

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Jurnal dengan judul “*Performance Comparison of LEACH and LEACH-C Protocols in Wireless Sensor Networks*” oleh Ala’a Al-Shaikh, Hebatallah Khatab, dan Al-Sharaeh melakukan perbandingan protokol LEACH dan LEACH-C menggunakan Matlab. Penelitian tersebut membandingkan jumlah paket data yang dikirim, jumlah *node* mati di setiap putaran, dan disipasi energi. Penelitian ini menggunakan beberapa parameter untuk pengujian, antara lain total energi 20 J, energi terbuang 19,62 J, sisa energi 0,38 J, jumlah *node* 100 *node* + 1 BS, energi awal 0.2 J, dan *round* 500. Hasil yang diperoleh pada penelitian tersebut adalah parameter yang paling penting untuk penghematan energi yaitu disipasi energi, karena memperpanjang masa pakai jaringan dilakukan salah satunya dengan memakai lebih sedikit energi pada *node* sensor. Selanjutnya ditemukan bahwa LEACH-C membuang lebih sedikit energi dibandingkan dengan LEACH dengan persentase total sisa energi sebesar 1,9 % [3].

Penelitian yang dilakukan oleh Faiz Satria Syukri, Andrian Rakhmatsyah, dan Sidik Prabowo yang berjudul “Analisis dan Simulasi *Clustering Node* Menggunakan Algoritma LEACH” meneliti tentang pemodelan wilayah Waduk Kedong Ombo menggunakan sistem *clustering* dengan LEACH sebagai algoritmanya. Pada penelitian ini dilakukan pengujian menggunakan 2 macam skenario yaitu homogen dan heterogen. Skenario pengujian homogen dilakukan untuk mengetahui seberapa banyak sensor *node* yang bertahan, melihat nilai *throughput*, energi, dan *packet loss* dengan karakteristik energi yang sama. Sedangkan pada pengujian heterogen terdapat 2 kelompok sensor *node* yang memiliki karakteristik yang berbeda. Penelitian ini direpresentasikan skala wilayah waduk dengan area 150x150 meter meliputi jumlah *node* sebanyak 50 dan 100, serta area 250x250 meter dengan jumlah *node* 517 dan 1034. Hasil yang didapatkan pada penelitian ini menunjukkan bahwa jumlah energi awal sensor *node* dan jarak antar sensor *node* mempengaruhi konsumsi energi dan komunikasi yang berlangsung [4].

Jurnal dengan judul “*Analisis Protocol Low Energy Adaptive Clustering Hierarchy pada Wireless Sensor Network*” oleh Afif melakukan penelitian tentang protokol LEACH pada WSN dengan menganalisa konsumsi energi, *delay*, dan *throughput*. Pengujian dilakukan sebanyak 5 kali percobaan. Penelitian ini menggunakan beberapa parameter untuk pengujian, antara lain sisa energi 100 *node* 275 J, energi awal 5 J, total energi 500 J, dan *round* 100. Hasil yang diperoleh pada penelitian ini adalah peningkatan signifikan dapat terjadi apabila *node* dikontrol jarak dan penempatannya dengan persentase total sisa energi sebesar 55 % [5].

Jurnal dengan judul “*Analisis Perbandingan Konsumsi Daya dan Masa Hidup Jaringan pada Protokol Routing LEACH dan HEED di Wireless Sensor Network*” oleh Farida Fitri Kusumastuti, Ida Wahidah, dan Ratna Mayasari melakukan perbandingan protokol LEACH dan HEED pada WSN. Penelitian tersebut membandingkan konsumsi daya dan masa hidup jaringan dengan skenario perubahan jumlah *node*. Penelitian ini menggunakan beberapa parameter untuk pengujian, antara lain energi awal 0,25 J, total sisa energi 90 *node* 9,98 J, dan *round* 100. Hasil yang didapatkan pada penelitian tersebut adalah protokol LEACH cocok digunakan pada jaringan skala kecil dengan persentase total sisa energi sebesar 44,35 %, sedangkan protokol HEED lebih baik digunakan untuk skala yang lebih besar [6].

Penelitian yang dilakukan oleh Khamis Aa, Chai Senchun, Zang Baihai, dan Cui Lingguo dengan judul “*Combined Metrics-Clustering Algorithm based on LEACH-C*” meneliti tentang mengkombinasikan metrik *clustering* berdasarkan algoritma LEACH-C. Penelitian tersebut menggunakan pengelompokan energi paling rendah, jaringan dibagi menjadi kelompok energi seimbang. Pemilihan CH dengan mempertimbangkan daya dan jarak komunikasi untuk memperpanjang masa pakai jaringan. Penelitian ini menggunakan beberapa parameter untuk pengujian, antara lain total energi 50 J, *round* 100, dan energi tersisa 17 J. Hasil yang didapatkan pada penelitian tersebut yaitu terciptanya *hole network* dan *node* yang dikecualikan dapat dihindari dengan menggunakan metode pengelompokan energi paling rendah. Pendekatan ini tidak mengharuskan *clustering* pada setiap *round* tetapi didasarkan pada batas energi yang rendah pada CH. Pada penelitian ini diketahui untuk persentase total sisa energi sebesar 34 % [2].

Pada penelitian yang dilakukan oleh penulis ini menguji algoritma LEACH-C pada Wireless Sensor Network. Pada penelitian ini menggunakan skenario perubahan jumlah *node* yaitu 50, 100, dan 150 *node* dengan software Matlab untuk simulasinya. Algoritma LEACH-C disimulasikan pada luas area 100 x 100 meter dengan jumlah 100 *round*. Parameter yang dianalisis yaitu konsumsi energi, paket yang diterima, dan *packet loss*. Hasil simulasi algoritma LEACH-C dapat dilihat bahwa dengan banyaknya jumlah *node* mempengaruhi kinerja jaringan. Simulasi jaringan pada algoritma protokol LEACH-C termasuk ke dalam kategori bagus karena jumlah persentase *packet loss* tidak mencapai 5%.

Tabel 1.1 Perbandingan Penelitian Kajian Pustaka.

No.	Judul dan Tahun Terbit	Protokol Routing	Metode	Hasil
1.	Judul : <i>Performance Comparison of LEACH and LEACH-C Protocols in Wireless Sensor Networks</i> Tahun : 2018	Protokol LEACH dan LEACH-C	<ul style="list-style-type: none"> • Simulasi menggunakan Matlab • Pengujian dengan jumlah <i>round</i> 500 • Pengujian dengan jumlah <i>node</i> 100 + 1 BS • Parameter yang diuji yaitu <i>packet sent</i>, <i>dead node</i>, dan sisa energi 	LEACH membuang lebih banyak energi daripada LEACH-C, yang memberikan LEACH-C keunggulan dibandingkan LEACH karena memperpanjang masa pakai WSN dengan persentase total sisa energi sebesar 1,9 %.
2.	Judul : Analisis dan Simulasi <i>Clustering Node</i> Menggunakan	Protokol LEACH	<ul style="list-style-type: none"> • Simulasi menggunakan Matlab 	Protokol LEACH memerlukan energi yang lebih besar dari energi awal (0.5 J).

No.	Judul dan Tahun Terbit	Protokol Routing	Metode	Hasil
	Algoritma LEACH Tahun : 2017		<ul style="list-style-type: none"> • Pengujian dengan jumlah <i>node</i> 50 dan 100 pada luas area 150x150 meter, <i>node</i> 517 dan 1034 pada luas area 250x250 meter • Parameter yang diuji yaitu energi, <i>throughput</i>, dan <i>packet loss</i> • Pengujian menggunakan topologi homogen dan heterogen • Pengujian dengan jumlah <i>round</i> 100, 200, 300, 500, 700 	Semakin banyak jumlah sensor maka semakin banyak energi tersisa pada suatu wilayah yang ditentukan.
3.	Judul : Analisis <i>Protocol Low Energy Adaptive Clustering Hierarchy</i> pada <i>Wireless Sensor Network</i> Tahun : 2017	Protokol LEACH	<ul style="list-style-type: none"> • Pengujian dengan jumlah <i>node</i> 10, 30, 50, 80, 100 dengan 5 percobaan • Parameter yang diuji yaitu konsumsi energi, <i>delay</i>, <i>throughput</i> 	Protokol LEACH melakukan kinerja dengan hasil maksimal pada pengujian 100 <i>node</i> , semakin banyak <i>node</i> maka sedikit energi yang digunakan dengan persentase total

