

**ROUTING PROTOCOL WIRELESS SENSOR NETWORK DENGAN
METODE FUZZY SUGENO**

SKRIPSI

TEKNIK ELEKTRO

KONSENTRASI TEKNIK TELEKOMUNIKASI

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik



ABDAN SYAKUR AMRULLAH

NIM. 145060301111017

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2021



LEMBAR PENGESAHAN
ROUTING PROTOCOL WIRELESS SENSOR NETWORK DENGAN METODE
FUZZY SUGENO

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK TELEKOMUNIKASI

Diajukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



ABDAN SYAKUR AMRULLAH

NIM. 145060301111017

Skripsi ini telah disetujui oleh dosen pembimbing
Pada tanggal 27 Juli 2021

Dosen Pembimbing I

Rusmi Ambarwati, S.T., M.T.
NIP. 19720204 200003 2 002

Dosen Pembimbing II

Ir. Wahyu Adi Priyono, M.T.
NIP. 19600518 198802 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro



Muslim, S.T., M.T., Ph.D.

NIP. 19741203 200012 1 001

JUDUL SKRIPSI :
ROUTING PROTOCOL WIRELESS SENSOR NETWORK DENGAN METODE
FUZZY SUGENO

Nama Mahasiswa : Abdan Syakur Amrullah

NIM : 145060301111017

Program Studi : Teknik Elektro

Konsentrasi : Teknik Telekomunikasi

KOMISI PEMBIMBING :

Dosen Pembimbing 1 : Rusmi Ambarwati, S.T., M.T.

Dosen Pembimbing 2 : Ir. Wahyu Adi Priyono, M.T.

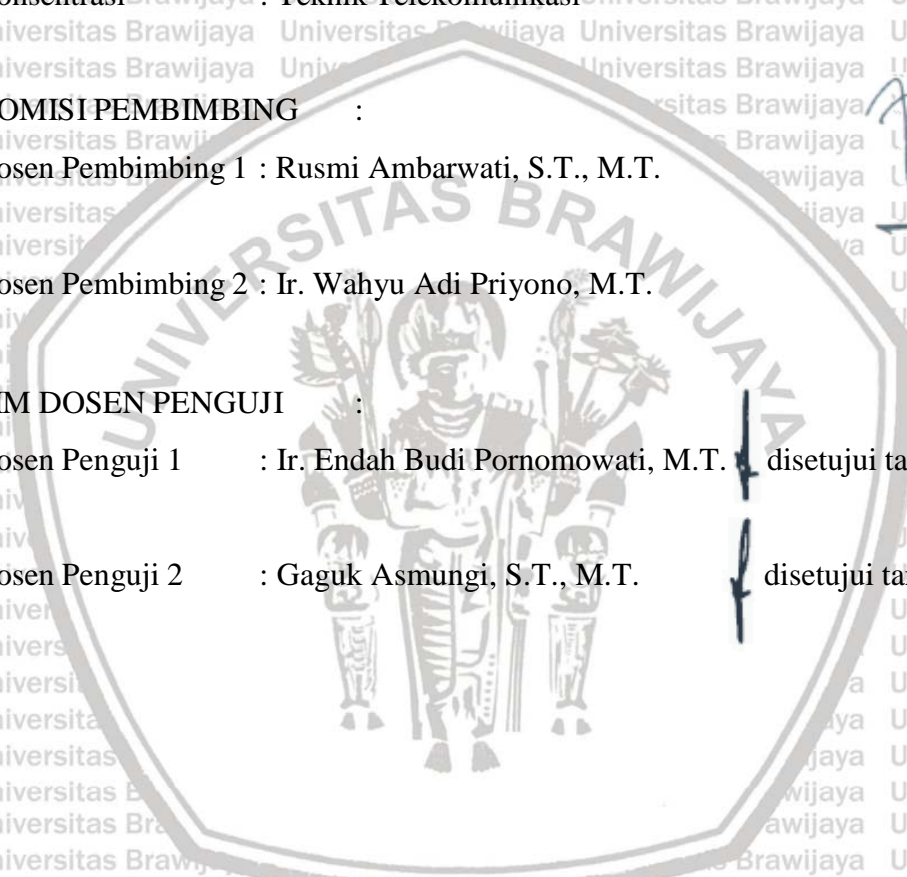
TIM DOSEN PENGUJI :

Dosen Penguji 1 : Ir. Endah Budi Pornomowati, M.T. **disetujui tanggal 26 Juli 2021**

Dosen Penguji 2 : Gaguk Asmungi, S.T., M.T. **disetujui tanggal 27 Juli 2021**

Tanggal Ujian : 22 Juli 2021

SK Penguji : No. 1253 Tahun 2021



[Handwritten signatures]





***Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda tercinta***

LEMBAR KEASLIAN

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan ditulis di dalam Naskah Skripsi ini adalah asli dari pemikiran saya, tidak terdapat karya ilmiah yang pernah diajukan orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis dan diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apalagi ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 28 Juni 2021

Mahasiswa,



Abdan Syakur Amrullah

NIM 145060301111017



RIWAYAT HIDUP

Penulis bernama lengkap Abdan Syakur Amrullah, lahir di kota Kediri, Provinsi Jawa Timur pada tanggal 11 Agustus 1996. Penulis merupakan anak pertama dari Bapak Taat Heru Sunarko dan Ibu Elis Yuliani. Pada tahun 2008, penulis berhasil menyelesaikan masa studi di SD YPPSB 1 Sangatta. Kemudian penulis melanjutkan sekolah tingkat pertama dan lulus pada tahun 2011 di SMP Ar Risalah Kediri. Selanjutnya penulis melanjutkan studi di SMA Ar Risalah Kediri.

Pada tahun 2014 penulis melanjutkan studi perguruan tinggi di Program Sarjana Teknik Elektro Fakultas Teknik Universitas Brawijaya. Selama menjadi mahasiswa di universitas brawijaya penulis aktif mengikuti organisasi Workshop divisi Robotika sub divisi KRPAI berkaki. Dalam organisasi Workshop divisi Robotika sub divisi KRPAI berkaki penulis bertugas sebagai *junior electrical designer* di tahun 2015/2016, *senior electrical designer* di tahun kedua, lalu menjadi pelatih dan penasihat *electrical designer* di tahun ketiga. Selain itu penulis juga aktif mengikuti kegiatan lomba tingkat nasional seperti KRI 2017 divisi Kontes Robot Pemadam Api Indonesia - Berkaki.

Malang, 28 Juli 2021

Penulis

RINGKASAN

Abdan Syakur Amrullah, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Juli 2021, *Routing Protocol Wireless Sensor Network* dengan Metode Fuzzy Sugeno, Dosen Pembimbing: Rusmi Ambarwati, S.T., M.T. dan Ir. Wahyu Adi Priyono, M.T.

Wireless Sensor Network (WSN) atau Sensor Jaringan Nirkabel merupakan sebuah jaringan *wireless* yang mengandung beberapa perangkat terdistribusi yang otonom dan menggunakan sensor untuk secara bersama memonitor suatu keadaan fisik atau kondisi lingkungan. WSN terdiri atas sejumlah besar *sensor node* yang bebas. Setiap node bertujuan untuk mengirim informasi menuju *base station* melalui node-node terdekatnya.

Masalah yang ada pada *node* sensor yaitu keterbatasan pada energi dan *bandwith* yang mengharuskan konsiderasi terhadap energi pada semua tingkatan layer di jaringan. *Routing* adalah protokol yang digunakan untuk mendapatkan rute dari satu jaringan ke jaringan lainnya. Rute ini disebut *route*, dan informasi *route* dapat ditetapkan secara dinamis ke *router* lain atau secara statis ditetapkan ke *router* lain. *Router* menggunakan informasi ini untuk membangun dan memperbaiki tabel peruteannya. Jenis lain dari *routing protocol* adalah *cluster based*. Beberapa *cluster* dibentuk dan *cluster-head* dipilih. Ketika *routing*, sensor mengirim paket ke *head cluster* pertama dan *head-cluster* kemudian mengambil tanggung jawab meneruskan paket ke *sink*. *Cluster* berisi *cluster head* dan semua *node* langsung. *Nonhead node* berkomunikasi dengan yang lain melalui *head cluster* tersebut.

Dari permasalahan di atas, penulis merancang sebuah aplikasi simulasi Matlab khusus protokol *routing* menggunakan *fuzzy* Sugeno. Dengan *routing protocol indicator-based* setiap node menyusun indikatornya masing-masing untuk membantu prosen *routing*. Kemudian dengan menerapkan fungsi keanggotaan, setiap node dapat menentukan *Cluster Head* (CH) nya masing-masing sesuai hasil pembobotannya. Kedua, dengan metode *Fuzzy* Sugeno, setiap node akan menentukan rute pengiriman data paling efisien untuk dapat menghemat daya dan waktu sehingga sampai pada *sink*.

Hasil yang didapat dari penelitian ini adalah dengan 50 node bercakupan 200 m² yang tersebar dalam wilayah 500 m², jumlah *cycle* maksimum adalah 1000 *cycle*, dan *sink* berada di tengah area. Seluruh *packet* informasi diterima saat putaran ke-282 atau dalam waktu 4 menit 42 detik. Rata-rata maksimal penggunaan daya setiap node adalah 13 mAh (dengan asumsi setiap node berkapasitas 100mAh) dan berkurang seiring putaran yang terjadi karena semakin sedikit node yang aktif dalam sistem.

Kata kunci: *Wireles Sensor Network, Roting Protocol, Indicator-Based, Fuzzy Sugeno.*

SUMMARY

Abdan Syakur Amrullah, *Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, July 2021, Routing Protocol Wireless Sensor Network with Fuzzy Sugeno Method, Supervisor: Rusmi Ambarwati, S.T., M.T. and Ir. Wahyu Adi Priyono, M.T.*

Wireless Sensor Network (WSN) or Wireless Sensor Network is a wireless network that contains several autonomous distributed devices and uses sensors to jointly monitor a physical condition or environmental condition. The WSN consists of a large number of independent sensor nodes. Each node aims to send information to the base station through the nearest nodes.

The problem that exists in the sensor node is the limitation of energy and bandwidth which requires consideration of energy at all layers of the network. Routing is a protocol used to get a route from one network to another. These routes are called routes, and route information can be dynamically assigned to other routers or statically assigned to other routers. Routers use this information to build and repair their routing tables. Another type of routing protocol is cluster based. Several clusters are formed and cluster-heads are selected. During routing, the sensor sends the packet to the cluster head first and the cluster head then takes on the responsibility of forwarding the packet to the sink. The cluster contains the cluster head and all direct nodes. Nonhead nodes communicate with each other through the cluster head.

From the problems above, the author designed a special Matlab simulation application for routing protocols using fuzzy Sugeno. With an indicator-based routing protocol, each node compiles its own indicators to assist the routing process. Then by applying the membership function, each node can determine its respective Cluster Head (CH) according to its weighting results. Second, with the Fuzzy Sugeno method, each node will determine the most efficient data transmission route to save power and time so that it arrives at the sink.

The results obtained from this study are 50 nodes covering 200 m² spread over an area of 500 m², the maximum number of cycles is 1000 cycles, and the sink is in the middle of the area. All information packets are received during the 282nd round or within 4 minutes 42 seconds. The maximum average power usage of each node is 13 mAh (assuming each node has a capacity of 100mAh) and decreases with rotation as there are fewer active nodes in the system.

Keywords: *Wireless Sensor Network, Routing Protocol, Indicator-Based, Fuzzy Sugeno.*

KATA PENGANTAR

Puji syukur Penulis panjatkan kepada Allah SWT karena berkat limpahan rahmat-Nya Penulis dapat menyelesaikan Laporan Skripsi yang berjudul "*Routing Protocol Wireless Sensor Network* dengan Metode *Fuzzy Sugeno*". Laporan ini dibuat dengan tujuan untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Konsentrasi Teknik Elektronika Fakultas Teknik Universitas Brawijaya Malang.

Dalam penyusunan Laporan ini tidak sedikit hambatan yang penulis hadapi, namun penulis menyadari bahwa kelancaran dan penyusunan Laporan ini berkat bantuan, dorongan, dan bimbingan dari berbagai pihak baik secara langsung maupun tidak langsung, untuk itu penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Ayahanda Taat Heru Sunarko dan Ibunda Elis Yuliani selaku orang tua atas segala motivasi, nasehat, kasih sayang, perhatian dan kesabarannya dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini.
2. Ashauma Aksanul Muttaqin dan Annisa Dyah Rahayu, adik-adikku yang kubanggakan.
3. Bapak Ir. Wahyu Adi priyono, M.T., dan Ibu Rusmi Ambarwati, S.T., M.T., selaku dosen pembimbing yang telah memberikan bimbingan serta arahan dalam penyelesaian skripsi ini.
4. Tim KRPAI, anggota Workshop Robotika, teman Konsentrasi Teknik Telekomunikasi (Paket c) dan teman Angkatan 2014 Teknik Elektro UB (DIODA) yang meluangkan waktunya untuk berdiskusi dengan penulis.
5. Seluruh pihak terlibat yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari masih terdapat kekurangan dalam penulisan tugas akhir ini, untuk itu penulis berharap adanya masukan dan saran agar tugas akhir ini bisa menjadi lebih baik.

Semoga tugas akhir ini bermanfaat bagi penulis dan juga pembaca.

Malang, 28 Juli 2021

Penulis

DAFTAR ISI

KATA PENGANTAR i

DAFTAR ISI ii

DAFTAR GAMBAR iv

DAFTAR TABEL vi

DAFTAR LAMPIRAN viii

BAB 1 PENDAHULUAN 1

 1.1. Latar Belakang 1

 1.2. Rumusan Masalah 2

 1.3. Batasan Masalah 3

 1.4. Tujuan Penelitian 3

 1.5. Manfaat Penelitian 3

BAB 2 TINJAUAN PUSTAKA 4

 2.1. Protokol Routing pada WSN 4

 2.2. Basis Klaster 7

 2.3. Logika *Fuzzy* 8

 2.3.1 Fungsi Keanggotaan 10

 2.3.2 Metode Sugeno 16

 2.4. Jaringan Sensor Nirkabel 17

BAB 3 METODOLOGI PENELITIAN 20

 3.1. Perancangan Sistem 20

 3.1.1 Perancangan *Fuzzy Logic* 20

 3.1.2 Pembangkitan Bilangan Random 21

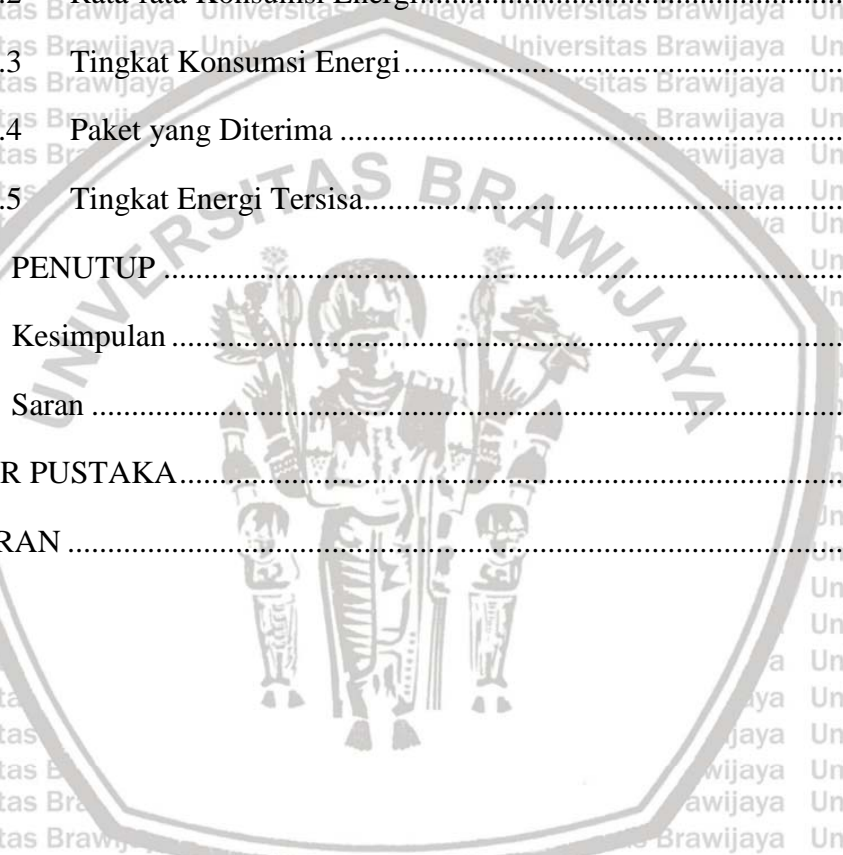
 3.1.3 Fungsi Membership 21

 3.1.4 Perancangan Rule Base System 26

 3.1.5 Perancangan Defuzzifikasi 27

 3.2. Pengujian Sistem 28

3.3.	Analisa Hasil & Pengambilan Kesimpulan	28
BAB 4 HASIL DAN PEMBAHASAN		
4.1.	Umum	30
4.2.	Perhitungan <i>Fuzzy</i>	30
4.3.	Implementasi pada Program Aplikasi	36
4.4.	Analisa Hasil	38
4.4.1	Node hidup	38
4.4.2	Rata-rata Konsumsi Energi	39
4.4.3	Tingkat Konsumsi Energi	40
4.4.4	Paket yang Diterima	40
4.4.5	Tingkat Energi Tersisa	41
BAB 5 PENUTUP		
5.1.	Kesimpulan	43
5.2.	Saran	44
DAFTAR PUSTAKA		
LAMPIRAN		
		47



DAFTAR GAMBAR

Gambar 2.1 Contoh Protokol Routing 4

Gambar 2.2 Algoritma *Routing Protocol* pada WSN..... 6

Gambar 2.3 *Routing Protocol* pada *Cluster Based* 7

Gambar 2.4 Algoritma *Cluster-Based* pada WSN 8

Gambar 2.5 Algoritma Sistem *Fuzzy*..... 9

Gambar 2.6 Representasi Linear Naik..... 11

Gambar 2.7 Representasi Kurva Segitiga..... 11

Gambar 2.8 Parameter *Local Distance* 12

Gambar 2.9 Parameter RPSN 13

Gambar 2.10 Parameter RPS 14

Gambar 2.11 Parameter SD 14

Gambar 2.12 Parameter RCSN..... 15

Gambar 2.13 Blok Diagram Penggunaan Metode *Fuzzy Sugeno*..... 17

Gambar 2.14 Contoh Arsitektur *Wireless Sensor Network*..... 18

Gambar 3.1 Diagram Alir *Fuzzy Sugeno* 20

Gambar 3.2 Fungsi Keanggotaan Segitiga 22

Gambar 3.3 Parameter *Local Distance* 22

Gambar 3.4 Parameter RPSN 23

Gambar 3.5 Parameter RPS 24

Gambar 3.6 Parameter SD 25

Gambar 3.7 Parameter RCSN..... 25

Gambar 3.8 Diagram *Fuzzy Output (Z)*..... 26

Gambar 3.9 Diagram Alir IF – THEN..... 27

Gambar 4.1 Contoh Graf 5 Node 30

Gambar 4.2 Representasi Kurva Segitiga..... 31

Gambar 4.3 Diagram Alir Perhitungan *Fuzzy* 32

Gambar 4.4 Graf Hasil Proses *Fuzzy Sugeno* 36

Gambar 4.5 Penyebaran Node 37

Gambar 4.6 Hasil Routing 38

Gambar 4.7 Grafik *Alive Nodes*..... 39

Gambar 4.8 Grafik *Average Energy Consumption*..... 39

Gambar 4.9 Grafik *Energy Consumption Level* 40

Gambar 4.10 Grafik *Recieved-Packets*..... 41

Gambar 4.11 Grafik *Remaining Energy Level*..... 41





DAFTAR TABEL

Table 4.1 Data Routing dan Hasil *Fuzzy* 30



DAFTAR LAMPIRAN

Lampiran 1 Listing Program Aplikasi Matlab 47

Lampiran 2 Rules 61



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Wireless Sensor Network (WSN) atau Jaringan Sensor Nirkabel merupakan sebuah jaringan *wireless* yang mengandung beberapa perangkat terdistribusi yang otonom dan menggunakan sensor untuk secara bersama memonitor suatu keadaan fisik atau kondisi lingkungan. WSN terdiri atas sejumlah besar *sensor node* yang bebas. Setiap *node* memiliki kemampuan untuk mengirim, menerima dan merasakan. Akan tetapi, mereka terkendala terhadap kemampuan yang mereka miliki seperti kemampuan mengolah, memori, *bandwith*, dan sebagainya (Sari, Muhtada'i, & Santoso, 2008). *Node* sensor memiliki keterbatasan pada energi dan *bandwith* yang mengharuskan konsiderasi terhadap energi pada semua tingkatan layer di jaringan. Layer pada jaringan bertujuan untuk menemukan cara agar konsumsi energi *routing* dan pengiriman data ke *sink* dapat diefisiensi agar dapat meningkatkan jangka hidup jaringan (Warrier & Kumar, 2016).

