.4.

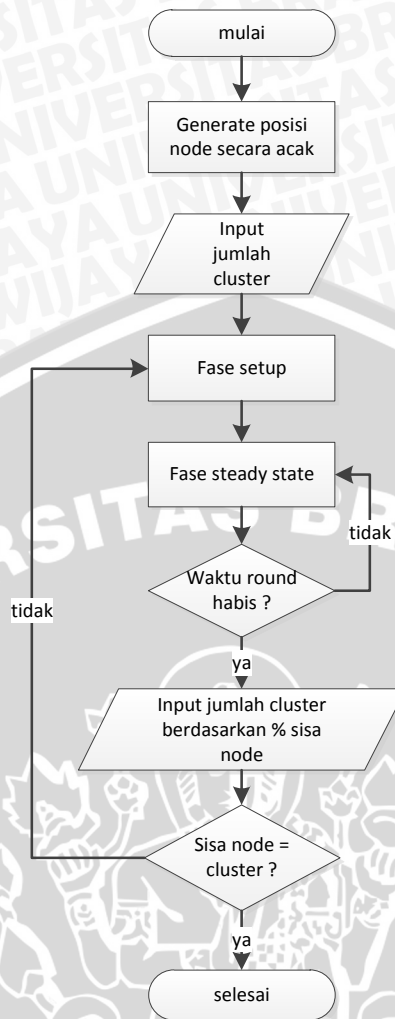


Gambar 3. 4 Diagram alir *cluster* statis

Sumber : [Perancangan]

2. Cluster Dinamis

Tahap pengujian ini dilakukan dengan cara mensimulasikan protokol *routing* LEACH-C, dimana jumlah *cluster* yang dibentuk untuk setiap *round* tidak selalu sama. Jumlah *cluster* ditentukan oleh tingkat *node sensor* yang masih hidup pada setiap *round*, semakin sedikit *node sensor* yang masih hidup maka jumlah *cluster* yang dibentuk akan semakin sedikit pula. Diagram alir simulasi pada protokol ini dapat dilihat dari gambar 3.5.

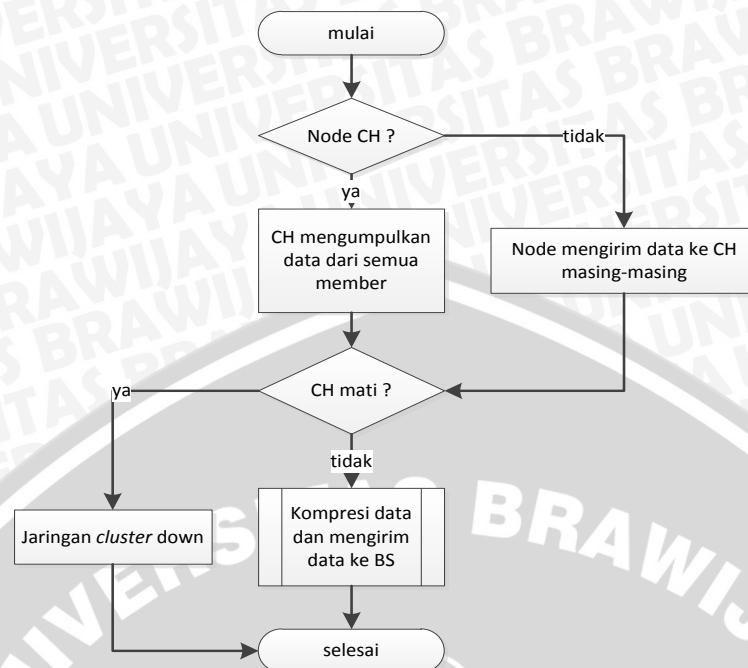


Gambar 3.5 Diagram alir *cluster dinamis*

Sumber : [Perancangan]

3. Backup CH

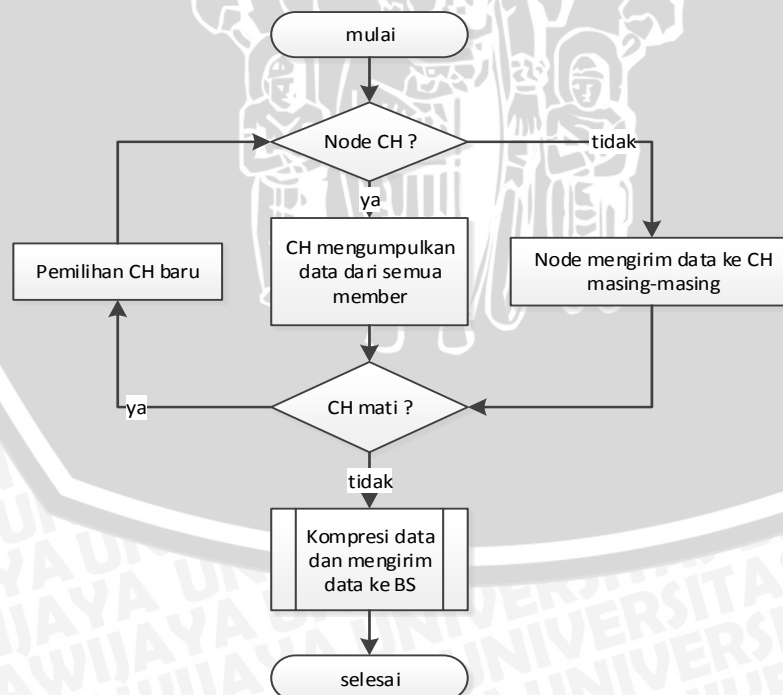
Tahap pengujian ini dilakukan dengan cara mensimulasikan protokol *routing* LEACH-C dengan menambahkan sistem *backup* CH pada skenario *cluster* statis dan dinamis. *Backup* CH berjalan pada fase *steady state*, jika terjadi fault terhadap CH (CH mati) sebelum *round* selesai. Proses berjalannya fase *steady state* tanpa menggunakan *backup* CH dapat dilihat pada gambar 3.6.



Gambar 3. 6 Diagram alir tanpa sistem *backup* CH

Sumber : [Perancangan]

Sedangkan untuk proses berjalannya fase *steady state* yang menggunakan *backup* CH dapat dilihat pada gambar 3.7.



Gambar 3. 7 Diagram alir dengan sistem *backup* CH

Sumber : [Perancangan]

Sistem *backup* CH berjalan dengan menjadikan *node* pada *cluster* tersebut dengan jarak terdekat dengan BS yang akan menjadi CH sementara hingga *round* selesai. Sistem pemilihan CH sementara ditentukan berdasarkan pertimbangan jarak terdekat dengan BS karena besar kecilnya konsumsi energi pengiriman data bergantung pada total data dan juga jarak pengirim dengan penerima. Setelah tahapan implementasi maka data file log diolah menggunakan microsoft excel kemudian dilakukan analisis. Hasil analisis akan digunakan untuk kesimpulan. Analisis protokol *routing* LEACH-C pada WSN menggunakan 4 parameter yaitu *packet delivery ratio*, *packet loss*, *lifetime* jaringan dan konsumsi energi untuk setiap simulasi.

3.1.5. Pengambilan Kesimpulan

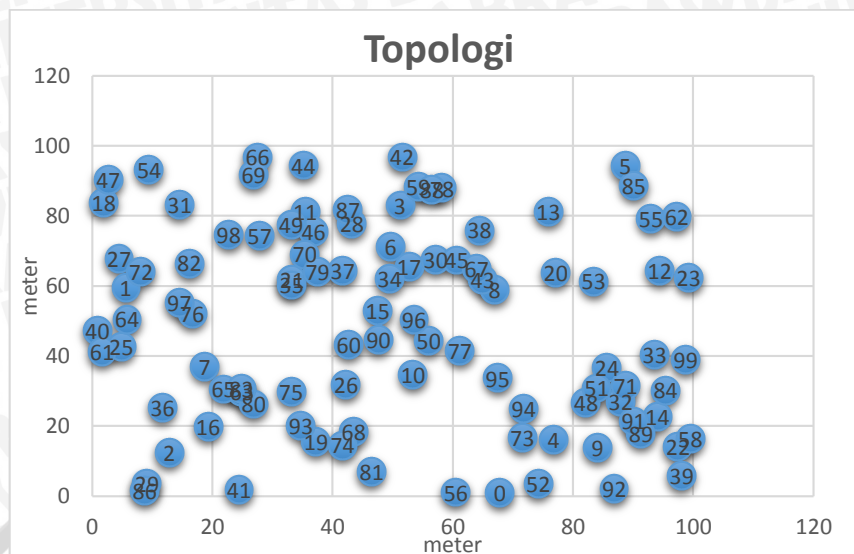
Pengambilan kesimpulan dilakukan setelah analisis data. Kesimpulan merupakan hasil akhir yang diperoleh dari kesesuaian teori dan praktek yang telah dilakukan. Selanjutnya penulisan saran untuk memperbaiki kesalahan yang terjadi untuk penelitian selanjutnya.

3.2. Perancangan

Tahapan perancangan simulasi pada jaringan WSN menggunakan protokol *routing* LEACH-C sebagai berikut.

3.2.1 Perancangan Lingkungan

Pada penelitian ini topologi jaringan *node-node sensor* dibuat secara *random* dalam area 100x100 meter dengan posisi BS (x:50, y:175) yang berada diluar area *node sensor*. Posisi *node-node sensor* dapat dilihat pada gambar 3.8.



Gambar 3. 8 Topologi node sensor

Sumber : [Perancangan]

Gambar 3.8 merupakan topologi jaringan WSN, dimana setiap titik adalah posisi *node-node sensor*. gambar 3.8 digenerate dari sekumpulan nilai posisi koordinat x dan y dari setiap *node* dapat dilihat pada lampiran 1.

3.2.2 Perancangan Simulasi

Perancangan simulasi WSN menggunakan protokol *routing* LEACH-C untuk setiap skenario memiliki parameter-parameter yang sama, parameter-parameter simulasi ini dapat dilihat pada table 3.1.

Tabel 3. 1 Parameter simulasi

Parameter	Nilai
Area sensor	100 x 100 meter
Node sensor	100 node
Energi tiap node	1 Joule
Bs location	x: 50 , y: 175
Bandwidth	1Mbps
Packet	500bytes
Waktu round	10s

Energi <i>beamforming</i>	5 nJ/bit
Energi Elektronik	50 nJ/bit
Waktu simulasi	1000
Jumlah <i>cluster</i>	5 <i>cluster</i> dan 5% <i>cluster</i> dari <i>node</i> yang masih hidup.
Ketinggian antenna	1.5m
Frekuensi	914Mhz
Kecepatan (c)	300.000.000 m/s

Dari parameter-parameter diatas maka jarak *crossover*, nilai transmisi amplifikasi pada model propagasi *free space* dan *two ray ground*, dimana nilai panjang gelombang dapat dihitung berdasarkan persamaan-persamaan sebelumnya diantaranya,

Untuk menentukan panjang gelombang (λ) dapat dihitung dari persamaan 2.8 dengan hasil sebagai berikut,

$$\lambda = \frac{3 \cdot 10^8}{914 \cdot 10^6} = 0.32822757 \text{ m/s} \quad (3.1)$$

Setelah panjang gelombang ditemukan, maka nilai energi amplifikasi model propagasi *free space* dapat dihitung dari persamaan 2.9 dengan hasil sebagai berikut,

$$\mathcal{E}_{friss_amp} = \frac{(6 \cdot 10^{-6}) \cdot (4 \cdot 3.1415926)^2}{(1 \cdot 10^6) \cdot 1 \cdot 1 \cdot 0.32822757^2} = 1.02542 \cdot 10^{-11} \text{ J/bit/m}^2 \quad (3.2)$$

Begitu juga nilai energi amplifikasi model propagasi *two ray ground* dapat dihitung dari persamaan 2.10 dengan hasil sebagai berikut,

$$\mathcal{E}_{two_ray_amp} = \frac{6 \cdot 10^{-6}}{(1 \cdot 10^6) \cdot 1 \cdot 1 \cdot 1.5^2 \cdot 1.5^2} = 1.3037 \cdot 10^{-15} \text{ J/bit/m}^4 \quad (3.3)$$

Setelah nilai panjang gelombang, ϵ_{friss_amp} dan $\epsilon_{two_ray_amp}$ didapatkan, maka langkah selanjutnya adalah menentukan parameter $d_{crossover}$ untuk membedakan penggunaan energi amplifikasi berdasarkan jaraknya menggunakan persamaan 2.11 atau persamaan 2.12.

$$d_{crossover} = \frac{(4 * 3.1415926)^2 * 1 * 1.5^2 * 1.5^2}{0.32822757^2} = 88.687159 \text{ m} \quad (3.4)$$

Atau menggunakan nilai energi amplifikasi untuk menentukan jarak *crossover*, sehingga dapat dilihat dari perhitungan sebagai berikut,

$$d_{crossover} = \sqrt{\frac{1.02542 * 10^{-11}}{1.3037 * 10^{-15}}} = 88.687159 \text{ m} \quad (3.5)$$

Sehingga untuk energi yang dibutuhkan melakukan pengiriman data dari *source node* ke *destination node* ditentukan dari energi elektronik, total data, energi amplifikasi yang digunakan dan jarak antar node, dimana energi elektronik merupakan bagian dari *digital coding*, *modulation* dan *filtering* signal sebelum dikirim ke tranmisi *amplifier*.

Sedangkan untuk pengolahan aggregasi data energi yang dibutuhkan bergantung pada banyaknya jumlah anggota dalam *cluster* dan juga besar data. Sebagai contoh untuk menghitung energi *beamforming* dengan anggota *cluster* 20 dan besar data 500 bytes dari tiap *node* maka dapat dilihat dari perhitungan sebagai berikut,

$$\begin{aligned} E_{aggregasi} &= l * JmlMember * E_{bf} \\ E_{aggregasi} &= (500 * 8) * 20 * (50 * 10^{-9}) = 4 * 10^{-4} \end{aligned} \quad (3.6)$$

Nilai persamaan 3.6 menggunakan satuan joule, maka untuk konversi satuan ke Watt didapat dari persamaan Watt= Joule/detik.