No.	Judul dan Tahun Terbit	Protokol Routing	Metode	Hasil
				sisa energi sebesar 55 %, delay menurun dan throughput semakin meningkat
4.	Judul : Analisis Perbandingan Konsumsi Daya dan Masa Hidup Jaringan pada Protokol <i>Routing</i> LEACH dan HEED di <i>Wireless Sensor Network</i> Tahun : 2016	Protokol LEACH dan HEED	<ul style="list-style-type: none"> • Simulasi menggunakan Matlab • Pengujian dengan jumlah <i>node</i> 10, 30, 50, 70, 90 <i>node</i> dengan luas area 100x100 meter • Parameter yang diuji yaitu energi, throughput, dan packet loss • Pengujian menggunakan topologi homogen • Pengujian dengan jumlah <i>round</i> 100, 200, 300, 400 	<ul style="list-style-type: none"> • LEACH cocok digunakan pada jaringan skala kecil dengan persentase total sisa energi sebesar 44,35% • HEED mempunyai performansi lebih baik daripada LEACH untuk kedua parameter • Rata-rata <i>lifetime</i> HEED lebih tinggi daripada LEACH karena jumlah <i>dead node</i> lebih lama
5.	Judul : <i>Combined</i>	Protokol LEACH-C	• Pengujian dengan jumlah node 100	Algoritma <i>clustering-metric</i>

No.	Judul dan Tahun Terbit	Protokol Routing	Metode	Hasil
	<p><i>Metrics-Clustering Algorithm based on LEACH-C</i></p> <p>Tahun : 2015</p>		<p>node dengan luas area 100x100 meter</p> <ul style="list-style-type: none"> • Parameter yang diuji yaitu <i>node</i> hidup, data <i>sent</i>, energi • Pengujian dengan jumlah <i>round</i> 400 	<p>gabungan dapat menyeimbangkan beban antar <i>node</i> dan memperpanjang periode stabilitas dan waktu hidup jaringan.</p> <p>Diketahui untuk persentase total sisa energi sebesar 34 %</p>
6.	<p>Judul : Analisis Performansi Algoritma LEACH-C pada Wireless Sensor Network</p> <p>Tahun : 2023</p>	Protokol LEACH-C	<ul style="list-style-type: none"> • Pengujian menggunakan jumlah node 50, 100, dan 150 node • Luas area simulasi 100x100 meter dengan jumlah 100 round • Software simulasi menggunakan Matlab • Parameter yang dianalisis yaitu konsumsi energi, paket yang diterima, dan packet loss. 	<p>Pada hasil konsumsi energi, algoritma LEACH-C masih menyimpan 80% energi dan simulasi parameter jaringan LEACH-C termasuk ke dalam kategori bagus karena jumlah persentase packet loss tidak mencapai 15%.</p>

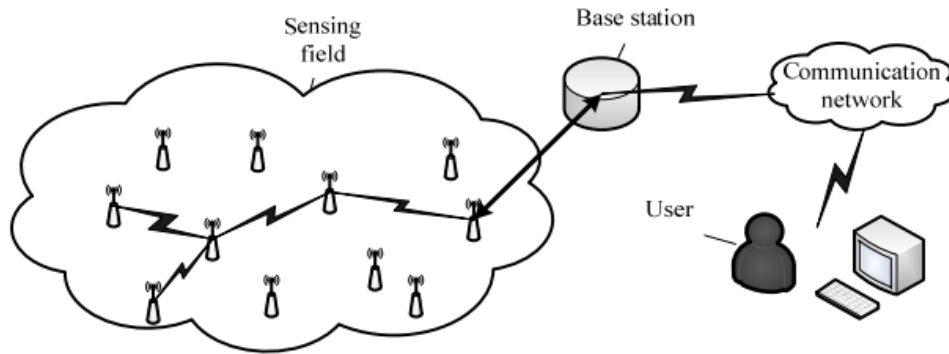
2.2 DASAR TEORI

Pada penelitian ini, penulis membutuhkan teori-teori yang akan digunakan pada topik penelitian. Pada bagian dasar teori penulis menggunakan referensi dari jurnal, buku, skripsi dan *website*.

2.2.1 WIRELESS SENSOR NETWORK (WSN)

Wireless Sensor Network atau jaringan sensor nirkabel merupakan suatu infrastruktur jaringan nirkabel yang memerlukan sensor untuk memonitor fisik maupun kondisi daerah sekitar seperti gerakan, temperatur, getaran, gelombang elektromagnetik, tekanan, suara, serta yang lainnya. Tiap-tiap *node* dalam *wireless sensor network* umumnya dilengkapi dengan radio *transceiver* ataupun perlengkapan komunikasi *wireless* yang lain, mikrokontroler kecil, dan sumber daya [7]. Suatu sensor mempunyai keahlian untuk memproses informasi dan berkomunikasi penginderaan yang disebut sebagai *node* dalam WSN. *Node* sensor memproses komputasi data yang belum dianalisis serta mengirimkan informasi yang dibutuhkan [8].

Node sensor yang mempunyai ukuran kecil pada WSN tersebar pada suatu daerah dengan jumlah yang besar. *Node* sensor tersebar dengan tujuan untuk menangkap fenomena yang akan diteliti. Jumlah *node* yang tersebar dapat ditetapkan sesuai dengan kebutuhan serta bergantung dari beberapa aspek seperti luas wilayah, kemampuan sensing *node*, kemampuan jangkauan *node* dan yang lainnya. Masing-masing *node* mempunyai keahlian untuk mengumpulkan informasi serta merutinkannya menuju ke *Base Station* (BS) ataupun *sink node* serta berkomunikasi dengan *node* yang lainnya [7] seperti yang ditunjukkan dalam arsitektur WSN pada Gambar 2.1.



Gambar 2.1 Arsitektur Wireless Sensor Network [9]

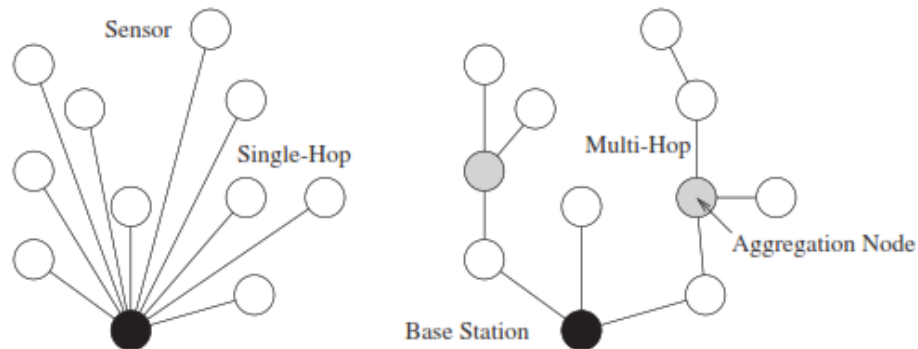
Keahlian *node* sensor pada WSN sangat bermacam-macam, yaitu *node* sensor sederhana mampu memantau fenomena fisik tunggal, sedangkan perangkat yang lebih kompleks mampu menyatukan banyak metode sensor yang berbeda. *Node* sensor pada WSN juga dapat berbeda dalam segi kemampuan komunikasinya misalnya, menggunakan suatu teknologi *ultrasound*, inframerah, ataupun menggunakan frekuensi radio yang kecepatan serta latensi data yang bermacam-macam. Sedangkan sensor sederhana hanya mampu mengumpulkan dan mengkomunikasikan data tentang wilayah yang diamati. Selain itu pada perangkat yang lebih kuat (perangkat dengan kapasitas pemrosesan, energi, serta penyimpanan yang besar) dapat melakukan fungsi pemrosesan dan agregasi yang ekstensif. Fungsi-fungsi ini seringkali memiliki tanggung jawab tambahan di WSN. Misalnya fungsi tersebut dapat membentuk *backbone* komunikasi yang dapat dilakukan oleh fungsi lain dari sensor daya untuk merespon BS. Beberapa fitur dapat mengakses teknologi bantuan tambahan, seperti penerima *Global Positioning System* (GPS), sehingga sistem dapat menentukan lokasi pengguna secara tepat. Namun, sistem ini sering mengkonsumsi banyak daya untuk *node* sensor berbiaya rendah [1]. Beberapa karakteristik yang terdapat pada WSN yaitu :

- a. Mempunyai jumlah *node* sensor yang sangat banyak.
- b. *Node* sensor rentan terhadap kegagalan.
- c. Topologi jaringan sensor nirkabel sangat sering berubah.
- d. *Node* sensor menggunakan siaran *broadcast*.
- e. *Node* sensor mempunyai sumber energi, kapasitas komputasi dan memori yang terbatas.

- f. Karena besarnya biaya *overhead* dan jumlah sensor yang banyak, *node* sensor mungkin tidak memiliki identifikasi global (ID) [10].

WSN mempunyai beberapa arsitektur komunikasi yaitu *Single-hop* dan *Multi-hop* yang digambarkan Gambar 2.2, dan penjelasannya sebagai berikut :

- Single hop without clustering*. Jaringan pada sistem *cluster* ini tidak harus membentuk suatu *cluster*, jadi tiap *node* yang akan mengirimkan data akan langsung terkirim ke *sink*.
- Multi hop without clustering*. Pada sistem *clustering* ini, *node* akan mengirimkan tiap informasi ke *sink* dengan cara melewati *node* tetangganya yang terdekat
- Single hop with clustering*. Pada *cluster* ini tiap *node* akan bergabung ke dalam suatu *cluster*, kemudian *node* akan mengirimkan informasinya langsung ke *cluster head*, setelah itu *cluster-head* mengirimkan data pada *sink*.
- Multihop with clustering*. Sistem pada *cluster* ini hampir sama dengan *single hop with clustering*, namun tidak seluruh *cluster-head* mengirimkan datanya langsung ke *sink*. Terdapat beberapa *cluster-head* yang mengirimkan data melintasi *cluster-head* tetangganya untuk bisa mencapai *sink*.