Routing adalah protokol yang digunakan untuk mendapatkan rute dari satu jaringan ke jaringan lainnya. Rute ini disebut *route*, dan informasi *route* dapat ditetapkan secara dinamis ke *router* lain atau secara statis ditetapkan ke *router* lain. Protokol *routing* adalah komunikasi antar *router*. Protokol *routing* memungkinkan *router* untuk berbagi informasi tentang jaringan dan koneksi antar *router*. *Router* menggunakan informasi ini untuk membangun dan memperbaiki tabel peruteannya. Oleh karena itu, teknik *routing* yang efisien sangat dibutuhkan untuk memperpanjang umur jaringan.

Dalam WSN, parameter yang digunakan dalam protokol *routing* adalah parameter *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN). Memilih *routing* yang tepat merupakan suatu permasalahan pada WSN. Semua *routing* memiliki tujuan yang sama, yaitu agar jaringan tetap dapat berjalan, meningkatkan jangka hidup sensor, mengurangi kompleksitas serta *delay* dari informasi dan meningkatkan performa WSN (Warrier & Kumar, 2016). Di WSN, *routing* sangat menantang karena beberapa karakteristik alami WSN yang membedakannya dari jaringan komunikasi biasa, misalnya, jaringan *ad hoc*. Salah satu masalah yang muncul adalah beberapa *node* dapat merasakan data yang sama di sekitar suatu fenomena, lalu lintas data

memiliki redundansi yang signifikan di dalamnya. Redundansi seperti itu perlu dimanfaatkan oleh protokol *routing* untuk meningkatkan *bandwidth* dan efisiensi energi. Selain itu, *node* sensor di WSN sangat dibatasi dalam hal daya transmisi, pasokan energi, kapasitas pemrosesan, dan *bandwidth*. Oleh karena itu, mereka membutuhkan *routing* yang cermat dan pengelolaan sumber daya untuk mengurangi konsumsi energi total dengan cara mengoptimalkan pencarian *cluster head* (CH).

Penggunaan metode *Fuzzy Sugeno* dalam penelitian ini adalah sebagai pencarian pengoptimalan nilai fungsi keanggotaan. Dalam penelitian ini, penulis menyajikan sebuah aplikasi khusus protokol *routing* menggunakan *Fuzzy Sugeno*, yang mempertimbangkan parameter *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN) untuk menentukan *cluster head* (CH) yang sesuai untuk menyelesaikan permasalahan di atas dengan optimal (efektif dan efisien) dengan cara memberikan pertimbangan bobot nilai yang spesifik pada tiap jalur antar *node* dan *cluster head* (CH).

Alasan penggunaan *Fuzzy Sugeno* dalam penelitian ini adalah karena logika *Fuzzy Sugeno* memiliki karakteristik dan keunggulan yang dapat digunakan multi parameter dalam menangani permasalahan yang bersifat tidak pasti dan kebenaran parsial (Kusumadewi dan Purnomo, 2010).

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka penulis memiliki rumusan masalah pada penelitian ini berupa :

1. Bagaimana merancang sebuah algoritma aplikasi khusus protokol *routing* dengan metode *Fuzzy Sugeno* menggunakan Matlab.
2. Bagaimana penerapan metode *Fuzzy Sugeno* pada *routing protocol* WSN dengan menggunakan parameter *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN) dalam mengoptimalkan pencarian *cluster head* (CH) dengan memberikan pertimbangan bobot nilai yang spesifik pada tiap jalur antar *node* dan *cluster head* (CH).

1.3. Batasan Masalah

Untuk menghindari penyimpangan pada penelitian ini, maka penulis menetapkan batasan masalah sebagai berikut:

1. Sistem yang dibuat dalam penelitian ini menggunakan *software* Matlab.
2. Sistem yang dibuat dalam penelitian ini adalah *Fuzzy Sugeno* dengan menggunakan parameter *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN).
3. Jumlah node sebanyak 50 node dalam area seluas 500 m².
4. Posisi *sink* tepat berada di tengah area.
5. Cakupan setiap node seluas 200 m².
6. Jumlah cycle maksimal sistem adalah 1000 cycle.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk menerapkan *routing protocol* WSN menggunakan metode *Fuzzy Sugeno* dengan parameter *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN), yang memberikan pertimbangan bobot nilai yang spesifik pada tiap jalur antar *node* dan *cluster head* (CH).

1.5. Manfaat Penelitian

Manfaat yang dapat diperoleh penulis dengan penerapan *wireless sensor network* menggunakan *Fuzzy Sugeno* adalah :

1. Dengan digunakannya metode *Fuzzy Sugeno* algoritma ini memberikan kinerja yang lebih baik untuk *graph* atau *node* skala besar dibandingkan algoritma *Rectangular* serta mendapatkan jalur yang paling efektif dan efisien.

BAB 2

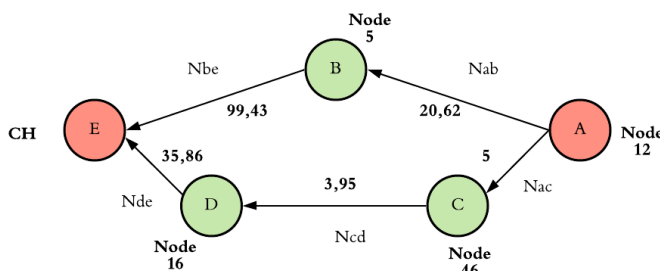
TINJAUAN PUSTAKA

Bab ini berisi landasan teori sebagai parameter rujukan untuk dilaksanakannya penelitian ini. Adapun landasan teori tersebut adalah *Routing Protocol*, *Fuzzy Sugeno* dan *Wireless Sensor Network (WSN)*.

2.1. Protokol Routing pada WSN

Routing adalah aktivitas penting di jaringan, digunakan untuk memilih jalur antara dua *host* untuk berkomunikasi satu sama lain. Dengan memilih jalur terbaik untuk menghubungkan dua *host* untuk berkomunikasi, aktivitas ini diselesaikan oleh protokol *routing* di *router*. Terdapat berbagai protokol *routing* pada setiap jenis jaringan dan jaringan *Mannet*. Dalam daftarnya, protokol *routing* dibagi menjadi dua jenis, yaitu *proactive (table-driven)* dan *reactive (on-demand)* (Azinar & Sari, 2015).

Routing adalah protokol yang digunakan untuk mendapatkan rute dari satu jaringan ke jaringan lainnya. Rute ini disebut *route*, dan informasi *route* dapat ditetapkan secara dinamis ke *router* lain atau secara statis ditetapkan ke *router* lain. Protokol *routing* adalah komunikasi antar *router*. Protokol *routing* memungkinkan *router* untuk berbagi informasi tentang jaringan dan koneksi antar *router*. *Router* menggunakan informasi ini untuk membangun dan memperbaiki tabel peruteannya. Gambar dibawah berisi 5 node yang dimana *cluster head* adalah tujuannya dan titik awalnya adalah *Node 12* atau A. Terdapat dua rute yaitu A-B-E dan A-C-D-E. Rute A-C-D-E memiliki nilai bobot jaringan sebesar 44,81 yang lebih rendah dari pada rute A-B-E yang memiliki nilai bobot jaringan sebesar 120,05, sehingga *router* memilih rute A-C-D-E. Seperti yang ditunjukkan pada Gambar 2.1 di bawah ini.



Gambar 2.1 Contoh Protokol Routing
Sumber: Perancangan

Salah satu masalah terpenting dalam jaringan sensor nirkabel adalah menyediakan layanan antara sensor dan *sink*. Meskipun jaringan sensor mirip dengan jaringan seluler yang mengatur dirinya sendiri, mereka pada dasarnya berbeda dalam banyak aspek. Jaringan sensor memiliki banyak fungsi unik dan perlu dikembangkan lebih lanjut. Secara umum, *routing protocol* di WSN dibagi menjadi dua yaitu:

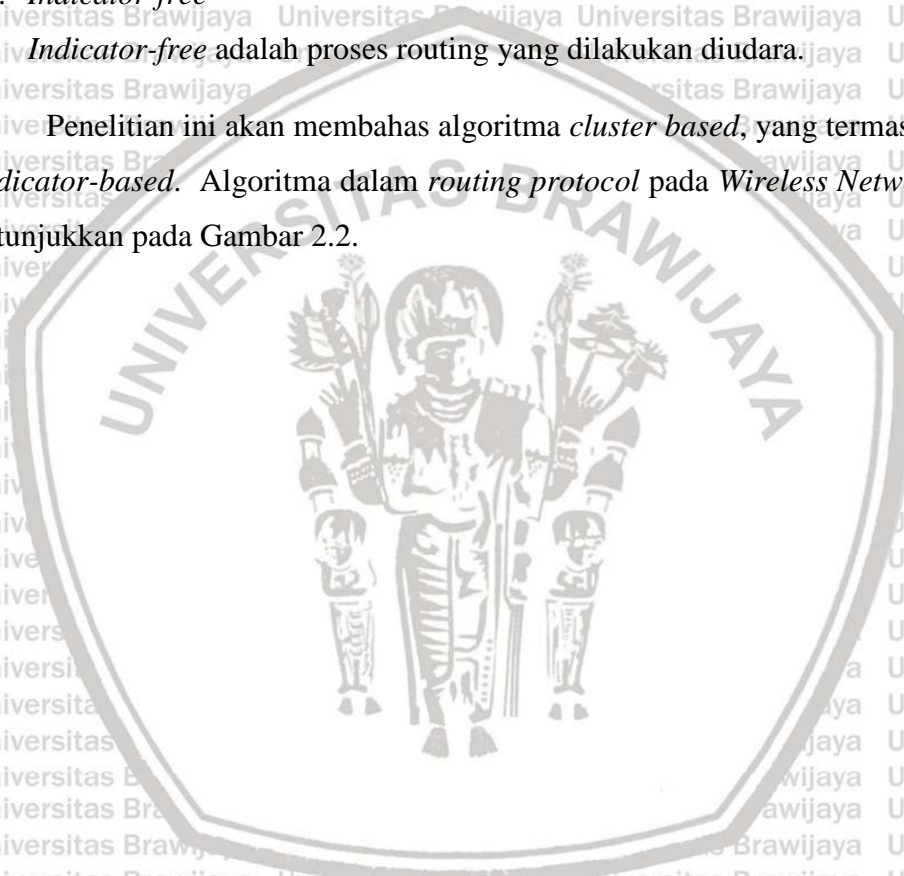
A. *Indicator-based*

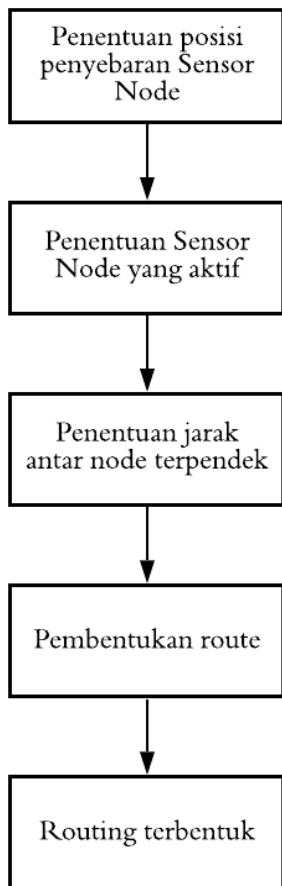
Indicator-based adalah fase inisialisasi, berdasarkan algoritma tersebut setiap *node* menyusun *indicator* untuk membantu proses *routing*.

B. *Indicator-free*

Indicator-free adalah proses *routing* yang dilakukan diudara.

Penelitian ini akan membahas algoritma *cluster based*, yang termasuk dalam kategori *indicator-based*. Algoritma dalam *routing protocol* pada *Wireless Network Sensor* (WSN) ditunjukkan pada Gambar 2.2.





Gambar 2.2 Algoritma *Routing Protocol* pada WSN
 Sumber: Perancangan

Berdasarkan Gambar 2.2 di atas, maka dapat didimpulkan bahwa langkah-langkah *routing protocol* pada WSN adalah sebagai berikut:

1. Penentuan posisi penyebaran *Sensor Node*

Pada bagian ini akan dilakukan algoritma pencarian posisi/lokasi dari *node sensor*, dimana parameter penentuan posisi *node* berdasarkan daya (*Remaining Battery Power of Sensor(RPS)*) yang dimiliki setiap *node*.

2. Penentuan *Sensor Node* yang aktif

Pada bagian ini akan dilakukan pengecekan apakah *sensor node* yang telah ditemukan masih aktif atau tidak. Jika aktif maka akan dilanjutkan pada proses selanjutnya, parameter dalam penentuan ini masih dari daya (RPS) setiap *node*.

3. Penentuan jarak antar node terpendek

Pada bagian ini akan dihitung parameter jarak antar node (*Local Distance(LD)*), kemudian akan ditentukan jarak terpendek dari masing-masing route (*Base Station Distance(SD)*). Jika sudah ditemukan maka akan dilanjutkan pada proses selanjutnya,

4. Pembentukan *route*

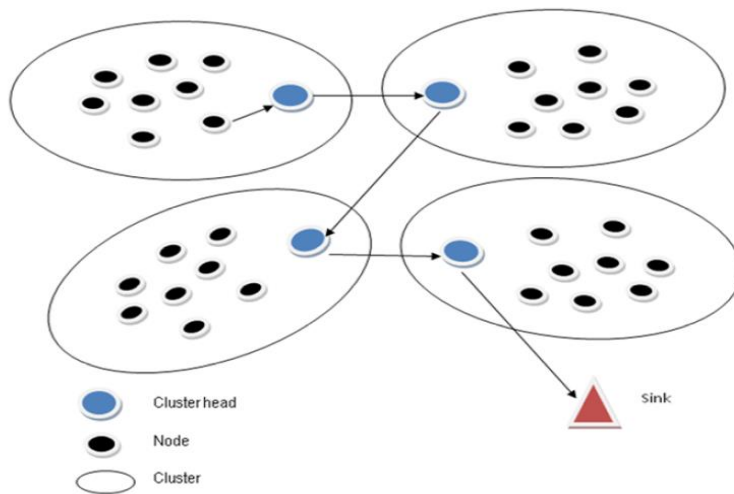
Jika sudah ditemukan jarak terpendek antar node, maka akan dilakukan algoritma pembentukan *route* (rute). Parameter *Rate of recurrent Communication of Sensor Node*(RCSN), *Residual Power of Sensor Node*(RPSN), LD, SD, dan RPS diproses dalam algoritma Fuzzy Sugeno.

5. *Routing* terbentuk

Jika sudah dilakukan hingga poin ke-4, maka *routing* dianggap selesai dan telah berhasil terbentuk.

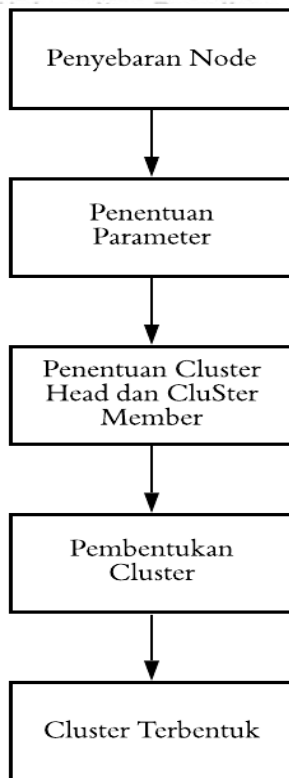
2.2. **Basis Klaster**

Jenis lain dari *routing protocol* adalah basis klaster (*cluster based*). Beberapa *cluster* dibentuk dan *cluster-head* dipilih. Ketika *routing*, sensor mengirim paket ke *head cluster* pertama dan *head-cluster* kemudian mengambil tanggung jawab meneruskan paket ke *sink*. *Cluster* berisi *cluster head* dan semua *node* langsung. *Nonhead node* berkomunikasi dengan yang lain melalui *head cluster* tersebut. Ilustrasi komunikasi pada metode *cluster base* dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Routing Protocol* pada *Cluster Based*
 Sumber: Arrozaqi (2011)

Algoritma dalam *cluster-based* untuk menentukan Cluster Head dan Member Cluster Head/Cluster Member ditunjukkan pada Gambar 2.4, dimana cluster tersebut berdasarkan parameter **daya** yang dimiliki setiap *node*.



