

SKRIPSI

**PENGEMBANGAN SINKRONISASI WAKTU DI *WIRELESS
SENSOR NETWORK* DENGAN ALGORITMA *FLOODING
TIME SYNCHRONIZATION PROTOCOL***



Felicia Christiany

NPM: 2015730006

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019**

UNDERGRADUATE THESIS

DEVELOPMENT OF TIME SYNCHRONIZATION IN
WIRELESS SENSOR NETWORK WITH *FLOODING TIME
SYNCHRONIZATION PROTOCOL*



Felicia Christiany

NPM: 2015730006

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019

LEMBAR PENGESAHAN

**PENGEMBANGAN SINKRONISASI WAKTU DI *WIRELESS
SENSOR NETWORK* DENGAN ALGORITMA *FLOODING
TIME SYNCHRONIZATION PROTOCOL***

Felicia Christiany

NPM: 2015730006

Bandung, 15 Mei 2019

Menyetujui,

Pembimbing

Elisati Hulu, M.T.

Ketua Tim Penguji

Anggota Tim Penguji

Kristopher David Harjono, M.T.

Pascal Alfadian, M.Comp.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENGEMBANGAN SINKRONISASI WAKTU DI *WIRELESS SENSOR NETWORK* DENGAN ALGORITMA *FLOODING TIME SYNCHRONIZATION PROTOCOL*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 15 Mei 2019

Meterai Rp. 6000

Felicia Christiany
NPM: 2015730006

ABSTRAK

Wireless Sensor Network (WSN) terdiri atas node-node sensor yang saling berhubungan. Akan tetapi, waktu antar node sensor tersebut tidak saling tersinkronisasi karena setiap node sensor menggunakan waktu setelan standarnya masing-masing saat baru dinyalakan. Pada umumnya, aplikasi-aplikasi WSN memerlukan waktu yang sinkron antar node sensor agar operasi mereka dapat berjalan dengan baik. Proses sinkronisasi waktu mengkonsumsi daya yang besar, sementara daya node sensor sangat terbatas. Oleh karena itu, diperlukan sebuah algoritma yang dapat mensinkronisasikan waktu antar node sensor di WSN secara efisien.

Pada penelitian ini, algoritma sinkronisasi waktu yang digunakan adalah *Flooding Time Synchronization Protocol* (FTSP). Algoritma ini menggunakan satu buah node sensor yang dinamakan *root* untuk menjadi sumber referensi utama bagi node sensor lainnya dalam WSN. *Root* akan menyebarkan pesan sinkronisasi pada node-node sensor yang berada dalam jangkauannya. Setelah node sensor tersinkronisasi, mereka dapat ikut menyebarkan pesan sinkronisasi sehingga cakupan sinkronisasi akan semakin meluas. Dengan menggunakan konsep *flooding*, algoritma ini dapat mencakup area jaringan yang sangat luas yang terdiri atas banyak node sensor. Selain itu, penggunaan rumus perhitungan perbedaan waktu pada FTSP dan pemakaian banyak *time-stamp* dalam sebuah pesan sinkronisasi dapat menghasilkan tingkat akurasi yang tinggi.

Perangkat lunak telah dibangun dan diuji untuk memastikan performansi algoritma FTSP. Hasil performansi kemudian dibandingkan dengan perangkat lunak sinkronisasi waktu dengan algoritma *Reference Broadcast Synchronization* (RBS) yang dibangun pada penelitian lain. Meskipun tidak menangani seluruh parameter yang terdapat pada algoritma FTSP karena keterbatasan node sensor, algoritma FTSP tetap menghasilkan tingkat akurasi yang lebih baik dibandingkan dengan algoritma RBS. Dengan demikian, terbukti bahwa pengembangan ide dari algoritma RBS yang diterapkan oleh algoritma FTSP berhasil meningkatkan akurasi sinkronisasi waktu.

