

RELIABLE FAULT-TOLERANT PADA JARINGAN SENSOR NIRKABEL BERBASIS CLUSTERING

M Ziki Elfirman¹⁾, Supeno Djanali²⁾, dan Radityo Anggoro³⁾

¹⁾Universitas Muhammadiyah Banjarmasin

^{2,3)}Jurusan Teknik Informatika, Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember (ITS)

Kampus ITS, Keputih-Sukolilo, Surabaya-60111

e-mail: m_ziki@yahoo.com¹⁾, supeno@its.ac.id²⁾, onggo@if.its.ac.id³⁾

ABSTRAK

Pada umumnya Jaringan Sensor Nirkabel (JSN) digunakan untuk meneliti lingkungan yang keras dan sulit dijangkau oleh manusia oleh karena itu JSN rentan terhadap *fault*. Metode *Reliable Fault-Tolerant* (RFT) diusulkan untuk mengatasi permasalahan ini. Metode ini terdiri dari dua bagian yaitu proses *clustering* dan mekanisme *fault-tolerant*. Pada proses *clustering*, pemilihan *Cluster Head* (CH) dilakukan oleh *Base Station* (BS) berdasarkan data yang didapat dari node-node pada JSN. Selanjutnya, mekanisme *fault-tolerant* akan memastikan apakah CH dapat mengakomodasi semua node di dalam klasternya. Jika terjadi *fault* pada CH, *Cluster Member* (CM) yang memiliki jarak terdekat ke BS akan dipilih sebagai CH yang baru. Agar penggunaan energi lebih merata, pada setiap *round* dilakukan pergantian CH, member yang memiliki energi terbesar akan dipilih sebagai CH yang baru. Dari hasil yang diperoleh, jumlah data yang didapatkan pada metode RFT hingga 12 persen lebih baik dari metode LEACH-C.

Kata Kunci: Jaringan sensor nirkabel, clustering, cluster head, cluster member, fault-tolerant, reliable.

ABSTRACT

Wireless Sensor Network (WSN) generally is used to examine harsh environment and difficult to reach by humans therefore WSN is prone to fault. Reliable Fault-Tolerant (RFT) method is proposed to overcome this problem. This method consists of two parts, namely the clustering process and fault-tolerant mechanisms. In the clustering process, Cluster Head (CH) election is conducted by the Base Station (BS) based on data obtained from the nodes in the WSN. Furthermore, fault-tolerant mechanisms will determine whether CH can accommodate all of the nodes in its cluster. If there is a fault in CH, Cluster Member (CM) which has the shortest distance to the BS will be chosen as the new CH. In order to use energy more evenly, at each round made the turn CH, the member that has the most energy will be chosen as the new CH. From the results obtained, the amount of data obtained in the method RFT up to 12 percent better than LEACH-C method.

Keywords: Wireless sensor network, clustering, cluster head, cluster member, fault-tolerant, reliable.

I. PENDAHULUAN

Dengan kemajuan teknologi nirkabel peneliti dapat melakukan penelitian tanpa berada langsung di area penelitian. Hal ini sangat berguna dan memungkinkan dilakukannya penelitian di daerah yang berbahaya dan beresiko tinggi yang tidak dapat dilakukan sebelumnya. Salah satu teknologi nirkabel yang populer adalah Jaringan Sensor Nirkabel (JSN). JSN merupakan sebuah jaringan nirkabel yang terdiri dari banyak sensor node yang secara acak ataupun manual diletakkan pada sebuah area target [1]. Node-node sensor ini memiliki kemampuan untuk mengumpulkan informasi mengenai lingkungan sekitar dan mempunyai kemampuan untuk berkomunikasi dan mengirimkan data ke sensor lainnya ataupun ke *Base Station* (BS). Node sensor pada umumnya berukuran kecil dan memiliki energi yang terbatas oleh karena itu banyak penelitian yang menjadikan penghematan energi sebagai fokus utama dari penelitiannya.

Salah satu pendekatan yang dilakukan untuk menghemat energi pada JSN yaitu dengan mengoptimasi *routing*. Ada beberapa metode *routing* pada JSN, *clustering* disebut sebagai salah satu metode terbaik karena terbukti dapat meningkatkan umur dari jaringan [4] dan dapat menghemat penggunaan energi secara keseluruhan [1]. Pada metode *clustering* node-node yang tersebar dikelompokkan menjadi beberapa kelompok kecil kemudian dipilih sebuah node dengan kriteria tertentu sebagai *Cluster-Head* (CH). CH atau *gateway* ini berfungsi sebagai node pengumpul atau data aggregator dari node-node lain dikelompoknya untuk kemudian mengirimkan data yang sudah terkumpul tersebut langsung ke BS atau ke CH lainnya. Terdapat banyak tujuan dari penggunaan metode *clustering* ini beberapa diantaranya yaitu efisiensi energi, skalabiliti, *routing* dan sebagainya.

Pada pengaplikasiannya, JSN sering digunakan untuk meneliti sebuah lingkungan dimana akses merupakan hal yang sulit dan bahkan mustahil untuk manusia. Oleh karena itu *fault-tolerant* merupakan karakteristik yang esensial untuk JSN [5]. *Fault-tolerant* merupakan faktor penunjang utama agar didapatkan sebuah jaringan yang *reliable*. Pada penelitian ini JSN yang *reliable* merupakan tujuan utama yang ingin dicapai. Meskipun demikian, efisiensi energi tetap menjadi bagian penting yang ingin dicapai pada penelitian ini.