3.2.3 Perancangan Protokol Routing LEACH-C

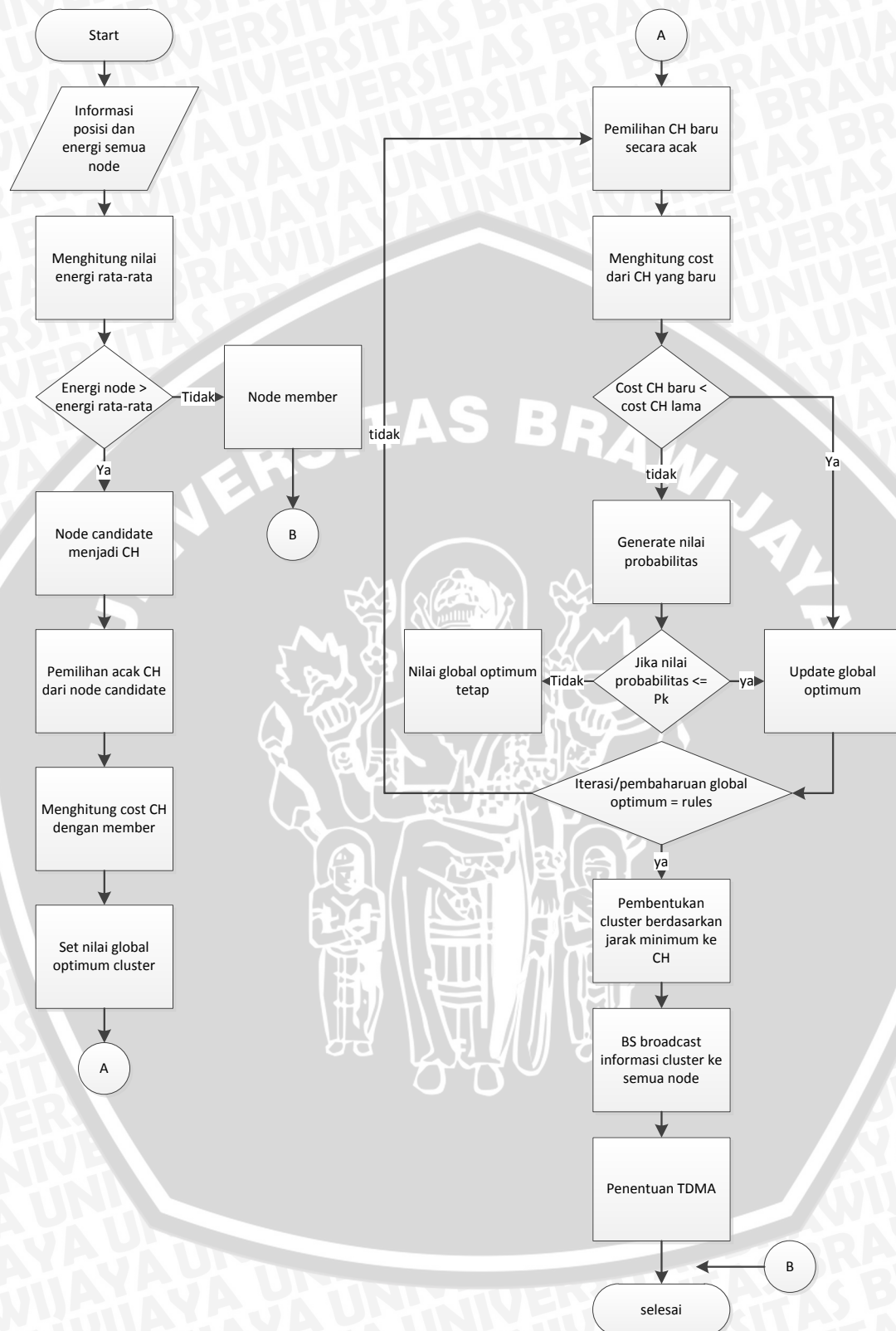
Perancangan protokol *routing* LEACH-C pada NS2 tidak dapat dibuat dengan mudah seperti mensimulasikan jaringan menggunakan simulator lain. Dalam perancangannya diperlukan mengerti tentang 2 bahasa program untuk simulasi, diantaranya TCL/Otcl dan C++. Bagian-bagian dalam perancangan diantaranya adalah,

- Pembuatan *node* secara *random* pada NS2 menggunakan perintah yang dibuat dari bahasa TCL seperti *source code* 3.1,

```
set outf [open "new100nodes.txt" w]
set num_nodes 100
set rng [new RNG]
$rng seed 0
set lim 100
for {set i 1} {$i <= $num_nodes} {incr i} {
    set x [$rng uniform 0 $lim]
    set y [$rng uniform 0 $lim]
    puts $outf "$x\t$y"
}
```

Source code 3. 1 Pembentukan *node* posisi *random*

- Menentukan jumlah *cluster* yang diinginkan pada file script
- Bagian-bagian fase *setup* pada algoritma LEACH-C dimana setiap *node* mengirimkan data lokasi dan energi ke BS, kemudian BS melakukan proses pemilihan *cluster* yang optimal berdasarkan algoritma *simulated annealing*. Setelah *cluster* optimal telah didapatkan maka BS akan mengirimkan informasi tersebut ke semua *node* untuk membentuk *cluster* berdasarkan informasi dari BS. Proses fase *setup* dapat dilihat pada gambar 3.9.



Gambar 3. 9 Fase setup algoritma *simulated annealing*

Sumber : [Perancangan]

- Setelah *cluster* terbentuk maka tahap selanjutnya adalah fase *steady state*. Pada fase ini dimana setiap CM mengirimkan data ke CH masing berdasarkan TDMA. CH menunggu data dari CM hingga semua data anggota dalam *cluster* terkumpul secara keseluruhan, maka tahap selanjutnya CH melakukan agregasi data. Setelah agregasi data selesai kemudian data dikirimkan ke BS.

Secara garis besar proses simulasi protokol *routing* LEACH-C pada WSN dapat dilihat dari tahap-tahap yang sudah dijelaskan diatas. Dengan tahap-tahap proses pembentukan *cluster*, pemilihan CH berdasarkan algoritma *simulated annealing* yang terdapat di BS, pengiriman data dari setiap *node* berdasarkan TDMA.



BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan implementasi, pengambilan data dan analisis kinerja protokol *routing* LEACH-C pada WSN yang dilakukan dengan menggunakan simulator NS2.

4.1 Implementasi Protokol *Routing* LEACH-C

Pada tahapan ini dilakukan implementasi dari masing-masing skenario. Implementasi dilakukan berdasarkan 3 skenario yaitu *cluster* statis, *cluster* dinamis dan *backup* CH dengan topologi *node random* yang telah dirancang sebelumnya pada bab 3.

4.1.1. Implementasi *Cluster* Statis

Implementasi *cluster* statis untuk protokol LEACH-C dilakukan dengan pengaturan *source code* pembentukan *cluster* pada BS untuk setiap awal *round*. Pada saat pembentukan *cluster*, jumlah *cluster* yang digunakan sama dengan input jumlah *cluster* awal. Pembentukan *cluster* pada BS dapat dilihat pada *source code* 4.1.

```
Application/BSApp instproc BSsetup {} {
    global ns_ opt
    $self instvar total_

    if {$total_ > $opt(num_clusters)} {
        [$self agent] transfer_info [expr $opt(nn) - 1] \
            $opt(num_clusters) \
            $opt(bs_setup_iters) \
            $opt(bs_setup_max_eps)

        [$self agent] BSsetup
    } else {
        puts "Only received info from $total_ nodes. - mit/uAMPS/ns-
bsapp.tcl"
        puts "There are currently $opt(nn_) alive ==> \
            $opt(num_clusters) cluster-heads needed. - mit/uAMPS/ns-
bsapp.tcl"
```

```
"sens_finish"
}
set total_ 0
$ns_ at [expr [$ns_ now] + $opt(ch_change)] "$self BSsetup"
}
```

Source code 4. 1 Proses pembentukan *cluster* statis pada BS

dilihat dari *source code* 4.1 di atas tidak terdapat penentuan jumlah *cluster* untuk *round* selanjutnya, maka jumlah *cluster* yang dibentuk sama dengan nilai jumlah *cluster* yang dimasukan pada awal simulasi.

4.1.2. Implementasi *Cluster* Dinamis

Implementasi *cluster* dinamis untuk protokol LEACH-C dilakukan dengan pengaturan *source code* pembentukan *cluster* pada BS untuk setiap awal *round*. Pada saat pembentukan *cluster*, jumlah *cluster* yang digunakan disesuaikan dengan prosentase *node* yang masih hidup pada setiap *round*. Pembentukan *cluster* dinamis dapat dilihat pada *source code* 4.2.

```
Application/BSApp instproc BSsetup {} {
    global ns_ opt
    $self instvar total_

    if { $opt(total_alive) == 0 } {
        set opt(jnn_) [expr $opt(nn)-1]
    } else {
        set opt(jnn_) $opt(total_alive)
    }

    set opt(num_clusters) [expr int(ceil(double($opt(pro_cluster) *
double($opt(jnn_))/100.0)))]
    set opt(spreading) [expr $opt(num_clusters)+1]
    # Spread-spectrum packet transmission time
    set opt(ss_slot_time) [expr $opt(slot_time) * $opt(spreading)]
    # Maximum TDMA frame time (if all nodes in one cluster)
    set opt(frame_time) [expr $opt(ss_slot_time) *
$opt(total_alive)]

    if {$total_ > $opt(num_clusters)} {
```

```

    [$self agent] transfer_info [expr $opt(nn) - 1] \
                                $opt(num_clusters) \
                                $opt(bs_setup_iters) \
                                $opt(bs_setup_max_eps)
    [$self agent] BSsetup
} else {
    puts "Only received info from $total_nodes. - mit/uAMPS/ns-
bsapp.tcl"
    puts "There are currently $opt(nn) alive ==> \
        $opt(num_clusters) cluster-heads needed. - mit/uAMPS/ns-
bsapp.tcl"
    "sens_finish"
}
set total_ 0
$ns_ at [expr [$ns_ now] + $opt(ch_change)] "$self BSsetup"
}

```

Source code 4. 2 Proses pembentukan *cluster* dinamis pada BS

source code 4.2 di atas menunjukkan bahwa terdapat pembentukan ulang untuk jumlah *cluster*, *spreading factor*, dan waktu TDMA.

4.1.3. Implementasi Backup CH

Implementasi sistem *backupCH* untuk protokol LEACH-C dilakukan dengan pengaturan *source code* pengiriman data pada setiap *node* member. Pada saat pengiriman data ke CH dilakukan pengecekan ketersediaan CH, jika CH mati maka sistem *backup* berjalan dengan melakukan pengecekan jarak semua *node* member pada *cluster* tersebut. *Node* member yang memiliki jarak terdekat akan menjadi CH sementara hingga *round* tersebut habis. Sistem *backup CH* dapat dilihat pada *source code 4.3*.

```

Application/LEACH instproc sendData {} {

    global ns_ bs node_ opt DATA MAC_BROADCAST BYTES_ID newch rng_

    $self instvar next_change_time_ frame_time_ end_frm_time_
    $self instvar currentCH_ dist_ code_ alive_ TDMAschedule_

```

```

set nodeID [$self nodeID]
set msg [list [list $nodeID , [$ns_ now]]]
set mac_dst $MAC_BROADCAST
set link_dst $currentCH_
# Use DS-SS to send data messages to avoid inter-cluster
interference.
set spreading_factor $opt(spreading)
set datasize [expr $spreading_factor * \
               [expr [expr $BYTES_ID * [llength $msg]] +
$opt(sig_size)]]
$self WakeUp

if {[[$node_($currentCH_) set rca_app_] set alive_] == 1} {
    $self send $mac_dst $link_dst $DATA $msg $datasize $dist_
    $code_
} else {
    close $aout
    return
}
} else {
    set i [$self nodeID]
    set in [lsearch $TDMAschedule_ $i]
    set panjang [llength $TDMAschedule_]
    set lost [expr $in + 1]
    puts $llost "$currentCH_ $lost [$ns_ now]"

    set indx [lsearch $TDMAschedule_ $currentCH_]
    set TDMAschedule_ [lreplace $TDMAschedule_ $indx $indx]
    set closest 200
    set y 0
    # cari member tdma dgn jarak terdekat ke bs
    foreach anggota $TDMAschedule_ {
        set dist_ [nodeToBSDist $node_($anggota) $bs]
        if {$dist_ < $closest} {
            set closest $dist_
            set y $anggota
        }
    }
}

```

```

set newch $y
set currentCH_ $newch
}

# Must transmit data again during slot in next TDMA frame.
set xmitat [expr [$ns_ now] + $frame_time_]
if {$salive_ && [expr $xmitat + $end_frm_time_] < \
    [expr $next_change_time_ - 10 *
$opt(ss_slot_time)]} {
    $ns_ at $xmitat "$self sendData"
}

set sense_energy [expr $opt(Esense) * $opt(sig_size) * 8]
[$self getER] remove $sense_energy

if {$currentCH_ != $nodeID} {
    $self GoToSleep
}
}

```

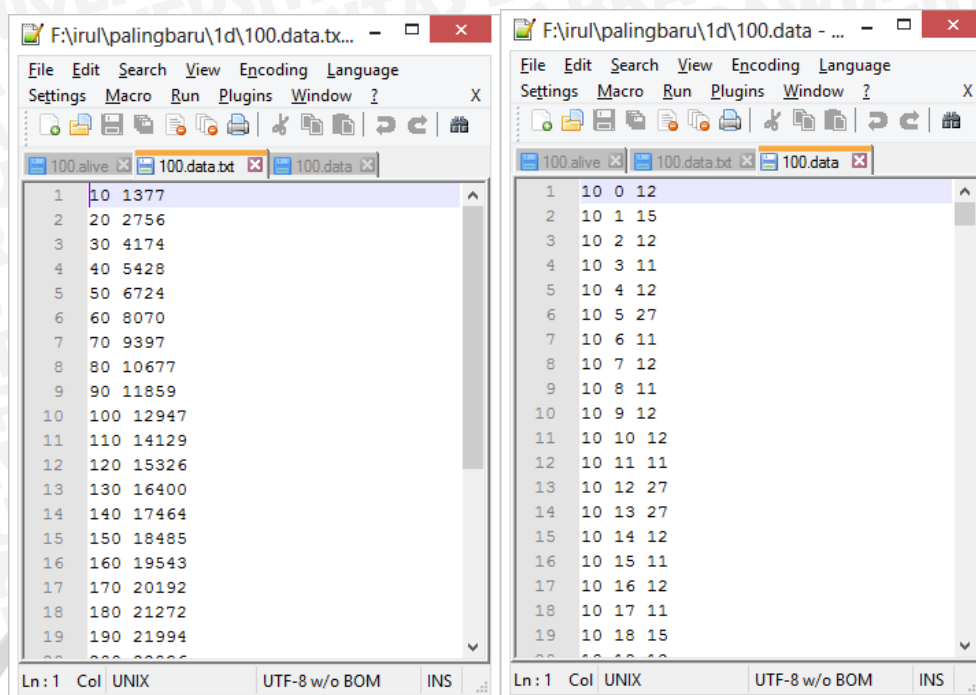
Source code 4.3 Proses pengiriman data dan *backupCH*

4.2 Pengambilan Data

Proses pengambilan data dilakukan dengan mengambil file log yang telah dibuat untuk kelangsungan analisis yang dibutuhkan. Pengambilan data dilakukan ketika setelah melakukan proses simulasi. Dari hasil pengambilan data yang dilakukan, dapat diketahui *lifetime* jaringan, banyaknya *packet* data yang dapat dikirim selama jaringan masih hidup dan konsumsi energi tiap *node* atau secara keseluruhan.

1. *Packet delivery ratio*

Untuk menganalisis *packet delivery ratio* terlebih dahulu mengambil file log data seperti pada gambar 4.1.