Kata-kata kunci: Jaringan Sensor Nirkabel, Sinkronisasi Waktu, *Flooding Time Synchronization Protocol*, *Reference Broadcast Synchronization*

ABSTRACT

Wireless Sensor Network (WSN) consists of interconnected sensor nodes. However, the clock between sensor nodes is not synchronized as each sensor node uses their own default clock when it is just turned on. In general, WSN applications require synchronous clock between sensor nodes so that their operations can run well. The clock synchronization process requires large power consumption of sensor node while its energy is very limited. Hence, a time synchronization algorithm is needed to efficiently synchronize the clock between sensor nodes in WSN.

The Flooding Time Synchronization Protocol (FTSP) is used in this study as the time synchronization algorithm. This algorithm uses a sensor node called root to be the main reference source for other sensor nodes in WSN. Root will broadcast a synchronization message to sensor nodes that are within its range. When those sensor nodes are synchronized, they can participate in broadcasting the synchronization message so that the synchronization coverage will be expanded. By using the flooding concept, this algorithm can cover a very wide network area consisting of many sensor nodes. In addition, the use of clock difference calculation formula on FTSP and the use of many time-stamps in a synchronization message can produce a high level of accuracy.

The software has been built and tested to ensure the performance of the FTSP algorithm. The performance results are then compared with the time synchronization software using Reference Broadcast Synchronization (RBS) that was built in other studies. Although this software does not handle all the parameters contained in FTSP algorithm because of the limitations of the sensor nodes, the FTSP algorithm still produces better accuracy than the RBS algorithm. Thus, it is proven that the development of the idea from RBS algorithm applied by the FTSP algorithm succeeded in increasing the accuracy of clock synchronization.

Keywords: Wireless Sensor Network, Time Synchronization, Flooding Time Synchronization Protocol, Reference Broadcast Synchronization

Dipersembahkan kepada orang tua, Informatika UNPAR, diri sendiri, serta semua orang yang berperan dalam pembuatan skripsi ini

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih-Nya yang begitu besar, penulis dapat menyelesaikan tugas akhir yang berjudul "Pengembangan Sinkronisasi Waktu di *Wireless Sensor Network* dengan Algoritma *Flooding Time Synchronization Protocol*" dengan baik dan tepat waktu. Penulis berharap agar tugas akhir ini dapat memberikan inspirasi dan manfaat bagi pembaca yang hendak melakukan dan mengembangkan penelitian terkait tugas akhir ini. Penulis juga mengucapkan terima kasih kepada seluruh pihak yang telah mendukung dan membantu penulis dalam menyelesaikan tugas akhir ini, yaitu:

1. Keluarga yang selalu memberikan dukungan kepada penulis untuk tidak menyerah.
2. Bapak Elisati Hulu sebagai dosen pembimbing yang selalu sedia memberikan nasihat, motivasi, dan bimbingan dengan sabar, juga rela mempercayakan peralatan node sensornya pada penulis.
3. Bapak Kristopher David Harjono dan Bapak Pascal Alfadian Nugroho selaku dosen penguji yang telah memberikan koreksi dan saran yang membangun sehingga tugas akhir ini dapat menjadi lebih baik.
4. Ibu Natalia yang telah menyediakan waktu dan memberikan pengarahan saat penulis membutuhkan konsultasi.
5. Joshua Riyadi sebagai teman diskusi dalam menghadapi topik tugas akhir yang serupa dan berkaitan, juga selalu rela membantu meminjamkan laptop saat penulis perlu melakukan pengujian.
6. Jonathan Alva dan Dandy Unggana sebagai teman seperjuangan dalam menghadapi tugas akhir yang menggunakan node sensor, juga rela membantu meminjamkan laptop saat penulis perlu melakukan pengujian.
7. Sutyoso, Thoby, Edrick, dan Yudhis yang telah membantu memberikan masukan dan ide pada penulis saat mengalami kesulitan.
8. Fakhry, Kezia, Matthew, dan Irvan yang seringkali ikut memeriahkan suasana dalam ruang lab skripsi.
9. Seluruh teman-teman Informatika angkatan 2015 yang telah memberikan pengalaman-pengalaman terbaik semasa kuliah.