Ada banyak penelitian yang membahas masalah *fault-tolerant* JSN berbasis *clustering* seperti penelitian [6] yang terdiri dari dua fase yaitu deteksi dan perbaikan. Proses deteksi dilakukan dengan cara pertukaran pesan secara periodik antara CH. Apabila terdeteksi adanya *fault* akan dilakukan perbaikan sesuai dengan *fault* yang terdeteksi.

Penelitian lain yang membahas masalah *fault-tolerant* adalah penelitian [5]. Pada penelitian [5] terdapat tiga fase yaitu deteksi *fault*, diagnosis *fault* dan perbaikan *fault*. Ketiga fase ini disebut *Fault-tolerant Service* (FTS). Setiap CH menjalankan FTS setelah proses *clustering* selesai dilakukan. Fase pertama dari FTS, yaitu deteksi *fault*, dilakukan dengan cara pertukaran pesan secara periodik. CH bertanggung jawab untuk mengirimkan pesan ke node-node pada klasternya yaitu berupa pesan *Heartbeat* (HB) dan *Summary* (SY). Selain itu CH juga mengirimkan pesan *Update* (UD) secara periodik ke CH tetangga. Apabila terjadi *fault* maka akan dilakukan proses diagnosis *fault* yang merupakan fase kedua dari FTS. Pada proses diagnosis ini akan ditentukan apakah *fault* terjadi pada *Cluster Member* (CM) atau pada CH. Fase terakhir yaitu fase perbaikan *fault*. Proses perbaikan dilakukan sesuai dengan diagnosis *fault* yaitu perbaikan CM dan perbaikan CH.

Penelitian lain yang membahas masalah *fault-tolerant* pada JSN berbasis *clustering* adalah penelitian [1]. Algoritma yang diusulkan pada penelitian tersebut dinamakan *Distributed Fault-tolerant Clustering Algorithm* (DFCA). Secara garis besar DFCA terdiri dari beberapa tahap. Pada awalnya CH dan CM ditentukan sebelumnya dimana CH memiliki energi yang jauh lebih besar dibanding CM. BS kemudian memberikan ID untuk masing-masing node. CH dapat menghitung jarak ke BS dan menyediakan jadwal TDMA. Node dibagi ke dalam beberapa kelompok, COset adalah kelompok node-node yang memiliki setidaknya satu *gateway*, UnCOset adalah kelompok node-node yang tidak memiliki *gateway* dan kelompok node yang memiliki *gateway* dan berada dalam jangkauan UnCOset disebut BackupSet. Node di kelompok UnCOset akan menyebarkan pesan HELP untuk meminta *backup*. Node pada kelompok BackupSet akan membalas pesan ini dan menjadi relay untuk node tersebut. Karena CH memiliki energi yang lebih besar dibanding node lainnya maka tidak ada proses pengecekan apakah energi yang dimiliki CH cukup untuk mengakomodasi semua CM.

Setelah proses *clustering* selesai, CH mengumpulkan data dari CM, mengagregasi data tersebut kemudian mengirimkannya ke BS. Apabila terdeteksi adanya *fault* pada CH maka anggota pada klaster tersebut menyebarkan pesan HELP. Apabila mendapatkan balasan dari CH node tersebut akan masuk kembali ke kelompok COset, apabila tidak mendapatkan balasan maka akan masuk ke kelompok UnCOset. Selanjutnya proses diulangi seperti pada saat proses *clustering*.

Dari penelitian-penelitian yang telah diuraikan diusulkan sebuah metode untuk mengatasi permasalahan pada JSN berbasis *clustering* dan mengatasi kelemahan-kelemahan yang ada pada penelitian sebelumnya. Kelemahan utama yaitu pada pemilihan CH. Pada penelitian [3] tidak dijelaskan berapa jumlah CH yang harus dipilih dan proses pemilihan CH cukup membebani jaringan karena node perlu melakukan proses perhitungan-perhitungan. Kelemahan lain yang ada pada penelitian terkait yaitu tidak adanya pengecekan apakah CH mampu mengakomodasi semua member pada klasternya.

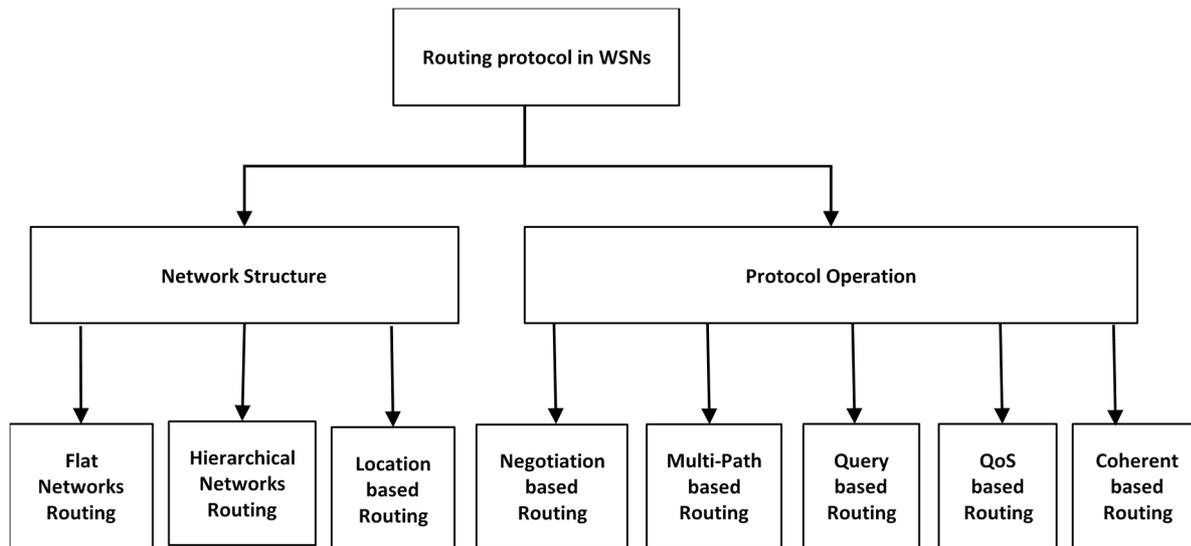
Penelitian ini mengacu pada dua masalah yang telah diuraikan yaitu proses *clustering* dan mekanisme *fault-tolerant*. Pada penelitian ini CH ditentukan oleh BS berdasarkan informasi yang didapat dari node pada jaringan. Setelah proses *clustering* CH melakukan perhitungan untuk memastikan energinya cukup untuk mengakomodasi semua CM. Apabila terjadi *fault* pada CH, CM dengan bobot terbaik akan dipilih menjadi CH yang baru. Pada setiap *round* ada kemungkinan terjadinya pergantian CH.