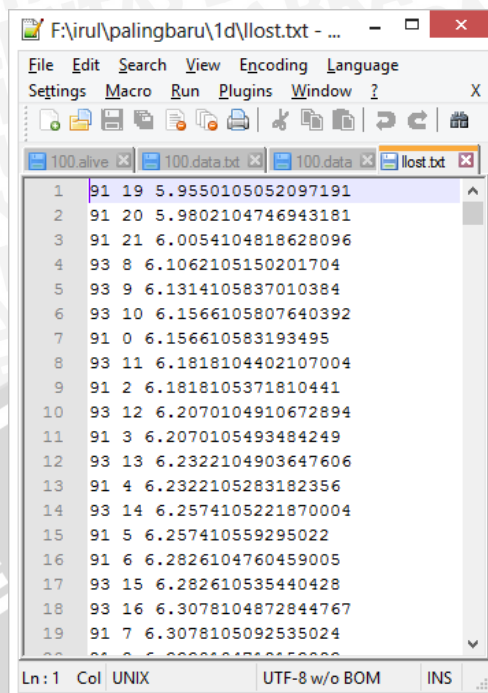
Gambar 4. 1 File log data

Sumber : [Hasil Pengujian]

dari gambar 4.1 diatas terdapat 2 file log data, diantaranya berisi berapa banyak tiap *node* mengirimkan *packet* data dalam selang waktu tertentu dan total *packet* data yang dikirim oleh keseluruhan *node* dalam selang waktu tertentu. Pembacaan dari file diatas untuk angka [10 1377] nilai 10 menyatakan waktu dalam *second* dan untuk nilai 1377 menyatakan total data yang dikirimkan oleh seluruh *node*. Sedangkan untuk angka [10 0 12] nilai 10 menyatakan waktu dalam *second*, nilai 0 adalah *nodeID* yang mengirimkan data dan nilai 12 adalah total data yang dikirimkan. Kemudian dari file log tersebut diolah ke dalam program microsoft excel dan dicari rata-rata *packet delivery ratio*.

2. Packet loss

Untuk menganalisis *packet loss* terlebih dahulu mengambil file log llost yang berisi informasi *node* member yang tidak sempat mengirimkan data ke CH, karena CH mati sebelum *round* selesai. Contoh file log llost yang telah dibuat dapat dilihat pada gambar 4.2.



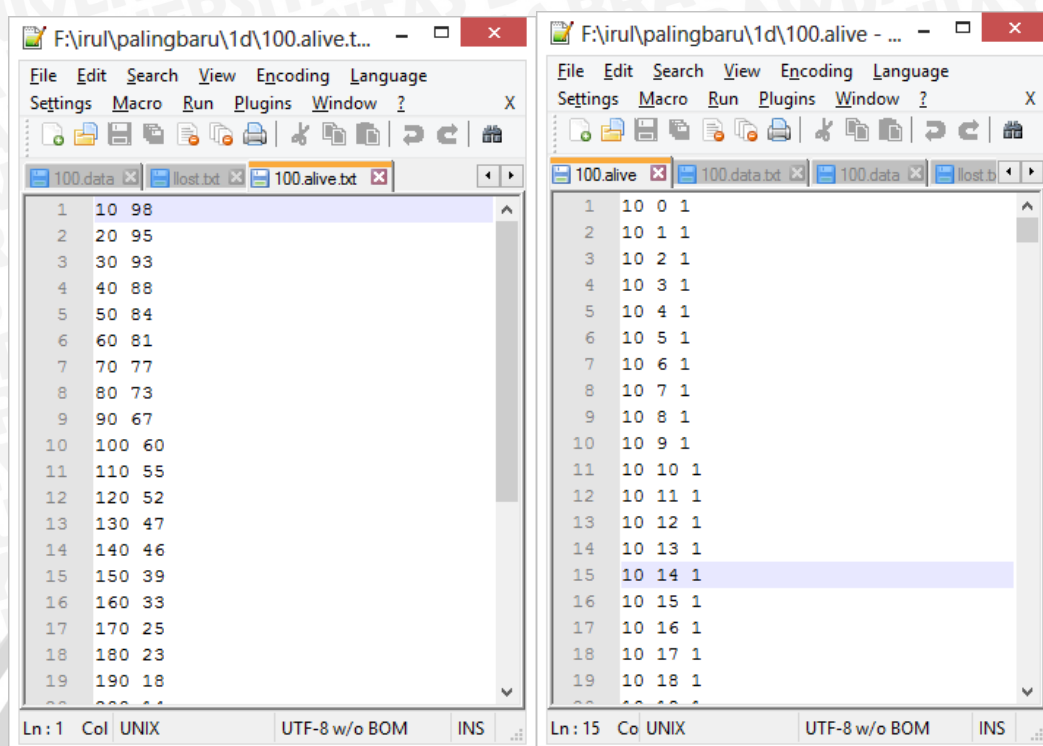
Gambar 4. 2 File log llost

Sumber : [Hasil Pengujian]

dari gambar 4.2 di atas maka didapatkan data informasi *node* member yang tidak sempat mengirimkan data ke CH dalam satu siklus pengiriman pada tiap *round*. Siklus pengiriman adalah waktu yang dibutuhkan untuk semua *node* mengirimkan satu data ke CH masing-masing, dalam satu *round* terdiri dari banyak siklus pengiriman selama sisa waktu *round*. Pembacaan nilai pada file diatas untuk angka [91 19 5.9550105052097191] nilai 91 adalah *nodeID* yang menjadi CH, nilai 19 merupakan *nodeID* yang tidak dapat mengirimkan file dikarenakan CH mati dan nilai 5.9550105052097191 merupakan waktu dalam *second* pada saat *nodeID* 19 tidak dapat mengirimkan data ke CH. Kemudian dari file log tersebut diolah ke dalam program microsoft excel dan dicari rata-rata *packet loss*.

3. *Lifetime* jaringan

Untuk menganalisis *lifetime* jaringan terlebih dahulu mengambil file log 100.alive, 100.alive.txt seperti pada gambar 4.3 dan chmati.txt yang telah dibuat seperti pada gambar 4.4.



Gambar 4. 3 File log alive dan chmati

Sumber : [Hasil Pengujian]

dari gambar 4.3 di atas terdiri dari 2 file log *alive* yaitu 100.alive.txt dan 100.alive. File log 100.alive.txt berisi lama kinerja jaringan hingga jumlah *node* yang masih hidup cara pembacaan data pada file ini untuk angka [10 98] angka 10 merupakan waktu dalam *second* dan angka 98 merupakan total *node* yang masih aktif. File log 100.alive berisi status *on/off* setiap *node* yang dijelaskan berdasarkan angka yang terdapat pada file sebagai contoh angka [10 0 1] angka 10 menunjukkan waktu dalam *second*, angka 0 merupakan *nodeID*-nya, angka 1 merupakan status dari *nodeID* tersebut dengan angka 1 menunjukkan status *on* dan 0 menunjukkan status *off* atau mati dikarenakan kehabisan energi.

```

F:\ru\palingbaru\1d\CHmati.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
TDMASchedule.150.txt TDMASchedule.160.txt TDMASchedule.170.txt New Text Document.txt CHmati.txt
1 91 5.9499999999999869 10 4.0500000000000131 5 0.81000000000000261 0.81000000000000261 9.189999999999977
2 93 6.0999999999999863 10 3.9000000000000137 5 0.78000000000000269 1.5900000000000052 8.409999999999948
3 89 15.250000000000082 20 4.749999999999983 5 0.94999999999998364 2.5399999999999889 17.460000000000012
4 36 16.250000000000096 20 3.7499999999999041 5 0.74999999999998079 3.2899999999999698 16.710000000000029
5 10 17.400000000000112 20 2.599999999999877 5 0.51999999999997759 3.8099999999999472 16.190000000000055
6 16 25.450000000000227 30 4.5499999999997733 5 0.90999999999995462 4.7199999999999021 25.280000000000097
7 48 26.100000000000236 30 3.8999999999997641 5 0.77999999999995284 5.4999999999998552 24.500000000000146
8 86 34.000000000000206 40 5.9999999999997939 5 1.1999999999999589 6.6999999999998145 33.300000000000182
9 9 34.900000000000155 40 5.0999999999998451 5 1.0199999999999689 7.7199999999997839 32.280000000000214
10 83 36.000000000000092 40 3.9999999999999076 5 0.7999999999999815 8.5199999999997651 31.480000000000235
11 33 36.350000000000072 40 3.6499999999999275 5 0.72999999999998555 9.2499999999997513 30.750000000000249
12 2 38.699999999999939 40 1.3000000000000611 5 0.26000000000001222 9.5099999999997635 30.490000000000236
13 0 43.999999999999638 50 6.0000000000003624 5 1.20000000000000726 10.7099999999999836 39.290000000000163
14 32 45.599999999999547 50 4.4000000000004533 5 0.880000000000009071 11.589999999999927 38.410000000000075
Normal text file length: 7456 lines: 73 Ln: 1 Col: 105 Sel: 104 | 0 UNIX UTF-8 w/o BOM INS
  
```

Gambar 4. 4 File log Chmati.txt

Sumber : [Hasil Pengujian]

Pada gambar 4.4 menunjukkan perhitungan efektifitas waktu yang digunakan dalam 1 *round*. Sedangkan efektifitas waktu yang digunakan dalam tiap *round* pada file Chmati.txt diperuntukkan untuk scenario pengujian tanpa *backupCH*. Pembacaan file log chmati.txt untuk angka [91 5.9499999999999869 10 4.0500000000000131 5 0.81000000000000261 0.81000000000000261 9.189999999999977] dapat diartikan sebagai berikut :

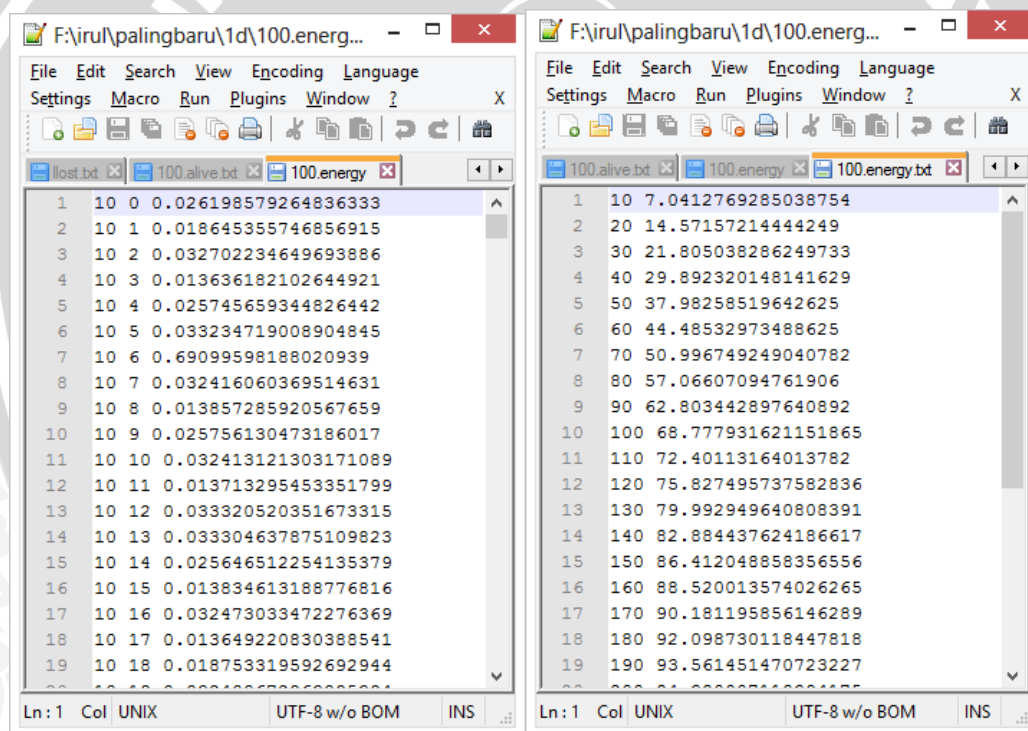
1. Angka 91 merupakan nodeID yang berperan sebagai CH
2. Angka 5.9499999999999869 merupakan waktu kapan *nodeCH* mati dikarenakan kehabisan energi.
3. Angka 10 merupakan alokasi waktu yang diberikan pada jaringan untuk beroperasi pada fase *setup* dan fase *steady state* dalam satuan *second*.
4. Angka 4.0500000000000131 merupakan sisa alokasi waktu yang terbuang dari alokasi waktu yang diberikan (*round*).
5. Angka 5 merupakan jumlah *cluster* yang digunakan pada waktu itu.
6. Angka 0.81000000000000261 merupakan waktu yang terbuang untuk keseluruhan jaringan, dimana nilai ini didapat dari sisa alokasi waktu dibagi jumlah *cluster* yang digunakan.
7. Angka 0.81000000000000261 yang kedua merupakan nilai komulatif waktu yang terbuang, dimana nilai ini akan bertambah sampai simulasi jaringan berakhir berdasarkan pada poin pembacaan data sebelumnya yaitu pada poin no 6 diatas.

8. Angka 9.1899999999999977 merupakan waktu digunakan secara efektif oleh seluruh node dalam jaringan.

Sehingga dari gambar 4.3 dan 4.4 di atas didapatkan berapa jumlah node yang masih aktif dan juga berapa efektifitas waktu yang digunakan dari waktu yang diberikan untuk setiap *round*. Kemudian dari file log tersebut diolah ke dalam program microsoft excel dan dicari rata-rata *lifetime* jaringan.

4. Konsumsi energi

Untuk menganalisis konsumsi energi setiap *node* atau keseluruhan *node* dalam jaringan terlebih dahulu mengambil file log *energy* yang telah dibuat seperti pada gambar 4.5



Gambar 4. 5 File log *energy*

Sumber : [Hasil Pengujian]

dari gambar 4.5 di atas didapatkan 2 file log *energy* yaitu 100.energy.txt dan 100.energy, dimana file log 100.energy berisi konsumsi energi tiap *node* dalam melakukan pengiriman informasi ke BS untuk setiap *round* pada fase *setup* dan total konsumsi energi dari setiap *node* untuk mengirim data ke CH atau CH ke BS dalam

satu *round*. File log 100.energy.txt berisi informasi total konsumsi energi untuk semua *node* dalam tiap *round*. Pembacaan untuk informasi pada file log 100.energy bahwa angka [10 0 0.026198579264836333] menunjukkan angka 10 sebagai waktu saat ini dalam *second*, angka 0 adalah *nodeID* dan angka 0.026198579264836333 merupakan energi yang telah digunakan. Sedangkan untuk informasi pada file log 100.energy.txt bahwa angka [10 7.0412769285038754] menunjukkan angka 10 sebagai waktu saat itu dan angka 7.0412769285038754 merupakan total energi yang digunakan oleh keseluruhan node pada waktu tersebut. Kemudian dari file log tersebut diolah ke dalam program microsoft excel dan dicari rata-rata konsumsi energi.

4.3 Hasil Pengolahan Data

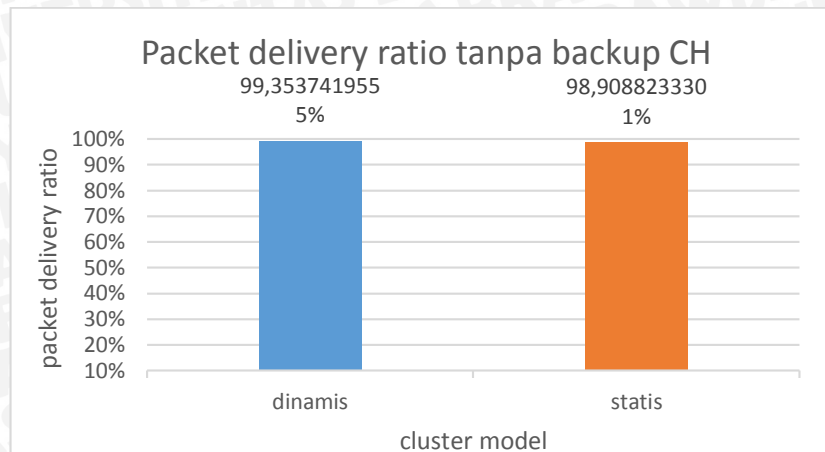
Setelah pengambilan data maka dilakukan pengolahan data yang meliputi *packet delivery ratio*, *total packet*, *packet loss*, *lifetime* jaringan, konsumsi energi.