Gambar 2.2 Komunikasi *single-hop* dan *multi-hop* pada WSN [1]

Wireless sensor network mempunyai kelebihan yaitu kombinasi diantara sensor, komputasi, serta komunikasi menjadi suatu perlengkapan yang terintegrasi. Kemampuannya yang dapat menyebarkan *node-node* yang sangat kecil dengan jumlah besar secara tersusun serta terkonfigurasi dengan sendirinya. Jaringan nirkabel *ad-hoc* memiliki persamaan dengan jaringan sensor nirkabel. Persamaan yang utama yaitu sama-sama menggunakan metode *multi-hop*. Selain itu WSN juga mempunyai perbedaan dengan jaringan nirkabel *ad-hoc* diantaranya adalah :

- a. *Node* yang digunakan pada WSN lebih banyak yaitu mencapai ratusan atau bahkan ribuan *node* dibandingkan dengan jaringan *ad-hoc*.
- b. *Node* sensor dibatasi dalam proses perhitungan, energi serta sumber tempat penyimpanan dibanding dengan *ad-hoc*.
- c. *Node* sensor dapat digunakan dengan jarak jauh dari manusia dengan waktu yang panjang.
- d. *Node* sensor yang berdekatan kerap mendeteksi peristiwa yang sama dalam suatu area dan mengirimkan informasi yang sama ke *base station* [11].

Pada jaringan *Wireless Sensor Network* juga memiliki berbagai tantangan dan kendala yang unik. Kendala ini memengaruhi desain WSN, yang mengarah ke protokol dan algoritma yang berbeda dari pasangannya di sistem terdistribusi lainnya. Kendala desain pada WSN itu antara lain :

1) Energi

Kendala yang paling sering dikaitkan dengan desain jaringan sensor adalah *node* sensor beroperasi dengan *budget* energi yang terbatas. Biasanya, mereka diberdayakan melalui baterai, yang harus diganti atau diisi ulang (misalnya, menggunakan tenaga surya) saat habis. Untuk beberapa *node*, tidak ada *opsi* yang sesuai, yaitu mereka akan dibuang begitu saja setelah sumber energinya habis. Apakah baterai dapat diisi ulang atau tidak secara signifikan mempengaruhi strategi yang diterapkan pada konsumsi energi. Untuk baterai yang tidak dapat diisi ulang, *node* sensor harus dapat beroperasi hingga waktu misinya terlewati atau baterai dapat diganti. Lamanya waktu misi bergantung pada jenis aplikasi.

2) *Self-management*

Self-management merupakan watak dari banyak aplikasi jaringan sensor yang wajib beroperasi di wilayah terpencil serta area yang keras, tanpa sokongan infrastruktur ataupun mungkin pemeliharaan serta perbaikan. Oleh sebab itu, *node* sensor wajib mengendalikan dirinya sendiri dalam perihal mereka mengonfigurasi dirinya sendiri, beroperasi serta bekerjasama dengan *node* lain, serta menyesuaikan diri dengan kegagalan, pergantian area, serta pergantian rangsangan area tanpa campur tangan manusia.

3) *Wireless Networking*

Ketergantungan pada jaringan nirkabel serta komunikasi memunculkan beberapa tantangan untuk perancang jaringan sensor. Misalnya, atenuasi menghalangi jangkauan sinyal radio, ialah sinyal frekuensi radio (RF) memudar (ialah penyusutan energi) dikala merambat lewat media serta dikala melewati rintangan.

4) *Decentralized Management*

Skala besar serta hambatan tenaga dari banyak jaringan ensor nirkabel membuatnya tidak mungkin untuk mengandalkan algoritma terpusat (misalnya dieksekusi di base station) buat mengimplementasikan pemecahan manajemen jaringan semacam manajemen topologi ataupun perutean. Kebalikannya, node sensor wajib bekerjasama dengan tetangganya buat membuat keputusan lokal, ialah tanpa pengetahuan global. Dampaknya, hasil dari algoritma terdesentralisasi (ataupun terdistribusi) ini tidak hendak maksimal, namun bisa jadi lebih hemat tenaga daripada pemecahan terpusat. Pertimbangkan perutean sebagai contoh untuk solusi terpusat dan terdesentralisasi. *Base station* dapat mengumpulkan informasi dari semua *node* sensor, menetapkan rute yang optimal (misalnya dalam hal energi), dan menginformasikan setiap *node* tentang rutenya. Namun, *overhead* dapat menjadi signifikan, terutama jika topologi sering berubah. Alih-alih, pendekatan terdesentralisasi memungkinkan setiap *node* membuat keputusan perutean berdasarkan informasi lokal yang terbatas (misalnya, daftar *node* tetangga, termasuk jarak mereka ke *base station*). Sementara pendekatan desentralisasi ini dapat menyebabkan rute yang tidak optimal, biaya manajemen dapat dikurangi secara signifikan.

5) *Design Constraints*

Tujuan utama desain sensor nirkabel merupakan menghasilkan fitur yang lebih kecil, lebih murah, serta lebih efektif. Didorong oleh kebutuhan buat melaksanakan aplikasi spesial dengan sedikit mengkonsumsi tenaga, tipikal node sensor mempunyai kecepatan pemrosesan serta kapasitas penyimpanan sistem pc dari sebagian dekade yang kemudian. Kebutuhan akan faktor bentuk yang kecil dan konsumsi energi yang rendah juga melarang integrasi banyak komponen yang diinginkan, seperti GPS *receiver*. Batasan dan persyaratan ini

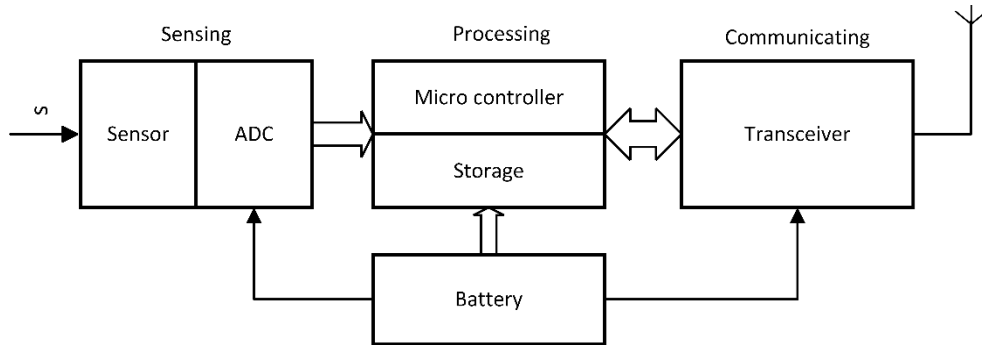
juga memengaruhi desain perangkat lunak di berbagai tingkatan, misalnya, sistem operasi harus memiliki jejak memori yang kecil dan harus efisien dalam tugas manajemen sumber dayanya. Tetapi, minimnya fitur-fitur keras Mutahir (misalnya, sokongan buat eksekusi paralel) memfasilitasi desain sistem pembedahan yang kecil serta efektif. Hambatan fitur keras sensor pula mempengaruhi desain banyak protokol serta algoritma yang dieksekusi dalam WSN.

6) *Security*

Banyak jaringan sensor nirkabel mengumpulkan informasi sensitif. Pengoperasian *node* sensor jarak jauh dan tanpa pengawasan meningkatkan keterpaparannya terhadap intrusi dan serangan jahat. Komunikasi nirkabel memudahkan musuh untuk menguping transmisi sensor. Salah satu ancaman keamanan yang paling menantang adalah serangan *denial-of-service*, yang bertujuan untuk mengganggu pengoperasian jaringan sensor yang benar. Ini dapat dicapai dengan menggunakan berbagai serangan, termasuk serangan *jamming*, di mana sinyal nirkabel berdaya tinggi digunakan untuk mencegah komunikasi sensor yang berhasil. Konsekuensinya bisa parah dan bergantung pada jenis aplikasi jaringan sensor. Meskipun ada banyak teknik dan solusi untuk sistem terdistribusi yang mencegah serangan atau mengandung tingkat dan kerusakan serangan tersebut, banyak dari ini menimbulkan kebutuhan komputasi, komunikasi, dan penyimpanan yang signifikan, yang seringkali tidak dapat dipenuhi oleh *node* sensor yang dibatasi sumber daya. Akibatnya, jaringan sensor membutuhkan solusi baru untuk pembuatan dan distribusi kunci, otentikasi *node*, dan kerahasiaan [1].