Gambar 2.4 Algoritma *Cluster-Based* pada WSN
 Sumber: Perancangan

Berikut tahapan algoritma perancangan *Fuzzy Sugeno* dalam sistem berdasarkan Gambar 2.4:

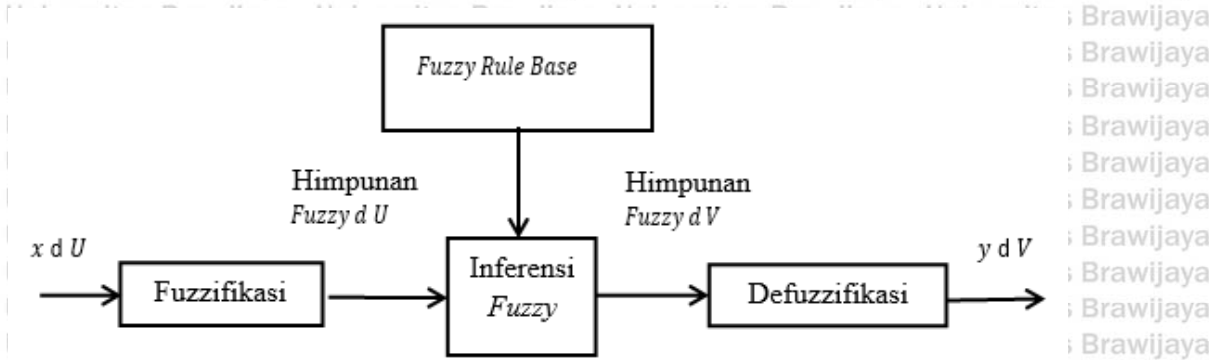
- a. Penentuan parameter dengan setiap node membangkitkan nilai – nilai random sebagai parameter penentuan bobot setiap node, yang mana bobot tersebut adalah daya yang dimiliki oleh setiap *node* (parameter RPS).
- b. Penentuan Cluster Head dan Cluster Member dimulai dengan menentukan CH dengan memilih node yang memiliki nilai parameter RCSN yang telah ditetapkan, dan node lainnya yang belum terpilih ditetapkan menjadi Cluster Member.
- c. Pembentukan Cluster, begitu para Cluster Head terpilih, para CH akan menyebarkan pesan *advertisement*. Berdasarkan dari kekuatan sinyal yang diterima (parameter LD dan RPSN), setiap *node* yang bukan CH akan memilih CH-nya

2.3. Logika *Fuzzy*

Logika fuzzy adalah cabang dari sistem kecerdasan buatan (Artificial Inteligent) yang meniru kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian

dijalankan oleh mesin. Algoritma ini digunakan dalam berbagai aplikasi pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Logika fuzzy menginterpretasikan statemen yang samar menjadi sebuah pengertian yang logis.

Algoritma atau langkah-langkah dalam sistem fuzzy ditunjukkan pada Gambar 2.5 di bawah ini.



Gambar 2.5 Algoritma Sistem Fuzzy

Sumber: Wang (1997)

Dalam Gambar 2.5 tersebut, dapat disimpulkan bahwa komponen - komponen fuzzy dan penjelasan tahapan sistem fuzzy adalah sebagai berikut:

- Himpunan Fuzzy

Himpunan fuzzy merupakan suatu pengembangan lebih lanjut tentang konsep himpunan dalam matematika. Himpunan Fuzzy adalah rentang nilai-nilai. Masing-masing nilai mempunyai derajat keanggotaan (*membership*) antara 0 sampai dengan 1. Ungkapan logika Boolean menggambarkan nilai-nilai “benar” atau “salah”. Logika fuzzy menggunakan ungkapan misalnya: “sangat lambat”, “agak sedang”, “sangat cepat” dan lain-lain untuk mengungkapkan derajat intensitasnya (Kusumadewi and Purnomo, 2010). Input dalam penelitian ini adalah RPSN, Local Distance, SD, RPS dan RCSN yang akan menghasilkan output routing protocol WSN.

- Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dijabarkan sebelum diolah oleh pengendali fuzzy harus diubah terlebih dahulu ke dalam variabel fuzzy. Melalui fungsi keanggotaan yang telah disusun maka nilai-nilai masukan tersebut menjadi informasi

fuzzy yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut fuzzifikasi (Kusumadewi and Purnomo 2010).

- **Inferencing (Rule Base)**

Pada umumnya, aturan-aturan *fuzzy* dinyatakan dalam bentuk “*IF...THEN*” yang merupakan inti dari relasi *fuzzy*. Relasi *fuzzy* dinyatakan dengan *R* juga disebut implikasi *fuzzy* (Kusumadewi and Purnomo 2010). Untuk mendapatkan aturan “*IF.....THEN*” ada dua cara utama :

1. Menanyakan ke operator manusia yang dengan cara manual telah mampu mengendalikan sistem tersebut, dikenal dengan “*human expert*”.
2. Dengan menggunakan algoritma pelatihan berdasarkan data-data masukan dan keluaran.

Berikut contoh rule “*IF...THEN*” yang ada:

“IF RPSN high LD medium SD medium RPS high RCSN large THEN rhigh”

“IF RPSN high LD far SD high RPS high RCSN large THEN vhigh”

- **Defuzzifikasi**

Keputusan yang dihasilkan dari proses penalaran (*inferencing “IF..Then”*) masih dalam bentuk *fuzzy*, yaitu berupa derajat keanggotaan keluaran. Hasil ini harus diubah kembali menjadi variabel numerik *nonfuzzy* melalui proses defuzzifikasi (Kusumadewi and Purnomo 2010).

2.3.1 Fungsi Keanggotaan

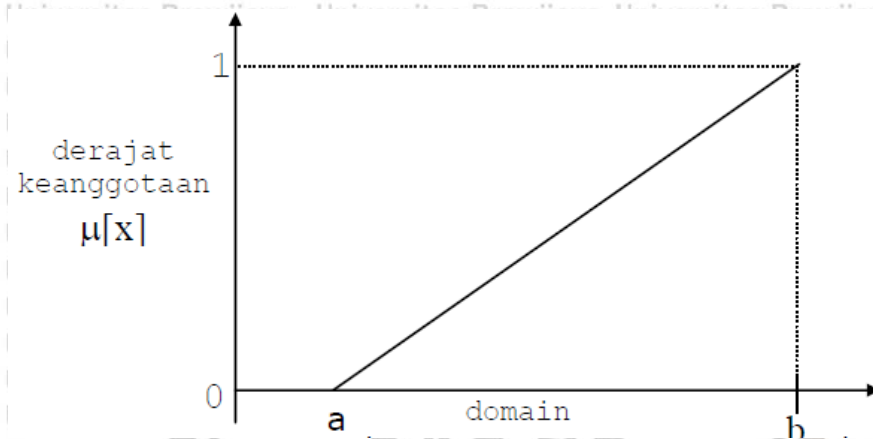
Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk

mendekati suatu konsep yang kurang jelas. Ada dua keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi (Kusumadewi dan Purnomo, 2010).



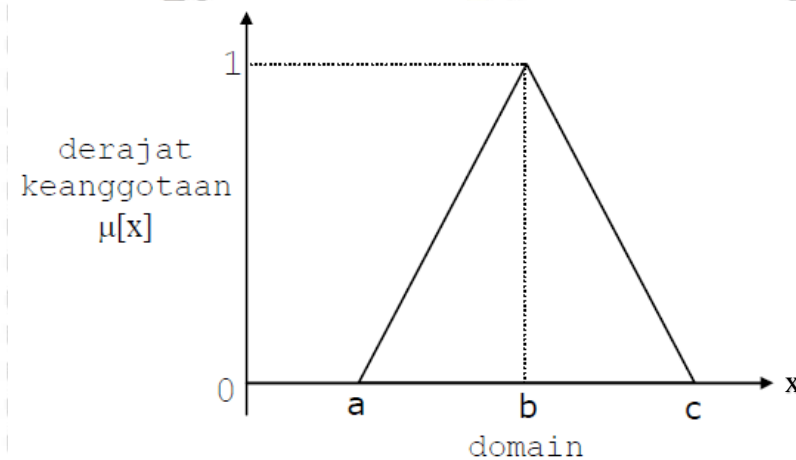
Gambar 2.6 Representasi Linear Naik
Sumber: Khairina (2019)

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a) / (b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2-1)$$

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti terlihat pada Gambar 2.7 (Kusumadewi and Purnomo 2010).



Gambar 2.7 Representasi Kurva Segitiga
Sumber: Khairina (2019)

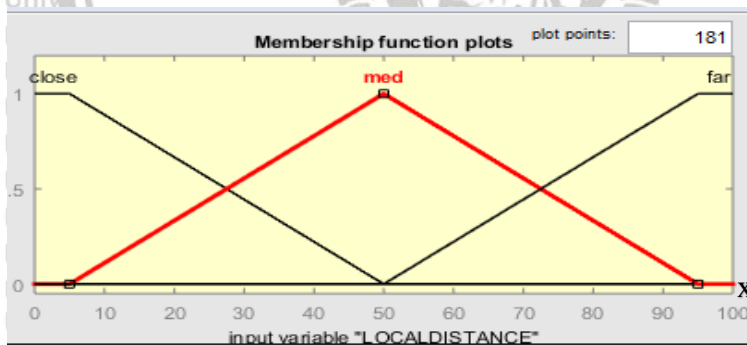
Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (b - x)/(c - b); & b \leq x \leq c \end{cases} \quad (2-2)$$

Pada penelitian ini, fungsi keanggotaan yang digunakan berdasarkan *input* dari *Local Distance* (LD), *Residual Power of Sensor Nodes* (RPSN), *Remaining Battery Power of Sensor* (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN) dan menghasilkan *output* berupa titik kurva yang terdiri dari medium, mlow, flow, low, vlow, vhigh, high, rhhigh, dan mhigh. Kemudian akan dimasukkan sebagai parameter data *fuzzy* dengan menggunakan 50 node dan 1000 *looping* sehingga nilai yang dihasilkan optimal untuk menghasilkan efisiensi energi dalam *Wireless Sensor Network*. Node yang dipakai sebanyak 50 akan ditentukan apakah akan menjadi *cluster head* ataupun *cluster member*. Kemudian akan dibentuk *cluster* melalui 1000 *looping* yang dilakukan. Selain itu dead node akan ditentukan juga ketika *looping* berjalan.

1. Fungsi Keanggotaan *Local Distance*

Himpunan fuzzy keanggotaan *Local Distance* disajikan dalam Gambar 2.8



Gambar 2.8 Parameter *Local Distance*

Sumber: Perancangan

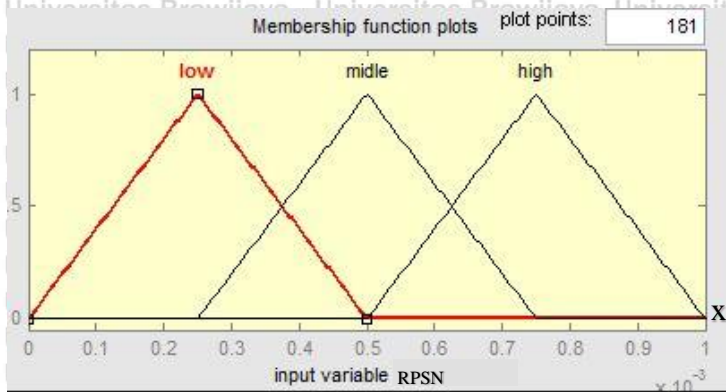
$$\mu \text{ LD (close)} = \begin{cases} 0; & C \leq 5 \\ \frac{C-5}{50-5}; & 5 \leq C \leq 50 \\ 1; & C \geq 50 \end{cases} \quad (2-3)$$

$$\mu \text{ LD (med)} = \begin{cases} 0; & M \leq 50 \\ \frac{M-50}{95-50}; & 50 \leq M \leq 95 \\ 1; & M \geq 95 \end{cases} \quad (2-4)$$

$$\mu_{LD}(\text{far}) = \begin{cases} 0; & F \leq 95 \\ \left(\frac{F-95}{100-95}\right); & 95 \leq F \leq 100 \\ 1; & F \geq 100 \end{cases} \dots\dots\dots (2-5)$$

2. Fungsi Keanggotaan RPSN

Berikut parameter RPSN pada Gambar 2.9



Gambar 2.9 Parameter RPSN
Sumber: Perancangan

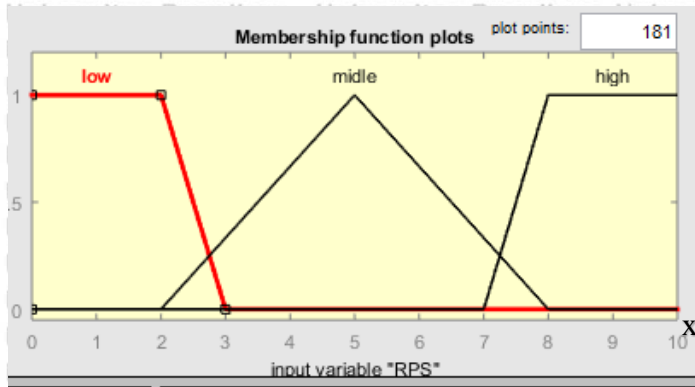
$$\mu_{Low} = \begin{cases} 0; & L \leq 0 \text{ atau } L \geq 0,5 \\ \left(\frac{L}{0,25}\right); & 0 \leq L \leq 0,25 \\ \left(\frac{0,25-L}{0,5-0,25}\right); & 0,25 \leq L \leq 0,5 \end{cases} \dots\dots\dots (2-6)$$

$$\mu_{Middle} = \begin{cases} 0; & M \leq 0,25 \text{ atau } M \geq 0,75 \\ \left(\frac{M-0,25}{0,5-0,25}\right); & 0,25 \leq M \leq 0,5 \\ \left(\frac{0,75-M}{0,75-0,5}\right); & 0,5 \leq M \leq 0,75 \end{cases} \dots\dots\dots (2-7)$$

$$\mu_{High} = \begin{cases} 0; & H \leq 0,5 \text{ atau } H \geq 1 \\ \left(\frac{H-0,5}{0,75-0,5}\right); & 0,5 \leq H \leq 0,75 \\ \left(\frac{1-M}{1-0,75}\right); & 0,75 \leq H \leq 1 \end{cases} \dots\dots\dots (2-8)$$

3. Fungsi Keanggotaan RPS

Himpunan fuzzy keanggotaan RPS disajikan dalam Gambar 2.10



Gambar 2.10 Parameter RPS

Sumber: Perancangan

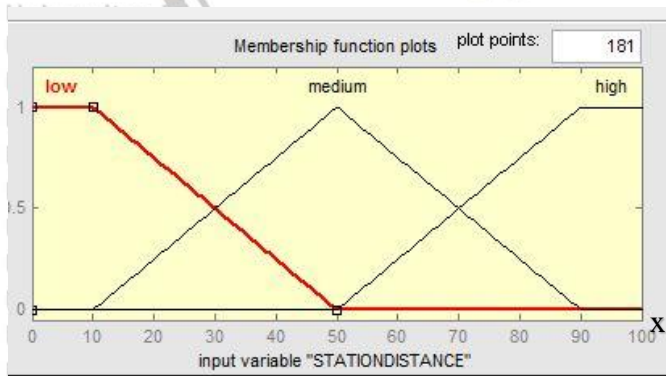
$$\mu_{RPS}(\text{Low}) = \begin{cases} 0; & L \leq 2 \\ \left(\frac{L-2}{3-2}\right); & 2 \leq L \leq 3 \\ 1; & L \geq 3 \end{cases} \dots\dots\dots (2-9)$$

$$\mu_{RPS}(\text{Middle}) = \begin{cases} 0; & M \leq 5 \\ \left(\frac{S-5}{8-5}\right); & 5 \leq M \leq 8 \\ 1; & S \geq 8 \end{cases} \dots\dots\dots (2-10)$$

$$\mu_{RPS}(\text{High}) = \begin{cases} 0; & H \leq 8 \\ \left(\frac{H-8}{10-8}\right); & 8 \leq H \leq 10 \\ 1; & C \geq 10 \end{cases} \dots\dots\dots (2-11)$$

4. Fungsi Keanggotaan SD

Himpunan fuzzy keanggotaan SD ditunjukkan dalam Gambar 2.11 *Station distance* diatur menjadi 3 himpunan fuzzy yaitu *high, middle* dan *low*.



Gambar 2.11 Parameter SD

Sumber: Perancangan

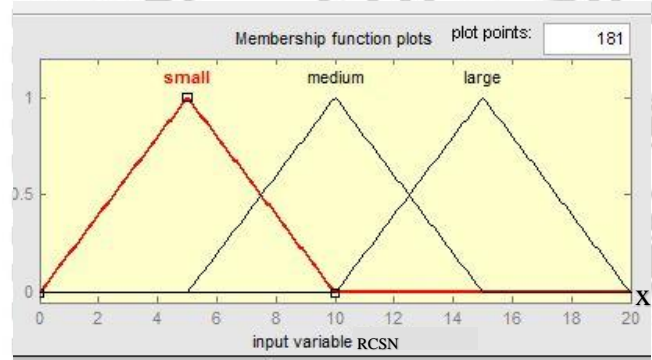
$$\mu_{SD}(\text{Low}) = \begin{cases} 0; & L \leq 10 \\ \left(\frac{L-10}{50-10}\right); & 10 \leq L \leq 50 \\ 1; & L \geq 50 \end{cases} \dots\dots\dots (2-12)$$

$$\mu_{SD}(\text{Medium}) = \begin{cases} 0; & M \leq 50 \\ \left(\frac{M-50}{90-50}\right); & 50 \leq M \leq 90 \\ 1; & M \geq 90 \end{cases} \dots\dots\dots (2-13)$$

$$\mu_{SD}(\text{High}) = \begin{cases} 0; & H \leq 90 \\ \left(\frac{H-90}{100-90}\right); & 90 \leq H \leq 100 \\ 1; & H \geq 100 \end{cases} \dots\dots\dots (2-14)$$

5. Fungsi Keanggotaan RCSN

Himpunan fuzzy keanggotaan RCSN disajikan dalam Gambar 2.12



Gambar 2.12 Parameter RCSN
Sumber: Perancangan

$$\mu_{RCSN}(\text{Small}) = \begin{cases} 0; & S \leq 0 \text{ atau } S \geq 10 \\ \left(\frac{S}{5}\right); & 0 \leq S \leq 5 \\ \left(\frac{5-S}{10-5}\right); & 5 \leq S \leq 10 \end{cases} \dots\dots\dots (2-15)$$

$$\mu_{RCSN}(\text{Medium}) = \begin{cases} 0; & M \leq 5 \text{ atau } M \geq 15 \\ \left(\frac{S-5}{10-5}\right); & 5 \leq M \leq 10 \\ \left(\frac{10-S}{15-10}\right); & 10 \leq M \leq 15 \end{cases} \dots\dots\dots (2-16)$$

$$\mu_{RCSN}(\text{Large}) = \begin{cases} 0; & L \leq 10 \text{ atau } L \geq 20 \\ \left(\frac{L-10}{15-10}\right); & 10 \leq L \leq 15 \\ \left(\frac{15-L}{20-15}\right); & 15 \leq L \leq 20 \end{cases} \dots\dots\dots (2-17)$$

2.3.2 Metode Sugeno

Fuzzy metode Sugeno merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk *IF–THEN*, dimana output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Jika X adalah sebuah koleksi obyek-obyek yang dinotasikan dengan x , maka himpunan Fuzzy A dalam X adalah sebuah himpunan pasangan berurutan $\dot{A} = \{x, \mu\dot{A} \mid (x) \in X\}$. Notasi $\mu\dot{A}(x)$ disebut fungsi atau derajat dalam A yang memetakan X ke ruang M yang terletak pada rentang $[0, 1]$, bila M hanya memuat dua titik 0 dan 1, maka A adalah bukan Fuzzy dan $\mu\dot{A}(x)$ serupa dengan fungsi himpunan non Fuzzy (Zimmerman, 1991). Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Model Sugeno menggunakan fungsi keanggotaan *Singleton* yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada nilai *crisp* yang lain. Untuk Orde 0 dengan rumus (Kusumadewi dan Purnomo, 2010) :

$$\begin{aligned}
 & \text{IF } (x_1 \text{ is } a_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_N \text{ is } A_N) \dots\dots\dots (2-18) \\
 & \text{THEN } z = k,
 \end{aligned}$$

Dengan :

- A_i adalah himpunan *fuzzy* ke i sebagai *antaseden* (alasan),
- \circ adalah operator *fuzzy* (*AND* atau *OR*)

k merupakan konstanta tegas sebagai *konsekuen* (kesimpulan).