4.3.1 Packet delivery ratio

Dari pengolahan data yang dilakukan maka diperoleh nilai *packet delivery ratio*. *Packet delivery ratio* didapatkan dari jaringan WSN menggunakan protokol *routing* LEACH-C dalam skenario *cluster* statis, *cluster* dinamis dan *backup* CH. Untuk tabel pengolahan data *packet delivery ratio* dapat dilihat pada lampiran 2.

4.3.1.1 Packet delivery ratio tanpa Backup CH

Untuk *packet delivery ratio* pada skenario *cluster* statis dan dinamis tanpa sistem *backup* CH dapat dilihat pada gambar 4.6.



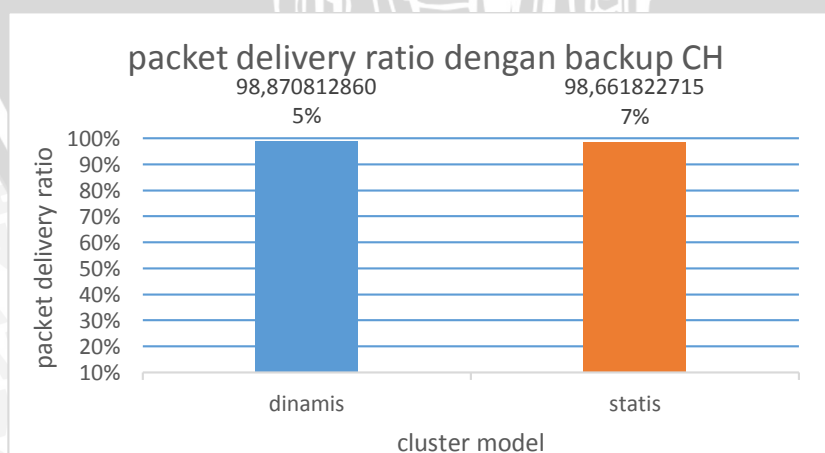
Gambar 4. 6 *Packet delivery ratio rata-rata tanpa backupCH*

Sumber : [Hasil Pengujian]

dari gambar 4.6 diatas didapatkan nilai *packet delivery ratio* dari kinerja protokol *routing* LEACH-C untuk *cluster* statis memiliki rata-rata *packet delivery ratio* yang lebih kecil sekitar 0.4% dibandingkan pada skenario *cluster* dinamis. *Packet delivery ratio* pada *cluster* dinamis 99,3537419555% sama dengan 29456 *packet* data, sedangkan untuk *cluster* statis 98,9088233301% sama dengan 20431,2 *packet* data.

4.3.1.2 *Packet delivery ratio dengan Backup CH*

Untuk *packet delivery ratio* pada skenario *cluster* statis dan dinamis sistem *backup* CH dapat dilihat pada gambar 4.7.



Gambar 4. 7 *Packet delivery ratio rata-rata dengan backupCH*

Sumber : [Hasil Pengujian]

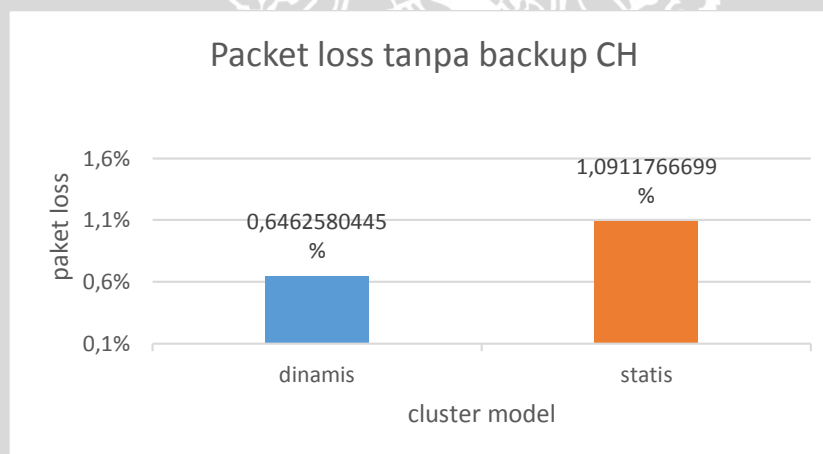
dari gambar 4.7 diatas didapatkan nilai *packet delivery ratio* dari kinerja protokol *routing* LEACH-C pada *cluster* statis memiliki rata-rata *packet delivery ratio* yang lebih kecil sekitar 0.2% dibandingkan pada *cluster* dinamis. *packet delivery ratio* pada *cluster* dinamis 98,8708128605% sama dengan 28211,6 *packet* data, sedangkan untuk *cluster* statis 98,6618227157% sama dengan 18358,4 *packet* data.

4.3.2 Packet Loss

Dari pengolahan data yang dilakukan maka diperoleh nilai *packet loss*. *Packet loss* didapat dari jaringan WSN menggunakan protokol *routing* LEACH-C dalam skenario *cluster* statis, *cluster* dinamis dan *backup* CH. Untuk tabel pengolahan data *packet loss* dapat dilihat pada lampiran 3.

4.3.2.1 Packet Loss tanpa Backup CH

Untuk *packet loss* pada skenario *cluster* statis dan dinamis tanpa sistem *backup* CH dapat dilihat pada gambar 4.8.



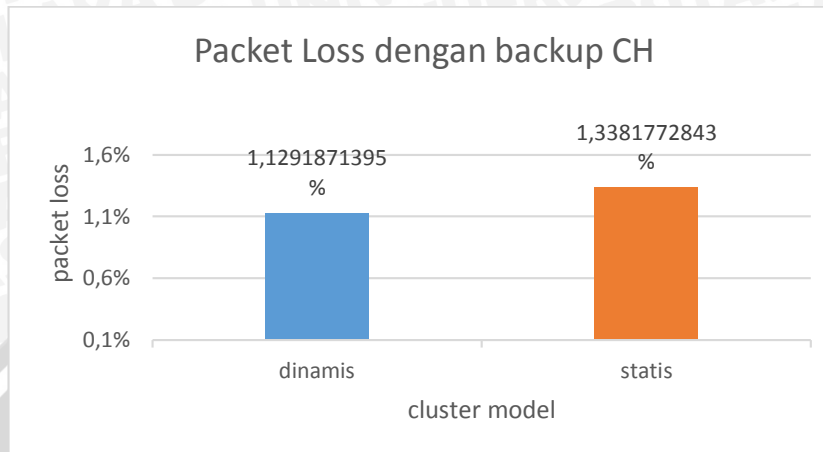
Gambar 4. 8 *Packet loss* rata-rata tanpa *backup*CH

Sumber : [Hasil Pengujian]

dari gambar 4.8 diatas didapatkan nilai *packet loss* dari kinerja protokol *routing* LEACH-C untuk *cluster* statis memiliki rata-rata *packet loss* yang lebih besar sekitar 0.3% dibandingkan pada skenario *cluster* dinamis. *Packet loss* pada *cluster* dinamis 0,6462580445% sama dengan 191,6 *packet* data, sedangkan untuk *cluster* statis 1,0911766699% sama dengan 225,4 *packet* data.

4.3.2.2 Packet Loss dengan Backup CH

Untuk *packet loss* pada skenario *cluster* statis dan dinamis dengan sistem *backupCH* dapat dilihat pada gambar 4.9.



Gambar 4. 9 *Packet loss* rata-rata dengan *backupCH*

Sumber : [Hasil Pengujian]

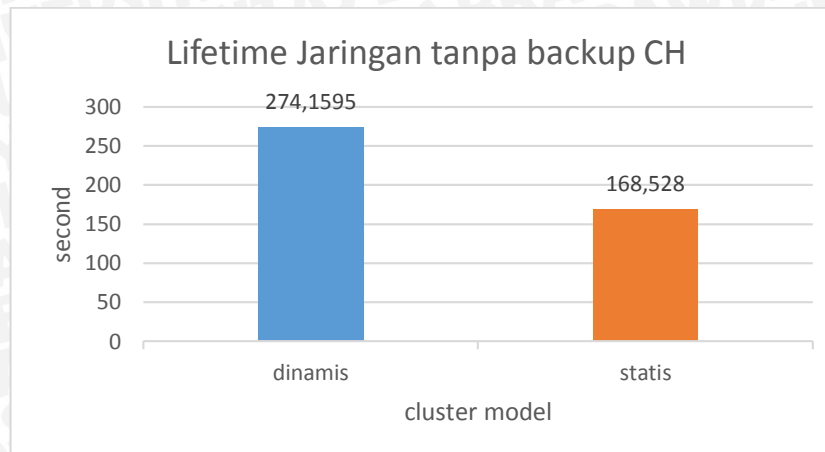
dari gambar 4.9 diatas didapatkan nilai *packet loss* dari kinerja protokol *routing* LEACH-C untuk *cluster* statis memiliki rata-rata *packet loss* yang besar sekitar 0.2% dibandingkan pada skenario *cluster* dinamis. *Packet loss* pada *cluster* dinamis 1,1291871395% sama dengan 322,2 *packet* data, sedangkan untuk *cluster* statis 1,3381772843% sama dengan 249 *packet* data.

4.3.3 Lifetime Jaringan

Dari pengolahan data yang dilakukan diperoleh nilai *lifetime* jaringan WSN menggunakan protokol *routing* LEACH-C dengan skenario *cluster* statis, dinamis dan *backupCH*. Untuk tabel pengolahan data *lifetime* jaringan dapat dilihat pada lampiran 2.

4.3.3.1 Lifetime Jaringan tanpa Backup CH

Untuk *lifetime* jaringan pada skenario *cluster* statis dan dinamis tanpa sistem *backup* CH dapat dilihat pada gambar 4.10.



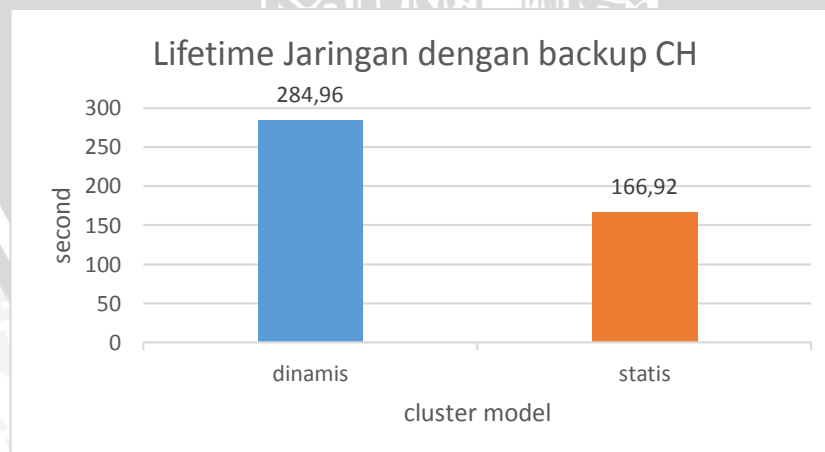
Gambar 4. 10 Lifetime jaringan rata-rata tanpa backupCH

Sumber : [Hasil Pengujian]

Dari gambar 4.10 diatas didapatkan nilai *lifetime* jaringan dari kinerja protokol *routing* LEACH-C untuk *cluster* dinamis memiliki rata-rata *lifetime* jaringan yang lebih lama sekitar 62.68% dengan 105,6315 *second* dibandingkan pada skenario *cluster* statis.

4.3.3.2 Lifetime Jaringan dengan Backup CH

Untuk *lifetime* jaringan pada skenario *cluster* statis dan dinamis dengan sistem *backup* CH dapat dilihat pada gambar 4.11.



Gambar 4. 11 Lifetime jaringan rata-rata dengan backupCH

Sumber : [Hasil Pengujian]

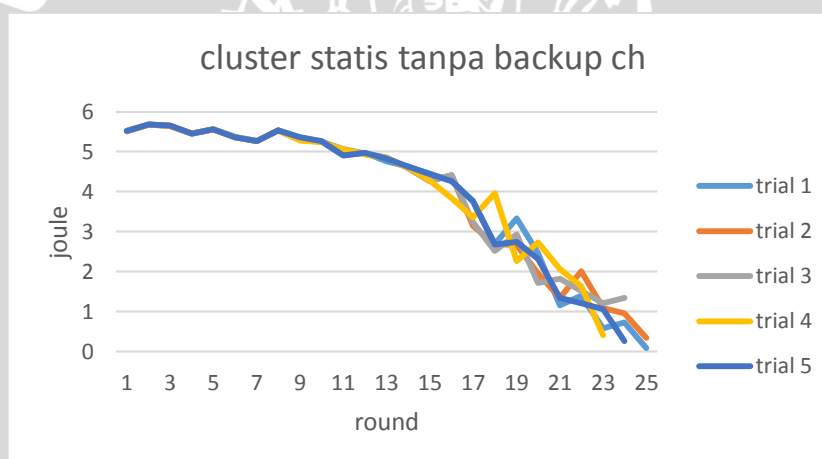
Dari gambar 4.11 diatas didapatkan nilai *lifetime* jaringan dengan sistem *backup* CH untuk kinerja protokol *routing* LEACH-C untuk *cluster* dinamis memiliki rata-rata *lifetime* jaringan yang lebih lama sekitar 71,72% dengan 118,04 *second* dibandingkan pada skenario *cluster* statis.

4.3.4 Konsumsi Energi

Dari pengolahan data yang dilakukan diperoleh nilai konsumsi energi jaringan WSN menggunakan protokol *routing* LEACH-C dengan skenario *cluster* statis, dinamis dan *backup* CH. Untuk tabel pengolahan data konsumsi energi dapat dilihat pada lampiran 2.

4.3.4.1 Konsumsi Energi tanpa Backup CH

Untuk konsumsi energi pada skenario *cluster* statis tanpa sistem *backup* CH dapat dilihat pada gambar 4.12.

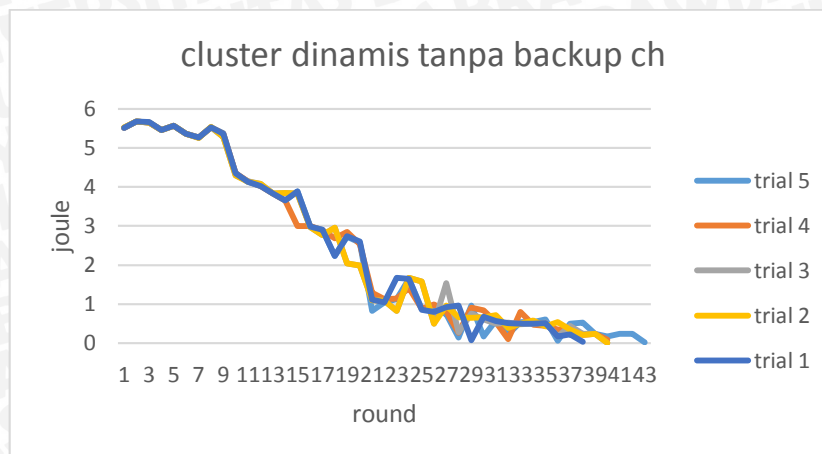


Gambar 4. 12 Konsumsi energi *cluster* statis tanpa *backup*CH

Sumber : [Hasil Pengujian]

dilihat dari gambar 4.12 diatas didapatkan nilai konsumsi energi untuk setiap *round* dengan 5 kali percobaan. Konsumsi energi rata-rata 4,255415 Joule dengan konsumsi energi antara 0,081395 Joule sampai 5,679591 Joule pada setiap *round*.