Namun tantangan terbesar dalam WSN adalah kekurangan energi dari perangkat sensor. *Node* bergantung pada daya baterainya untuk melakukan beberapa tugas seperti merasakan, memproses, mengirim, dan menerima data. Karena dimensi mini dari *node* sensor, ukuran baterai juga sangat kecil yang menyimpan lebih sedikit energi. Praktisnya, tidak mungkin untuk mengganti baterai kosong dari *node* sensor karena lebih sering sensor ditempatkan di daerah terpencil untuk mengambil informasi. Selama komunikasi, karena energi total *node* sensor

hilang, *node* menjadi mati dan proses berhenti. Oleh karena itu, sumber daya energi yang langka ini perlu digunakan dengan sangat efisien untuk meningkatkan masa hidup WSN. Pada *direct transmission*, paket data dikirim langsung dari *node* ke *base station* (BS). Karena banyak energi yang hilang dalam waktu singkat dan jaringan segera mati [12].



Gambar 2.3 Komponen node sensor pada *Wireless Sensor Network* [12]

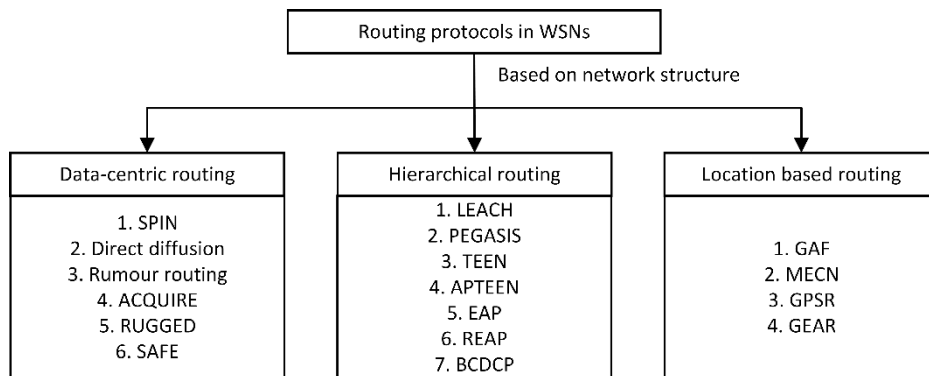
Gambar 2.3 menunjukkan komponen dari *node* sensor pada *Wireless Sensor Network*. Sensor jarak jauh mengatur *hub* brilian memiliki kapasitas korespondensi, persiapan modul dengan memori, unit pendeteksi itu sendiri, dan modul daya yang memungkinkan pengoperasian mandiri. Unit pendeteksi dapat dibuat dari setidaknya satu sensor dan konverter sederhana ke terkomputerisasi (ADC). Ukuran *hub* sensor dapat berkisar dari ukuran kotak sepatu hingga ukuran molekul kecil yang sangat kecil tergantung pada aplikasi tertentu. Biaya *hub* sensor berkisar dari beberapa *dolar* hingga beberapa *sen*. Ukuran dan harga sangat bergantung pada sumber dayanya dan sensor itu sendiri. Ini mengarah pada berbagai model dan kegunaan sensor *hub*. Sebuah sistem sensor mungkin memiliki setidaknya satu sink *hub* (juga disebut sebagai *base station*) yang menyerahkan informasi ke dunia luar. Dalam beberapa kasus yang tidak biasa, sistem dapat menjadi tidak tenggelam, yaitu tidak memiliki *sink hub* [13].

2.2.2 ROUTING PADA WIRELESS SENSOR NETWORK

Protokol *routing* pada WSN memiliki tujuan umum untuk memanfaatkan sumber daya *node* sensor yang terbatas secara efisien untuk memperpanjang masa pakai jaringan. Teknik perutean yang berbeda dapat diadopsi untuk aplikasi yang berbeda berdasarkan persyaratannya. Aplikasi dapat menjadi kritis terhadap waktu atau membutuhkan pembaruan berkala, mereka mungkin memerlukan data yang

akurat atau jaringan yang tahan lama dan kurang tepat, mereka mungkin memerlukan aliran data yang berkelanjutan atau keluaran yang digerakkan oleh peristiwa. Metode perutean bahkan dapat ditingkatkan dan diadaptasi untuk aplikasi tertentu [14].

Secara umum protokol *routing* pada *wireless sensor network* terbagi dalam tiga jenis yaitu *data-centric(flat-based) routing*, *hierarchical-based routing*, dan *location-based routing* tergantung dari struktur jaringan. Dalam *data-centric routing*, seluruh *node* akan diberikan fungsi dan tugas yang sama, semua *node* secara fungsional sama dan berasosiasi dalam merutekan *query* yang diterima dari *base station* ke *event*. Pada *hierarchical-based routing node*, akan diberikan dua fungsi yang berbeda di dalam jaringan yaitu sebagai *cluster head (CH)* dan *cluster-member (CM)*, beberapa *node* menambahkan tanggung jawab untuk mengurangi beban pada *node* lain dalam jaringan. Sedangkan untuk *location-based routing*, memerlukan posisi *node* untuk mengatur pengiriman data pada jaringan, pengetahuan tentang posisi *node* sensor dieksploitasi untuk merutekan kueri dari stasiun pangkalan ke acara [14] [15].



Gambar 2.4 Klasifikasi protokol *routing* pada *Wireless Sensor Network* [14]

Berdasarkan gambar 2.4 penjelasan klasifikasi protokol *routing* pada WSN sebagai berikut :

1) *Data-Centric Routing*

Dalam jumlah besar *node* sensor disebarkan di suatu wilayah sehingga tidak dapat dipahami untuk menetapkan pengidentifikasi global untuk setiap *node*. Hal ini menyebabkan pengembangan teknik perutean berbasis *query* yang dikenal sebagai protokol *routing data-centric*. Berbasis *query*, *base station* mengirimkan *query* ke wilayah tertentu di jaringan yang datanya diperlukan.

Query dikirim ke *node* sensor acak dari *base station*, dan harus diteruskan ke wilayah yang dituju. *Node* sensor di wilayah mengumpulkan data penginderaan mereka dan mengarahkan kembali ke *base station* di sepanjang jalur sebaliknya yang ditemukan pada langkah sebelumnya. Contoh protokol *routing data-centric* yaitu *Sensor Protocols for Information via Negotiation (SPIN)*, *Direct diffusion*, *Rumor routing*, *Active Query Forwarding in Sensor Networks (ACQUIRE)*, *Routing on Fingerprint Gradient in Sensor Networks (RUGGED)*, dan *SAFE*.

2) *Hierarchical-Based Routing*

Hierarchical-based routing adalah prosedur mengatur *router* secara hierarkis. Sebuah protokol hirarkis memungkinkan seorang administrator untuk memanfaatkan *router* yang cepat dan berdaya penuh sebagai *router backbone*, dan *router* yang lebih lambat dan berdaya rendah dapat digunakan untuk tujuan akses. Dengan cara ini, *router* akses membentuk tingkat pertama dari hirarki, dan *router backbone* membentuk tingkat kedua. Protokol *hierarchical* berusaha untuk menjaga lalu lintas lokal tetap lokal, yaitu mereka tidak akan meneruskan lalu lintas ke *backbone* jika tidak diperlukan untuk mencapai tujuan. Contoh protokol *routing hierarchical-based routing* yaitu *Low Energy Adaptive Clustering Hierarchy (LEACH)*, *Power Efficient Gathering in Sensor Information System (PEGASIS)*, *Threshold sensitive energy efficient protocols (TEEN)*, *Adaptive Threshold sensitive energy efficient protocols (APTEEN)*, *Energy Aware routing Protocol (EAP)*, *Ring based Energy Adaptive Protocol (REAP)*, dan *BCDCP*.

3) *Location-Based Routing*

Algoritma *routing* berdasarkan lokasi geografis merupakan subjek penelitian penting dalam WSN. Mereka menggunakan informasi lokasi untuk memandu penemuan dan pemeliharaan perutean serta penerusan paket, sehingga memungkinkan pemilihan perutean terbaik, mengurangi konsumsi energi, dan mengoptimalkan seluruh jaringan. Melalui tiga aspek yang melibatkan skema pembatasan *flooding*, skema partisi area virtual dan skema pemilihan rute terbaik, pentingnya informasi lokasi terlihat dalam algoritma perutean. Contoh protokol *routing location-based* yaitu *Geographic Adaptive Fidelity (GAF)*,

Minimum Energy Communication Network (MECN), Greedy Perimeter Stateless Routing (GPSR), dan Geographic and Energy Aware Routing (GEAR) [14].