Bandung, Mei 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 <i>Wireless Sensor Network</i> [1] [2] [3]	5
2.1.1 Node Sensor	5
2.1.2 Sistem Operasi Pada WSN	6
2.1.3 Protokol <i>Stack</i>	7
2.1.4 Protokol Komunikasi	8
2.1.5 Topologi Jaringan	9
2.1.6 Arsitektur Jaringan	10
2.2 Node Sensor <i>Preon32 Shuttle</i> [4] [5]	11
2.3 <i>Library</i> WSN pada <i>Java</i> [6]	12
2.3.1 Kelas <i>AT86RF231</i>	12
2.3.2 Kelas <i>Frame</i>	13
2.3.3 Kelas <i>FrameIO</i>	13
2.3.4 Kelas <i>USART</i>	13
2.4 Jenis-jenis <i>Delay</i> Waktu [7]	14
2.5 Penerapan WSN dalam Dunia Nyata [8]	15
2.6 Sinkronisasi Waktu pada WSN [2] [9]	16
2.6.1 <i>Network Time Protocol</i>	16
2.6.2 <i>Reference Broadcast Synchronization</i>	16
2.6.3 <i>Timing-Sync Protocol for Sensor Networks</i>	17
2.7 <i>Flooding Time Synchronization Protocol</i> [2] [7] [10]	17
2.7.1 Penggunaan <i>Time-stamp</i> Pada FTSP	17
2.7.2 Konten Pesan <i>Broadcast</i>	18
2.7.3 Metode Pemilihan <i>Root</i>	18
2.7.4 Metode Sinkronisasi Waktu	19
2.7.5 Kelebihan dan Kekurangan Algoritma FTSP	23

3	ANALISIS MASALAH	25
3.1	Deskripsi Karakteristik WSN	25
3.1.1	Karakteristik WSN untuk Area Sempit	25
3.1.2	Karakteristik WSN untuk Area Luas	26
3.2	Analisis Algoritma FTSP pada Perangkat Lunak	27
3.2.1	Tahap Inisiasi	27
3.2.2	Format Pesan	27
3.2.3	Tahap Pengiriman dan Penerimaan Pesan	27
3.2.4	Tahap Penanganan <i>Delay</i> Waktu	28
3.2.5	Tahap Sinkronisasi Node Sensor	28
3.2.6	Tahap Terminasi	29
3.3	Analisis Diagram Kelas	29
3.4	Analisis Diagram <i>Use Case</i>	30
3.5	Skenario Pengguna	30
4	PERANCANGAN	33
4.1	Perancangan Masukan dan Keluaran	33
4.2	Perancangan Format Pesan Sinkronisasi	33
4.3	Perancangan Interaksi Antarmuka dan Node Sensor	34
4.4	Perancangan Detil Diagram Kelas	36
4.4.1	Kelas <i>CMD</i>	36
4.4.2	Kelas <i>Root</i>	37
4.4.3	Kelas <i>SensorNode</i>	39
5	IMPLEMENTASI DAN PENGUJIAN	41
5.1	Implementasi	41
5.1.1	Lingkungan Perangkat Keras	41
5.1.2	Lingkungan Perangkat Lunak	41
5.1.3	Tampilan Masukan dan Keluaran Perangkat Lunak	42
5.1.4	Modul-modul Perangkat Lunak	44
5.2	Pengujian	47
5.2.1	Hasil Pengujian Fungsional Perangkat Lunak Sinkronisasi Waktu dengan Algoritma FTSP	47
5.2.2	Hasil Pengujian Eksperimental Algoritma FTSP	49
5.2.3	Perbandingan Hasil Analisis Performansi Algoritma FTSP dengan Algoritma RBS	55
6	KESIMPULAN DAN SARAN	59
6.1	Kesimpulan	59
6.2	Saran	59
	DAFTAR REFERENSI	61
	A KODE PROGRAM	63
	B HASIL EKSPERIMEN	71