II. KAJIAN PUSTAKA

Jaringan sensor nirkabel merupakan sebuah jaringan nirkabel yang terdiri dari banyak sensor node yang secara acak ataupun manual diletakkan pada sebuah area target [1-3]. Tujuan utama dari JSN yaitu mengumpulkan informasi mengenai sebuah area yang sulit dijangkau khususnya oleh manusia. Ada beberapa komponen di dalam JSN yaitu *Base Station* (BS) dan node sensor. Pada penelitian ini node sensor dibagi menjadi dua sesuai tugas dari node sensor tersebut yaitu *Cluster Head* (CH) dan *Cluster Member* (CM). Karakteristik utama dari BS yaitu mempunyai energi yang jauh lebih besar dan lebih baik seperti *power grid* [7] dan mempunyai area jangkauan yang luas yang dapat mencakup semua node sensor.

A. Routing pada JSN

Berikut adalah taksonomi dari protokol *routing* di JSN.



Gambar. 1. Taksonomi protokol *routing* di JSN (Al-Karaki, dkk., 2004)

Alur data yang terjadi pada JSN 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 JSN merupakan hal yang penting. Pada penelitian ini akan digunakan *hierarchical network routing* atau yang biasa disebut *cluster-based routing*. *Cluster-based routing* memiliki kelebihan yang berhubungan dengan *scalability* dan efisiensi komunikasi [8].

B. Clustering di JSN

Clustering di JSN 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 JSN 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.

C. Fault-tolerant JSN

Fault-tolerant JSN adalah JSN yang memiliki kemampuan untuk mengatasi permasalahan yang terjadi. Permasalahan yang ada pada JSN umumnya terjadi karena ketidakmampuan sebuah node untuk menjalankan fungsinya. Berdasarkan lama waktu dari *fault*, terdapat tiga jenis *fault* yaitu: *transient* atau sementara, *intermittent* atau berselang dan *permanent* atau tetap [5].

Metode *Fault-tolerant* dapat dibagi menjadi beberapa kelompok yaitu [10]:

- *Fault prevention*
Fault prevention adalah metode untuk mencegah atau menghindari terjadinya *fault*.
- *Fault detection*
Fault detection adalah metode untuk mengetahui apakah terjadi *fault* di jaringan dengan menggunakan parameter-parameter yang berbeda.
- *Fault isolation*
Fault isolation adalah metode untuk menghubungkan berbagai indikasi *fault* yang terjadi dan mengusulkan berbagai hipotesis *fault*.
- *Fault identification*
Fault identification adalah metode untuk menguji setiap hipotesis yang diusulkan untuk memastikan lokasi *fault* dan mengenali *fault* tersebut.
- *Fault recovery*
Fault recovery adalah metode untuk mengatasi *fault* yang terjadi.

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

D. Reliability di JSN

Reliability di JSN merupakan ukuran untuk mengetahui tingkat keberhasilan data yang dikirim oleh node ke BS. Umumnya semakin meningkatnya *reliability* maka semakin meningkat pula penggunaan energi [12]. Sebagai contoh, *routing protocol* dapat mengirimkan paket melalui banyak jalur agar lebih *reliable*. Pengiriman paket ke beberapa jalur tentu memerlukan energi yang lebih banyak dibandingkan dengan mengirimkan paket hanya ke satu jalur.

Agar JSN berbasis *clustering reliable* maka *availability* dari CH perlu dimaksimalkan karena CH merupakan node yang bertanggung jawab atas keberhasilan pengiriman paket dari node ke BS. Umumnya node-node di JSN akan diset pada mode *sleep* jika tidak sedang melakukan pengiriman atau *sensing* data. Untuk tetap dapat melakukan penghematan energi dan *reliable* CH akan terus berada dalam keadaan siap menerima paket dari *member*. *Reliability* di JSN dapat diukur dengan membandingkan jumlah CH yang aktif pada setiap *round*. Apabila terdapat kluster yang tidak memiliki CH maka *reliability* dari metode yang dimaksud berkurang.

E. Time Division Multiple-Access (TDMA)

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 [11]. TDMA dapat menghemat energi yaitu dengan cara membiarkan node tetap dalam status *sleep* [13]. Selain itu juga TDMA mengurangi penggunaan energi dengan menghilangkan *collision* dan menghindari *idle listening*.