2.2.3 CLUSTERING NODE

Kumpulan dari banyaknya sensor, dimana dari banyaknya sensor tersebut yang terletak didalam *cluster* wajib mempunyai kemiripan atau kedekatan disebut *clustering*. Konfigurasi *clustering* dapat dilihat pada Gambar 2.2. Terdapat banyak pendekatan untuk menciptakan sebuah *cluster* diantaranya yaitu membuat aturan yang dapat memerintah anggota dalam grup bersumber pada tingkat kedekatan diantara anggota dan *cluster nodenya*. Pendekatan berikutnya yaitu dengan membuat sekumpulan fungsi yang dapat mengukur sebagian parameter sebagai fungsi dari suatu proses pengklasteran [4]. Jaringan *sigle-tier* dapat memicu kelebihan beban pada *gateway* dengan meningkatnya kepadatan sensor. Kelebihan beban dapat menyebabkan latensi dalam komunikasi dan *tracking* yang tidak memadai. Arsitektur *sigle-tier* juga tidak dapat diskalakan untuk set sensor yang lebih luas karena sensor umumnya tidak dapat melakukan komunikasi jarak jauh. *Clustering* pada WSN sangat berkontribusi dalam skalabilitas sistem secara keseluruhan, masa pakai (*lifetime*), maupun efisiensi energi. Skalabilitas yang dimaksud yaitu perlunya penyeimbangan beban dan penggunaan sumber energi yang efisien. Penerapan *wireless sensor network* sangat luas seperti bidang militer, kesehatan, industri, infrastruktur dan pertanian.

Parameter yang berhubungan dengan seluruh prosedur *clustering* pada WSN berfungsi sebagai sarana yang mendasar untuk perbandingan serta pengkategorian lebih lanjut dari protokol *clustering*. Beberapa parameter *clustering* yaitu :

a. Jumlah *cluster*

Dalam algoritma *clustering random* dan probabilistik baru, proses pemilihan dan pembentukan CH secara alami memvariasikan jumlah *cluster*. Namun dalam beberapa metode jumlah *cluster* telah ditentukan sebelumnya.

b. Komunikasi antar *cluster*

Beberapa teknik *clustering* mengasumsikan bahwa komunikasi antar sensor dan CH ditentukan secara langsung. Namun pada saat ini komunikasi *cluster multi-hop* banyak diperlukan, yaitu pada saat jumlah *node* sensor sangat besar dan jumlah saluran terbatas.

c. *Mobilitas node dan cluster head*

Dengan asumsi *node* sensor diam dan CH diam, biasanya mengarah ke *cluster* yang stabil dengan manajemen jaringan *intercluster* dan *intracluster* yang mudah. Sebaliknya, apabila *node* dan CH diasumsikan secara *mobile*, keanggotaan *cluster* harus berubah secara dinamis pada basis *node-by-node*. Ini menjadikan *cluster* untuk berkembang dari waktu ke waktu.

d. *Jenis dan peran node*

Beberapa model jaringan yang diusulkan seperti lingkungan heterogen, mengasumsikan bahwa CH memiliki sumber daya komputasi dan komunikasi yang jauh lebih banyak daripada model lainnya. Dalam model jaringan yang paling umum, semua *node* memiliki fungsi yang sama dan hanya beberapa sensor yang digunakan yang disebut CH.

e. *Pemilihan cluster head*

Dalam beberapa algoritma untuk lingkungan heterogen, *node* CH dapat ditentukan sebelumnya. Namun dalam banyak kasus untuk lingkungan homogen, CH menyebar secara probabilistik atau sepenuhnya secara acak.

f. *Metodologi pembentukan cluster*

Dalam teknik pendekatan terbaru, apabila CH hanya sebuah *node* sensorik normal dan efisiensi waktu adalah parameter utama, *clustering* dilakukan terdistribusi tanpa koordinasi. Beberapa pendekatan sebelumnya telah menggunakan pendekatan terpusat dengan menggunakan satu atau lebih koordinator untuk mempartisi secara mandiri seluruh jaringan dan mengelola anggota *cluster*.

g. *Multiple level*

Beberapa metode telah memperkenalkan konsep tingkat *cluster* berjenjang untuk mencapai distribusi daya dan konsumsi daya yang lebih baik.

h. *Kompleksitas algoritma*

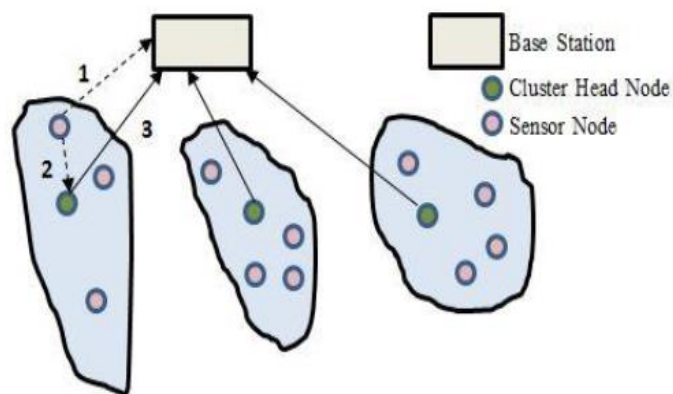
Dalam algoritma terbaru, penyelesaian secara cepat dari eksekusi protokol adalah salah satu tujuan utama. Oleh karena itu, kompleksitas waktu dari sebagian besar prosedur pembentukan *cluster* saat ini adalah konstan. Namun dalam beberapa protokol sebelumnya, kompleksitas waktu dapat bervariasi tergantung pada jumlah total sensor dalam jaringan yang berfokus pada kriteria lain.

i. *Overlapping*

Beberapa protokol sangat mementingkan konsep tumpang tindih (*overlapping*) *node* pada *cluster* yang berbeda. Namun sebagian besar protokol yang dikenal masih mencoba meminimalkan atau menghilangkan tumpang tindih [16].

2.2.4 LOW ENERGY ADAPTIVE CLUSTERING HIERARCHY-CENTRALIZED (LEACH-C)

LEACH-C adalah protokol perutean yang terpusat yang merupakan pengembangan dari protokol LEACH. LEACH-C merupakan jenis pembaruan dimana pemilihan *cluster head* dan perhitungan *cluster* dilakukan di *base station* [17]. Pada algoritma LEACH-C, *node* sensor dikelompokkan ke dalam sebuah *cluster* atau lebih pada sebuah jaringan, dimana setiap *cluster* memiliki sebuah kepala *cluster* (*cluster head*) dan mempunyai dua tugas ganda yaitu, memantau data dan mengontrol pengiriman data dari *node* sensor ke BS. Arsitektur LEACH-C dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur LEACH-C [18]

Namun, alih-alih memilih sendiri *node* secara acak sebagai CH, algoritma terpusat (*centralized*) dilakukan oleh *sink* di LEACH-C. *Sink* mengumpulkan

informasi lokasi dari *node*, dan kemudian mem-*broadcast* (menyebarkan) keputusannya tentang *node* mana yang akan bertindak sebagai CH kembali ke *node*. Performa LEACH-C secara keseluruhan lebih baik daripada LEACH karena memindahkan tugas pembentukan *cluster* ke *sink*. Namun, LEACH-C peka terhadap lokasi *sink*. Setelah *cost* energi untuk berkomunikasi dengan *sink* menjadi lebih tinggi daripada total energi untuk pembentukan *cluster*, LEACH-C tidak lagi memberikan kinerja yang baik. *Sink* mungkin terletak jauh dari jaringan di sebagian besar aplikasi WSN. Jadi, ketergantungan pada lokasi *sink* merupakan kelemahan utama dari LEACH-C [19]. Perbandingan antara LEACH dan LEACH-C dijabarkan pada Tabel 2.1.

Tabel 2.1 Perbedaan antara LEACH dan LEACH-C [20]

LEACH	LEACH-C
Merupakan algoritma pengelompokan terdistribusi	Merupakan algoritma pengelompokan terpusat
Setiap <i>node</i> dapat memilih dirinya sendiri sebagai <i>cluster head</i> yang independen dari <i>node</i> lain	<i>Cluster head</i> dipilih oleh <i>base station</i>
<i>Cluster head</i> dipilih berdasarkan <i>Probabilistic threshold</i> yaitu dipilih secara acak oleh <i>node</i>	BS menjalankan algoritma pembentukan kluster terpusat untuk memilih CH berdasarkan tingkat energi sebuah <i>node</i> dan jaraknya dari BS
Tidak ada jaminan bahwa jumlah <i>cluster head</i> yang diinginkan akan terbentuk atau pemerataan <i>cluster head</i> dalam jaringan	Jumlah <i>cluster head</i> yang diinginkan akan dibuat dan didistribusikan secara merata di antara <i>node</i> dalam jaringan
Tahap <i>set up</i> terdiri dari pemilihan <i>cluster head</i> secara acak, sehingga setiap <i>node</i> menjadi <i>cluster head</i> minimal satu kali	Setiap <i>node</i> mungkin tidak mendapat kesempatan untuk menjadi CH, dan <i>node</i> yang sama dapat menjadi CH untuk putaran berikutnya karena BS mengambil kendali jaringan
<i>Lifetime</i> jaringan akan lebih sedikit dibandingkan dengan LEACH-C	<i>Lifetime</i> jaringan akan lebih lama dibandingkan dengan LEACH