DAFTAR GAMBAR

2.1	<i>Wireless Sensor Networks</i>	5
2.2	Sensor kelembaban dan node sensor	6
2.3	Struktur node sensor	6
2.4	Protokol <i>stack</i> pada WSN	7
2.5	Komunikasi <i>single-hop</i> dan <i>multi-hop</i> dalam WSN	9
2.6	Topologi Linier	9
2.7	Topologi <i>Star</i>	9
2.8	Topologi <i>Mesh</i>	10
2.9	Topologi <i>Fully-Connected Mesh</i>	10
2.10	Arsitektur <i>flat</i> pada WSN	11
2.11	Arsitektur hirarkis pada WSN	11
2.12	Contoh node sensor <i>Mica</i> , <i>Mica2</i> , dan <i>Preon32</i>	12
2.13	Jenis-jenis <i>delay</i> yang muncul di lapisan atas jaringan	14
2.14	Contoh pendeteksi kebakaran hutan dengan menggunakan WSN	15
2.15	Konten paket data yang ditransmisikan	18
2.16	Menentukan <i>root</i>	20
2.17	<i>Root</i> menyebar pesan sinkronisasi	20
2.18	<i>Reference point</i> masuk pada tabel	21
2.19	Mengestimasi <i>reference points</i> pada tabel	21
2.20	Sebagian node-node sensor telah tersinkronisasi	22
2.21	Node sensor yang telah tersinkronisasi dapat mulai menyebar pesan sinkronisasi	22
2.22	Seluruh node sensor telah tersinkronisasi	22
3.1	Denah WSN untuk area sempit	26
3.2	Denah WSN untuk area luas dengan jangkauan minimum 2 <i>hop</i> untuk <i>root</i> dapat menjangkau node sensor terjauh	26
3.3	Denah WSN untuk area luas dengan jangkauan minimum 3 <i>hop</i> untuk <i>root</i> dapat menjangkau node sensor terjauh	27
3.4	Diagram kelas untuk pemrograman antarmuka	29
3.5	Diagram kelas untuk pemrograman node sensor <i>root</i>	29
3.6	Diagram kelas untuk pemrograman node sensor biasa	30
3.7	Diagram <i>use case</i> untuk aplikasi sinkronisasi waktu yang dibangun	30
4.1	<i>Sequence diagram</i> untuk fitur mulai dan berhenti sinkronisasi	35
4.2	<i>Sequence Diagram</i> untuk fitur keluar dari program	36
4.3	Kelas <i>CMD</i>	36
4.4	Kelas <i>Root</i>	37
4.5	Kelas <i>SensorNode</i>	39
5.1	Tampilan awal antarmuka	42
5.2	Tampilan saat menjalankan fitur "1"	43
5.3	Tampilan saat menjalankan fitur "2"	43
5.4	Tampilan saat menjalankan fitur "0"	44

5.5	Contoh tampilan node sensor pada antarmuka	48
5.6	Grafik hasil akhir rata-rata selisih waktu untuk eksperimen ke-1 sampai 5	54
5.7	Grafik hasil selisih waktu untuk 20 kali sinkronisasi pada eksperimen ke-6	54
5.8	Grafik perbandingan akurasi waktu dengan menggunakan 2 referensi	55
5.9	Grafik perbandingan akurasi waktu dengan menggunakan 5 referensi	56
5.10	Grafik perbandingan akurasi waktu dengan menggunakan 10 referensi	56
5.11	Grafik perbandingan akurasi waktu dengan menggunakan 15 referensi	56
5.12	Grafik perbandingan akurasi waktu dengan menggunakan 25 referensi	57
5.13	Grafik perbandingan akurasi waktu pada WSN area sempit	57
5.14	Grafik perbandingan akurasi waktu pada WSN area luas	58
B.1	Hasil sinkronisasi waktu dengan 3 <i>reference point</i>	71
B.2	Tampilan ke-1 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-1	72
B.3	Tampilan ke-2 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-1	73
B.4	Tampilan ke-3 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-1	74
B.5	Tampilan ke-1 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-2	75
B.6	Tampilan ke-2 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-2	76
B.7	Tampilan ke-3 node-node sensor pada <i>console</i> untuk hasil eksperimen ke-2	77
B.8	Tampilan ke-1 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-3	78
B.9	Tampilan ke-2 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-3	78
B.10	Tampilan ke-1 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-3	79
B.11	Tampilan ke-2 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-3	79
B.12	Tampilan ke-1 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-4	80
B.13	Tampilan ke-2 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-4	80
B.14	Tampilan ke-1 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-4	81
B.15	Tampilan ke-2 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-4	81
B.16	Tampilan ke-1 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-5	82
B.17	Tampilan ke-2 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-5	82
B.18	Tampilan ke-3 node sensor AFAP dan BEBA pada <i>console</i> untuk hasil eksperimen ke-5	83
B.19	Tampilan ke-1 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-5	83
B.20	Tampilan ke-2 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-5	84
B.21	Tampilan ke-3 node sensor AFPE dan EFEP pada <i>console</i> untuk hasil eksperimen ke-5	84
B.22	Tampilan ke-1 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	85
B.23	Tampilan ke-2 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	85
B.24	Tampilan ke-3 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	86
B.25	Tampilan ke-4 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	86
B.26	Tampilan ke-5 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	87
B.27	Tampilan ke-6 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	87