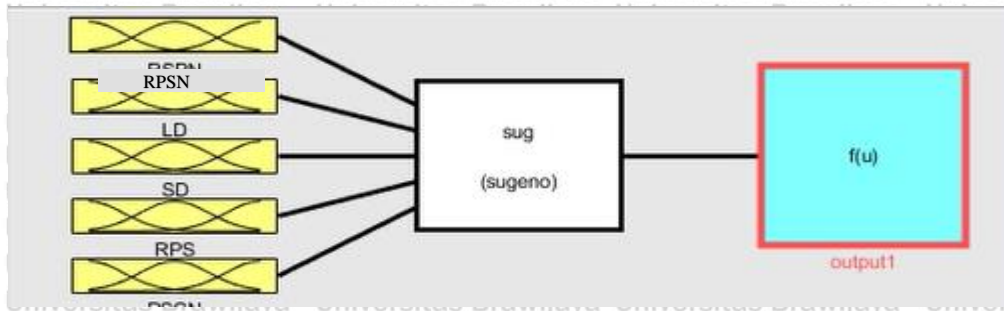
Sedangkan rumus Orde 1 adalah:

$$\begin{aligned}
 & \text{IF } (X_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (X_N \text{ is } A_N) \dots\dots\dots (2-19) \\
 & \text{THEN } z = p_1 * x_1 + \dots + p_N * x_N + q,
 \end{aligned}$$

Dengan:

- A_i adalah himpunan *fuzzy* ke i sebagai *antaseden*,
- \circ adalah operator *fuzzy* (*AND* atau *OR*),
- p_i adalah konstanta ke i
- q juga merupakan *konstanta* dalam *konsekuen*.

Penggunaan metode *Fuzzy Sugeno* dalam skripsi ini adalah sebagai pencarian pengoptimalan nilai fungsi keanggotaan. Data *input* dan *output* dalam penggunaan metode *Fuzzy Sugeno* ditunjukkan oleh Blok Diagram pada Gambar 2.13.



Gambar 2.13 Blok Diagram Penggunaan Metode *Fuzzy Sugeno*

Sumber: Perancangan

Pada Gambar 2.13 menunjukkan bahwa *input* yang digunakan adalah *Local Distance* (LD) yang merupakan fitur yang digunakan oleh router untuk menentukan jalur tiap *nodes* dalam *cluster*, *Residual Power of Sensor Nodes* (RPSN) yaitu pengaturan diri sendiri yang dimiliki setiap kepala *cluster* (CH) karena mengkonsumsi banyak energi, *Remaining Battery Power of Sensor* (RPS) adalah sisa baterai yang dimiliki setiap sensor, *Station Distance* (SD) yaitu jarak setiap *nodes* terhadap *cluster head* (CH), dan *Rate of recurrent Communication of Sensor Node* (RCSN) yang berarti parameter untuk mengukur intensitas komunikasi antara tiap *nodes* terhadap *cluster head* (CH) yang mana *input* yang tertera digunakan sebagai parameter daya yang dimiliki setiap *nodes* yang selanjutnya akan diproses menggunakan *Fuzzy Sugeno* dan akan dihasilkan *output* berupa nilai fungsi keanggotaan pada setiap masing-masing *input*-an.

Proses defuzzyfikasi *Sugeno* menggunakan *weight average*. Dengan *weighted average*, setiap nilai hasil inferensi terhadap aturan *fuzzy* (*W*) dikalikan dengan nilai diagram *fuzzy output* (*Z*), kemudian dijumlahkan. Hasil penjumlahan tersebut kemudian dibagi dengan penjumlahan final *output*. *Weight average* menghasilkan sebuah nilai bilangan real yang dijadikan bobot pada setiap node Cluster Head Selection WSN

$$\text{Final Output} = \frac{\sum_{i=1}^N w_i z_i}{w_i} \dots \dots \dots (2-20)$$

Dengan:

W = nilai hasil inferensi *fuzzy*

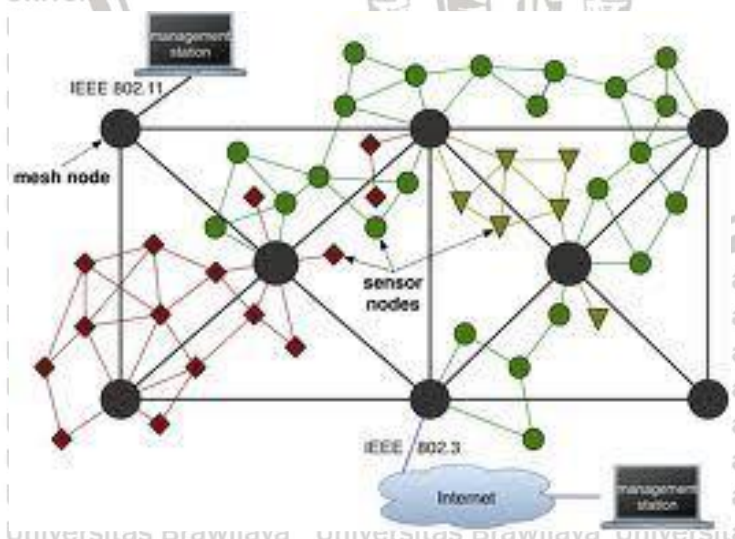
Z = nilai diagram *fuzzy output*

2.4. Jaringan Sensor Nirkabel

Jaringan Sensor Nirkabel atau *Wireless Sensor Network* (WSN) merupakan sebuah jaringan *wireless* yang mengandung beberapa perangkat terdistribusi yang otonom dan

menggunakan sensor untuk secara bersama memonitor suatu keadaan fisik atau kondisi lingkungan. WSN terdiri atas sejumlah besar *sensor node* yang bebas. Setiap *node* memiliki kemampuan untuk mengirim, menerima dan merasakan. Akan tetapi, mereka terkendala terhadap kemampuan yang mereka miliki seperti kemampuan mengolah, memori, *bandwith*, dan sebagainya (Sari, Muftada'i, & Santoso, 2008). *Node* sensor memiliki keterbatasan pada energi dan *bandwith* yang mengharuskan pertimbangan terhadap energi pada semua tingkatan layer di jaringan. Layer pada jaringan bertujuan untuk menemukan cara agar konsumsi energi *routing* dan pengiriman data ke *sink* dapat diefisiensi agar dapat meningkatkan jangka hidup jaringan (Warrier & Kumar, 2016).

Konsep jaringan sensor nirkabel (*Wireless Sensor Networks*), secara sederhana, merupakan perpaduan fungsi penginderaan (*sensing*), fungsi komputasi, dan fungsi komunikasi. Memadukan sensor, actuator, radio, dan CPU ke dalam suatu jaringan sensor dan aktuator nirkabel membutuhkan pemahaman rinci pada kemampuan dan keterbatasan setiap komponen perangkat keras di dalamnya di samping pemahaman rinci tentang teknologi jaringan dan teori sistem tersebar. Setiap titik harus dirancang agar memberikan sekumpulan fungsi dasar atau primitif untuk menciptakan jejaring yang memiliki fungsi-fungsi lebih besar dengan tetap memenuhi kebutuhan akan penghematan daya dan ruang (ukuran) (Jati, A. S., Puspitorini, O., & Siswandari, 2016). Contoh arsitekturnya seperti gambar 2.14.



Gambar 2.14 Contoh Arsitektur *Wireless Sensor Network*
 Sumber: Jati (2016)

Untuk WSN dalam penelitian ini, parameter yang digunakan dalam protokol *routing* adalah parameter *Local Distance (LD)*, *Residual Power of Sensor Nodes (RPSN)*, *Remaining*

Battery Power of Sensor (RPS), *Station Distance* (SD), dan *Rate of recurrent Communication of Sensor Node* (RCSN). Semua parameter *routing* memiliki tujuan yang sama, yaitu agar jaringan tetap dapat berjalan, meningkatkan jangka hidup sensor, mengurangi kompleksitas serta *delay* dari informasi dan meningkatkan performa WSN. Oleh karena itu, mereka membutuhkan *routing* yang cermat dan pengelolaan sumber daya untuk mengurangi konsumsi energi total dengan cara mengoptimalkan pencarian *cluster head* (CH).



BAB 3

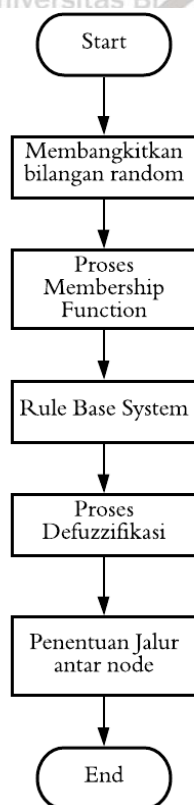
METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan tahapan penelitian dan perancangan sistem yang digunakan adalah perancangan sistem, pengujian sistem, analisa hasil dan pengambilan kesimpulan. Dengan adanya metodologi penelitian ini diharapkan dapat memberikan petunjuk dalam merumuskan masalah penelitian.

3.1. Perancangan Sistem

3.1.1 Perancangan Fuzzy Logic

Perancangan Model Algoritma *Fuzzy Sugeno*:



Gambar 3.1 Diagram Alir *Fuzzy Sugeno*

Sumber: Perancangan

Berikut tahapan algoritma perancangan *Fuzzy Sugeno* dalam sistem berdasarkan

Gambar 3.1:

- a. Membangkitkan bilangan *random* (secara acak). Jadi setiap nilai *random* menjadi masukan setiap parameter yang digunakan. Jadi berapapun nilainya tergantung dari program yang me-*random* nilai-nilai yang dijadikan parameter.
- b. Proses *membership function*.
- c. Penentuan *rule base system* dengan menentukan terlebih dahulu nilai diagram *fuzzy output* (Z).
- d. Proses defuzzifikasi. Nilai *fuzzy* ini akan mewakili setiap *node* yang akan digunakan.
- e. Setiap *node* yang akan digunakan mempunyai nilai *fuzzy* yang akan menjadi nilai pertimbangan untuk menentukan jalur optimal yang akan dilewati.

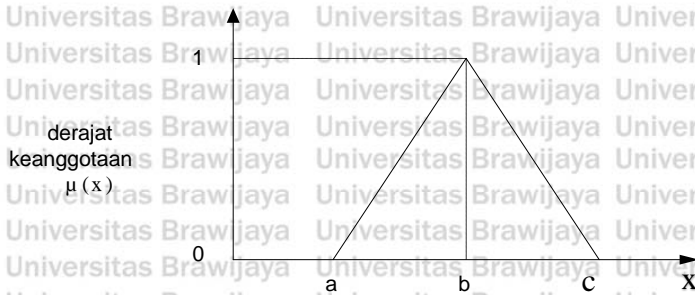
3.1.2 Pembangkitan Bilangan Random

Pada tahap pembangkitan bilangan random, setiap nilai *random* menjadi masukan setiap parameter yang digunakan. Pada penelitian ini parameter yang digunakan adalah :

1. *Rate of recurrent Communication of Sensor Node* (RCSN) sebagai jumlah kali sebuah node berkomunikasi dengan CH-nya. Cakupan nilai random antara 0 -20.
2. *Residual Power of Sensor Nodes* (RPSN) digunakan dalam skema *self organizing* karena CH mengkonsumsi banyak daya dari *node* sensor karena CH memiliki *responsibility* agregasi data, pengolahan data, *routing* data. Cakupan nilai random antara 0 – 1, dan nilai 1 sama dengan 10 mAh dengan asumsi 1 node menggunakan 100 mAh.
3. *Local distance* (LD) merupakan jumlah jarak antara node A dan node yang berada dalam jarak r. Cakupan nilai random antara 0 – 100, dan nilai 1 sama dengan 1 meter.
4. *Base station distance* (SD) digunakan untuk jarak dari CH ke *base station*. Cakupan nilai random antara 0 – 100, dan nilai 1 sama dengan 1 meter.
5. *Remaining Battery Power of Sensor* (RPS) digunakan untuk mengetahui sisa baterai yang berada di sensor. Cakupan nilai antara 0 -10, dan nilai 1 sama dengan 10 mAh dengan asumsi 1 node menggunakan 100 mAh.

3.1.3 Fungsi Membership

Representasi kurva segitiga adalah sebagai berikut :



Gambar 3.2 Fungsi Keanggotaan Segitiga
 Sumber: Khairina (2019)

Dengan fungsi keanggotaan sebagai berikut :

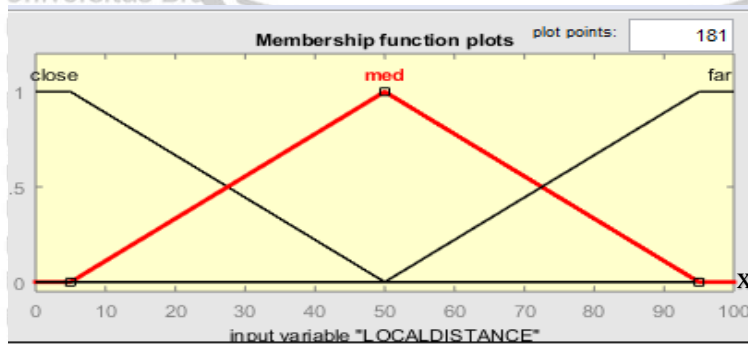
$$\mu(x) = \begin{cases} 0 & x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a) & a \leq x \leq b \\ (b-x)/(c-b) & b \leq x \leq c \\ 1 & x \geq c \end{cases} \quad (3-1)$$

- a = nilai dominan terkecil yang mempunyai derajat keanggotaan nol;
- b = nilai dominan yang mempunyai derajat keanggotaan satu;
- c = nilai dominan terbesar yang mempunyai derajat keanggotaan nol;
- x = nilai input atau output yang akan diubah ke dalam bilangan fuzzy.

Fungsi untuk memetakan kembali nilai Fuzzy menjadi nilai *crisp* yang menjadi output/input solusi permasalahan.

1. Fungsi Keanggotaan *Local Distance*

Himpunan fuzzy keanggotaan *Local Distance* disajikan dalam Gambar 3.3



Gambar 3.3 Parameter *Local Distance*
 Sumber: Perancangan

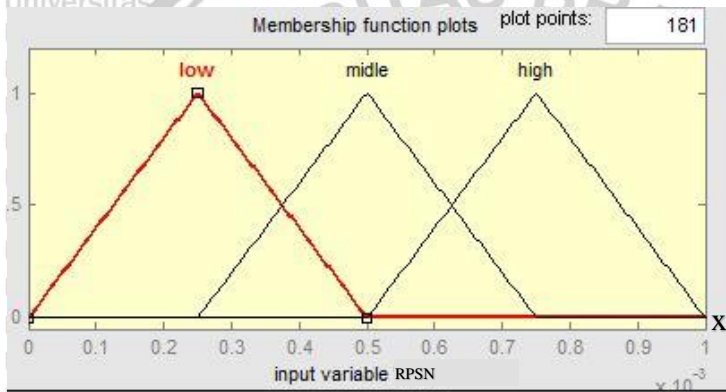
$$\mu_{LD}(\text{close}) = \begin{cases} 0; & c \leq 5 \\ \left(\frac{c-5}{50-5}\right); & 5 \leq c \leq 50 \\ 1; & c \geq 50 \end{cases} \dots\dots\dots (3-2)$$

$$\mu_{LD}(\text{med}) = \begin{cases} 0; & M \leq 50 \\ \left(\frac{M-50}{95-50}\right); & 50 \leq M \leq 95 \\ 1; & M \geq 95 \end{cases} \dots\dots\dots (3-3)$$

$$\mu_{LD}(\text{far}) = \begin{cases} 0; & F \leq 95 \\ \left(\frac{F-95}{100-95}\right); & 95 \leq F \leq 100 \\ 1; & F \geq 100 \end{cases} \dots\dots\dots (3-4)$$

2. Fungsi Keanggotaan RPSN

Berikut parameter RPSN pada Gambar 3.4



Gambar 3.4 Parameter RPSN

Sumber: Perancangan

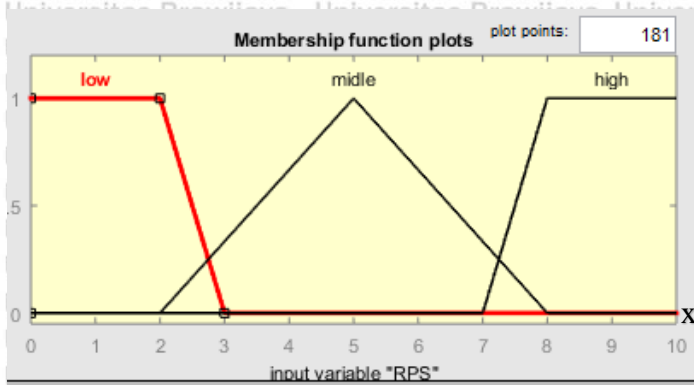
$$\mu_{Low} = \begin{cases} 0; & L \leq 0 \text{ atau } L \geq 0,5 \\ \left(\frac{L}{0,25}\right); & 0 \leq L \leq 0,25 \\ \left(\frac{0,25-L}{0,5-0,25}\right); & 0,25 \leq L \leq 0,5 \end{cases} \dots\dots\dots (3-5)$$

$$\mu_{Midle} = \begin{cases} 0; & M \leq 0,25 \text{ atau } M \geq 0,75 \\ \left(\frac{M-0,25}{0,5-0,25}\right); & 0,25 \leq M \leq 0,5 \\ \left(\frac{0,75-M}{0,75-0,5}\right); & 0,5 \leq M \leq 0,75 \end{cases} \dots\dots\dots (3-6)$$

$$\mu_{High} = \begin{cases} 0; & H \leq 0,5 \text{ atau } H \geq 1 \\ \left(\frac{H-0,5}{0,75-0,5}\right); & 0,5 \leq H \leq 0,75 \\ \left(\frac{1-M}{1-0,75}\right); & 0,75 \leq H \leq 1 \end{cases} \dots\dots\dots (3-7)$$