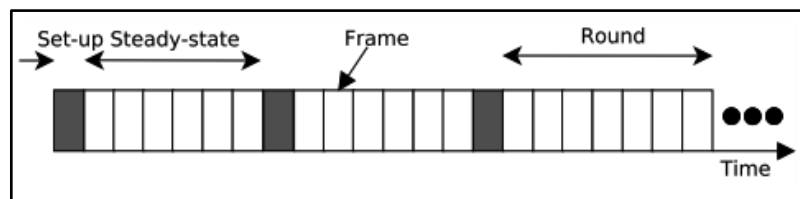
LEACH	LEACH-C
Disipasi energi awal akan lebih sedikit dibandingkan dengan LEACH-C	Disipasi energi awal akan lebih banyak dibandingkan dengan LEACH
Sinyal data yang diterima di BS akan lebih sedikit dibandingkan dengan LEACH-C	Sinyal data yang diterima di BS akan lebih banyak dibandingkan dengan LEACH
Disipasi energi total akan lebih banyak dibandingkan dengan LEACH-C	Disipasi energi total akan lebih sedikit dibandingkan dengan LEACH

LEACH-C memproses sejumlah putaran (*round*) dimana masing-masing putaran terdiri dari dua fase yaitu fase *setup* dan fase *steady-state*. Agar tidak menguras energi dari beberapa *node* sensor, terutama *node cluster-head*, LEACH-C menggunakan algoritma pengelompokan (*clustering*) terpusat untuk memilih CH pada fase *set-up*. Keunggulan dari LEACH-C adalah dapat mengalokasikan konsumsi energi secara menyeluruh antar *node* sensor dengan menempatkan CH ke pusat *cluster*. Setiap *node* sensor harus dimuat dengan perangkat penerima GPS namun tidak menjamin keseimbangan konsumsi energi dari seluruh jaringan sensor [4] [21]. Karakteristik dari algoritma protokol LEACH-C yaitu :

- a. LEACH-C merupakan algoritma *clustering* terpusat.
- b. Pemilihan *cluster head* berada di *base station*.
- c. *Base station* menjalankan algoritma *clustering* sentralized untuk memilih CH dari tingkat energi *node* dan jarak dari BS.
- d. Jumlah *cluster head* yang akan dibuat didistribusikan secara merata pada *node* jaringan.
- e. Tidak setiap *node* memiliki kesempatan untuk dapat menjadi CH, dan *node* yang sama dapat memiliki kesempatan untuk menjadi CH pada putaran berikutnya ketika BS mengontrol jaringan.
- f. Dibandingkan dengan LEACH, LEACH-C memiliki *lifetime* pada jaringan yang lebih tinggi.
- g. Disipasi energi saat *stae-up* akan lebih besar dibandingkan dengan LEACH.
- h. Pada LEACH-C, BS akan lebih banyak menerima sinyal data dibandingkan dengan LEACH.

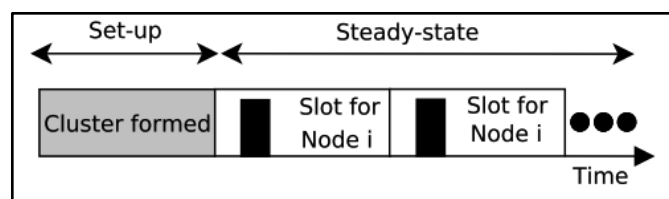
- i. Total disipasi energi pada LEACH-C lebih sedikit bila dibandingkan dengan LEACH [20].

Proses awal pada protokol LEACH-C dimana semua *node* mengirimkan informasi energi dan lokasinya ke *base station* di setiap putaran (*round*). Kemudian *base station* memutuskan *cluster head* dan anggota kluster dengan kemungkinan pengelompokan terbaik tergantung pada jarak antara *cluster head* dan anggota kluster. *Node-node* yang memiliki posisi yang sesuai sehingga biaya (*cost*) komunikasinya minimum dan sisa energinya di atas energi rata-rata jaringan, *node* tersebut akan dipilih sebagai *cluster head*. *Node* menyadari apakah mereka adalah *cluster head* atau anggota cluster, sehingga mereka menjadwalkan transmisi mereka menggunakan skema TDMA. *Cluster head* mengagregasi informasi yang datang dari *node* anggota ke *base station* pada interval reguler, di akhir *frame* TDMA di cluster tertentu. Proses ini diulangi di setiap putaran selama fase *setup*. Fase *Steady state* mirip dengan protokol LEACH [22].



Gambar 2.6 Proses *round* (putaran) pada LEACH-C [22].

Pada gambar 2.6 menunjukkan bagaimana proses protokol LEACH-C seiring waktu. Awalnya membentuk *cluster* selama fase *setup* dan kemudian setelah itu fase *steady-state* dimulai. Setiap fase *steady-state* diberi interval waktu tetap. Dalam interval ini, setiap anggota *cluster* mengirimkan data ke *cluster-head* masing-masing. Bergantung pada jumlah anggota yang ada di setiap *cluster*, setiap anggota akan mendapatkan slot TDMA secara berkala seperti yang ditunjukkan pada Gambar 2.7 [22].



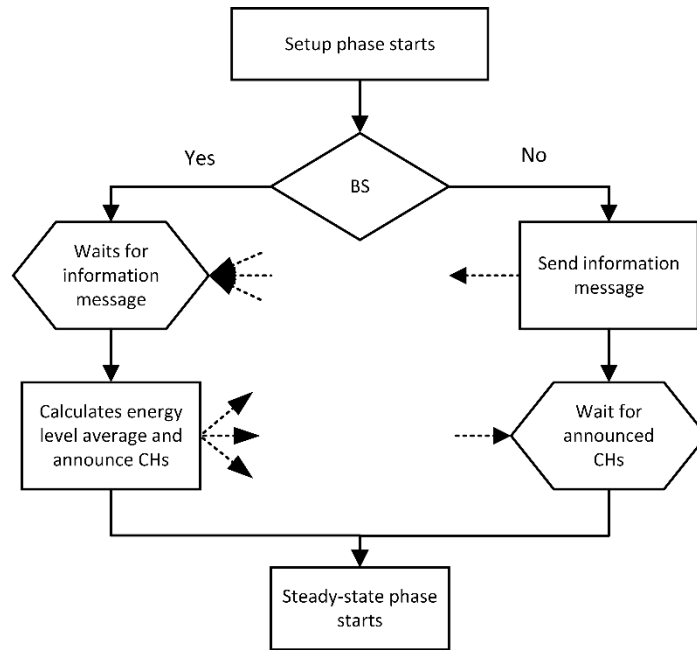
Gambar 2.7 Layout dari LEACH-C [22].

2.2.5 SET-UP PHASE

Fase *setup* adalah *fase* dimana seluruh *node* menginformasikan mengenai status *node*, energi yang tersisa, dan lokasi *node* kepada *Base Station* (BS). BS harus memastikan bahwa energi yang didistribusikan terbagi secara merata ke seluruh *node* agar dapat menentukan *cluster* yang optimal. Untuk memastikan bahwa energi dapat terdistribusi secara merata hal pertama yang harus dilakukan adalah menghitung energi rata-rata dari masing-masing *node*, apabila energi *node* kurang dari energi rata-rata, maka *node* tidak memiliki kesempatan untuk menjadi *cluster head* pada putaran tersebut. Untuk menghitung energi rata-rata setiap *round* dapat dilihat pada persamaan 2.1.

$$\text{energi rata - rata} = \frac{\text{jumlah energi tiap node}}{\text{jumlah node}} \quad (2.1)$$

Selanjutnya *Base Station* memilih *node* CH secara *random* berdasarkan jumlah cluster. Kemudian menentukan bit cost dengan menghitung jarak dari *node* CH ke *non-CH* sebagai nilai global optimum awal. Setelah itu BS secara *random* menentukan CH baru dari *node* yang sudah didapatkan berdasarkan energi rata-rata. Selanjutnya menghitung kembali jarak CH yang baru dengan jarak *non-CH* untuk dibandingkan dengan nilai global optimum awal yang baru. Apabila nilai *cost* lebih tinggi atau sama dengan nilai global optimum baru, maka langkah selanjutnya membandingkan nilai probabilitas *random* (dari 0 sampai 1), jika nilai probabilitas lebih kecil atau sama dengan probabilitas P_k , maka nilai *cost* tersebut menjadi nilai global optimum, dan seterusnya sampai jumlah maksimum iterasi dan tingkat pembaruan global optimum tercapai.