Untuk konsumsi energi pada skenario *cluster* dinamis tanpa sistem *backup* CH dapat dilihat pada gambar 4.13.



Gambar 4. 13 Konsumsi energi *cluster* dinamis tanpa *backup*CH

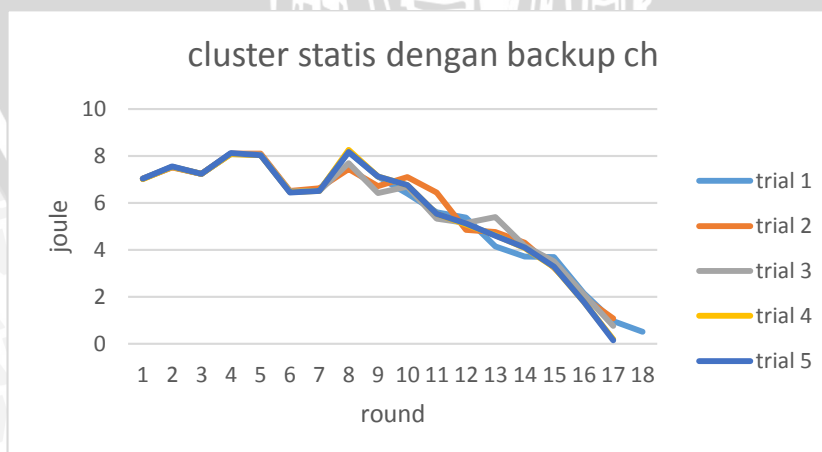
Sumber : [Hasil Pengujian]

dilihat dari gambar 4.13 diatas didapatkan nilai konsumsi energi untuk setiap *round* dengan 5 kali percobaan. Konsumsi energi rata-rata 2,546545 Joule konsumsi energi antara 0,004281 Joule sampai dengan 5,680292 Joule pada setiap *round*.

Dilihat dari hasil konsumsi energi pada gambar 5.11 dan gambar 5.12 diatas untuk *cluster* statis memiliki rata-rata konsumsi energi untuk setiap *round* yang lebih besar sekitar 1,70887 Joule dibandingkan pada skenario *cluster* dinamis.

4.3.4.2 Konsumsi Energi dengan *Backup* CH

Untuk konsumsi energi pada skenario *cluster* statis dengan sistem *backup* CH dapat dilihat pada gambar 4.14.

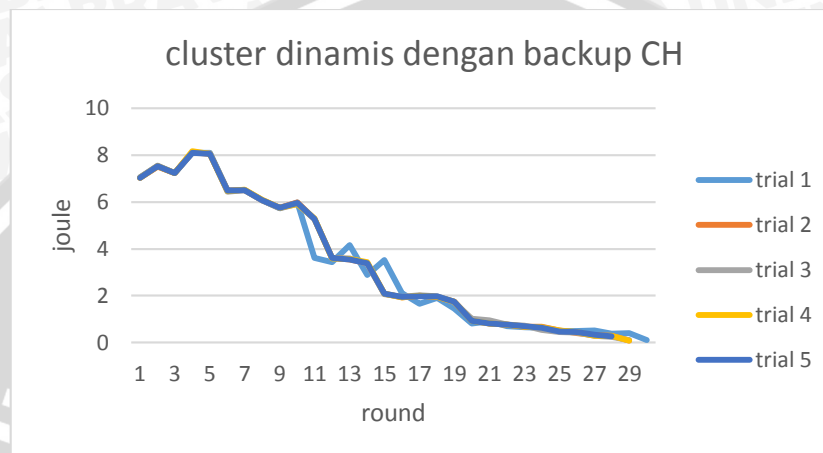


Gambar 4. 14 Konsumsi energi *cluster* statis dengan *backup* CH

Sumber : [Hasil Pengujian]

dilihat dari gambar 4.14 diatas didapatkan nilai konsumsi energi untuk setiap *round* dengan 5 kali percobaan. Konsumsi energi rata-rata 6,066067 Joule dengan konsumsi energi antara 1,076533 Joule sampai 8,121992 Joule pada setiap *round*.

Untuk konsumsi energi pada skenario *cluster* dinamis tanpa sistem *backup* CH dapat dilihat pada gambar 4.15.



Gambar 4. 15 Konsumsi energi *cluster* statis dengan *backup*CH

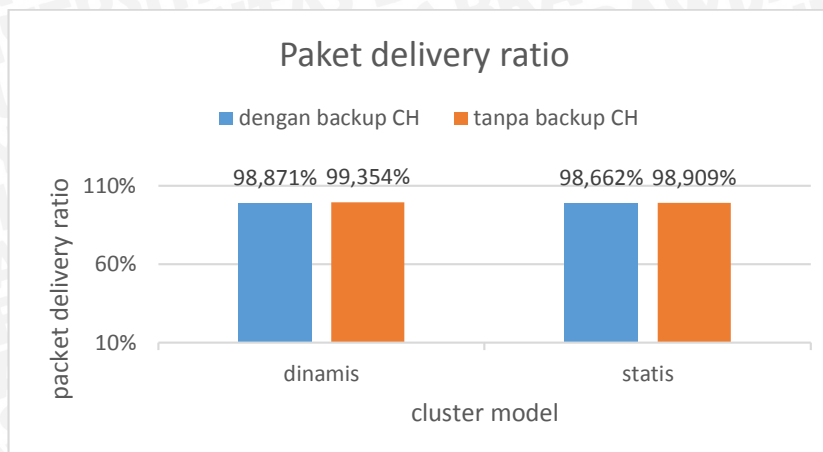
Sumber : [Hasil Pengujian]

dilihat dari gambar 4.15 diatas didapatkan nilai konsumsi energi untuk setiap *round* dengan 5 kali percobaan. Konsumsi energi rata-rata 3,564356 Joule konsumsi energi antara 0,113154 Joule sampai dengan 8,150443 Joule pada setiap *round*.

Dilihat dari hasil konsumsi energi pada gambar 4.14 dan gambar 4.15 diatas untuk *cluster* statis memiliki rata-rata konsumsi energi untuk setiap *round* yang lebih besar sekitar 2,501711 Joule dibandingkan pada skenario *cluster* dinamis.

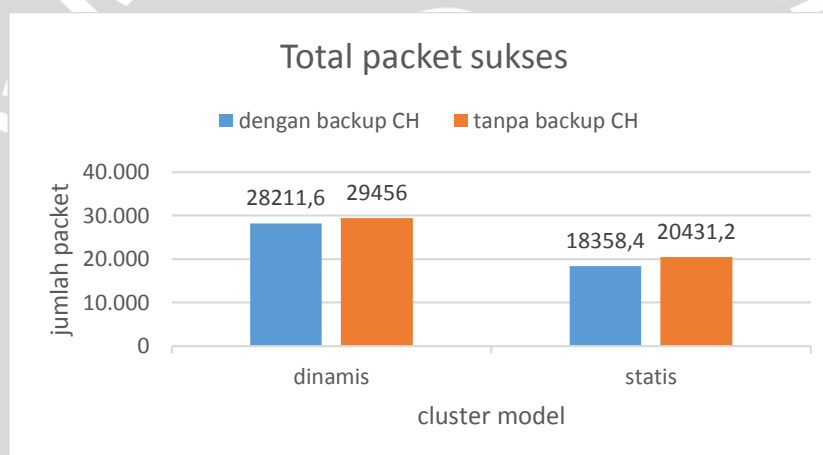
4.4 Analisis Protokol Routing LEACH-C

Berdasarkan implementasi dan pengolahan data yang telah dilakukan maka didapatkan grafik *packet delivery ratio*, *packet loss*, *lifetime* jaringan dan konsumsi energi. Kinerja protokol routing LEACH-C dibandingkan dengan setiap parameter untuk skenario *cluster* statis, *cluster* dinamis dan *backup* CH. Perbandingan *packet delivery ratio* dan jumlah total *packet* sukses untuk setiap skenario dapat dilihat pada gambar 4.16 dan gambar 4.17.



Gambar 4. 16 *Packet delivery ratio* dengan dan tanpa *backup CH*

Sumber : [Hasil Pengujian]

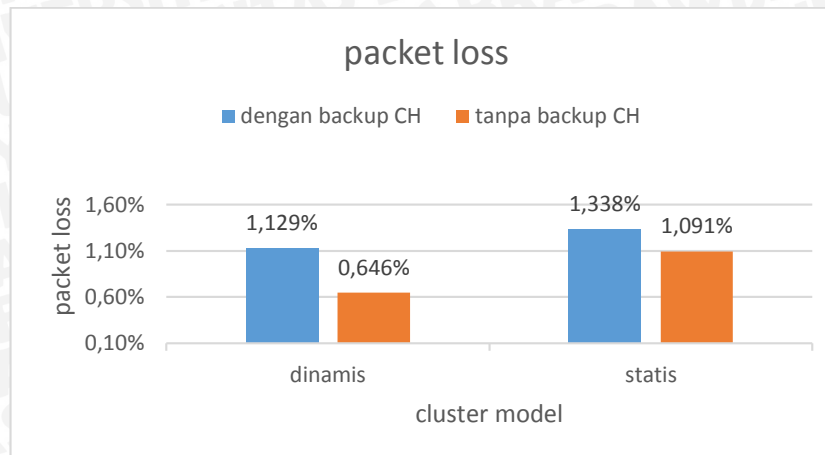


Gambar 4. 17 *Total packet sukses* dengan dan tanpa *backupCH*

Sumber : [Hasil Pengujian]

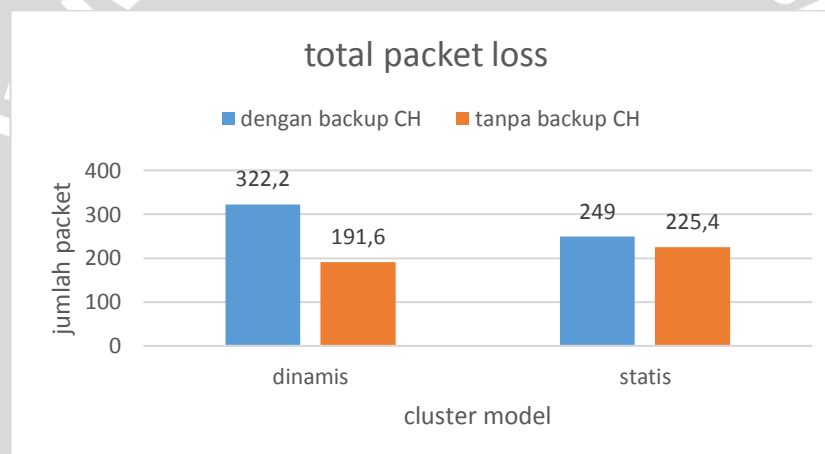
dari gambar 4.16 dan 4.17 diatas mendapatkan bahwa dari segi *packet delivery ratio* untuk setiap skenario tidak menunjukkan perbedaan yang signifikan antara penggunaan *backup CH* dan tanpa penggunaan *backup CH*. Sedangkan nilai *packet delivery ratio* untuk skenario *cluster* dinamis tanpa *backup CH* memiliki nilai yang tertinggi 99.354% dengan total data terbanyak yaitu 29456 *packet*.

Perbandingan *packet loss* dan jumlah total *packet loss* untuk setiap skenario dapat dilihat pada gambar 4.18 dan gambar 4.19.



Gambar 4. 18 *Packet loss* dengan dan tanpa *backupCH*

Sumber : [Hasil Pengujian]

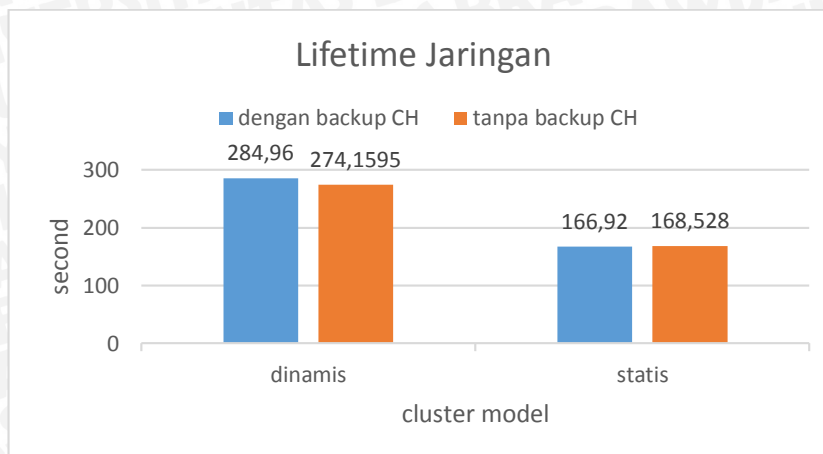


Gambar 4. 19 *Total packet loss* dengan dan tanpa *backupCH*

Sumber : [Hasil Pengujian]

dari gambar 4.18 dan 4.19 diatas dapat disimpulkan bahwa *packet loss* untuk penggunaan sistem backup CH pada skenario yang sama tidak menunjukkan perbedaan yang signifikan, tapi nilai *packet loss* untuk skenario *cluster* dinamis tanpa *backup CH* memiliki nilai *packet loss* terendah 0.646% dengan total data loss sebanyak 191,6 *packet*.

Perbandingan *lifetime* jaringan untuk setiap skenario dapat dilihat pada gambar 4.20.

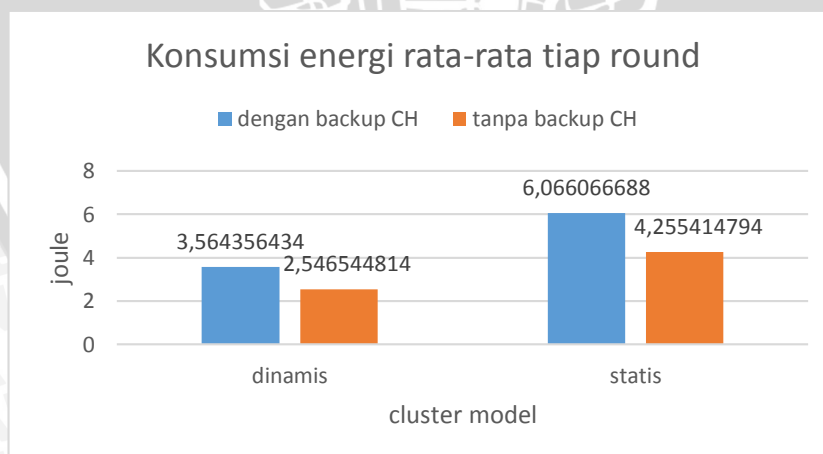


Gambar 4. 20 *Lifetime jaringan dengan dan tanpa backupCH*

Sumber : [Hasil Pengujian]

dari gambar 4.20 diatas dapat disimpulkan bahwa *lifetime* jaringan untuk *cluster* model yang sama tidak menunjukkan perbedaan yang signifikan antara penggunaan backup CH dan tanpa backup CH. Sedangkan nilai *lifetime* jaringan untuk skenario *cluster* dinamis dengan *backup* CH memiliki nilai *lifetime* jaringan 3,9395% lebih baik dengan total *lifetime* jaringan 284,96 second. Namun berbeda dengan *cluster* statis yang mengalami penurunan kinerja *lifetime* jaringan 0.34% pada penggunaan sistem *backup*CH.