F. Low Energy Adaptive Clustering Hierarchy-Centralized (LEACH-C)

Metode LEACH-C merupakan pengembangan dari metode LEACH yang menggunakan algoritma *Simulated Annealing* untuk mencari K optimal kluster [9]. Algoritma LEACH-C dibagi menjadi dua fase yaitu fase *setup* dan fase *steady state*. Pada fase *setup* masing-masing node mengirimkan informasi mengenai lokasi dan residual energi ke BS. BS menggunakan algoritma *Simulated Annealing* untuk menentukan CH dan menemukan kluster yang optimal. Setelah CH dan kluster ditentukan, BS akan menyebarkan informasi yang berisi CH dan informasi kluster ke semua node.

Pada fase *steady state*, node hanya mengirimkan data sekali ke CH berdasarkan jadwal TDMA yang sudah ditentukan. CH menerima data dari setiap anggota kluster dan melakukan agregasi data. Selanjutnya, CH akan mengirimkan data yang sudah diagregasi ke BS menggunakan DS-SS *spreading code*. Penggunaan DS-SS *spreading code* bertujuan untuk mengurangi kemungkinan terjadinya interferensi antar kluster. Setelah *round* berakhir maka akan dilakukan *reclustering* ulang. Proses akan berjalan seperti pertama kali aplikasi dijalankan.

III. METODOLOGI

Bab ini memaparkan tentang langkah-langkah yang dilakukan pada penelitian yang dilakukan.

A. Perumusan Masalah

Masalah utama yang menjadi fokus pada penelitian ini adalah mengatasi *fault* mungkin terjadi pada JSN berbasis *clustering*. *Fault* yang dimaksud pada penelitian ini yaitu node yang tidak memiliki kluster pada proses *clustering* dan tidak berfungsinya CH. Untuk mengatasi *fault* tersebut penulis mengusulkan sebuah metode *Reliable Fault Tolerant* (RFT).

B. Perancangan Metode

Metode RFT ini menggunakan pendekatan yang umum yang sering digunakan pada JSN berbasis *clustering*, khususnya penelitian [1], yaitu proses *clustering* dengan menggunakan parameter residual energi dan jarak ke BS dan mekanisme *fault-tolerant* apabila terjadi *fault* pada CH.

Pada proses *clustering*, BS mengirimkan pesan yang berisi ID untuk masing-masing node dan energi yang diperlukan untuk mengirimkan pesan ke masing-masing node tersebut. Dengan informasi ini node mengetahui besar energi yang diperlukan untuk mengirimkan data ke BS. Selanjutnya, masing-masing node mengirimkan informasi koordinat dan residual energi ke BS. Apabila semua data node diterima oleh BS maka akan dijalankan proses pemilihan kluster oleh BS. Data kluster yang dihasilkan dikirim ke semua node melalui pesan *broadcast*.

Node yang terpilih sebagai CH akan mengecek apakah mempunyai energi yang cukup untuk mengakomodasi semua *member*. CH akan mengirimkan update ke BS apabila energi yang dimiliki tidak cukup untuk mengakomodasi semua *member*. *Update* ini berisi *list* node *member* yang akan dikeluarkan dari kluster. *List* node yang dikeluarkan dari kluster didapatkan dengan menjalankan simulasi hingga terpenuhi syarat yang ditentukan.

Syarat yang dimaksud yaitu pada saat CH mati di *round* pertama, node terakhir yang mengirimkan data ke CH sebelum CH mati berada pada posisi maksimal 50% dari TDMA. Sebagai contoh kluster dengan 5 *member* dan jadwal TDMA {1 2 3 4 5}. Pada *round* pertama, setelah beberapa kali pengiriman CH masih menerima data dari

node 1 dan 2 kemudian CH kehabisan energi.

Karena data yang diterima CH tidak lebih dari setengah total member pada kluster tersebut maka CH tetap dianggap mampu mengakomodasi semua member. Pada setiap *round* CH dapat mengirimkan data berkali-kali selama semua data dari member sudah diterima dan memiliki energi yang cukup untuk mengirimkan data tersebut ke BS. Proses pengecekan kluster ini diilustrasikan ke dalam diagram alir seperti pada Gambar 2.

Proses pengecekan kluster ditujukan untuk mengetahui apakah CH memiliki energi yang cukup untuk mengakomodasi semua membernya. Proses ini juga dilakukan untuk mendapatkan perhitungan energi yang sebenarnya karena model energi yang akan digunakan tidak memungkinkan untuk dilakukannya perhitungan penggunaan energi diawal simulasi. Setelah proses pengecekan dilakukan dan terdapat node yang perlu dikeluarkan dari kluster, CH akan mengirimkan update ke BS. BS akan menjalankan algoritma pemilihan kluster lagi dan mengirimkan hasilnya ke semua node.

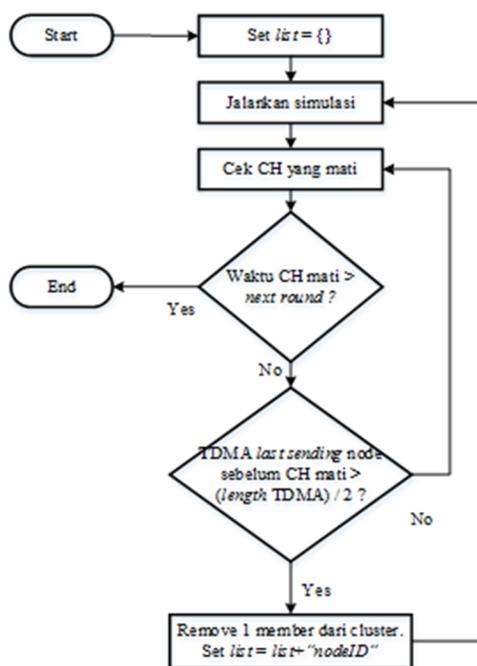
Setelah kluster terbentuk maka akan ditentukan TDMA berdasarkan data kluster dari BS. Semua CM akan terus mengirimkan data ke CH sesuai dengan slot waktu yang ditentukan dan berada dalam keadaan *sleep* diluar slot waktu pengiriman. Jika CH kehabisan energi sebelum *round* berakhir maka akan dipilih CM yang memiliki jarak terdekat ke BS sebagai CH yang baru. Pada awal setiap *round* akan dilakukan perputaran CH, node yang memiliki energi terbanyak pada kluster tersebut akan dipilih sebagai CH yang baru. Metode ini diilustrasikan ke dalam diagram alir seperti pada Gambar 3.