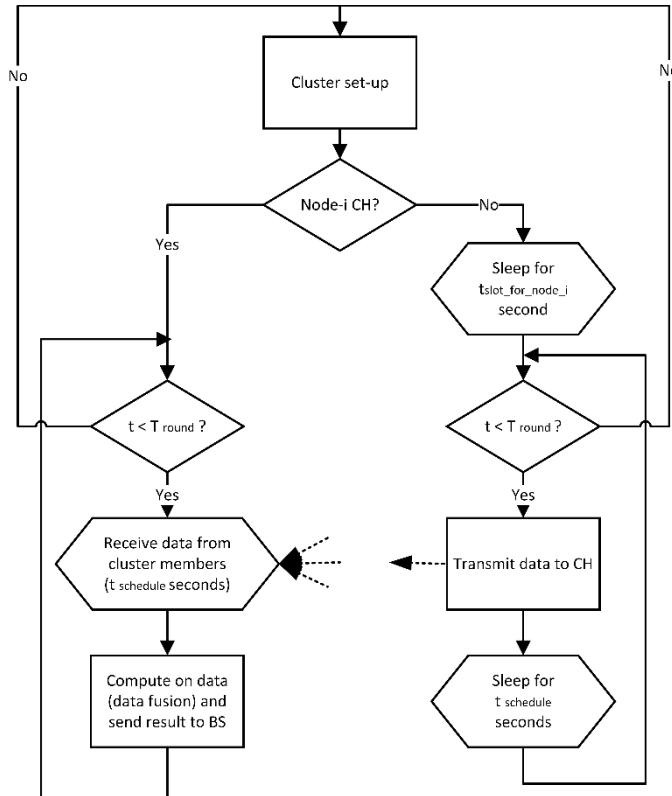
Gambar 2.8 Flowchart *set-up phase* dari protokol LEACH-C [3]

Pada Gambar 2.8 merupakan *flowchart* (alur) dari fase *set-up* protokol LEACH-C. LEACH-C menggunakan algoritma pengelompokan terpusat untuk memilih CH dalam fase *set-up*. Alih-alih memilih CH secara acak, semua *node* WSN mengirim informasi mereka, yang merupakan lokasi dan tingkat energi mereka saat ini, ke BS. Setelah itu, BS menghitung rata-rata tingkat energi semua *node*. Setiap *node* yang memiliki energi lebih dari rata-rata yang dihitung akan memiliki peluang untuk menjadi *cluster head*. Ketika BS menerima informasi yang dikirim oleh *node* dan menghitung energi rata-rata, BS memutuskan *node* mana yang akan menjadi CH dan mengumumkannya ke seluruh WSN [3].

2.2.6 STEADY-STATE PHASE

Fase *steady-state* merupakan fase dimana WSN menggabungkan data dan mentransfernya ke *base station* (BS). Proses transmisi dapat dilakukan apabila *cluster* dan jadwal TDMA sudah terbentuk. Waktu yang diperlukan CH untuk mengirim data ke BS berdasarkan banyaknya *node* dalam *cluster*, sehingga durasi waktu untuk transmisi setiap *node* berukuran tetap. Fase *steady state* memerlukan waktu lebih lama dari fase *setup*, karena transfer data yang dilakukan melalui radio transmisi secara intensif. Setiap *node non cluster head* yang mempunyai data untuk di transfer akan mengirimkan datanya ke CH. Sementara itu radio akan dimatikan

hingga waktu transmisi selesai untuk menghemat energi yang ada di *node non cluster head*. CH juga harus menjaga penerimanya agar selalu siap untuk menerima data dari *node cluster*. Apabila CH sudah menerima seluruh data, CH akan melakukan pemrosesan sinyal untuk menghasilkan sinyal tunggal komposit lalu mengirimkannya ke BS.



Gambar 2.9 Flowchart *steady-state phase* dari protokol LEACH-C [3]

Pada Gambar 2.9 merupakan *flowchart* (alur) dari fase *steady-state* protokol LEACH-C. Fase *steady-state* dilakukan dengan cara yang sama seperti pada protokol LEACH. Modifikasi LEACH yang diwakili oleh protokol LEACH-C ini dimaksudkan untuk memperpanjang masa pakai WSN dan menurunkan disipasi energi. BS menerima data lokasi dari *node*, memutuskan *node* mana yang akan dipilih sebagai CH, dan mengiklankannya ke seluruh WSN. Dalam LEACH-C, *node* dengan tingkat energi lebih besar dari batas yang telah ditentukan, yaitu tingkat energi rata-rata yang dihitung oleh BS, dipilih menjadi CH oleh BS. Berbeda dengan LEACH, dimana setiap *node* memiliki kesempatan untuk menjadi CH pada putaran yang berbeda, tidak semua *node* pada LEACH-C memiliki kesempatan yang sama [3].

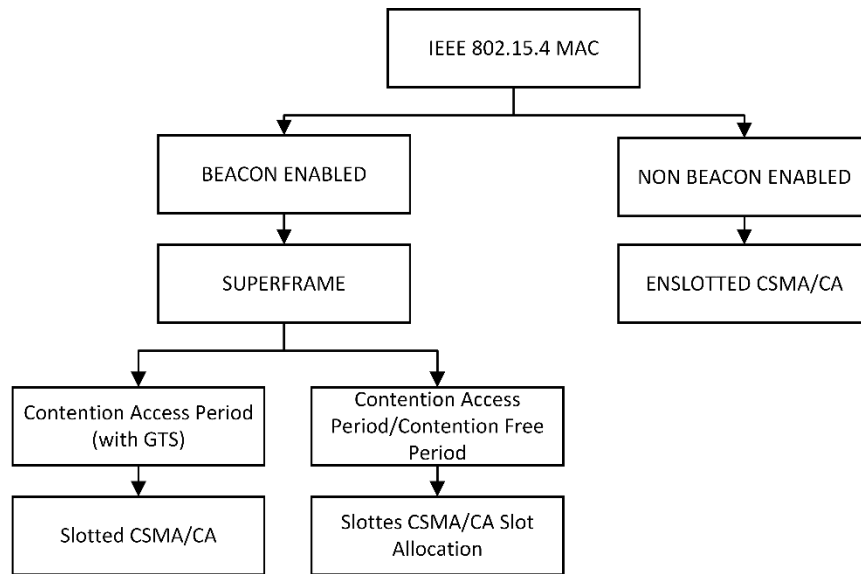
2.2.7 STANDAR IEEE 802.15.4

WSN menggunakan standar IEEE 802.15.4. Pada standar ini mengatur penggunaan frekuensi aplikasi WSN pada *band* ISM yaitu untuk pada kawasan Eropa menggunakan frekuensi 868 MHz, pada kawasan Amerika menggunakan 915 MHz, dan untuk global menggunakan frekuensi 2,4 GHz. Rangkaian standar IEEE 802.15.4 / ZigBee umumnya diakui sebagai teknologi pilihan untuk aplikasi yang melibatkan jaringan sensor, karena kemampuannya untuk memastikan komunikasi yang andal, berdaya rendah, dan hemat biaya [23].

Standar IEEE 802.15.4 dibuat untuk perangkat berdaya rendah yang beroperasi pada pita frekuensi 868 MHz, 915 MHz, dan 2,45 GHz. Kecepatan data yang didukung oleh standar ini adalah 20, 40, dan 250 kbps; agak sederhana dibandingkan dengan protokol lain seperti IEEE 802.11 (misalnya, IEEE 802.11a menawarkan kecepatan data hingga 54 Mbps). Sebelum standar ini dikembangkan, *ZigBee Alliance* mengerjakan teknologi komunikasi berbiaya rendah untuk laju data rendah dan konsumsi daya rendah. IEEE dan *ZigBee Alliance* akhirnya bergabung dan *ZigBee* telah menjadi nama komersial untuk teknologi IEEE 802.15.4. Sementara IEEE 802.15.4 telah digunakan secara luas di WSN, ada sejumlah masalah dengan standar ini. Sebagai contoh, sementara pertukaran pesan dan operasi didefinisikan dengan baik untuk topologi *star*, standar tidak secara jelas mendefinisikan operasi pendekatan *peer-to-peer*. Dalam lingkup WSN yang besar, tidak mungkin semua perangkat dapat menggunakan koordinator PAN yang sama. Meskipun standar memungkinkan komunikasi antar koordinator PAN, hal ini lagi-lagi tidak didefinisikan dengan baik [1].

Dua jenis *node* dibentuk oleh IEEE 802.15.4 yaitu *Full Function Device* (FFD) dan *Reduced Function Device* (RFD). *Node* FFD dapat berkomunikasi dengan FFD dan RFD, dapat berupa Koordinator PAN atau perangkat akhir tetapi RFD hanya dapat berkomunikasi dengan FFD yang terkait. *Node* RFD hanya bisa berupa perangkat akhir. Ada tiga topologi yang dapat dibentuk oleh *node* FFD dan RFD yaitu topologi *star*, topologi *mesh* dan *peer to peer*. Dalam topologi bintang, setiap FFD berkomunikasi dengan RFD yang terkait. Dalam koordinator jaringan *Mesh* memulai parameter jaringan. Itu harus mengaktifkan tautan *peer-to-peer*

penuh. Selain itu pada tipe ketiga, kami mengintegrasikan hanya satu koordinator PAN (FFD) meskipun ada banyak FFD yang terkait dengan jaringan [24].