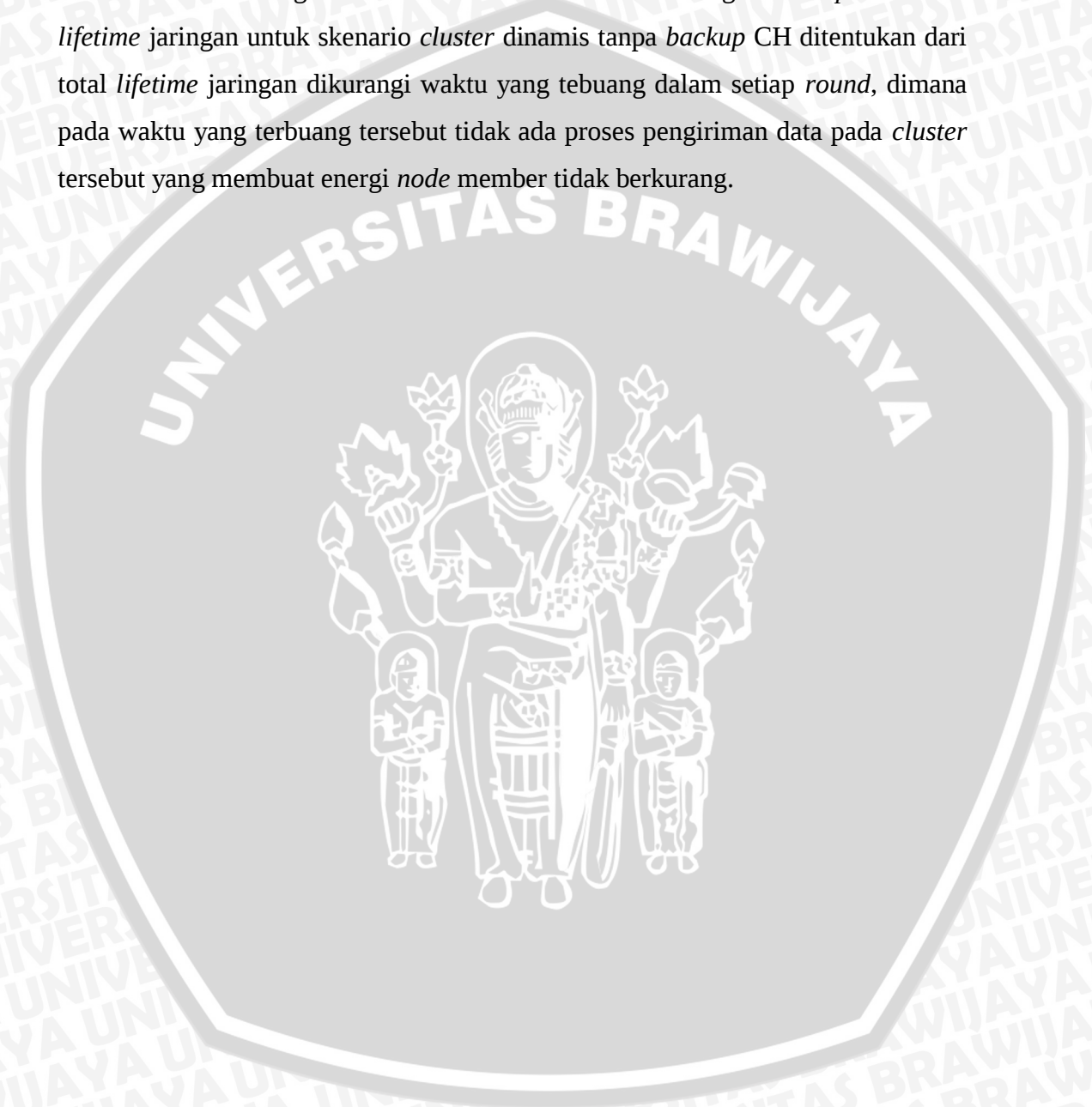
Perbandingan konsumsi energi untuk setiap skenario dapat dilihat pada gambar 4.21.



Gambar 4. 21 Konsumsi energi dengan dan tanpa *backup*CH

Sumber : [Hasil Pengujian]

dari gambar 4.21 diatas menunjukkan bahwa konsumsi energi terbesar pada skenario *cluster* statis untuk setiap *round*, sehingga *lifetime* jaringan pada skenario tersebut sangat singkat. Konsumsi energi terendah untuk setiap *round* terjadi pada skenario *cluster* dinamis tanpa *backup* CH, tapi nilai *lifetime* jaringan untuk skenario ini lebih singkat dari skenario *cluster* dinamis dengan *backup* CH. Nilai *lifetime* jaringan untuk skenario *cluster* dinamis tanpa *backup* CH ditentukan dari total *lifetime* jaringan dikurangi waktu yang tebuang dalam setiap *round*, dimana pada waktu yang terbuang tersebut tidak ada proses pengiriman data pada *cluster* tersebut yang membuat energi *node* member tidak berkurang.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. Pada protokol *routing* LEACH-C, *cluster* dinamis memiliki kinerja yang lebih baik dengan peningkatan parameter *packet delivery ratio* 0.4%, total *packet* 60,5% dan *lifetime* jaringan 62.68%, namun menurunkan parameter *packet loss* 0,4% dan konsumsi energi 67.11% dibandingkan dengan *cluster* statis,
2. Pada protokol *routing* LEACH-C, penggunaan sistem *backupCH* pada *cluster* dinamis mengalami peningkatan parameter *lifetime* jaringan 3,94%, namun menurunkan total *packet* data 4,23% dibandingkan *cluster* dinamis tanpa *backupCH*. Sedangkan penggunaan sistem *backupCH* pada *cluster* statis dapat menurunkan nilai parameter *packet delivery ratio* 0,34% dan *lifetime* jaringan 0,95%, namun mengalami peningkatan *packet loss* 0,34% dan konsumsi energi 42,5%,
3. Pada protokol *routing* LEACH-C, untuk parameter *packet delivery ratio*, total *packet*, *packet loss* dan konsumsi energi paling optimal diperoleh pada skenario pembentukan *cluster* dinamis tanpa menggunakan sistem *backupCH* dengan nilai parameter *packet delivery ratio* 99,354%, total *packet* 29.456, *packet loss* 0,646% dan konsumsi energi rata-rata untuk setiap *round* sebesar 2,546J. Sedangkan untuk parameter *lifetime* jaringan yang optimal diperoleh pada skenario pembentukan *cluster* dinamis menggunakan *backupCH* selama 284,96s.

5.2 Saran

Saran untuk pengembangan sistem WSN selanjutnya antara lain :

1. Untuk pengembangan lebih lanjut mencoba diujikan dengan menggunakan metode pembentukan *cluster* yang lain, yang kemudian hasilnya dibandingkan dengan metode pembentukan *cluster simulated annealing*.

2. Untuk pengembangan lebih metode *backup* CH mengambil dari bobot antara jarak ke BS dan sisa energi agar fase *Steady State* berjalan lebih optimal untuk setiap *round*.
3. Untuk penelitian selanjutnya dapat dilakukan dengan membandingkan protokol *routing* berbasis *cluster* yang lain seperti LEACH, T-LEACH, AZR-LEACH, H-LEACH, dll.



DAFTAR PUSTAKA

Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H. (2000). Energyefficient communication protocol for wireless microsensor networks. *Proc 33rd Hawaii International Conference on Sistem Sciences* (pp. 1–10).

Syarif, M.I., Djanali, S., & Shiddiqi, A.M. (2011). Resource Aware Data Stream dan Frequent Item Dengan Distance Vector Routing Pada Wirelss Sensor Networks. *Seminar Nasional Pasca Sarjana XI ITS*. Surabaya.

Abad, M.F.K. & Jamali, M.A.J. (2011). Modify LEACH Algorithm for Wireless Sensor Network. *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 5, No 1.

Anastasi, G., Conti, M., Francesco, M.D. & Passarella, A. (2008). *Energy Conservation in Wireless Sensor Networks : A Survey*.

Azharuddin, M., Kuila, P., & Jana, P. K. (2013). A distributed fault-tolerant clustering algorithm for Wireless Sensor Networks. In *Advances in Computing, Communications and Informatics (ICACCI)*, 2013 International Conference on (pp. 997-1002). IEEE.

Younis, O., Krunz, M., & Ramasubramanian, S. (2006). *Node clustering in Wireless Sensor Networks: recent developments and deployment challenges*. Network, IEEE, 20(3), 20-25.

Afsar, M. M. (2014). Maximizing the reliability of clustered sensor networks by a fault-tolerant service. In *Electrical and Computer Engineering (CCECE)*, 2014 IEEE 27th Canadian Conference on (pp. 1-8). IEEE.

Gupta, G., & Younis, M. (2003). Fault-tolerant *clustering* of *Wireless Sensor Networks*. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE* (Vol. 3, pp. 1579-1584). IEEE.

Michal'ík, M. (2013). *Base station for Wireless Sensor Network*. Masaryk University. Brno, Czech Republic.

Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in *Wireless Sensor Networks: a survey*. *Wireless communications, IEEE*, 11(6), 6-28.

Giannattasio, G., Erfanian, J., Wills, P., Nguyen, H. Q., Croda, T., Rauscher, K., Fernando, X., Pavlidou, N., Daniel, K. W. (2009). *A Guide to the Wireless Engineering Body of Knowledge (WEBOK)*. IEEE Communications Society. John Wiley & Sons. Hal. 4



LAMPIRAN

Lampiran 1 Topologi

NodeID	X	Y	NodeID	X	Y
0	67.837825076579	0.953653222394014	50	55.968540839836	44.3263267373323
1	5.610634761681	59.5673870572668	51	83.906006107063	30.9128477847729
2	12.847064953692	12.4737983627588	52	74.199836083781	3.56209604235463
3	51.243302110230	83.0155942975616	53	83.447930488478	61.3618835626923
4	76.816643763714	16.0925414953811	54	9.487527613289	93.2204855108729
5	88.698179828328	94.4149853169987	55	92.960198034048	79.2755930122339
6	49.698749394016	71.1633239272811	56	60.493715647838	1.0870899544503
7	18.752563846648	37.0406073690581	57	27.881951736231	74.271631601393
8	66.948251550527	58.9997790097257	58	99.528413638253	16.4732080960987
9	84.093159662603	13.7604613386841	59	54.228790409038	88.4384806214079
10	53.398475681151	34.7515692630557	60	42.668555044880	43.1404478117546
11	35.437237767241	81.2523068307211	61	1.638886798936	41.1901082104025
12	94.395684168858	64.2824611460242	62	97.267456072042	79.77069564153
13	75.900641584723	81.1163760168088	63	24.913105892443	29.8315279324685

14	94.025061975245	22.7416399040918	64	5.782040490667	50.5416135073367
15	47.483933645992	53.0243885018976	65	21.849933230248	30.6090108727146
16	19.265831783072	19.7613818197331	66	27.509063960756	96.7801976002660
17	52.722785367036	65.422820470027	67	63.979926735153	64.9052565288288
18	1.909147948916	83.8665231987212	68	43.507665695393	18.4784889772899
19	37.249374872655	15.680148832351	69	26.849475981132	91.7742018549582
20	77.114107076597	63.7601259461418	70	35.349628299172	69.0825258703355
21	33.241755344552	61.7928638876382	71	88.727382099595	31.5756797937563
22	97.444312226700	14.0589539492777	72	8.121474230719	64.0834343918056
23	99.232554295674	62.4387373041541	73	71.549202907621	16.7075309048907
24	85.631965047508	36.7027257740044	74	41.656597723093	14.5847000249590
25	4.864080625057	42.6727737033147	75	33.102920946247	29.7944790822428
26	42.221173849991	31.7758945430516	76	16.622054072387	52.0843752436732
27	4.485753273818	67.8926330375916	77	61.164900735563	41.4673043142386
28	43.204884670305	77.7508595854746	78	58.178306258367	87.8892764392724
29	9.115077419726	3.68771637030305	79	37.488581211068	64.1613501888519
30	57.134931048907	67.5559286342728	80	26.884297526853	26.3497081242267
31	14.544329705902	83.1574465535382	81	46.483757135684	7.01542608766603
32	87.932448735429	27.0079356743991	82	16.154014885497	66.5223071195754
33	93.579592552772	40.2996491828466	83	24.824923754123	30.7508537688995

34	49.493132368425	62.1550765177957	84	95.379973899284	30.2772480204129
35	33.177067634266	60.2348743752738	85	90.106802429122	88.6913625005126
36	11.702478682484	25.3155992018597	86	8.730517844078	1.63336219342116
37	41.669898173618	64.4830178769692	87	42.558253343477	81.8036340557986
38	64.442092443091	75.9828965999106	88	56.511843463644	87.4212413036363
39	98.054823837269	5.85994934936051	89	91.323620961664	17.8565664765688
40	0.874982588401	47.2045200165382	90	47.633418509566	44.7426022704423
41	24.442630365697	1.7647778157912	91	90.056318645392	21.4099018468567
42	51.564891474119	96.7865238882538	92	86.859871534100	2.03502946628026
43	64.973403497121	61.7752482470941	93	34.766157825834	20.0740745384591
44	35.148001525154	94.4527312156058	94	71.793812826180	25.0121765420828
45	60.627286350647	67.3761132021323	95	67.471484033145	33.6675201233790
46	36.826605739457	75.5734529698143	96	53.524006834963	50.4010014470671
47	2.670529672257	90.4490261759837	97	14.493734955086	55.2594336007067
48	82.141417955114	26.5940365971038	98	22.725265204313	74.5333212774867
49	33.105320405730	77.5564664870297	99	98.747841361327	38.9084607543928

Lampiran 2 *Packet delivery ratio, Lifetime Jaringan dan Konsumsi Energi*

Cluster statis tanpa backupCH

Percobaan 1

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,515033996	1216	1216
20	96	11,19218443	2456	1240
30	94	16,83495959	3718	1262
40	91	22,28684098	4814	1096
50	88	27,84906343	5892	1078
60	86	33,21629261	7055	1163
70	84	38,48122838	8249	1194
80	82	44,01485658	9449	1200
90	78	49,3743766	10562	1113
100	75	54,62452305	11720	1158
110	70	59,52528272	12870	1150
120	66	64,49406191	13935	1065
130	61	69,25970318	14844	909
140	55	73,88857268	15732	888
150	51	78,34137971	16773	1041
160	49	82,60445288	17791	1018
170	43	86,37502831	18468	677
180	38	89,06744894	18817	349
190	30	92,40321994	19334	517
200	22	94,8631549	19755	421
210	17	96,01489204	20030	275
220	14	97,40986874	20267	237
230	9	97,99531315	20419	152
240	6	98,72749094	20570	151
240,4	5	98,80888602	20579	9