C. Mekanisme Pengujian

Parameter ujicoba yang lain akan mengikuti parameter ujicoba pada penelitian [9]. Untuk mengevaluasi performa dari metode yang diusulkan perlu dilakukan perbandingan terhadap parameter berikut:

- Jumlah data yang diterima BS.
- Waktu yang dicapai sebelum semua node kehabisan energi. Waktu dapat mewakili jumlah *round* yang dicapai karena lama setiap *round* adalah 10 detik.
- Tingkat *reliability* dari metode *fault-tolerant*. Pada penelitian ini parameter untuk mengukur *reliable* tidaknya sebuah metode hanya berdasarkan status CH. Metode dianggap *reliable* apabila pada setiap *round* selalu ada CH yang aktif. Apabila terjadi *fault* pada CH pada satu *round* dan segera diganti oleh member dari kluster tersebut pada *round* yang sama maka metode tetap dianggap *reliable*.

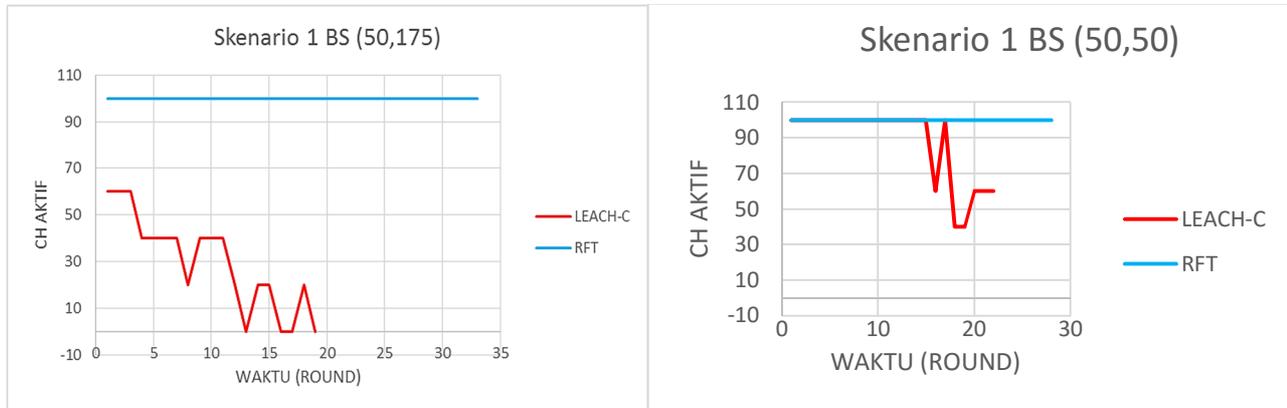
Percobaan akan dilakukan dengan menggunakan lima skenario yang berbeda. Masing-masing skenario akan dijalankan sebanyak sepuluh kali, lima kali dengan posisi BS pada koordinat (50,50) dan lima kali dengan posisi BS pada koordinat (50,175). Analisa dari hasil ujicoba yang akan dilakukan menggunakan parameter evaluasi yang sudah ditentukan yaitu jumlah data, waktu dan tingkat *reliability*.



Gambar. 2. Diagram Alir Proses Pengecekan Kluster

TABEL II
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 1 BS (50,50)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	27857	278	27857	278	27843	277	27871	278	27872	279	-
LEACH-C	26525	220	27057	230	26820	230	26686	220	26632	230	-
RFT-P	27837	278	27863	278	27876	277	27855	278	27850	279	-



(a)Koordinat BS (50,175)

(b) Koordinat BS (50,50)

Gambar. 4. Persentase Jumlah CH Aktif Tiap Metode Pada Skenario 1

Berdasarkan TABEL I dan II dapat dilihat peningkatan jumlah data dan waktu pada metode RFT dan RFT-P dibandingkan dengan metode LEACH-C. Peningkatan data pada metode RFT dan RFT-P disebabkan karena selalu ada CH yang aktif disetiap *round*. Peningkatan jumlah *round* pada metode RFT dan RFT-P disebabkan karena alokasi waktu pengiriman data yang sama, sedangkan pada metode LEACH-C alokasi waktu pengiriman sesuai dengan kondisi jumlah kluster yang selalu sama.

Jumlah kluster disemua *round* pada metode LEACH-C dapat dipertahankan karena adanya proses *reclustering* diawal setiapp *round*. Hal ini membuat metode LEACH-C lebih efisien dalam hal penggunaan energi untuk pengiriman data. Sebaliknya, pada metode RFT dan RFT-P alokasi waktu pengiriman ditentukan oleh pengaturan awal. Pada Gambar 4 dapat dilihat metode RFT selalui memiliki CH yang aktif pada setiapp *round* sedangkan pada metode LEACH-C, hanya beberapa kluster yang masih memiliki CH pada setiapp *round*.