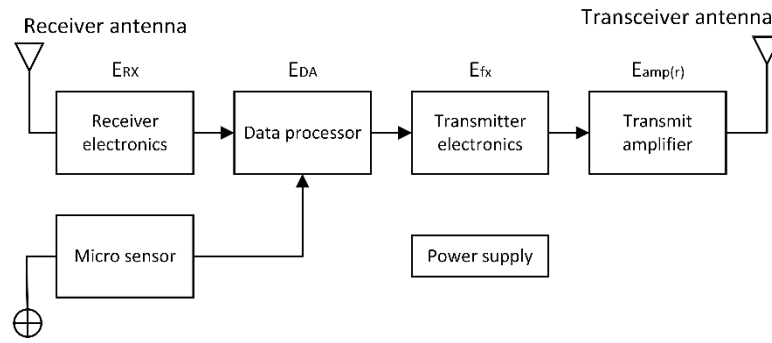


Gambar 2.10 Mekanisme MAC pada IEEE 802.15.4 [24]

Dua mode operasi tersedia di sublapisan MAC baik dalam mekanisme *beacon-enabled* atau *non-beacon-enabled* yang digambarkan oleh gambar 2.10. Dalam *beacon-enabled*, *slot* CSMA-CA diperbolehkan untuk transmisi data. Meskipun dalam mode *non-beacon-enabled*, CSMA-CA *unslotted* diaktifkan. Koordinator memilih antara dua mode. Dalam mekanisme pertama, frame *beacon* ditransmisikan secara teratur oleh koordinator untuk menyinkronkan RFD yang terkait dengannya, untuk menentukan PAN-nya dan untuk menggambarkan arsitektur *Superframe*. Dua suar berurutan dipisahkan oleh periode waktu yang diberi nama *Beacon Interval* (BI). Ini dibagi dalam periode aktif yang disebut juga *Superframe Duration* (SD) dan durasi tidak aktif [24].

2.2.8 ENERGY MODEL

Pada Gambar 2.11 merupakan model konsumsi energi radio. Bagian perangkat komunikasi *node* sensor dibagi menjadi penerima (*receiver*), pemancar (*transmitter*), dan penguat (*repeater*). Propagasi gelombang elektromagnetik dimodelkan menggunakan jarak antara *receiver* dan *transmitter*. Nilai konsumsi energi pada *transmitter* adalah hasil dari besarnya energi dari radio elektronik serta energi *amplifier*, sedangkan besarnya konsumsi energi *receiver* hanya dipengaruhi oleh energi radio elektronik.



Gambar 2.11 Komponen utama dalam konsumsi energi radio [25].

E_{RX} merupakan hilangnya energi per-bit pada sisi penerima, dan E_{TX} yaitu hilangnya energi per-bit pada saat transmisi. Dibutuhkannya energi oleh penguat transmisi untuk mempertahankan *signal-to-noise* yang telah diterima agar dapat mentransmisikan data dengan baik disebut dengan $E_{amp(r)}$, sedangkan E_{DA} atau energi agregasi data merupakan energi yang hilang pada saat pengoperasian data (agregasi) [26].

Konsumsi energi untuk pengiriman dan penerimaan data yang dipisahkan oleh jarak (d) menggunakan dua persamaan 2.2 [25] :

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{tx-amp}(k, d) \quad (2.2)$$

$$E_{Tx}(k, d) = E_{elec} * k + E_{amp} * k * d^2$$

Dan untuk penerimaan menggunakan persamaan 2.3 [25] :

$$E_{Rx}(k) = E_{Rx-elec}(k) \quad (2.3)$$

$$E_{Rx}(k) = E_{elec} * k$$

Sehingga $E_{Tx}(k, d)$ adalah energi total yang dihamburkan di pemancar dari *node* sumber untuk mengirimkan pesan dengan panjang d dan $E_{Rx}(k)$ adalah energi total disipasi (hilang) di dalam *receiver* dari *node* tujuan ke penerima dengan panjang k -bit. Pada persamaan 2.2 E_{elec} merupakan energi pada *node* sensor yang digunakan untuk mengoperasikan *circuit* perangkat radio, dan E_{amp} adalah energi penguat sinyal pengirim pada *receiver*. Beberapa parameter yang mempengaruhi hasil dari nilai E_{elec} dan E_{TX-amp} yaitu, panjang gelombang pengirim, *bit rate* (R_b), penguatan (G), tinggi antenna (h), dan sensitivitas penerima (P -*Thresh*).

Gelombang elektromagnetik dapat dihitung menggunakan dua cara yaitu, apabila jarak diantara *transmitter* dan *receiver* kurang dari nilai *threshold* maka model yang digunakan adalah *free space* (d_2). Jika jarak *transmitter* dan *receiver*

lebih besar dari nilai *threshold* maka model propagasi yang digunakan adalah *two-ray ground*.

Nilai transmit energi amplifier pada model propagasi *free space* (d_2) menggunakan persamaan 2.4 [25] :

$$\varepsilon_{fris_amp} = \frac{P_{R-Threshold} \cdot (4\pi)^2}{R_b \cdot G_t \cdot G_r \cdot \lambda^2} \quad (2.4)$$

Keterangan :

ε_{fris_amp} = energi *amplifier* (J/bit/m²).

PR-Threshold = minimum *power* penerima *packet* (W).

R_b = *bandwidth* (bit/detik).

G_t = antena pengirim *gain*.

G_r = antena penerima *gain*.

λ = panjang gelombang sinyal *carrier* (m).

Untuk menghitung nilai transmit amplifier pada model propagasi *two-ray ground* (d_4) menggunakan persamaan 2.5 [25] :

$$\varepsilon_{two_ray_amp} = \frac{P_{R-Threshold}}{R_b \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2} \quad (2.5)$$

Keterangan ;

$\varepsilon_{two_ray_amp}$ = energi *two-ray amplifier* (J/bit/m⁴).

PR-Threshold = minimum *power* penerima paket (W).

R_b = *bandwidth* (bit/detik).

G_t = antena pengirim *gain*.

G_r = antena penerima *gain*.

h_r = tinggi antena penerima (m).

h_t = tinggi antena pengirim (m).

$$E_{amp}(d) = \begin{cases} \varepsilon_{fs} d^2, & d \leq d_0 \\ \varepsilon_{mp} d^4, & d > d_0 \end{cases} \quad (2.6)$$

Keterangan :

ε_{amp} = energi amplifier (J/bit/m²)

2.2.9 QUALITY OF SERVICE (QoS)

Quality Of Service merupakan kemampuan untuk menyediakan kualitas layanan yang baik dari suatu jaringan dengan mengatasi *jitter* dan *delay*, serta menyediakan *bandwidth*. Parameter dari QoS adalah *jitter*, *latency*, *packet loss*, *throughput*, *echo cancellation*, MOS, dan PDD. Standarisasi QoS versi TIPHON TR 101 329 V2.1.1 (1999-06) [27] dapat dilihat pada tabel 2.2.

Tabel 2.2 Standarisasi QoS versi TIPHON.

Nilai	Persentase (%)	Indeks
3,8 - 4	95 - 100	Sangat memuaskan
3 - 3,79	75 - 94,75	Memuaskan
2 - 2,99	50 - 74,75	Kurang memuaskan
1 - 1,99	25 - 49,75	Jelek

Packet loss adalah parameter yang menggambarkan suatu kondisi yang menunjukkan banyaknya jumlah total paket yang hilang. *Packet loss* dapat terjadi karena *collision* dan *congestion* yang berpengaruh pada semua aplikasi karena proses transmisi akan menurunkan efisiensi jaringan secara keseluruhan walaupun jumlah *bandwidth* cukup tersedia untuk aplikasi tersebut [28]. Standarisasi *Packet Loss* versi TIPHON TR 101 329 V2.1.1 (1999-06) yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Standarisasi *Packet Loss* versi TIPHON.

Kategori Degradasi	<i>Packet Loss</i>	Indeks
Sangat bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

Untuk menghitung *packet loss* dapat menggunakan rumus :

$$packet\ loss = \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim} \times 100\% \quad (2.7)$$

2.2.10 MATLAB

Matrix Laboratory atau biasa dikenal dengan MATLAB adalah sebuah perangkat lunak berkemampuan tinggi yang cocok digunakan untuk komputasi numerik. Matlab merupakan sebuah alat yang mudah digunakan karena masalah dan solusi dijelaskan dalam notasi matematika sehingga memiliki jumlah pengguna yang signifikan di antara lain yaitu peneliti, mahasiswa dan lain sebagainya [29]. Fitur matlab yang sangat penting yaitu matlab adalah platform yang mudah digunakan bagi pengguna karena pengguna dapat mengembangkan fungsi kustom mereka sendiri serta kemudahannya dalam kemampuan pemrograman [30].

Di sisi lain, MATLAB dianggap sebagai bahasa dasar untuk beberapa simulator, seperti PROWLER. Selain itu, diperlukan alat lain yang harus dipasang seperti PiccSIM. Selain itu, MATLAB disertakan sebagai *toolbox* di simulator lain, misalnya LabVIEW. Mengenai fitur-fitur MATLAB, beberapa yang paling signifikan adalah kemudahan kemampuan pemrograman dan kepemilikan platform yang mudah yang menarik pengguna untuk mengembangkan fungsinya sendiri. Selain itu, MATLAB memiliki banyak *toolbox*, seperti *Control System Design*, *Aerospace*, *Fuzzy Logic*, *Statistics*, *Symbolic Computations*, *Communication*, dan beberapa lainnya [3].