3. Fungsi Keanggotaan RPS

Himpunan fuzzy keanggotaan RPS disajikan dalam Gambar 3.5



Gambar 3.5 Parameter RPS

Sumber: Perancangan

$$\mu_{\text{RPS}} (\text{Low}) = \begin{cases} 0 & ; L \leq 2 \\ \left(\frac{L-2}{3-2}\right) & ; 2 \leq L \leq 3 \\ 1 & ; L \geq 3 \end{cases} \dots\dots\dots (3-8)$$

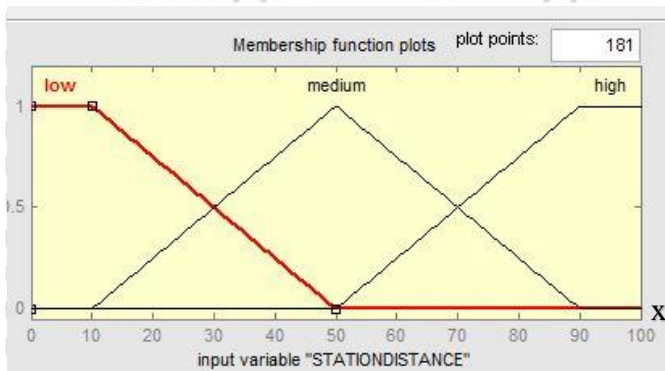
$$\mu_{\text{RPS}} (\text{Middle}) = \begin{cases} 0 & ; M \leq 5 \\ \left(\frac{M-5}{8-5}\right) & ; 5 \leq M \leq 8 \\ 1 & ; M \geq 8 \end{cases} \dots\dots\dots (3-9)$$

$$\mu_{\text{RPS}} (\text{High}) = \begin{cases} 0 & ; H \leq 8 \\ \left(\frac{H-8}{10-8}\right) & ; 8 \leq H \leq 10 \\ 1 & ; H \geq 10 \end{cases} \dots\dots\dots (3-10)$$

4. Fungsi Keanggotaan SD

Himpunan fuzzy keanggotaan SD ditunjukkan dalam Gambar 3.6 *Station distance*

diatur menjadi 3 himpunan fuzzy yaitu *high*, *middle* dan *low*.



Gambar 3.6 Parameter SD
Sumber: Perancangan

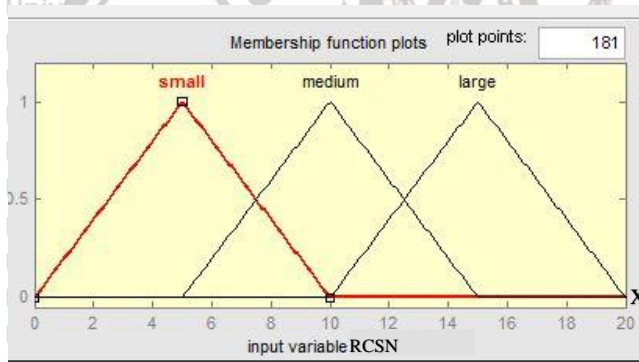
$$\mu_{SD}(\text{Low}) = \begin{cases} 0; & L \leq 10 \\ \left(\frac{L-10}{50-10}\right); & 10 \leq L \leq 50 \\ 1; & L \geq 50 \end{cases} \dots\dots\dots (3-11)$$

$$\mu_{SD}(\text{Medium}) = \begin{cases} 0; & M \leq 50 \\ \left(\frac{M-50}{90-50}\right); & 50 \leq M \leq 90 \\ 1; & M \geq 90 \end{cases} \dots\dots\dots (3-12)$$

$$\mu_{SD}(\text{High}) = \begin{cases} 0; & H \leq 90 \\ \left(\frac{H-90}{100-90}\right); & 90 \leq H \leq 100 \\ 1; & H \geq 100 \end{cases} \dots\dots\dots (3-13)$$

5. Fungsi Keanggotaan RCSN

Himpunan fuzzy keanggotaan RCSN disajikan dalam Gambar 3.7



Gambar 3.7 Parameter RCSN
Sumber: Perancangan

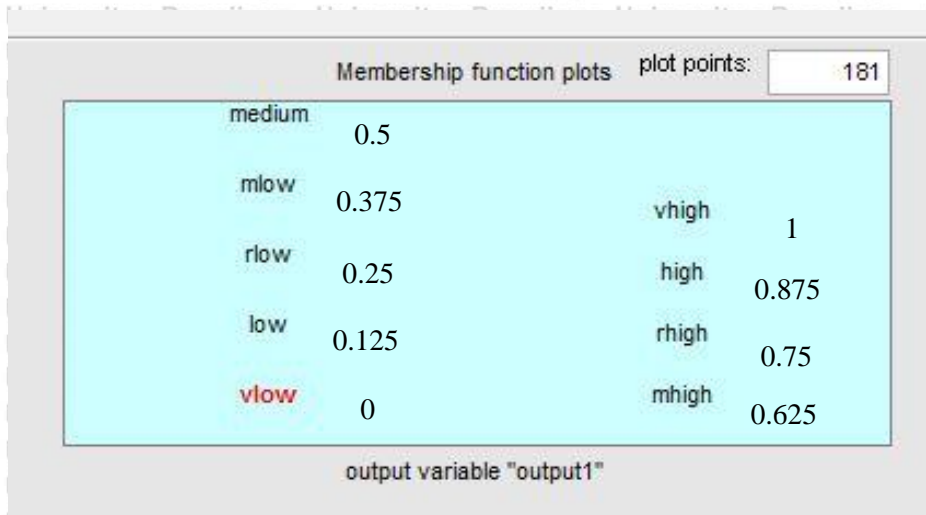
$$\mu_{RCSN}(\text{Small}) = \begin{cases} 0; & S \leq 0 \text{ atau } S \geq 10 \\ \left(\frac{S}{5}\right); & 0 \leq S \leq 5 \\ \left(\frac{5-S}{10-5}\right); & 5 \leq S \leq 10 \end{cases} \dots\dots\dots (3-14)$$

$$\mu_{RCSN}(\text{Medium}) = \begin{cases} 0; & M \leq 5 \text{ atau } M \geq 15 \\ \left(\frac{S-5}{10-5}\right); & 5 \leq M \leq 10 \\ \left(\frac{10-S}{15-10}\right); & 10 \leq M \leq 15 \end{cases} \dots\dots\dots (3-15)$$

$$\mu_{RCSN}(\text{Large}) = \begin{cases} 0; & L \leq 10 \text{ atau } L \geq 20 \\ \left(\frac{L-10}{15-10}\right); & 10 \leq L \leq 15 \\ \left(\frac{15-L}{20-15}\right); & 15 \leq L \leq 20 \end{cases} \dots\dots\dots (3-16)$$

3.1.4 Perancangan Rule Base System

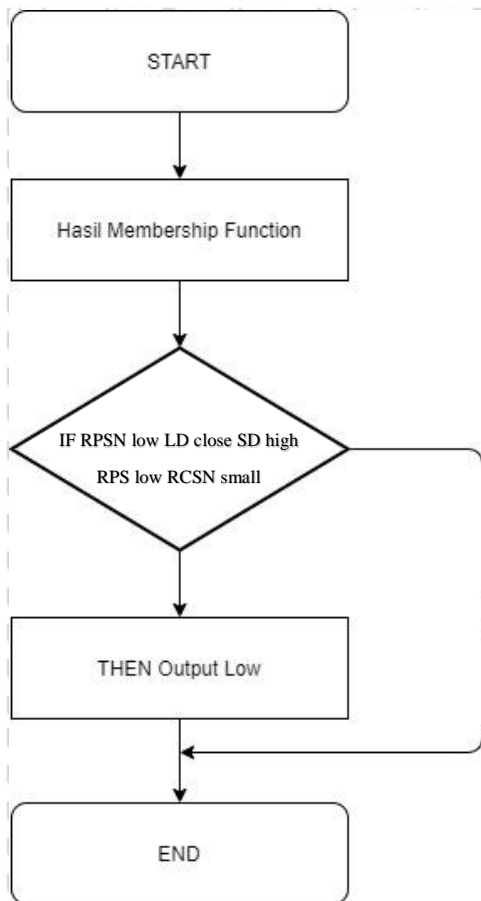
Setelah proses pembuatan fungsi keanggotaan, dilakukan pembuatan *rule base system*. Sebelum membuat *rule base system* tentukan dulu nilai diagram *fuzzy output* (Z). Diagram *Fuzzy output* disajikan dalam Gambar 3.8.



Gambar 3.8 Diagram *Fuzzy Output* (Z)
Sumber: Perancangan

Langkah selanjutnya pembentukan *rule base system*. *Rule base system* didapatkan dari ke-lima parameter yang masing – masing memiliki dua variabel parameter. Sedangkan untuk fungsi implikasi, fungsi yang digunakan adalah fungsi AND (fungsi MIN).

Berikut adalah diagram alir *IF – THEN* untuk Rule Base System:



Gambar 3.9 Diagram Alir IF – THEN
 Sumber: Perancangan

Berikut adalah rule yang ditetapkan:

1. If (RSPN is low) and (LD is close) and (SD is high) and (RPS is low) and (RSCN is small) then (output1 is low) (1)
2. If (RSPN is low) and (LD is close) and (SD is high) and (RPS is low) and (RSCN is medium) then (output1 is vlow) (1)
3. If (RSPN is low) and (LD is close) and (SD is high) and (RPS is low) and (RSCN is large) then (output1 is vlow) (1)
241. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is small) then (output1 is rhigh) (1)
242. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is medium) then (output1 is medium) (1)
243. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is large) then (output1 is medium) (1)

3.1.5 Perancangan Defuzzifikasi

Setelah proses rule base system akan diproses defuzzyfikasi menggunakan *weight average*. Dengan *weighted average*, setiap nilai hasil inferensi terhadap aturan *fuzzy* (W) dikalikan dengan nilai diagram fuzzy output (Z), kemudian dijumlahkan. Hasil penjumlahan tersebut kemudian dibagi dengan penjumlahan final output. *Weight average* menghasilkan sebuah nilai bilangan real yang dijadikan bobot pada setiap node Cluster Head Selection WSN.

$$\text{Final Output} = \frac{\sum_{i=1}^N w_i z_i}{w_i} \quad (3-17)$$

Dengan :

W = nilai inferensi fuzzy

Z = nilai diagram fuzzy output

3.2. Pengujian Sistem

Pada saat simulasi, penulis menggunakan *node* sebanyak 50 *node* dengan putaran (*round*) sebanyak 1000 kali. Sistem dapat menggunakan dua teknik dalam penentuan titik lokasi *node*, yaitu melalui teknik *random* (acak) dan melalui data *excel* (ekstensi .xls). Selain itu, *size area* dari *node* dapat diatur pada sistem dan dapat diatur sesuai dengan posisinya. Pada implementasi sistem, simulasi yang digunakan menggunakan teknik data *excel*.

3.3. Analisa Hasil & Pengambilan Kesimpulan

Analisa hasil yang dilakukan dalam penelitian ini meliputi analisa-analisa sebagai berikut:

a. Analisa Node Hidup

Analisa ini bertujuan untuk menentukan node hidup. Node hidup merupakan jumlah node yang masih aktif dalam setiap putaran.

b. Analisa Rata-rata Konsumsi Energi

Analisa ini bertujuan untuk menentukan rata-rata konsumsi energi. Rata-rata konsumsi energi merupakan nilai rata-rata energi yang dikonsumsi oleh setiap node dalam setiap putaran.

c. Analisa Tingkat Konsumsi Energi

Analisa ini bertujuan untuk menentukan tingkat konsumsi energi. Tingkat konsumsi energi merupakan jumlah energi yang dikonsumsi oleh sistem dalam setiap putaran.

d. Analisa Paket yang Diterima

Analisa ini bertujuan untuk menentukan paket yang diterima. Paket yang diterima merupakan jumlah paket yang diterima oleh *sink* dari node *cluster head* dalam setiap putaran.

e. Analisa Energi Tersisa

Analisa ini bertujuan untuk menentukan energi tersisa. Energi tersisa merupakan jumlah energi yang tersisa dalam sistem pada setiap putaran.

Dari hasil setiap analisa yang dilakukan maka penulis akan mengambil kesimpulan berdasarkan pebandingan nilai hasil pengujian dengan tujuan pengujian.

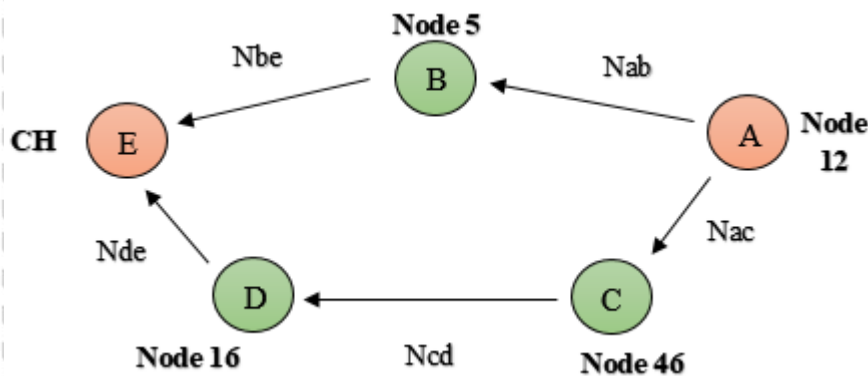


BAB 4

HASIL DAN PEMBAHASAN

4.1. Umum

Perhitungan *Fuzzy Sugeno* memberikan kinerja yang lebih baik untuk *graph* atau *node* skala besar dibandingkan algoritma *Rectangular* serta mendapatkan jalur yang paling efektif dan efisien. Di sini akan digunakan sampel 5 *node* saja dari total populasi 50 *node* karena terlalu banyak jika di tuliskan semuanya.



Gambar 4.1 Contoh Graf 5 Node
Sumber: Pengujian

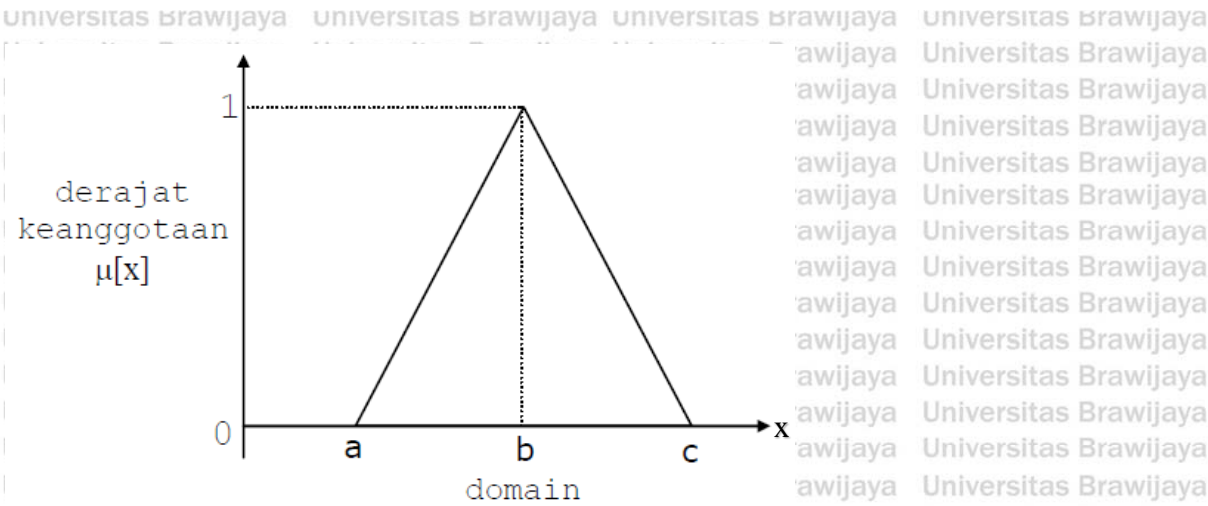
4.2. Perhitungan *Fuzzy*

Contoh perhitungan *fuzzy* dengan *node* awal yaitu *node* A ke *node* tujuan yaitu *node* E dengan rincian data parameter sebagai berikut:

Table 4.1 Data Routing dan Hasil *Fuzzy*

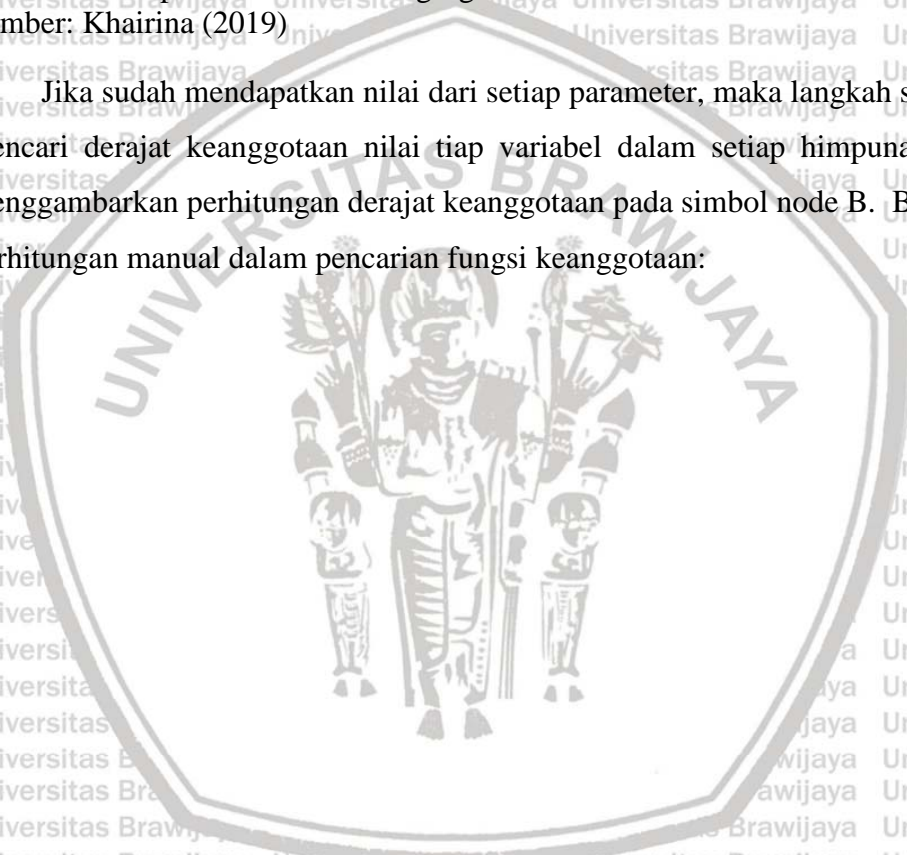
Simbol <i>Node</i>	Nomor <i>Node</i>	RPSN	LD	SD	RPS	RCSN	<i>Fuzzy</i> Output
A→B	12→5	0,24	12,4	18,9	1,95	2,77	20,62
A→C	12→46	0,46	91,1	7,5	0,53	18,8	5
B→E	5→CH	0,83	94	87,5	9,54	15,5	99,43
C→D	46→16	0,42	40,8	2,5	5,5	15,5	3,95
D→E	16→CH	0,46	45,7	48,9	4,93	14,8	35,86