Percobaan 2

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,515887296	1216	1216
20	96	11,19529186	2456	1240
30	94	16,84299016	3718	1262
40	91	22,29855227	4814	1096
50	88	27,85896577	5892	1078
60	86	33,21971947	7055	1163
70	84	38,48452137	8249	1194
80	81	44,02163846	9448	1199
90	78	49,31111777	10558	1110
100	75	54,56660659	11716	1158
110	69	59,62344718	12862	1146
120	66	64,57264158	13933	1071
130	62	69,42309883	14921	988
140	54	74,01001344	15780	859
150	51	78,27385787	16666	886
160	47	82,68255299	17637	971
170	40	85,8207049	18226	589
180	34	88,50313377	18736	510
190	27	91,16405309	19276	540
200	19	93,12279624	19548	272
210	16	94,45855074	19832	284
220	13	96,46783901	20065	233
230	10	97,55059038	20283	218
240	6	98,49838739	20423	140
242,2	5	98,83150978	20451	28

Percobaan 3

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,513208483	1216	1216
20	96	11,19430186	2456	1240
30	94	16,8457252	3718	1262
40	91	22,2981396	4813	1095
50	88	27,8585531	5891	1078
60	86	33,22086243	7037	1146
70	84	38,48408122	8231	1194
80	81	44,01145882	9431	1200
90	78	49,30014378	10541	1110
100	75	54,55314933	11699	1158
110	69	59,62144708	12840	1141
120	66	64,57707831	13911	1071
130	62	69,43174635	14895	984
140	54	74,0183672	15754	859
150	51	78,27994748	16640	886
160	47	82,68617868	17585	945
170	41	85,91716328	18267	682
180	35	88,42706294	18648	381
190	27	91,36085382	19241	593
200	22	93,07391148	19448	207
210	18	94,89355799	19670	222
220	13	96,4053134	19889	219
230	9	97,61608178	20122	233
236,1	5	98,95516224	20268	146

Percobaan 4

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	5,516270872	1216	1216
20	96	11,19586184	2456	1240
30	94	16,8413806	3718	1262
40	91	22,30051573	4813	1095
50	88	27,86475828	5891	1078
60	86	33,2274319	7037	1146
70	84	38,49192829	8231	1194
80	81	44,02523762	9431	1200
90	78	49,30958106	10541	1110
100	75	54,55339901	11698	1157
110	70	59,60761967	12832	1134
120	66	64,54365698	13893	1061
130	62	69,38727509	14882	989
140	54	73,98135687	15749	867
150	50	78,26120108	16636	887
160	47	82,10486347	17448	812
170	42	85,46480824	18132	684
180	32	89,41660391	18795	663
190	26	91,68462964	19240	445
200	21	94,40801859	19596	356
210	14	96,46646946	19974	378
220	8	98,09757409	20229	255
222,4	5	98,50442869	20295	66

Percobaan 5

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	5,517498753	1216	1216

20	96	11,19421943	2456	1240
30	94	16,84191773	3718	1262
40	91	22,29922105	4814	1096
50	88	27,86389183	5892	1078
60	86	33,22428553	7038	1146
70	84	38,48808657	8232	1194
80	82	44,01855488	9432	1200
90	78	49,37335912	10555	1123
100	75	54,63704183	11713	1158
110	70	59,54191934	12863	1150
120	65	64,50944569	13928	1065
130	61	69,3337374	14859	931
140	55	73,96139362	15747	888
150	51	78,39960517	16787	1040
160	49	82,66895197	17799	1012
170	43	86,42366338	18469	670
180	37	89,10433	18825	356
190	29	91,84977415	19343	518
200	24	94,16769464	19706	363
210	18	95,50284722	20060	354
220	13	96,70771753	20305	245
230	7	97,77228763	20537	232
231,3	5	98,030257	20563	26

Cluster dinamis tanpa backupCH

Percobaan 1

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	5,513182459	1216	1216
20	96	11,19258703	2456	1240
30	94	16,84822946	3718	1262

40	91	22,29824964	4808	1090
50	88	27,86312177	5886	1078
60	86	33,2237232	7032	1146
70	84	38,4844911	8226	1194
80	82	44,00398481	9426	1200
90	78	49,37017955	10548	1122
100	75	53,72026209	11639	1091
110	70	57,85457785	12712	1073
120	67	61,8822534	13805	1093
130	65	65,71349988	14896	1091
140	61	69,37199232	15776	880
150	56	73,26393511	16663	887
160	54	76,25031788	17565	902
170	51	79,14838417	18618	1053
180	47	81,38667798	19378	760
190	41	84,12180879	20162	784
200	39	86,72190017	21100	938
210	37	87,83495551	21790	690
220	33	88,88069143	22180	390
230	27	90,55785684	23011	831
240	24	92,20306412	23504	493
250	22	93,06100487	23803	299
260	20	93,85773855	24059	256
270	18	94,77830441	24731	672
280	14	95,74948185	25558	827
290	12	95,83384158	25640	82
300	9	96,50937723	26355	715
310	7	97,07990446	26951	596
320	6	97,5931501	27147	196
330	5	98,0893037	27339	192
340	5	98,5929594	27924	585

350	5	99,10224569	28509	585
360	3	99,28161049	28714	205
370	2	99,5120993	28999	285
371,35	1	99,5453736	29043	44

Percobaan 2

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	5,521415883	1216	1216
20	96	11,20114241	2456	1240
30	94	16,84666117	3718	1262
40	91	22,30160134	4814	1096
50	88	27,86808108	5892	1078
60	86	33,22790274	7038	1146
70	84	38,48656989	8232	1194
80	81	44,01917624	9431	1199
90	78	49,30489897	10541	1110
100	75	53,59817667	11613	1072
110	71	57,73319015	12685	1072
120	69	61,8084947	13796	1111
130	66	65,63908403	14815	1019
140	61	69,48186333	15814	999
150	56	73,31458103	16675	861
160	54	76,28670711	17619	944
170	52	79,05787042	18597	978
180	47	82,01701536	19492	895
190	44	84,06479876	20117	625
200	40	86,05042364	20645	528
210	36	87,14786616	21142	497
220	32	88,23236214	21542	400
230	30	89,05967114	22099	557

240	25	90,7325801	23111	1012
250	23	92,30741345	23662	551
260	20	92,80765938	23976	314
270	16	93,77769417	24743	767
280	15	94,44566669	24983	240
290	12	95,09663273	25443	460
300	9	95,76174471	26159	716
310	9	96,48197536	26915	756
320	8	96,87251226	27338	423
330	7	97,37705177	27538	200
340	5	97,95928249	27922	384
350	4	98,38861581	28421	499
360	3	98,92851376	28841	420
370	3	99,28073249	29282	441
380	2	99,50197659	29554	272
390	2	99,74400914	29890	336
390,5	1	99,75552688	29906	16

Percobaan 3

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,516244848	1216	1216
20	96	11,19353085	2456	1240
30	94	16,83897961	3718	1262
40	91	22,29382918	4814	1096
50	88	27,85849996	5892	1078
60	86	33,22060155	7038	1146
70	84	38,48317655	8232	1194
80	81	44,01474933	9432	1200
90	78	49,2917047	10542	1110
100	75	53,58656927	11614	1072

110	71	57,72713861	12686	1072
120	69	61,80505365	13784	1098
130	66	65,6341885	14803	1019
140	61	69,48239937	15799	996
150	56	73,31427103	16660	861
160	54	76,28855534	17604	944
170	52	79,05971865	18582	978
180	47	82,0173679	19479	897
190	44	84,06408764	20103	624
200	40	86,0572307	20649	546
210	36	87,15627206	21147	498
220	32	88,24131591	21544	397
230	30	89,07148027	22103	559
240	25	90,74406795	23116	1013
250	23	92,31890131	23666	550
260	21	92,81531523	23980	314
270	15	94,34917119	24588	608
280	14	94,63247506	24887	299
290	11	95,38904273	25366	479
300	8	96,00193758	26023	657
310	7	96,50136569	26559	536
320	6	97,02505516	26769	210
330	5	97,51869351	27129	360
340	5	98,02645869	27714	585
350	5	98,5551477	28299	585
360	3	98,72999137	28501	202
370	2	99,18328764	28864	363
380	2	99,42725915	29200	336
390	2	99,67123066	29536	336
390,2	1	99,67551199	29542	6

Percobaan 4

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,515566204	1216	1216
20	96	11,19585805	2456	1240
30	94	16,84355635	3718	1262
40	91	22,29654471	4808	1090
50	88	27,86303027	5886	1078
60	86	33,22211463	7032	1146
70	84	38,48910566	8226	1194
80	82	44,00915926	9426	1200
90	78	49,36890994	10554	1128
100	75	53,71513496	11657	1103
110	70	57,85455653	12740	1083
120	67	61,88055033	13843	1103
130	65	65,71047971	14934	1091
140	60	69,36709477	15803	869
150	57	72,37275188	16698	895
160	55	75,37870399	17776	1078
170	53	78,26274254	18912	1136
180	50	80,95014692	19736	824
190	47	83,80636107	20595	859
200	40	86,34400306	21461	866
210	38	87,64180806	22109	648
220	36	88,77697844	22625	516
230	34	89,91673137	23176	551
240	30	91,33016235	23669	493
250	28	92,20753146	23964	295
260	25	93,19929603	24444	480
270	22	94,01274351	24793	349
280	20	94,27980534	24955	162

290	15	95,19661725	25678	723
300	12	96,04050856	26452	774
310	10	96,57900928	26651	199
320	9	96,68412461	26761	110
330	6	97,47922098	27388	627
340	4	97,95588518	27563	175
350	3	98,39663625	27731	168
360	3	98,74699216	28172	441
370	3	99,09734808	28613	441
380	2	99,31863045	28915	302
390	2	99,56260195	29251	336
393,95	1	99,65962078	29385	134

Percobaan 5

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	5,511996189	1216	1216
20	96	11,18914663	2456	1240
30	94	16,83628086	3718	1262
40	91	22,29193358	4814	1096
50	88	27,85479541	5892	1078
60	86	33,21290917	7038	1146
70	84	38,47971327	8232	1194
80	82	44,00394544	9432	1200
90	78	49,37218657	10559	1127
100	75	53,72385598	11662	1103
110	70	57,8518646	12729	1067
120	67	61,88040706	13814	1085
130	65	65,71446784	14905	1091
140	61	69,37572933	15787	882
150	56	73,26882729	16680	893

160	54	76,25797783	17593	913
170	51	79,15773203	18648	1055
180	47	81,39602584	19408	760
190	42	84,1278482	20194	786
200	39	86,70664816	21111	917
210	36	87,54247445	21617	506
220	32	88,59440015	22010	393
230	27	89,72486705	22504	494
240	25	91,36814939	22999	495
250	23	92,25491996	23308	309
260	22	93,10460832	23728	420
270	19	93,85725181	23975	247
280	18	94,00241971	24070	95
290	15	94,97461706	24900	830
300	14	95,14357877	25065	165
310	12	95,7218369	25277	212
320	11	95,95021572	25517	240
330	7	96,49845895	25717	200
340	6	97,0356012	26318	601
350	6	97,64758469	26954	636
360	5	97,71692608	27025	71
370	5	98,22058179	27610	585
380	5	98,74506186	28195	585
390	3	98,98678432	28473	278
400	2	99,1653572	28698	225
410	2	99,4093287	29034	336
420	2	99,65330021	29370	336
421,05	1	99,67932627	29404	34

Cluster statis menggunakan *backupCH*

Percobaan 1

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,038215365	1377	1375
20	95	14,58264159	2757	1380
30	93	21,81417343	4175	1418
40	88	29,91288657	5432	1257
50	84	37,9573054	6722	1290
60	81	44,41311308	8061	1339
70	77	50,93907767	9384	1323
80	71	59,10306445	10546	1162
90	66	66,22991201	11759	1213
100	56	72,62696384	12868	1109
110	52	78,23800749	14052	1184
120	46	83,61938518	15157	1105
130	41	87,7698375	16093	936
140	27	91,48493309	16862	769
150	21	95,16653349	17653	791
160	10	97,30711513	18086	433
170	7	98,26373541	18292	206
176,5	5	98,7689284	18406	114

Percobaan 2

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,041866391	1377	1377
20	95	14,56145018	2754	1377
30	93	21,79814759	4172	1418
40	88	29,89574097	5431	1259
50	84	37,99295383	6726	1295

60	81	44,49870747	8068	1342
70	79	51,12163286	9439	1371
80	73	58,56788556	10697	1258
90	67	65,2937629	11936	1239
100	60	72,39044955	13066	1130
110	51	78,82554352	14157	1091
120	46	83,68801591	15303	1146
130	38	88,44402799	16399	1096
140	33	92,73890847	17428	1029
150	19	95,98300764	18094	666
160	9	97,93798244	18466	372
168,55	5	99,0145157	18694	228

Percobaan 3

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,031457159	1376	1376
20	95	14,57707436	2756	1380
30	93	21,81185541	4174	1418
40	88	29,90094947	5427	1253
50	84	37,96237445	6719	1292
60	81	44,45546343	8059	1340
70	77	50,96654515	9382	1323
80	71	58,65518993	10579	1197
90	68	65,08453278	11884	1305
100	60	71,78151595	13061	1177
110	57	77,1169082	14264	1203
120	50	82,27159982	15376	1112
130	38	87,66627239	16335	959
140	29	91,83972987	17206	871
150	17	95,34245624	17774	568

160	8	97,44087786	18266	492
166,95	5	98,20857943	18399	133

Percobaan 4

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,023052613	1375	1375
20	95	14,56425073	2753	1378
30	93	21,80376945	4171	1418
40	88	29,88686581	5426	1255
50	84	37,93017645	6714	1288
60	81	44,40294308	8058	1344
70	77	50,91922485	9383	1325
80	71	59,17100366	10543	1160
90	65	66,28684578	11755	1212
100	57	73,04062117	12827	1072
110	51	78,56382458	14005	1178
120	48	83,64136794	15178	1173
130	38	88,25685915	16065	887
140	30	92,32672336	17058	993
150	19	95,59650971	17754	696
160	7	97,36903211	18115	361
161,3	5	97,56129652	18148	33

Percobaan 5

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,044873161	1377	1377
20	95	14,59236146	2759	1382
30	93	21,82585737	4177	1418
40	88	29,94784923	5421	1244

50	84	37,9894429	6709	1288
60	81	44,44197369	8047	1338
70	77	50,9501193	9371	1324
80	71	59,11329963	10528	1157
90	65	66,24245789	11745	1217
100	57	73,00691474	12814	1069
110	50	78,52944224	13994	1180
120	48	83,65030911	15186	1192
130	37	88,24060649	16069	883
140	30	92,33303781	17081	1012
150	18	95,61731807	17772	691
160	6	97,39529483	18127	355
161,3	5	97,54195194	18145	18

Cluster dinamis menggunakan *backupCH*

Percobaan 1

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,041276929	1377	1377
20	95	14,57157214	2756	1379
30	93	21,80503829	4174	1418
40	88	29,89232015	5428	1254
50	84	37,9825852	6724	1296
60	81	44,48532973	8070	1346
70	77	50,99674925	9397	1327
80	73	57,06607095	10677	1280
90	67	62,8034429	11859	1182
100	60	68,77793162	12947	1088
110	55	72,40113164	14129	1182
120	52	75,82749574	15326	1197
130	47	79,99294964	16400	1074