B.28 Tampilan ke-7 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	88
B.29 Tampilan ke-8 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	88
B.30 Tampilan ke-9 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	89
B.31 Tampilan ke-10 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	89
B.32 Tampilan ke-11 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	90
B.33 Tampilan ke-12 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	90
B.34 Tampilan ke-13 node sensor pada <i>console</i> untuk hasil eksperimen ke-6	91
B.35 Tampilan node sensor AFAF pada <i>console</i> untuk hasil eksperimen ke-7	91
B.36 Tampilan node sensor EFEF pada <i>console</i> untuk hasil eksperimen ke-7	92
B.37 Tampilan node sensor BEBA pada <i>console</i> untuk hasil eksperimen ke-7	92
B.38 Tampilan node sensor BABE pada <i>console</i> untuk hasil eksperimen ke-7	93
B.39 Tampilan node sensor AFFE pada <i>console</i> untuk hasil eksperimen ke-7	93
B.40 Tampilan node sensor AFAF pada <i>console</i> untuk hasil eksperimen ke-8	94
B.41 Tampilan node sensor AFFE pada <i>console</i> untuk hasil eksperimen ke-8	94
B.42 Tampilan node sensor EFEF pada <i>console</i> untuk hasil eksperimen ke-8	95

DAFTAR TABEL

2.1	Tabel sistem operasi beserta kemampuan masing-masing	7
3.1	Skenario mulai sinkronisasi	31
3.2	Skenario berhenti sinkronisasi	31
3.3	Skenario keluar dari program	31
5.1	Spesifikasi perangkat keras	41
5.2	Spesifikasi node sensor	41
5.3	Spesifikasi perangkat lunak	41
5.4	Contoh isi tabel referensi	47
5.5	Tabel pengujian untuk 2 <i>reference point</i>	49
5.6	Tabel pengujian untuk 5 <i>reference point</i>	50
5.7	Tabel pengujian untuk 10 <i>reference point</i>	50
5.8	Tabel pengujian untuk 15 <i>reference point</i>	51
5.9	Tabel pengujian untuk 25 <i>reference point</i>	51
5.10	Tabel pengujian untuk 20 kali sinkronisasi	52
5.11	Tabel pengujian untuk WSN area luas dengan komunikasi minimum sebanyak 2 <i>hop</i> dari <i>root</i> menuju node sensor terjauh.	53
5.12	Tabel pengujian untuk WSN area luas dengan komunikasi minimum sebanyak 3 <i>hop</i> dari <i>root</i> menuju node sensor terjauh.	53

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jaringan Sensor Nirkabel atau *Wireless Sensor Network* (WSN) merupakan sebuah jaringan yang terdiri atas node-node sensor yang saling berhubungan. Node sensor dalam WSN merupakan sebuah perangkat yang mengandung sensor, perangkat komputasi, pengirim dan penerima sinyal, serta komponen daya. Sensor dalam node sensor dapat melakukan *sensing* dan menangkap berbagai data, seperti getaran, kelembaban, suhu, tekanan udara, dan lainnya. Informasi-informasi tersebut kemudian dapat diproses dan dikirim ke perangkat utama yang mengatur WSN.