Sumber: Hasil Pengujian



Gambar 4.2 Representasi Kurva Segitiga
Sumber: Khairina (2019)

Jika sudah mendapatkan nilai dari setiap parameter, maka langkah selanjutnya adalah mencari derajat keanggotaan nilai tiap variabel dalam setiap himpunan. Penulis akan menggambarkan perhitungan derajat keanggotaan pada simbol node B. Berikut merupakan perhitungan manual dalam pencarian fungsi keanggotaan:





Gambar 4.3 Diagram Alir Perhitungan *Fuzzy*
 Sumber: Pengujian

1. RPSN sebesar 0,83 berada pada area **High** maka dapat dihitung dengan menggunakan persamaan (2-8) yaitu sebagai berikut:

$$\mu_{\text{High}} = \begin{cases} 0 & ; H \leq 0,5 \text{ atau } H \geq 1 \\ \left(\frac{H - 0,5}{0,75 - 0,5} \right) & ; 0,5 \leq H \leq 0,75 \\ \left(\frac{1 - H}{1 - 0,75} \right) & ; 0,75 \leq H \leq 1 \end{cases}$$

Low : 0

Midle : 0

High : $(1 - 0,83)/0,25 = 0,68$

2. LD sebesar 94 berada pada area **Medium** dan **Far** maka dapat dihitung dengan menggunakan persamaan (2-4) dan (2-5) yaitu sebagai berikut:

$$\mu_{LD}(\text{med}) = \begin{cases} 0; & M \leq 50 \\ \left(\frac{M-50}{95-50}\right); & 50 \leq M \leq 95 \\ 1; & M \geq 95 \end{cases}$$

$$\mu_{LD}(\text{far}) = \begin{cases} 0; & F \leq 95 \\ \left(\frac{F-95}{100-95}\right); & 95 \leq F \leq 100 \\ 1; & F \geq 100 \end{cases}$$

Close : 0

Med : $(95 - 94)/45 = 0,02$

Far : $(94 - 50)/45 = 0,98$

3. SD sebesar 87,5 berada pada area **Medium** dan **High** maka dapat dihitung dengan menggunakan persamaan (2-13) dan (2-14) yaitu sebagai berikut:

$$\mu_{SD}(\text{Medium}) = \begin{cases} 0; & M \leq 50 \\ \left(\frac{M-50}{90-50}\right); & 50 \leq M \leq 90 \\ 1; & M \geq 90 \end{cases}$$

$$\mu_{SD}(\text{High}) = \begin{cases} 0; & H \leq 90 \\ \left(\frac{H-90}{100-90}\right); & 90 \leq H \leq 100 \\ 1; & H \geq 100 \end{cases}$$

Low : 0

Med : $(90 - 87,5)/40 = 0,22$

High : $(87,5 - 50)/40 = 0,94$

4. RPS 9,54 berada pada area **High** maka dapat dihitung dengan menggunakan persamaan (2-11) yaitu sebagai berikut:

$$\mu_{RPS}(\text{High}) = \begin{cases} 0; & H \leq 8 \\ \left(\frac{H-8}{10-8}\right); & 8 \leq H \leq 10 \\ 1; & C \geq 10 \end{cases}$$

Low : 0

Midle : 0

High : 1

5. RCSN sebesar 15,5 berada pada area **Large** maka dapat dihitung dengan menggunakan persamaan (2-17) yaitu sebagai berikut:

$$\mu_{\text{RCSN}}(\text{Large}) = \begin{cases} 0 & ; L \leq 10 \text{ atau } L \geq 20 \\ \left(\frac{L-10}{15-10}\right) & ; 10 \leq L \leq 15 \\ \left(\frac{15-L}{20-15}\right) & ; 15 \leq L \leq 20 \end{cases}$$

Small : 0

Med : 0

Large : $(20 - 15,5)/5 = 0,9$

6. Langkah selanjutnya adalah memasuki tahapan *rule evaluation*. *Rule evaluation* akan dilakukan pencarian yang cocok dengan derajat keanggotaan. *Rule evaluation* secara lengkap disajikan pada Tabel 4.1. Hasil *rule evaluation* yang cocok adalah sebagai berikut:

IF RPSN high LD medium SD medium RPS high RCSN large THEN rhigh

$$w_1 = \text{MIN}(0,68; 0,02; 0,22; 1, 0,9)$$

$$w_1 = 0,02$$

$$Z_1 = \text{rhigh} = 80$$

IF RPSN high LD far SD high RPS high RCSN large THEN vhigh

$$w_2 = \text{MIN}(0,68; 0,98; 0,94; 1, 0,9)$$

$$w_2 = 0,68$$

$$Z_2 = \text{vhigh} = 100$$

7. Langkah selanjutnya adalah melakukan perhitungan *weight average* untuk menghasilkan pengambilan keputusan *Fuzzy Sugeno* menggunakan persamaan (2-20), yaitu sebagai berikut:

$$\text{Final Output} = \frac{(w_1 \times z_1) + (w_2 \times z_2)}{w_1 + w_2}$$

$$\text{Final Output} = \frac{(0,02 \times 80) + (0,68 \times 100)}{0,02 + 0,68} = 99,43$$

Berikut *script* perhitungan *Fuzzy* yang ada dalam program Matlab:

```
NumOutputs=1
```

```
NumRules=243
```

```
AndMethod='min'
```

```
OrMethod='max'
```

```

ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtsum'

[Input1]
Name='RPSN'
Range=[0, 1]
NumMFs=3
MF1='low': 'trapmf', [0 0 0.25 0.5]
MF2='midle': 'trimf', [0.25 0.5 0.75]
MF3='high': 'trapmf', [0.5 0.75 1 1]

[Input2]
Name='LD'
Range=[0 100]
NumMFs=3
MF1='close': 'trapmf', [0 0 5 50]
MF2='med': 'trimf', [5 50 95]
MF3='far': 'trapmf', [50 95 100 100]

[Input3]
Name='SD'
Range=[0 100]
NumMFs=3
MF1='low': 'trapmf', [0 0 10 50]
MF2='medium': 'trimf', [10 50 90]
MF3='high': 'trapmf', [50 90 100 100]

[Input4]
Name='RPS'
Range=[0, 10]
NumMFs=3
MF1='low': 'trapmf', [0 0 2 3]
MF2='midle': 'trimf', [2 5 8]
MF3='high': 'trapmf', [7 8 10, 10]

[Input5]
Name='RCSN'
Range=[0, 20]
NumMFs=3
MF1='small': 'trapmf', [0 0 5 10]

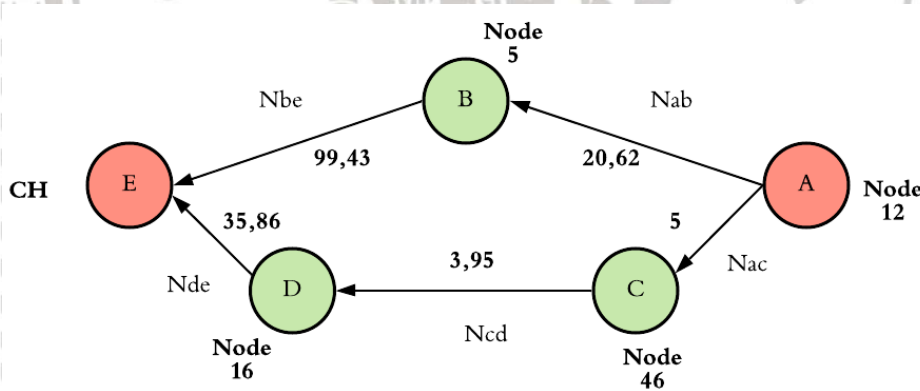
```




```

MF2='medium': 'trimf', [5 10 15]
MF3='large': 'trapmf', [10 15 20 20]
[Output1]
Name='output1'
Range=[0 1]
NumMFs=9
MF1='vlow': 'constant', [0]
MF2='low': 'constant', [0.125]
MF3='rlow': 'constant', [0.25]
MF4='mlow': 'constant', [0.375]
MF5='medium': 'constant', [0.5]
MF6='mhigh': 'constant', [0.625]
MF7='rhigh': 'constant', [0.75]
MF8='high': 'constant', [0.875]
MF9='vhigh': 'constant', [1]
    
```

Dari hasil perhitungan *fuzzy* yang disajikan pada Tabel 4.1, setiap jalur mempunyai *fuzzy output* yang secara otomatis dihitung oleh program disajikan pada Gambar 4.4 di bawah ini:



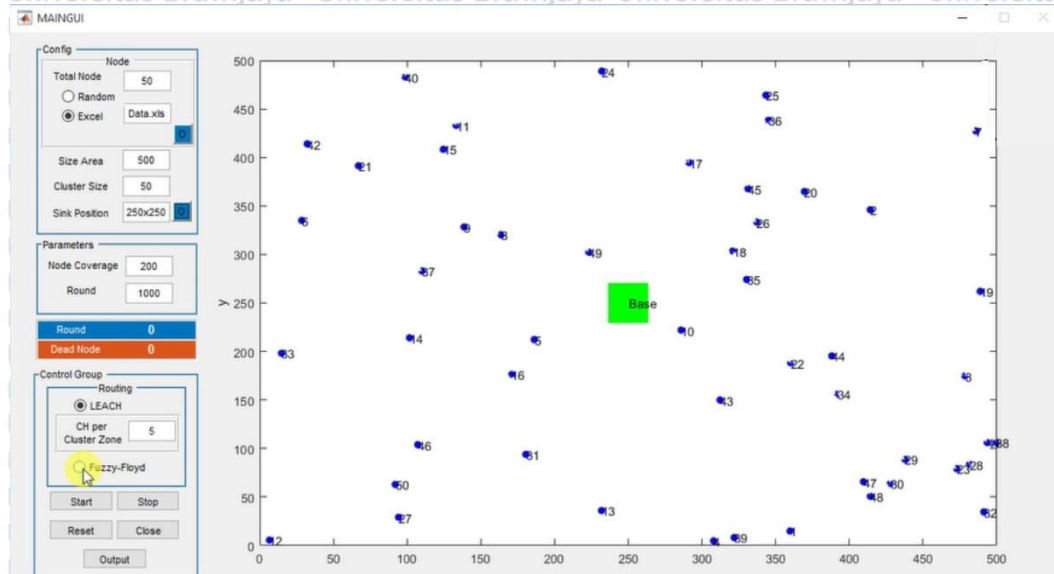
Gambar 4.4 Graf Hasil Proses Fuzzy Sugeno
 Sumber: Hasil Pengujian

Gambar 4.4 di atas menunjukkan bahwa jalur A-C-D-E lebih efektif, efisien, dan optimal daripada jalur A-B-E.

4.3. Implementasi pada Program Aplikasi

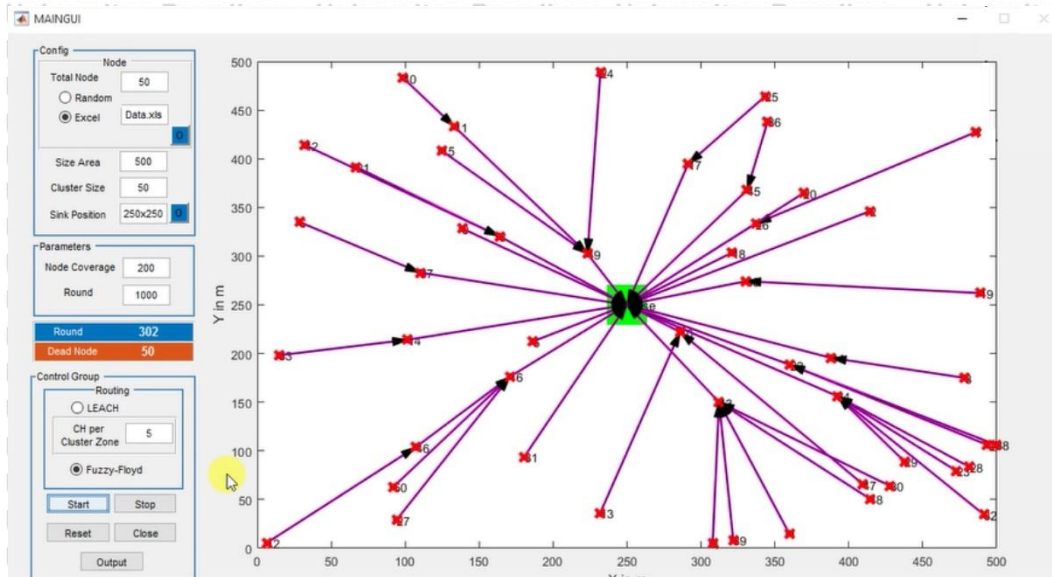
Hasil simulasi pada saat menggunakan sistem akan dijabarkan pada sub bab ini. Pada saat simulasi, penulis menggunakan *node* sebanyak 50 *node* dengan putaran (*round*) sebanyak 1000 kali. Sistem dapat menggunakan dua teknik dalam penentuan titik lokasi

node, yaitu melalui teknik *random* (acak) dan melalui data *excel* (ekstensi .xls). Selain itu, *size area* dari *node* dapat diatur pada sistem dan dapat diatur sesuai dengan posisinya. Pada implementasi sistem, simulasi yang digunakan menggunakan teknik data *excel*. Berikut ini merupakan konfigurasi awal dan penyebaran titik lokasi *node* pada saat simulasi berlangsung yang ditunjukkan pada Gambar 4.5 di bawah ini:



Gambar 4.5 Penyebaran Node
Sumber: Pengujian

Dengan menggunakan *Fuzzy Sugeno*, *node* akan mencari rute menuju *cluster head* (CH) dengan seoptimal mungkin dan diharapkan dapat mengurangi konsumsi energi pada saat protokol *routing* pada WSN berlangsung. Putaran akan otomatis dihentikan ketika semua *node* sudah mati (*dead node*). Berikut ini merupakan hasil akhir dari pencarian rute menuju *cluster head* yang digambarkan pada Gambar 4.6 di bawah ini:



Gambar 4.6 Hasil Routing
 Sumber: Pengujian

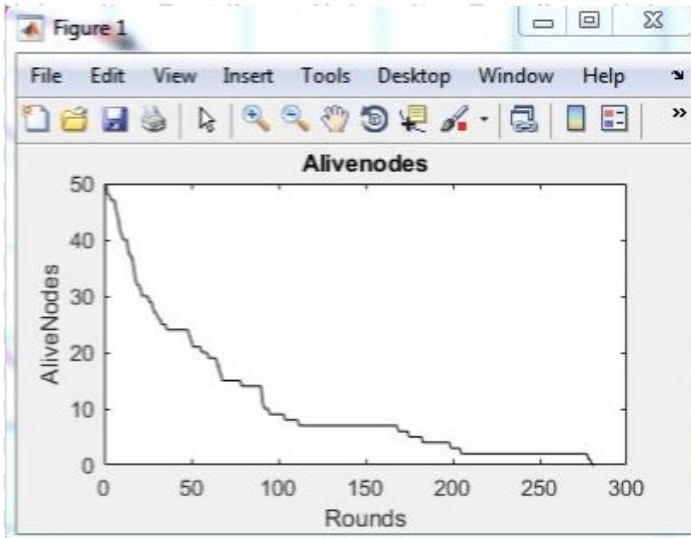
Berdasarkan Gambar 4.6 di atas, putaran (*round*) berhenti pada putaran 282 ketika *dead node* sudah berjumlah 50. Hasil untuk *node* dengan nomor 12 menuju *node base* (*Cluster Head*) menunjukkan kesesuaian dengan perhitungan manual yang ditunjukkan pada sub bab 4.1. Hal ini dapat disimpulkan bahwa pengujian program aplikasi mengalami tingkat akurasi 100% dan dinyatakan sukses atau akurat.

4.4. Analisa Hasil

Pada proses simulasi menggunakan program aplikasi tersebut menghasilkan sejumlah grafik yang dapat diamati.

4.4.1 Node hidup

Grafik *node* hidup (*alive nodes*) yang menggambarkan grafik *node* yang mati pada setiap putaran (*round*) ditunjukkan pada Gambar 4.7 yaitu sebagai berikut:

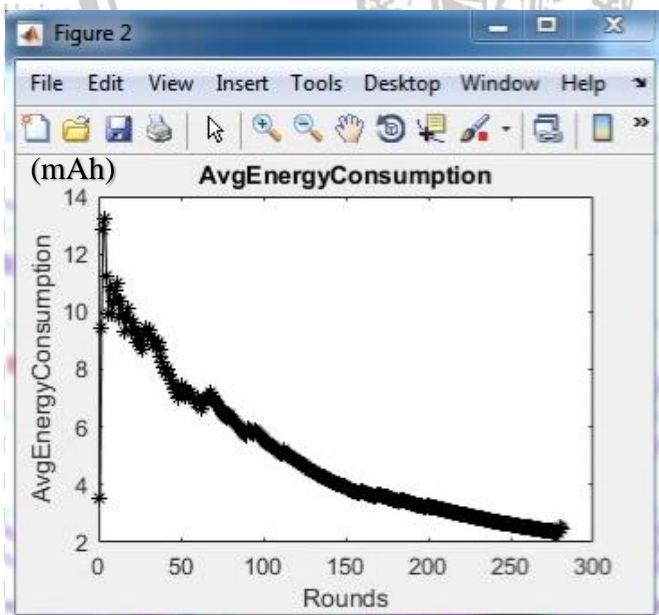


Gambar 4.7 Grafik *Alive Nodes*
 Sumber: Hasil Pengujian

Grafik diatas menunjukkan jumlah *Alive Nodes* dalam setiap *round* dan dapat di amati bahwa semua node telah mati pada saat putaran ke-282, dimana satu putaran memiliki waktu 1 detik sehingga sistem memerlukan waktu sebanyak 4 menit 42 detik.