140	46	82,88443762	17464	1064
150	39	86,41204886	18485	1021
160	33	88,52001357	19543	1058
170	25	90,18119586	20192	649
180	23	92,09873012	21272	1080
190	18	93,56145147	21994	722
200	14	94,38000712	22836	842
210	10	95,28425568	23605	769
220	9	95,99391343	24379	774
230	8	96,65711306	25089	710
240	8	97,29823488	25809	720
250	5	97,78696466	26359	550
260	4	98,2812884	26887	528
270	4	98,81211439	27407	520
280	3	99,18920226	27821	414
290	3	99,60623555	28262	441
294,7	1	99,71938969	28404	142

Percobaan 2

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,035325552	1375	1375
20	95	14,54142161	2751	1376
30	93	21,77119279	4169	1418
40	88	29,89267913	5410	1241
50	84	37,9413351	6700	1290
60	81	44,4234537	8045	1345
70	77	50,92873857	9372	1327
80	73	56,99806027	10651	1279
90	68	62,75137703	11838	1187
100	61	68,72736797	12933	1095

110	55	74,01191821	14129	1196
120	52	77,62121811	15306	1177
130	45	81,17155552	16366	1060
140	37	84,56506527	17304	938
150	35	86,65977273	18428	1124
160	32	88,58999576	19500	1072
170	30	90,59118347	20641	1141
180	26	92,5540757	21750	1109
190	19	94,30309592	22621	871
200	16	95,25155341	23474	853
210	13	96,06325431	24271	797
220	11	96,83167571	25029	758
230	7	97,5202779	25739	710
240	5	98,19052412	26361	622
250	5	98,70116683	26946	585
260	3	99,12018567	27414	468
270	2	99,43526877	27774	360
280	2	99,73182152	28110	336
282,75	1	99,81315128	28202	92

Percobaan 3

Waktu (second)	Total node	Konsumsi energi	Total packet	Data /round
10	98	7,041519469	1377	1377
20	95	14,59341406	2757	1380
30	93	21,82970028	4175	1418
40	88	29,92059627	5431	1256
50	84	37,9849859	6721	1290
60	81	44,42020945	8055	1334
70	77	50,93094974	9377	1322
80	73	56,99371251	10654	1277

90	67	62,72217959	11852	1198
100	61	68,64279697	12971	1119
110	55	73,90844744	14164	1193
120	52	77,51667338	15347	1183
130	46	81,10829494	16425	1078
140	38	84,47889384	17342	917
150	35	86,53689069	18452	1110
160	33	88,49075311	19524	1072
170	30	90,50333261	20677	1153
180	27	92,48752373	21800	1123
190	19	94,21555104	22642	842
200	17	95,23365553	23557	915
210	15	96,18795727	24407	850
220	11	96,94922659	25130	723
230	9	97,66699746	25851	721
240	4	98,21314847	26403	552
250	3	98,66949376	26923	520
260	3	99,09178879	27364	441
270	2	99,39529053	27714	350
280	2	99,64431935	28050	336
284,65	1	99,76043641	28208	158

Percobaan 4

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	7,020615346	1375	1375
20	95	14,55089362	2752	1377
30	93	21,78636654	4170	1418
40	88	29,93680971	5414	1244
50	84	37,98049695	6702	1288
60	81	44,42617856	8039	1337

70	77	50,94258742	9364	1325
80	73	57,02875058	10644	1280
90	68	62,78473722	11834	1190
100	61	68,70167959	12924	1090
110	55	73,99372299	14121	1197
120	52	77,59367854	15299	1178
130	45	81,15819051	16354	1055
140	37	84,59509628	17297	943
150	35	86,68449864	18412	1115
160	32	88,61210473	19483	1071
170	30	90,61252426	20624	1141
180	26	92,57143723	21728	1104
190	20	94,31290578	22583	855
200	16	95,24240316	23418	835
210	13	96,05978269	24218	800
220	11	96,82779939	24982	764
230	9	97,51816654	25702	720
240	5	98,16817315	26299	597
250	5	98,67881586	26884	585
260	4	99,12172042	27373	489
270	2	99,40811734	27697	324
280	2	99,7046701	28033	336
283,3	1	99,80320565	28145	112

Percobaan 5

Waktu (second)	Total node	Konsumsi energy	Total packet	Data /round
10	98	7,035591656	1375	1375
20	95	14,55396537	2750	1375
30	93	21,78799746	4168	1418
40	88	29,87406463	5426	1258

50	84	37,91848346	6716	1290
60	81	44,39730028	8059	1343
70	77	50,90373045	9386	1327
80	73	56,97305215	10666	1280
90	68	62,72209928	11853	1187
100	61	68,68920318	12945	1092
110	55	73,97023706	14138	1193
120	52	77,57531871	15312	1174
130	45	81,11812117	16375	1063
140	37	84,51463792	17312	937
150	35	86,60214017	18434	1122
160	33	88,55684911	19529	1095
170	30	90,54287062	20657	1128
180	26	92,5197481	21774	1117
190	20	94,27189501	22642	868
200	16	95,20939279	23483	841
210	13	96,0304475	24277	794
220	11	96,79748968	25043	766
230	9	97,49009715	25764	721
240	6	98,11842299	26343	579
250	5	98,58719314	26878	535
260	4	99,04137344	27384	506
270	3	99,3904267	27780	396
279,4	1	99,66116321	28099	319

Lampiran 3 Packet loss

Cluster dinamis

CH	backup CH					tanpa backup CH				
	Percobaan					Percobaan				
	1	2	3	4	5	1	2	3	4	5
0	0	0	0	0	0	0	0	0	0	0
1	4	4	4	4	4	0	0	0	0	0
2	2	2	2	1	2	1	1	1	1	1
3	27	1	1	27	1	0	0	0	0	0
4	7	7	7	7	7	1	1	1	16	1
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0
9	19	19	19	17	17	19	19	19	19	19
10	13	1	13	13	1	0	0	0	0	0
11	12	2	4	5	5	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	4	2	4	4	2	13	11	13	9	11
15	9	1	2	1	1	0	0	0	0	0
16	5	5	5	5	5	6	6	6	6	6
17	12	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	6	6	3	3	6	4	4	6	6	4
20	0	0	0	0	0	0	0	0	0	0
21	0	13	9	13	9	0	0	0	0	0
22	1	2	5	2	2	1	5	5	1	10
23	1	1	4	11	1	1	6	6	1	1
24	6	6	10	8	6	0	0	0	0	0
25	5	1	3	1	1	0	0	0	0	0
26	0	1	1	1	3	9	7	7	7	9
27	0	8	1	5	1	0	0	0	0	0
28	23	4	1	4	7	0	0	0	0	0
29	1	1	3	1	1	1	1	1	1	1
30	1	1	7	1	1	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	5	7	7	9	7	7	9	9	9	9
33	2	2	1	3	1	0	0	0	0	0
34	0	0	0	0	1	0	0	0	0	0
35	5	9	3	9	7	0	0	0	0	0
36	7	7	4	7	6	2	4	2	2	4
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0

39	0	0	0	0	0	0	0	0	0	0
40	1	1	2	1	1	0	0	0	0	0
41	0	1	0	0	1	6	3	3	1	1
42	13	1	5	10	1	0	0	0	0	0
43	5	0	0	0	0	0	0	0	0	0
44	0	7	1	1	7	0	0	0	0	0
45	0	0	1	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	1	9	10	9	1	0	0	0	0	0
48	1	1	1	1	1	19	19	19	19	19
49	0	0	0	0	0	0	0	0	0	0
50	1	1	1	1	1	0	0	0	0	0
51	7	7	1	10	7	5	5	5	3	5
52	1	1	1	1	1	0	0	0	0	0
53	3	3	3	3	3	0	0	0	0	0
54	0	1	0	1	3	0	0	0	0	0
55	1	1	1	3	1	0	0	0	0	0
56	3	1	1	1	1	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0
59	3	0	0	0	0	0	0	0	0	0
60	3	4	0	6	4	0	0	0	0	0
61	1	3	1	3	3	0	0	0	0	0
62	1	1	1	1	1	0	0	0	0	0
63	1	1	1	1	1	5	6	5	5	5
64	11	16	9	16	12	0	0	0	0	0
65	13	15	15	15	15	10	8	8	10	8
66	1	18	0	18	0	0	0	0	0	0
67	7	0	0	0	0	0	0	0	0	0
68	5	5	5	3	3	0	0	0	0	0
69	4	1	0	1	1	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0
71	1	1	1	1	3	1	1	1	1	1
72	0	0	0	0	0	0	0	0	0	0
73	3	1	1	1	1	7	9	9	5	7
74	5	3	6	6	6	0	0	0	0	0
75	1	11	1	11	11	0	0	0	0	0
76	9	1	1	1	1	0	10	10	0	0
77	5	1	3	3	1	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0
79	1	3	0	5	14	0	0	0	0	0
80	4	5	7	5	5	12	14	12	14	12
81	1	1	1	1	2	0	0	0	0	0
82	5	1	1	1	1	0	0	0	0	0
83	1	16	1	16	1	2	1	1	2	2
84	0	2	1	1	1	0	0	0	0	0

85	0	0	0	0	0	0	0	0	0	0
86	1	1	1	1	1	1	1	1	1	1
87	1	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0	0	0
89	9	9	11	11	11	1	4	4	4	4
90	6	8	9	8	8	0	0	0	0	0
91	19	19	19	19	19	19	19	18	19	19
92	6	1	4	6	6	0	0	0	0	0
93	8	10	8	10	10	8	10	10	8	10
94	4	1	1	1	1	16	14	19	18	16
95	0	0	0	0	0	0	0	0	0	0
96	14	1	1	1	1	0	0	0	0	0
97	1	1	2	1	1	1	0	0	1	1
98	0	0	0	0	0	0	0	0	0	0
99	4	1	1	1	1	0	0	0	0	0
Loss	374	311	265	380	281	179	199	202	190	188
sukses	28404	28202	28208	28145	28099	29043	29906	29542	29385	29404
Total	28778	28513	28473	28525	28380	29222	30105	29744	29575	29592
Loss	1,30%	1,09%	0,93%	1,33%	0,99%	0,61%	0,66%	0,68%	0,64%	0,64%

Cluster statis

CH	backup CH					tanpa backup CH				
	Percobaan					Percobaan				
	1	2	3	4	5	1	2	3	4	5
0	0	0	0	0	0	0	0	0	0	0
1	4	2	4	4	4	0	0	0	0	0
2	1	1	2	2	2	1	1	1	1	1
3	0	5	0	0	0	0	0	0	0	0
4	7	1	7	7	7	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	1	0	0	0	0	0
7	6	0	1	1	6	4	11	11	12	4
8	0	0	0	0	0	0	0	0	0	0
9	19	17	17	19	17	19	19	19	19	19
10	13	1	13	13	11	0	0	0	0	0
11	1	1	1	2	4	0	0	0	0	0
12	0	5	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	6	4	4	4	6	13	11	11	13	9
15	2	1	1	1	1	0	14	14	14	0
16	5	5	5	5	5	6	6	6	6	6
17	1	0	0	0	0	0	0	0	0	0
18	1	0	0	5	1	0	0	0	0	0
19	1	2	2	8	1	1	6	6	8	1
20	0	0	0	0	0	0	0	0	0	0
21	0	6	1	0	1	0	0	0	0	0

22	2	6	0	1	2	0	0	0	0	0
23	3	5	2	5	5	3	1	1	1	5
24	10	1	10	8	8	0	0	0	0	0
25	5	6	1	1	1	0	0	0	0	0
26	1	1	1	0	1	9	9	7	9	9
27	5	6	1	0	0	0	0	0	0	0
28	10	3	0	1	3	0	0	0	0	0
29	0	0	0	0	0	1	1	1	1	1
30	0	0	0	2	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	7	7	7	9	9	7	7	7	9	9
33	2	2	2	2	1	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0
35	0	0	10	0	0	0	0	0	0	0
36	4	4	6	6	6	2	2	2	2	4
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	1	1	1	1	0	0	0	0	0
40	1	0	3	1	1	0	0	0	0	0
41	0	0	1	1	1	1	5	5	5	1
42	0	0	1	0	0	0	0	0	0	0
43	0	1	5	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	1	0	0	0	0	0	0	0	0	0
46	0	0	4	1	4	0	0	0	0	0
47	0	1	0	0	0	0	0	0	0	0
48	1	1	1	1	1	19	19	19	19	19
49	1	0	0	1	1	0	0	0	0	0
50	1	6	1	1	1	0	0	0	0	0
51	1	1	2	1	2	3	5	3	5	3
52	1	0	8	1	1	0	0	0	0	0
53	10	0	1	10	8	0	0	0	0	0
54	0	0	1	1	1	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0
56	1	3	1	1	1	1	0	0	0	1
57	0	0	0	0	0	0	0	0	0	0
58	1	3	1	1	1	1	0	0	0	1
59	0	0	0	0	0	0	0	0	0	0
60	6	1	1	7	6	0	0	0	0	0
61	1	1	4	3	1	0	0	0	0	0
62	0	1	0	0	0	0	0	0	0	0
63	1	12	1	1	1	5	5	5	6	3
64	1	0	1	1	1	0	0	0	0	0
65	15	13	13	15	15	8	8	8	8	8
66	1	1	1	1	1	0	0	0	0	0
67	0	9	12	0	0	0	0	0	0	0

68	0	5	6	1	3	0	0	0	0	0
69	6	1	1	7	7	0	0	0	0	0
70	0	1	0	0	0	0	0	0	0	0
71	3	1	2	3	3	3	1	1	1	5
72	0	0	0	0	0	0	0	0	0	0
73	1	3	1	1	1	12	12	10	10	1
74	1	0	0	1	1	0	0	0	4	0
75	4	1	1	1	1	0	0	0	0	0
76	1	1	0	1	1	1	12	12	10	9
77	10	7	11	1	1	0	0	0	0	0
78	1	1	5	1	1	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0
80	7	4	4	5	7	12	12	12	14	12
81	1	1	1	1	2	0	0	0	0	0
82	0	0	1	0	0	0	0	0	0	0
83	1	1	1	1	16	8	8	8	6	10
84	1	1	1	0	0	2	2	4	1	6
85	1	1	0	1	1	0	0	0	0	0
86	1	1	1	1	1	1	1	1	1	1
87	2	0	2	9	9	0	0	0	0	0
88	0	0	8	1	1	0	0	0	0	0
89	11	11	9	11	9	1	4	2	4	4
90	6	1	0	1	1	0	0	0	0	0
91	19	19	18	19	19	19	19	19	18	19
92	1	0	1	1	5	0	3	3	1	0
93	8	8	10	10	8	8	10	8	10	10
94	1	1	2	1	2	18	18	18	19	18
95	0	0	0	0	0	0	0	0	0	0
96	2	1	11	12	10	0	0	0	0	0
97	1	1	1	1	2	12	10	10	8	1
98	0	0	0	0	0	0	0	0	0	0
99	0	1	1	0	0	0	0	0	0	0
loss	250	220	262	249	264	202	243	235	246	201
success	18406	18694	18399	18148	18145	20579	20451	20295	20295	20563
total	18656	18914	18661	18397	18409	20781	20694	20530	20541	20764
loss	1,34%	1,16%	1,40%	1,35%	1,43%	0,97%	1,17%	1,14%	1,20%	0,97%