B. Ujicoba menggunakan Skenario 2

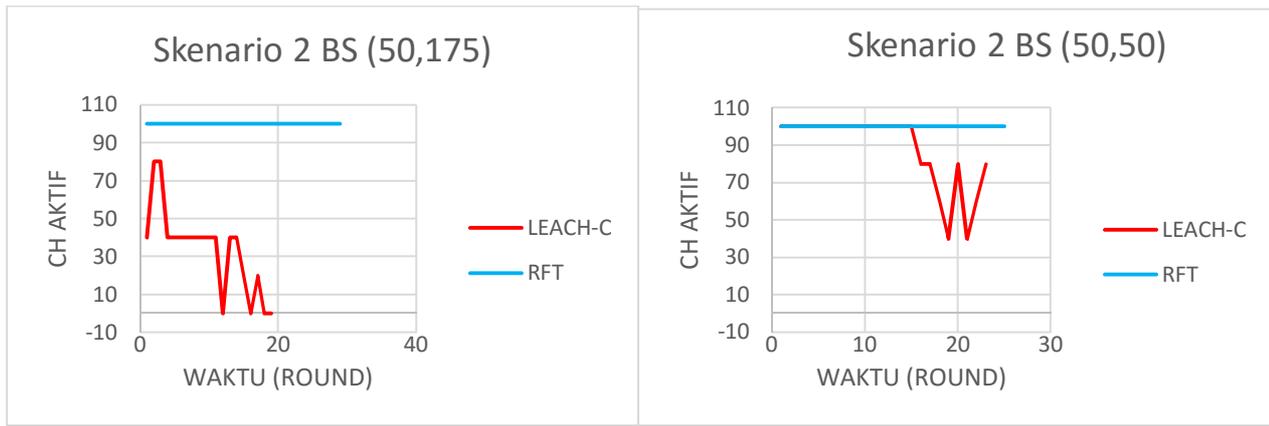
Pada Skenario 2 digunakan berkas topologi 100nodes_random1.txt. Penyebaran node pada topologi ini sedikit berbeda dibandingkan dengan skenario sebelumnya. Dengan menggunakan topologi ini dapat diketahui pengaruh dari perubahan koordinat node sensor terhadap hasil evaluasi. Agar hasil yang didapatkan valid, bukan merupakan hasil terbaik atau hasil terburuk, ujicoba dengan skenario ini dijalankan beberapa kali.

TABEL III
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 2 BS (50,175)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	17141	288	17220	291	17207	293	17196	287	17159	287	-
LEACH-C	15680	194	15571	194	15665	193	15262	190	15549	194	-
RFT-P	17231	288	17158	291	17230	293	17202	287	17150	287	-

TABEL IV
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 2 BS (50,50)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	27034	248	27019	250	27104	250	27050	250	27193	250	-
LEACH-C	25385	230	25530	230	25477	230	25617	230	25493	230	-
RFT-P	27050	248	27200	250	27050	250	27049	250	27065	250	-



(a) Koordinat BS (50,175)

(b) Koordinat BS (50,50)

Gambar. 5. Persentase Jumlah CH Aktif Tiap Metode Pada Skenario 2

Pada skenario ini distribusi node pada tiap kluster lebih baik dibandingkan dengan skenario 1. Selisih jumlah member antara kluster terbesar dengan kluster terkecil lebih sedikit dibandingkan skenario 1. Berdasarkan TABEL III dan TABEL IV dapat diketahui bahwa distribusi node yang baik dapat meningkatkan jumlah data yang diterima oleh BS.

C. Ujicoba menggunakan Skenario 3

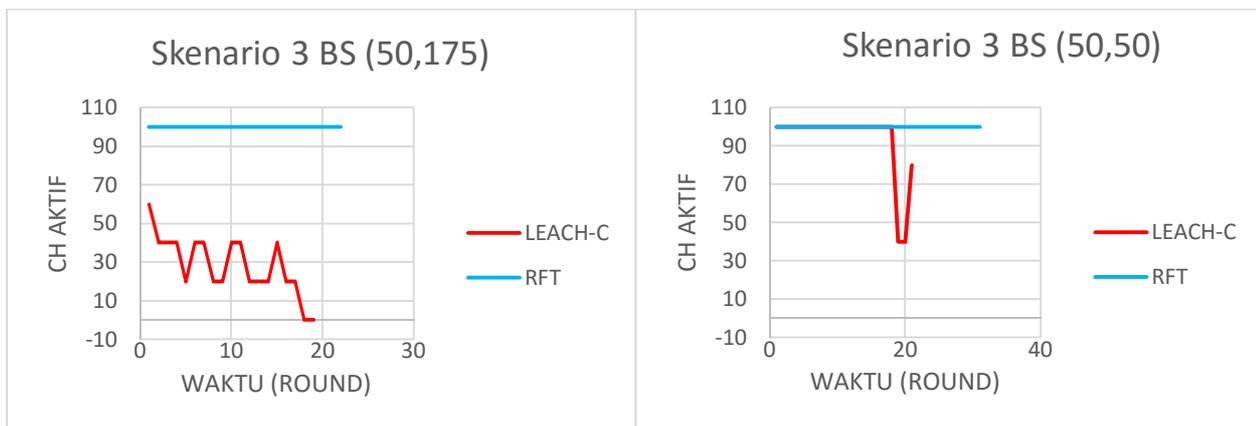
Pada Skenario 3 digunakan berkas topologi 100nodes_random2.txt. Penyebaran node disetiap kluster pada topologi ini hampir seimbang. Dengan menggunakan topologi ini dapat diketahui pengaruh dari perubahan koordinat node sensor terhadap hasil evaluasi. Agar hasil yang didapatkan valid, bukan merupakan hasil terbaik atau hasil terburuk, ujicoba dengan skenario ini dijalankan beberapa kali.