4.4.2 Rata-rata Konsumsi Energi

Pada saat simulasi berlangsung menghasilkan rata-rata konsumsi energi (*Average Energy Consumption*) dengan grafik seperti yang ditunjukkan Gambar 4.8 di bawah ini:

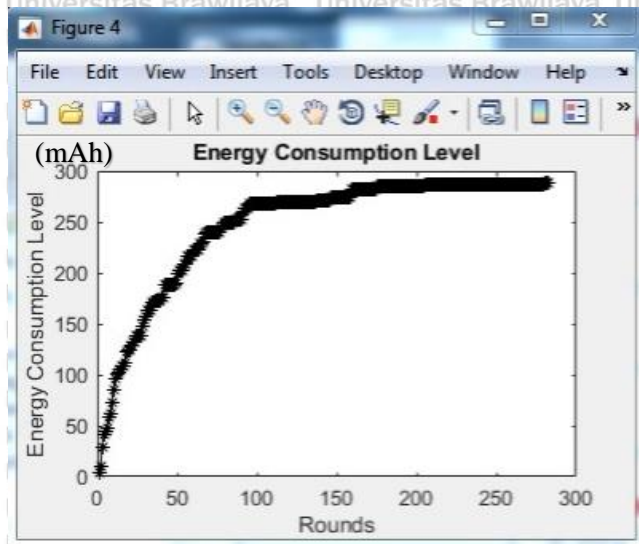


Gambar 4.8 Grafik *Average Energy Consumption*
 Sumber: Hasil Pengujian

Dari grafik di atas dapat diamati bahwa rata-rata konsumsi energi tiap node berkurang seiring putaran yang terjadi karena semakin sedikit node yang aktif.

4.4.3 Tingkat Konsumsi Energi

Pada saat simulasi berlangsung juga menghasilkan tingkat konsumsi energi (*Energy Consumption Level*) dengan grafik seperti yang ditunjukkan Gambar 4.9 di bawah ini.

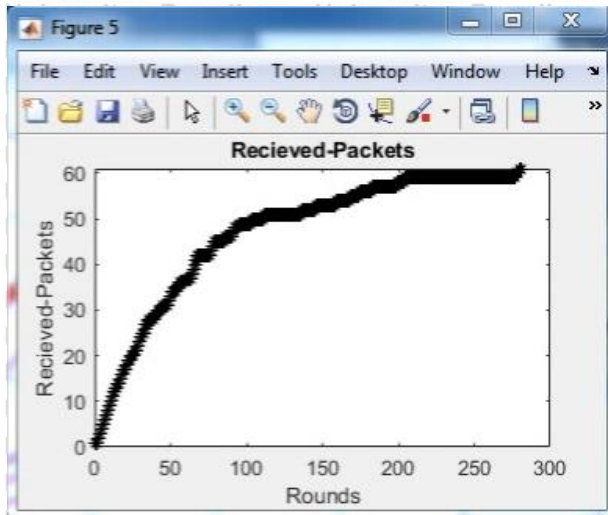


Gambar 4.9 Grafik *Energy Consumption Level*
 Sumber: Hasil Pengujian

Dari grafik di atas dapat diamati bahwa semakin banyak putaran yang terjadi maka tingkat tingkat konsumsi energi akan semakin melandai karena jumlah node yang aktif semakin sedikit.

4.4.4 Paket Yang Diterima

Pada saat simulasi berlangsung juga menghasilkan penerimaan paket (*Received-Packets*) dengan grafik seperti yang ditunjukkan Gambar 4.10 di bawah ini.



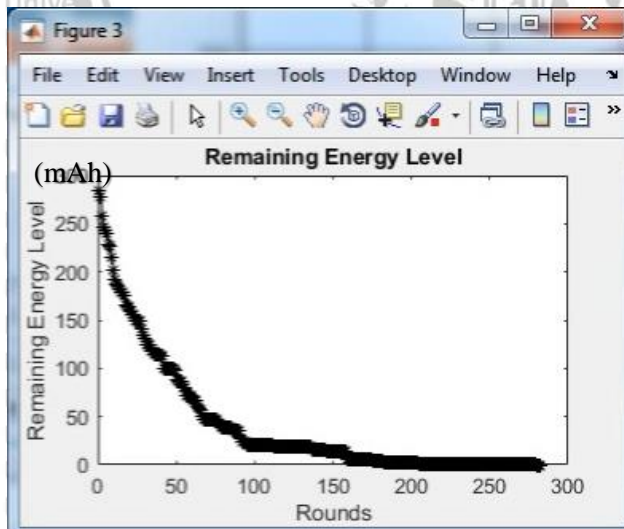
Gambar 4.10 Grafik *Recieved-Packets*

Sumber: Hasil Pengujian

Dari grafik di atas dapat diamati bahwa seluruh packet berhasil diterima saat putaran ke-282, atau dalam waktu 4 menit 42 detik.

4.4.5 Tingkat Energi Tersisa

Pada saat simulasi berlangsung juga menghasilkan tingkat energi yang tersisa (*Remaining Energy Level*) dengan grafik seperti yang ditunjukkan Gambar 4.11 di bawah ini.



Gambar 4.11 Grafik *Remaining Energy Level*

Sumber: Hasil Pengujian

Dari grafik di atas dapat diamati bahwa tingkat energi yang tersisa akan semakin berkurang seiring putaran yang terjadi dan mencapai angka nol pada putaran ke-282. Hal

itu terjadi karena sudah tidak ada lagi node yang aktif sehingga tidak ada lagi energi yang tersisa dalam sistem.



BAB 5

PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan meliputi, perancangan algoritma, perhitungan fuzzy, dan implementasi algoritma *routing protocol* dalam *Wireless Sensor Network* pada program aplikasi di Matlab, dapat disimpulkan bahwa:

1. Perancangan algoritma aplikasi khusus protokol routing menggunakan Fuzzy Sugeno dengan menggunakan Matlab berhasil dilakukan. Dengan kesimpulan setiap grafik pengamatan:
 - a. Grafik Alive Nodes menunjukkan bahwa semua node telah mati pada saat putaran ke-282, dimana satu putaran memiliki waktu 1 detik sehingga sistem memerlukan waktu sebanyak 4 menit 42 detik.
 - b. Grafik Average Energy Consumption menunjukkan bahwa rata-rata konsumsi energi tiap node berkurang seiring putaran yang terjadi karena semakin sedikit node yang aktif.
 - c. Grafik Energy Consumption Level menunjukkan bahwa semakin banyak putaran yang terjadi maka tingkat konsumsi energi akan semakin melandai karena jumlah node yang aktif semakin sedikit.
 - d. Grafik Received-Packets menunjukkan bahwa seluruh packet berhasil diterima saat putaran ke-282, atau dalam waktu 4 menit 42 detik.
 - e. Grafik Remaining Energy Level menunjukkan bahwa tingkat energi yang tersisa akan semakin berkurang seiring putaran yang terjadi dan mencapai angka nol pada putaran ke-282. Hal itu terjadi karena sudah tidak ada lagi node yang aktif sehingga tidak ada lagi energi yang tersisa dalam sistem.
2. Penerapan metode *Fuzzy Sugeno* pada *routing protocol* WSN dengan menggunakan parameter *Local Distance (LD)*, *Residual Power of Sensor Nodes (RPSN)*, *Remaining Battery Power of Sensor (RPS)*, *Station Distance (SD)*, dan *Rate of recurrent Communication of Sensor Node (RCSN)* cukup mengoptimalkan pencarian *cluster head (CH)* dengan memberikan pertimbangan bobot nilai yang spesifik pada tiap jalur antar *node* dan *cluster head (CH)*. Sehingga konsumsi energi dan waktu putaran sistem untuk mengirimkan paket ke *sink* dari setiap node

menjadi lebih optimal karena Fuzzy Sugeno memproses semua parameter pembobotan tiap jalur antar node yang ada.

5.2. Saran

Saran yang digunakan untuk pengembangan penelitian ini adalah :

Penerapan Fuzzy Sugeno pada routing protocol WSN dengan menggunakan parameter Local Distance (LD), Residual Power of Sensor Nodes (RPSN), Remaining Battery Power of Sensor (RPS), Station Distance (SD), dan Rate of recurrent Communication of Sensor Node (RCSN) dalam perangkat nyata bergantung pada spesifikasi dan kualitas setiap sensor. Maka dianjurkan untuk memilih setiap sensor dan router yang sesuai dengan kegunaan yang dibutuhkan dan medan yang ada.



DAFTAR PUSTAKA

- AlShawi, I. S., Yan, L., Pan, W., & Luo, B. (2012). Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm.
- Ando, H., Barolli, L., Durresi, A., Xhafa, F., & Koyama, A. (2010, November). An intelligent fuzzy-based cluster head selection system for WSNs and its performance evaluation for D3N parameter. In *2010 International Conference on Broadband, Wireless Computing, Communication and Applications* (pp. 648-653). IEEE.
- Arrozaqi, U. A. (2011). Simulasi routing protokol pada jaringan sensor nirkabel dengan menggunakan metode cluster based. *EEPIS final project*.
- Azinar, A. W., & Sari, D. N. (2015). Analisis Perbandingan Routing Protokol OLSR (Optimized Link State Routing) dan GRP (Geographic Routing Protocol) pada Wireless Sensor Network. *Proceeding SNTEKPAN, Oct*.
- Bidaki, M. and R. K. Tabbakh (2016). "Efficient fuzzy logic-based clustering algorithm for wireless sensor networks." *International Journal of Grid and Distributed Computing* 9(5): 79-88.
- Jati, A. S., Puspitorini, O., & Siswandari, N. A. (n.d.). *Karakterisasi Lingkungan Propagasi di Daerah Terbuka Untuk Aplikasi WSN*. 1-5.
- Khairina, N. (2019) <http://khairina.blog.uma.ac.id/wp-content/uploads/sites/394/2019/09/BAB-1-LOGIKA-FUZZY.pdf>, diakses tanggal 18 November 2020.
- Kodali, R. K., & Aravapalli, N. K. (2014, February). Multi-level LEACH protocol model using NS-3. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 375-380). Ieee.
- Kusumadewi, S. and H. Purnomo (2010). "Aplikasi Logika Fuzzy untuk pendukung keputusan." *Yogyakarta: Graha Ilmu*: 33-34.
- Natarajan, H. and S. Selvaraj (2014). A fuzzy based predictive cluster head selection scheme for wireless sensor networks. *International Conference on Sensing Technology*.
- Paul, A. K., & Sato, T. (2011, October). Effective data gathering and energy efficient communication protocol in wireless sensor network. In *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)* (pp. 1-5). IEEE.
- Sari, A. R., Muftada'i, N. R., Santoso, T. B. (2008). Analisa Kinerja dan Simulasi Clustering Penyebaran Node Pada Wireless Sensor Network menggunakan Algoritma K-MEANS. *Politeknik Elektronika Negeri Surabaya*, 1-5.
- Thein, M. C. M., & Thein, T. (2010, January). An energy efficient cluster-head selection for wireless sensor networks. In *2010 International Conference on Intelligent Systems, Modelling and Simulation* (pp. 287-291). IEEE.
- Wang, B., Shen, C., & Li, J. (2012). Study and improvement on leach protocol in wsn.
- Wang, Li-Xin, (1997). *A Course In Fuzzy Systems And Control International Edition*, Prentice-Hall Inc.

Warrier, M. M., & Kumar, A. (2016). An energy efficient approach for routing in wireless sensor networks. *Procedia Technology*, 25, 520-527.

Zimmermann. (1991). *Fuzzy Sets Theory and Its Applications*, Edisi 2. Kluwer Academic Publishers: Massachusetts.



LAMPIRAN

Lampiran 1 Listing Program Aplikasi Matlab

```

function varargout = MAINGUI(varargin)
% MAINGUI MATLAB code for MAINGUI.fig
% MAINGUI, by itself, creates a new MAINGUI or raises the existing
% singleton*.
%
% H = MAINGUI returns the handle to a new MAINGUI or the handle to
% the existing singleton*.
%
% MAINGUI('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in MAINGUI.M with the given input
% arguments.
%
% MAINGUI('Property','Value',...) creates a new MAINGUI or raises
% the existing singleton*. Starting from the left, property value
% pairs are applied to the GUI before MAINGUI_OpeningFcn gets
% called. An unrecognized property name or invalid value makes
% property application stop. All inputs are passed to
% MAINGUI_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
% one instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help MAINGUI
%
% Last Modified by GUIDE v2.5 09-Jul-2020 08:24:19
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @MAINGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @MAINGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
%
% --- Executes just before MAINGUI is made visible.
function MAINGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to MAINGUI (see VARARGIN)

```



```

% Choose default command line output for MAINGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MAINGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = MAINGUI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in startBtn.
function startBtn_Callback(hObject, eventdata, handles)
% hObject handle to startBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global E X Y Xb Yb N maxl SN Rc stopl
alpha=0.01;%0.001 energy per distance% 0.001
beta=0.045;%0.0015
Sector1=1;
nodes=str2double(get(handles.totalnodeEdit,'String'));% Total No. of Nodes
%E=(str2double(get(handles.initenergyEdit,'String'))).*ones(1,nodes);
E=randi(10,1,nodes);
nodesCH=str2double(get(handles.chzoneEdit,'String'));
EexL=E;
E1=E;
%% Ch selection
chalg1="chselalg.m";
chalg=strrep(chalg1,'.m','');
varName=matlab.lang.makeValidName(chalg);
chselalg=str2func(varName)
[CH]=chselalg(EexL,nodesCH)
%% Routing
if (get(handles.leachBtn,'Value') == 0 && get(handles.ffloydBtn,'Value')
== 0)
else
switch get(get(handles.uibuttongroup5,'SelectedObject'),'Tag')
case 'leachBtn', Route_type = 'leach';
case 'ffloydBtn', Route_type = 'fuzzy-floyd';
end
end
if(strcmp(Route_type,'fuzzy-floyd'))
Ralg1="shortpat.m";
Ralg=strrep(Ralg1,'.m','');
varName=matlab.lang.makeValidName(Ralg);
Ralgfun=str2func(varName)
end
%% EnergyFunction
Efun1="Energyfun.m";
Efun=strrep(Efun1,'.m','');
varName=matlab.lang.makeValidName(Efun);
Efun=str2func(varName)

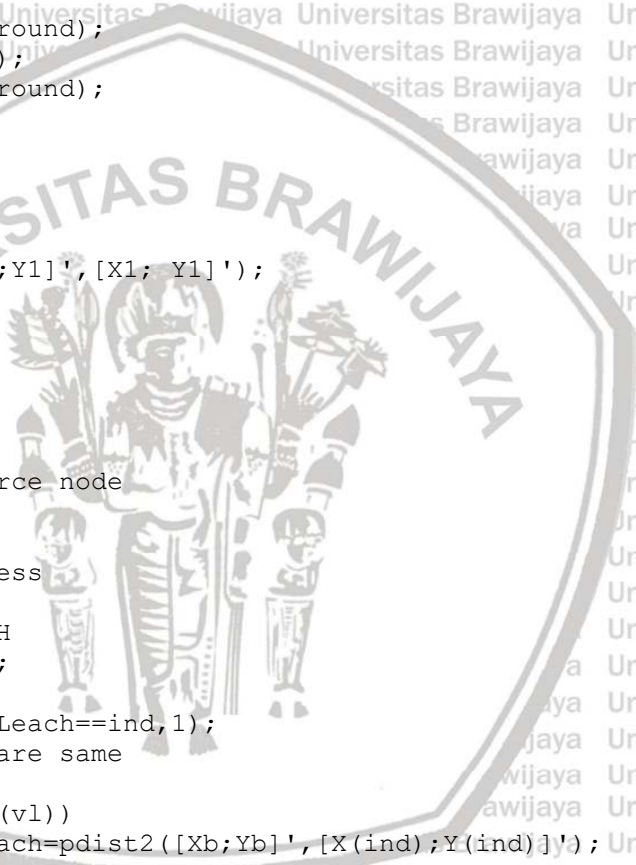
```

```

%% Route select
if(strcmp(Route_type,'leach'))
    A=1;
elseif(strcmp(Route_type,'fuzzy-floyd'))
    A=2;
end

%% ThroughputFn
Thfun1="Throughputfun.m";
Thfun=strrep(Thfun1,'.m','');
varName=matlab.lang.makeValidName(Thfun);
Thfun=str2func(varName)
%% Simulation Starts
round=str2double(get(handles.roundEdit,'String'));
roundDelay=.1;
%%
ipp=1;
Alivenodes=zeros(1,round);
AvgEc=zeros(1,round);
Throughput=zeros(1,round);
d=1;
apL=1;
%%
X1=[X Xb];
Y1=[Y Yb];
matrizP=pdist2([X1 ;Y1]',[X1; Y1]');
%%Routing Process
while(ipp<round)
    EneExL(ipp)=0;
    %%Leach
    %% Leach
    A1=randperm(N);
    Randomly select Source node
    ind=A1(3);
    pathL=[];
    %% Routing Process
    %% LEACH
    if(A==1)%% LEACH
        indLeach=CH;
        CH of LEACH
        v1=find(indLeach==ind,1);
        Verify node and CH are same
        v2=1;
        if(~isempty(v1))
            dist2Leach=pdist2([Xb;Yb]',[X(ind);Y(ind)]');
            %if same data direct transfer to sink
            valp=dist2Leach;
            dist2L=dist2Leach;
            indp=ind;
            v2=1;
            pathL1=[ind-1];
            Sink Node Assign as PathL=-1
        else
            for ib=1:numel(indLeach)
                dist22L(ib)=pdist2([X(indLeach(ib));Y(indLeach(ib))]',[X(ind);Y(ind)]');
                %verify which CH nearest to Node
            end
            [valp, indp]=min(dist22L);
        end
    end
    ipp=ipp+1;
end

```




```

dist1L=pdist2([X(indLeach(indp));Y(indLeach(indp))]',[X(ind);Y(ind)]');%
node to XH
dist2L=pdist2([Xb;Yb]',[X(indLeach(indp));Y(indLeach(indp))]',[X(ind);Y(ind)]'); % CH to
sink
v2=2;
pathL1=[ind indLeach(indp)-1];
end
dist1L(isempty(dist1L))=0;
dist2L(isempty(dist2L))=0;
apL=0;
if((valp < (Rc)) && EexL(ind)~=0)
pathL=pathL1;
apL=apL+1;
path11=pathL(1:end-1);
EexL(path11)=EexL(path11)-Efun(alpha,beta,dist1L,dist2L,A);
EneExL(ipp)=Efun(alpha,beta,dist1L,dist2L,A).*numel(path11);
end
end
%% fuzz-floyd
if(A==2)
Source = ind;
Dest=numel(X1);
Rc1=Rc;%4.*
% init fuzzy
RPSN = E(Source) * alpha * beta;
ld = matrizP(Source);
sd = matrizP(Source);
rps = E(Source) - EexL(Source);
RCSN = EexL(Source);
% eval fuzzy
fuzzf = ff(RPSN, ld, sd , rps, RCSN);
% shortest path
matrizP(matrizP>Rc1)=inf;
[pathP,cost]=Ralgun(Source, Dest,matrizP);
cost = cost + fuzzf;
costN=cost.*length(pathP);
dist1L=costN;
dist2L=0;
apL=0;
if(~isempty(pathP))
if(pathP(end)~=Dest)
pathP=[pathP Dest];
end
if(EexL(ind)~=0)
pathP(pathP==Dest)=-1;
pathL=pathP;
path11=pathP;
path11(path11==-1)=[];
apL=apL+1;
path11
EexL(path11)=EexL(path11)-
Efun(alpha,beta,dist1L,dist2L,A);
EneExL(ipp)=Efun(alpha,beta,dist1L,dist2L,A).*numel(path11);
end
end
end