TABEL V
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 3 BS (50,175)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	16860	216	16896	216	16806	217	16876	218	16855	220	-
LEACH-C	15718	190	15393	198	15722	190	15731	190	15280	195	-
RFT-P	16829	216	16850	216	16893	217	16806	218	16818	220	-

TABEL VI
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 3 BS (50,50)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	28486	302	28486	302	28497	300	28488	305	28475	302	-
LEACH-C	27308	230	27190	230	27154	220	27262	220	27202	230	-
RFT-P	28470	302	28475	302	28497	300	28484	305	28489	302	-



(a) Koordinat BS (50,175)

(b) Koordinat BS (50,50)

Gambar. 6. Persentase Jumlah CH Aktif Tiap Metode Pada Skenario 3

Dari hasil ujicoba dapat dilihat distribusi node yang lebih baik menghasilkan jumlah data yang lebih banyak. Seimbangnya penyebaran node pada tiap kluster juga mempengaruhi efisiensi waktu pengiriman, bahkan hampir seefisien metode LEACH-C.

Pada skenario ini terjadi peningkatan *reliability* dari metode LEACH-C, hanya pada *round* terakhir dimana tidak ada satupun CH yang hidup. Meskipun demikian tingkat *reliability* metode LEACH-C masih jauh dibawah metode RFT.

D. Ujicoba menggunakan Skenario 4

Pada Skenario 4 digunakan berkas topologi 100nodes_random3.txt. Pada skenario ini terdapat kluster yang yang memiliki anggota member berlebih sehingga perlu dilakukan pengurangan member. Dengan adanya pengurangan member ini dapat dianalisa pengaruhnya terhadap hasil ujicoba yang dilakukan.

Dari hasil ujicoba menggunakan skenario ini dapat dilihat bahwa pengurangan member membuat hasil dari metode RFT-P mengalami penurunan yang signifikan. Data yang dihasilkan oleh metode RFT-P bahkan jauh lebih sedikit dibandingkan dengan data yang dihasilkan oleh metode LEACH-C.

E. Ujicoba menggunakan Skenario 5

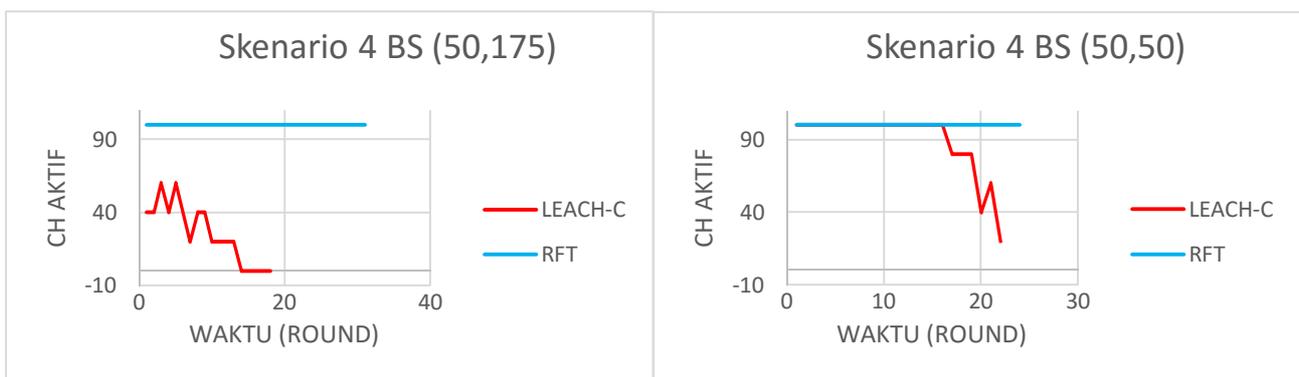
Pada skenario ini, metode RFT-P membentuk kluster baru untuk menampung member yang dikeluarkan dari klasternya. Dengan bertambahnya jumlah kluster dapat diketahui terjadi penurunan jumlah data pada metode RFT-P. Karena terdapat lebih dari satu node yang membentuk kluster baru maka penurunan jumlah data pada metode RFT-P lebih sedikit dibandingkan skenario sebelumnya.

TABEL VII
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 4 BS (50,175)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	16465	247	16478	247	16492	247	16394	245	16419	247	-
LEACH-C	15321	190	15356	190	15287	190	15051	190	15394	190	-
RFT-P	14061	242	14100	240	14079	240	14065	240	14102	240	0

TABEL VIII
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 4 BS (50,50)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	27088	238	27276	238	27165	238	27058	240	27233	237	-
LEACH-C	26438	217	26304	220	26528	219	26196	220	26665	220	-
RFT-P	27078	238	27286	238	27155	238	27068	240	27235	237	-



(a) Koordinat BS (50,175)

(b) Koordinat BS (50,50)