```



```

%
if(EexL(ind)<=0)
    EexL(ind)=0;
end
%
if(Sector1==1)
    if(~isempty(pathL))
        path1=pathL;
        for p =1:(size(path1,2))-1
            if(path1(p+1)==-1)
                line([X(path1(p)) Xb], [Y(path1(p)) Yb],
                    'Color','m','LineWidth',2,'LineStyle','-')
                arrow([X(path1(p)) Y(path1(p)) ], [Xb Yb ])
            else
                line([X(path1(p)) X(path1(p+1))], [Y(path1(p))
                    Y(path1(p+1))], 'Color','m','LineWidth',2,'LineStyle','-')
                arrow([X(path1(p)) Y(path1(p)) ], [X(path1(p+1))
                    Y(path1(p+1)) ])
            end
            hold on
        end
    end
end
% Recluster Leach
EexL(EexL<=0)=0;
indg3=find(EexL<=0)
if(A==2)
    [CH]=chselalg(EexL,nodesCH)
end
% Store Parameters
Ec2L(ipp)=sum(EexL); %
Energy Consumption
REc2L(ipp)=sum(E1) - sum(EexL); % Remaining
Energy Level
% Throughput Calc
RxData1L=apL;
if(ipp>1)
    ThroughputL(ipp)=Thfun(ThroughputL,RxData1L,ipp-1);
else
    ThroughputL(ipp)=RxData1L;
end
% AliveNodes Calc
hold on
plot(X(indg3),Y(indg3),'x','LineWidth',3,...
    'MarkerEdgeColor','r',...
    'MarkerFaceColor','r',...
    'MarkerSize',10);
xlabel('X in m')
ylabel('Y in m')
hold on
AlivenodesL(ipp)=N-numel(indg3);
set(handles.deadnodeText,'string',num2str(N-AlivenodesL(ipp)));
%Avg Energy Consumed
AvgEcL(ipp)=mean(EneExL);
if ((sum(AlivenodesL(:))== 0))>2

```



```

break;
end;
ipp=ipp+1;
set(handles.round1Text,'string',num2str(ipp));
pause(roundDelay)
if(stop1==1)
    stop1=0;
    break;
end
end
if(strcmp(Route_type,'leach'))
    Az=Route_type;
else
    inde=find(Ralg1=='.');
    Az='fuzzyfloyd';
end
deadnode = zeros(N);
deadnode = AlivenodesL;
fname=['-' Az];
assigname = ['_' Az];
save(['Hasil\AliveNodes' fname '.mat'],'AlivenodesL')
assignin('base',['Alivenodes' assigname],AlivenodesL)
save(['Hasil\AverageEnergyConsumpt' fname '.mat'],'AvgEcL')
assignin('base',['RemainingEnergy' assigname],Ec2L)
save(['Hasil\RemainingEnergy' fname '.mat'],'Ec2L')
assignin('base',['RemainingEnergy' assigname],REc2L)
save(['Hasil\EnergyConsumption' fname '.mat'],'REc2L')
assignin('base',['EnergyConsumption' assigname],REc2L)
save(['Hasil\Throughput',fname,'.mat'],'ThroughputL')
assignin('base',['RecievedPacket' assigname],ThroughputL)
%%
figure,
plot(1:numel(AlivenodesL)-1,AlivenodesL(1:end-1),'-k')
xlabel('Rounds')
ylabel('AliveNodes')
title('Alivenodes')
%%
figure,
plot(1:numel(AvgEcL)-1,AvgEcL(1:end-1),'-*k')
xlabel('Rounds')
ylabel('AvgEnergyConsumption')
title('AvgEnergyConsumption')
%%
figure,
plot(1:numel(Ec2L),Ec2L(1:end),'-*k')
xlabel('Rounds')
ylabel('Remaining Energy Level')
title('Remaining Energy Level')
%%
figure,
plot(1:numel(REc2L),REc2L(1:end),'-*k')
xlabel('Rounds')
ylabel('Energy Consumption Level')
title('Energy Consumption Level')
%%
% Throughput
figure,

```

```

plot(1:numel(ThroughputL)-1,ThroughputL(1:end-1),'-*k')
xlabel('Rounds')
ylabel('Recieved-Packets')
title('Recieved-Packets')
% --- Executes on button press in rstBtn.
function rstBtn_Callback(hObject, eventdata, handles)
% hObject handle to rstBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.axes1);
hold off;
cla reset;
global X Y N max1 RaDat
rand('seed',RaDat)
N=str2double(get(handles.totalnodeEdit,'String'));% Total No. of Nodes
t1=get(handles.sizeareaEdit,'String');
min1=0;
max1=str2double(1);
X = min1+(max1-min1)*rand(1,N);
Y = min1+(max1-min1)*rand(1,N);
%%
plot(X,Y,'o','LineWidth',1,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','b',...
      'MarkerSize',5');
xlabel('X in m')
ylabel('Y in m')
for i2 = 1:N
    text(X(i2), Y(i2), num2str(i2),'FontSize',10);
    hold on;
end
hold on
axes(handles.axes1);
xlabel('x');
ylabel('y');
guidata(hObject,handles);
% --- Executes on button press in closeBtn.
function closeBtn_Callback(hObject, eventdata, handles)
% hObject handle to closeBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close all;
closereq;
% --- Executes on button press in outputBtn.
function outputBtn_Callback(hObject, eventdata, handles)
% hObject handle to outputBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
fn=dir('NEWBUILD\Hasil\*.mat');
for ik=1:numel(fn)
    ind1=find(fn(ik).name=='-');
    ind2=find(fn(ik).name=='.'');

```




```

fname2{ik}=fn(ik).name(ind1+1:ind2-1);
end
fname=unique(fname2);
c=['r' 'g' 'b' 'k' 'c' 'y' 'm'];
%%
figure,
for ik=1:numel(fname)
    load(['NEWBUILD\Hasi\Alivenodes-' fname(ik) '.mat'])
    plot(1:numel(AlivenodesL)-1,AlivenodesL(1:numel(AlivenodesL)-1),['-*'
c(ik)])
        hold on
end
xlabel('Rounds')
ylabel('AliveNodes')
legend(fname)
title('Alivenodes')
%%
figure,
for ik=1:numel(fname)
    load(['NEWBUILD\Hasi\AvgEc-' fname(ik) '.mat'])
    plot(1:numel(AvgEcL)-1,AvgEcL(1:numel(AvgEcL)-1),['-*' c(ik)])
        hold on
end
xlabel('Rounds')
ylabel('AvgEnergyConsumption')
legend(fname)
title('AvgEnergyConsumption')
%%
figure,
for ik=1:numel(fname)
    load(['NEWBUILD\Hasi\Ec2-' fname(ik) '.mat'])
    plot(1:numel(Ec2L),Ec2L,['-*' c(ik)])
        hold on
end
xlabel('Rounds')
ylabel('Remaining Energy Level')
legend(fname)
title('Remaining Energy Level')
%%
figure,
for ik=1:numel(fname)
    load(['NEWBUILD\Hasi\Rec2-' fname(ik) '.mat'])
    plot(1:numel(Rec2L),Rec2L,['-*' c(ik)])
        hold on
end
xlabel('Rounds')
ylabel('Energy Consumption Level')
legend(fname)
title('Energy Consumption Level')
%%
% Throughput
figure,
for ik=1:numel(fname)
    load(['NEWBUILD\Hasi\Throughput-' fname(ik) '.mat'])
    plot(1:numel(ThroughputL)-1,ThroughputL(1:numel(ThroughputL)-1),['-*'
c(ik)])
        hold on
end

```



```

legend(fname)
xlabel('Rounds')
ylabel('Recieved-Packets')
title('Recieved-Packets')

function chzoneEdit_Callback(hObject, eventdata, handles)
% hObject handle to chzoneEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of chzoneEdit as text
% str2double(get(hObject,'String')) returns contents of chzoneEdit
as a double

% --- Executes during object creation, after setting all properties.
function chzoneEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to chzoneEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function coverageEdit_Callback(hObject, eventdata, handles)
% hObject handle to coverageEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of coverageEdit as text
% str2double(get(hObject,'String')) returns contents of coverageEdit
as a double

% --- Executes during object creation, after setting all properties.
function coverageEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to coverageEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function initenergyEdit_Callback(hObject, eventdata, handles)
% hObject handle to initenergyEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of initenergyEdit as text
% str2double(get(hObject,'String')) returns contents of
initenergyEdit as a double

% --- Executes during object creation, after setting all properties.
function initenergyEdit_CreateFcn(hObject, eventdata, handles)

```



```

% hObject handle to initenergyEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function roundEdit_Callback(hObject, eventdata, handles)
% hObject handle to roundEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of roundEdit as text
% str2double(get(hObject,'String')) returns contents of roundEdit
as a double
% --- Executes during object creation, after setting all properties.
function roundEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to roundEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function totalnodeEdit_Callback(hObject, eventdata, handles)
% hObject handle to totalnodeEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of totalnodeEdit as text
% str2double(get(hObject,'String')) returns contents of totalnodeEdit
as a double
% --- Executes during object creation, after setting all properties.
function totalnodeEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to totalnodeEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sizeareaEdit_Callback(hObject, eventdata, handles)
% hObject handle to sizeareaEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```




```

% Hints: get(hObject,'String') returns contents of sizeareaEdit as text
% str2double(get(hObject,'String')) returns contents of sizeareaEdit
as a double
% --- Executes during object creation, after setting all properties.
function sizeareaEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sizeareaEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function clustersizeEdit_Callback(hObject, eventdata, handles)
% hObject    handle to clustersizeEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of clustersizeEdit as text
% str2double(get(hObject,'String')) returns contents of
clustersizeEdit as a double
% --- Executes during object creation, after setting all properties.
function clustersizeEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to clustersizeEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sinkEdit_Callback(hObject, eventdata, handles)
% hObject    handle to sinkEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sinkEdit as text
% str2double(get(hObject,'String')) returns contents of sinkEdit as
a double
% --- Executes during object creation, after setting all properties.
function sinkEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sinkEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

% --- Executes on button press in plotnodeBtn.
function plotnodeBtn_Callback(hObject, eventdata, handles)
% hObject    handle to plotnodeBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
%clear all;
%close all;
warning off all;
restoredefaultpath;
addpath(genpath(pwd));
global X Y N max1 RaDat
axes(handles.axes1);
hold off;
cla reset;
excelsheet=get(handles.excelEdit,'String');
if (get(handles.randomRBtn,'Value') == 0 && get(handles.excelRBtn,'Value')
== 0)
else
    switch get(get(handles.uibuttongroup7,'SelectedObject'),'Tag')
        case 'randomRBtn', type = 1;
        case 'excelRBtn', type = 2;
    end
end
RaDat=round(100.*rand(1));
rand('seed',RaDat)
if(type==1)
    N=str2double(get(handles.totalnodeEdit,'String'));% Total No. of Nodes
    t1=get(handles.sizeareaEdit,'String');
    min1=0;
    max1=str2double(t1);
    X = min1+(max1-min1)*rand(1,N);
    Y = min1+(max1-min1)*rand(1,N);
    assignin('base','X',X);
    assignin('base','Y',Y);
else
    [num,txt,row] = xlsread(excelsheet);
    X=num(:,1);
    Y=num(:,2);
    X = X.';
    Y = Y.';
    N=numel(X);
    %assignin('base','X',X);
    %assignin('base','Y',Y);
end
%%
plot(X,Y,'o','LineWidth',1,...
    'MarkerEdgeColor','b',...
    'MarkerFaceColor','b',...
    'MarkerSize',5);
xlabel('X in m')
ylabel('Y in m')
for i2 = 1:N
    text(X(i2), Y(i2), num2str(i2),'FontSize',10);
    hold on;

```

```

end
hold on
axes(handles.axes1);
xlabel('x');
ylabel('y');
guidata(hObject,handles);

% --- Executes on button press in plotareaBtn.
function plotareaBtn_Callback(hObject, eventdata, handles)
% hObject handle to plotareaBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global X Y Xb Yb N max1 SN Rc
Rc=(str2double(get(handles.coverageEdit,'String')));
t1=get(handles.sinkEdit,'String');%sink
t2=strsplit(t1,'x');
Xb =str2double(t2(1));
Yb =str2double(t2(2));
t1=get(handles.clustersizeEdit,'String');%zone
gapX=str2double(t1);
gapY=str2double(t1);
hold on
plot(Xb,Yb,'s','LineWidth',1,...
'MarkerEdgeColor','g',...
'MarkerFaceColor','g',...
'MarkerSize',40');
xlabel('X in m')
ylabel('Y in m')
text(Xb, Yb, 'Base','FontSize',10);
hold on;
% cluster zone
zX=0:gapX:max1;
zY=0:gapY:max1;
id=zeros(1,N);
ik1=1;
for ik=1: numel(zX)-1
for ij=1: numel(zY)-1
rectangle('Position',[zX(ik) zY(ij) gapX gapY]);
points = [zX(ik) zY(ij); zX(ik)+gapX zY(ij);
zY(ij)+gapY ;zX(ik) zY(ij)+gapY ;zX(ik) zY(ij)];
[in,on]=inpolygon(X,Y,points(:,1),points(:,2));
ind=find(in==1);
SN(ik1).id=ind;
ik1=ik1+1;
end
end
axes(handles.axes1);
xlabel('x');
ylabel('y');
guidata(hObject,handles);

% --- Executes on button press in plotsinkBtn.
function plotsinkBtn_Callback(hObject, eventdata, handles)
% hObject handle to plotsinkBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject handle to checkbox1 (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox1
function chselEdit_Callback(hObject, eventdata, handles)
% hObject handle to chselEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of chselEdit as text
% str2double(get(hObject,'String')) returns contents of chselEdit
as a double
% --- Executes during object creation, after setting all properties.
function chselEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to chselEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in stopBtn.
function stopBtn_Callback(hObject, eventdata, handles)
% hObject handle to stopBtn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global stop1
stop1=1;

function excelEdit_Callback(hObject, eventdata, handles)
% hObject handle to excelEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of excelEdit as text
% str2double(get(hObject,'String')) returns contents of excelEdit
as a double
% --- Executes during object creation, after setting all properties.
function excelEdit_CreateFcn(hObject, eventdata, handles)
% hObject handle to excelEdit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```



- Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
205. If (RSPN is high) and (LD is med) and (SD is medium) and (RPS is high) and (RSCN is small) then (output1 is rhigh) (1) ; Brawijaya
206. If (RSPN is high) and (LD is med) and (SD is medium) and (RPS is high) and (RSCN is medium) then (output1 is rhigh) (1) ; Brawijaya
207. If (RSPN is high) and (LD is med) and (SD is medium) and (RPS is high) and (RSCN is large) then (output1 is mhigh) (1) ; Brawijaya
208. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is low) and (RSCN is small) then (output1 is vhigh) (1) ; Brawijaya
209. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is low) and (RSCN is medium) then (output1 is high) (1) ; Brawijaya
210. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is low) and (RSCN is large) then (output1 is rhigh) (1) ; Brawijaya
211. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is midle) and (RSCN is small) then (output1 is high) (1) ; Brawijaya
212. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is high) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya
213. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is midle) and (RSCN is large) then (output1 is mhigh) (1) ; Brawijaya
214. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is high) and (RSCN is small) then (output1 is rhigh) (1) ; Brawijaya
215. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is high) and (RSCN is medium) then (output1 is mhigh) (1) ; Brawijaya
216. If (RSPN is high) and (LD is med) and (SD is low) and (RPS is high) and (RSCN is large) then (output1 is large) (1) ; Brawijaya
217. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is low) and (RSCN is small) then (output1 is vhigh) (1) ; Brawijaya
218. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is low) and (RSCN is medium) then (output1 is rhigh) (1) ; Brawijaya
219. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is low) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya
220. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is midle) and (RSCN is small) then (output1 is high) (1) ; Brawijaya
221. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is midle) and (RSCN is medium) then (output1 is rhioh) (1) ; Brawijaya
222. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is midle) and (RSCN is large) then (output1 is mhigh) (1) ; Brawijaya
223. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is high) and (RSCN is small) then (output1 is rhigh) (1) ; Brawijaya
224. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is high) and (RSCN is medium) then (output1 is high) (1) ; Brawijaya
225. If (RSPN is high) and (LD is far) and (SD is high) and (RPS is high) and (RSCN is large) then (output1 is vhigh) (1) ; Brawijaya
226. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is low) and (RSCN is small) then (output1 is vhigh) (1) ; Brawijaya
227. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is low) and (RSCN is medium) then (output1 is rhigh) (1) ; Brawijaya
228. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is low) and (RSCN is large) then (output1 is high) (1) ; Brawijaya
229. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is midle) and (RSCN is small) then (output1 is high) (1) ; Brawijaya
230. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is midle) and (RSCN is medium) then (output1 is rhigh) (1) ; Brawijaya
231. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is midle) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya
232. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is high) and (RSCN is small) then (output1 is medium) (1) ; Brawijaya
233. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is high) and (RSCN is medium) then (output1 is medium) (1) ; Brawijaya
234. If (RSPN is high) and (LD is far) and (SD is medium) and (RPS is high) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya
235. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is low) and (RSCN is small) then (output1 is rhigh) (1) ; Brawijaya
236. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is low) and (RSCN is medium) then (output1 is mhigh) (1) ; Brawijaya
237. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is low) and (RSCN is large) then (output1 is large) (1) ; Brawijaya
238. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is midle) and (RSCN is small) then (output1 is rhioh) (1) ; Brawijaya
239. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is midle) and (RSCN is medium) then (output1 is high) (1) ; Brawijaya
240. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is midle) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya
241. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is small) then (output1 is rhigh) (1) ; Brawijaya
242. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is medium) then (output1 is medium) (1) ; Brawijaya
243. If (RSPN is high) and (LD is far) and (SD is low) and (RPS is high) and (RSCN is large) then (output1 is medium) (1) ; Brawijaya