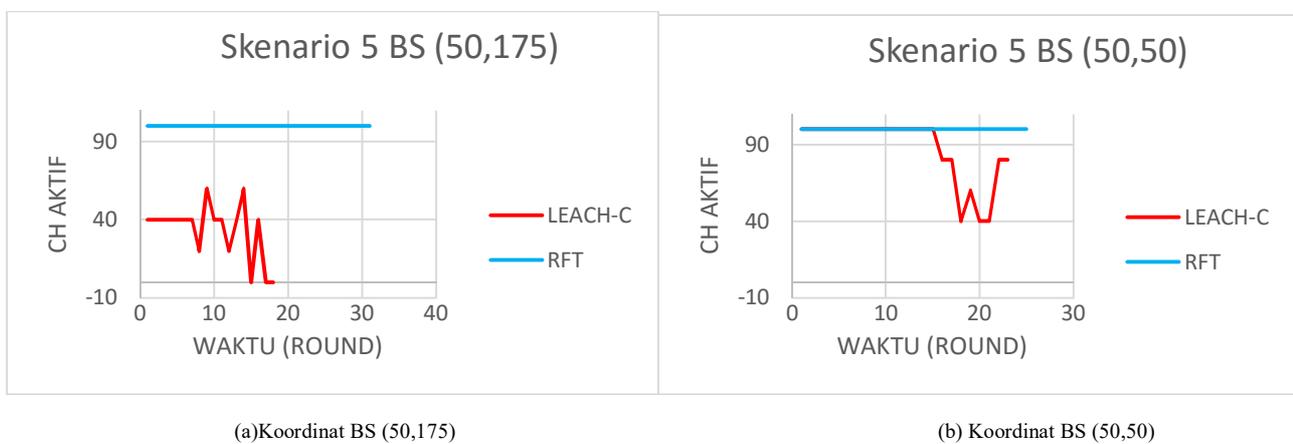
Gambar. 7. Persentase Jumlah CH Aktif Tiap Metode Pada Skenario 4

TABEL IX
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 5 BS (50,175)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	17229	306	17222	306	17206	306	17171	306	17229	306	-
LEACH-C	15508	190	15524	190	15307	190	15473	198	15405	198	-
RFT-P	14797	296	14851	297	14837	297	14834	296	14817	297	1,19

TABEL X
PERBANDINGAN JUMLAH DATA DAN WAKTU SKENARIO 5 BS (50,50)

Metode	Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4		Percobaan 5		List
	Data	Waktu									
RFT	28045	243	28055	245	28096	244	28052	246	28081	248	-
LEACH-C	26747	230	26830	240	26782	240	26714	230	26426	220	-
RFT-P	28055	243	28045	245	28080	244	28081	246	28050	248	-



Gambar. 8. Persentase Jumlah CH Aktif Tiap Metode Pada Skenario 5

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan dari penelitian dan analisa yang telah dilakukan adalah sebagai berikut:

1. Berdasarkan ujicoba yang dilakukan dapat diketahui bahwa proses *clustering* yang dilakukan di BS dengan menggunakan algoritma *clustering* yang baik dapat mengatasi permasalahan *fault* pada CM.
2. Berdasarkan hasil ujicoba, jumlah data terbanyak dihasilkan pada skenario 3 yang disebabkan karena distribusi node pada semua kluster hampir merata. Pada skenario ini data yang dihasilkan metode RFT mencapai 12% lebih banyak dibandingkan dengan metode LEACH-C.
3. Dari hasil ujicoba skenario 4 dan 5 dapat diketahui bahwa pengurangan member pada metode RFT-P menyebabkan penurunan jumlah data yang dihasilkan hingga 10% dibandingkan dengan metode LEACH-C. Hal ini disebabkan karena aggregasi data pada CH di kluster yang baru dibentuk tidak efisien.
4. Dari skenario-skenario ujicoba yang dilakukan dapat diketahui bahwa persentase jumlah CH yang aktif pada metode RFT selalu 100% dengan demikian tidak ada data dari member yang tidak terkirim ke BS. Berdasarkan jumlah CH yang aktif pada masing-masing metode dapat disimpulkan bahwa metode RFT jauh lebih *reliable* dibandingkan metode LEACH-C.

B. Saran

Beberapa saran yang dapat digunakan untuk dapat meningkatkan hasil dari metode RFT antara lain:

1. Penggunaan algoritma *clustering* yang lebih baik.
2. Model energi yang digunakan juga dapat diperbaiki agar lebih sesuai dengan skenario percobaan metode yang diusulkan.
3. Peningkatan yang signifikan akan didapatkan apabila dilakukan update nilai *spreading* sesuai dengan jumlah kluster yang tersisa.
4. Aplikasi dijalankan sampai semua node kehabisan energi.

DAFTAR PUSTAKA

- [1] 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.
- [2] Sadat, A., Karmakar, G., Zaslavsky, A., & Gaber, M. (2010). Reliable and energy efficient backup clustering scheme for wireless sensor networks. *In Proceedings of the International Conference on Information Networking (ICOIN)*.
- [3] Samanta, M., & Banerjee, I. (2014, March). Optimal load distribution of cluster head in fault-tolerant wireless sensor network. *In Electrical, Electronics and Computer Science (SCECS), 2014 IEEE Students' Conference on* (pp. 1-7). IEEE.
- [4] Younis, O., Krunz, M., & Ramasubramanian, S. (2006). Node clustering in wireless sensor networks: recent developments and deployment challenges. *Network, IEEE, 20(3)*, 20-25
- [5] 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.
- [6] 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.
- [7] Michal'ik, M. (2013). Base station for Wireless sensor network. Masaryk University. Brno, Czech Republic.
- [8] Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *Wireless communications, IEEE, 11(6)*, 6-28.
- [9] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on, 1(4)*, 660-670.
- [10] Paradis, L., & Han, Q. (2007). A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management, 15(2)*, 171-190.
- [11] 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.
- [12] Dâmaso, A., Rosa, N., & Maciel, P. (2014). Reliability of Wireless Sensor Networks. *Sensors, 14(9)*, 15760-15785.
- [13] Pantazis, N. A., Vergados, D. J., Vergados, D. D., & Douligieris, C. (2009). Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling. *Ad Hoc Networks, 7(2)*, 322-343.