Meskipun node sensor memiliki macam-macam kemampuan, pada umumnya node sensor tidak dapat menyimpan informasi mengenai waktu. Padahal, waktu merupakan atribut yang sangat dibutuhkan untuk aplikasi-aplikasi yang berhubungan dengan WSN. Waktu yang dimiliki oleh node-node dalam WSN itu pun harus tersinkronisasi. Contoh kasus yang menunjukkan pentingnya waktu yang sinkron adalah aplikasi WSN dalam bidang perkebunan. Aplikasi berfungsi untuk mengatur jadwal pembukaan dan penutupan *sprinkler* secara otomatis. Jika waktu pada node-node sensor yang bersangkutan tidak tersinkronisasi, maka untuk tanaman yang sama di tempat yang berbeda, jadwal penyiramannya dapat berbeda-beda juga. Padahal, seharusnya tanaman itu disiram secara bersamaan.

Node sensor membutuhkan usaha yang besar untuk melakukan sinkronisasi waktu, sementara sumber energi dan tempat penyimpanan node sensor sangat terbatas. Oleh karena itu, diperlukan sebuah algoritma yang dapat melakukan sinkronisasi waktu di WSN dengan pemakaian energi dan memori yang minimal. Terdapat beberapa algoritma yang telah dikembangkan, seperti *Flooding Time Synchronization Protocol* (FTSP), *Reference Broadcast Synchronization* (RBS), *Timing-sync Protocol for Sensor Networks* (TPSN), dan algoritma lainnya. Setiap algoritma memiliki kelebihan dan kekurangan masing-masing. Algoritma RBS tidak efisien bila digunakan di area jaringan yang terlalu luas. Algoritma TPSN tidak memiliki tingkat akurasi waktu yang tinggi jika terdapat cukup banyak node sensor di WSN.

Pada penelitian ini, algoritma yang akan digunakan adalah *Flooding Time Synchronization Protocol*. Algoritma ini sudah menangani jeda waktu yang timbul dengan menggunakan banyak *time-stamp* dalam perhitungannya. Selain itu, algoritma ini dapat digunakan dalam berbagai jenis topologi, seperti topologi bintang, jala, serta topologi lainnya. Algoritma ini menggunakan satu buah node sensor yang akan menjadi sumber penyedia waktu bagi node sensor lainnya dalam jaringan. Node sumber tersebut akan menyebarkan pesan sinkronisasi kepada node-node sensor lainnya yang dapat ia jangkau. Node sensor yang tidak terjangkau oleh node sumber akan mendapatkan pesan sinkronisasi melalui node sensor lainnya yang telah tersinkronisasi.

1.2 Rumusan Masalah

Adapun rumusan masalah dari deskripsi yang telah dipaparkan adalah:

- Bagaimana cara membangun perangkat lunak yang dapat melakukan sinkronisasi waktu di WSN?

- Bagaimana cara menerapkan algoritma *Flooding Time Synchronization Protocol* dalam perangkat lunak sinkronisasi waktu di WSN?
- Bagaimana permasalahan pada performansi algoritma *Flooding Time Synchronization Protocol* bila dibandingkan dengan algoritma *Reference Broadcast Synchronization* dalam melakukan sinkronisasi waktu di WSN?

1.3 Tujuan

Adapun tujuan penelitian dari rumusan masalah yang telah dipaparkan adalah:

- Membangun perangkat lunak yang dapat melakukan sinkronisasi waktu di WSN.
- Menerapkan algoritma *Flooding Time Synchronization Protocol* dalam perangkat lunak sinkronisasi waktu di WSN.
- Membandingkan hasil analisis performa algoritma *Flooding Time Synchronization Protocol* bila dibandingkan dengan *Reference Broadcast Synchronization* dalam melakukan sinkronisasi waktu di WSN.

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah aplikasi hanya menangani *delay* yang timbul di lapisan atas jaringan, hanya mengestimasi *clock offset*, pengiriman pesan bersifat tidak *reliable*, serta node sensor yang akan menjadi *root* telah ditetapkan sejak awal.

1.5 Metodologi

Langkah-langkah yang dilakukan dalam skripsi ini adalah:

1. Melakukan studi literatur mengenai karakteristik WSN.
2. Melakukan studi literatur mengenai cara kerja algoritma *Flooding Time Synchronization Protocol* dalam WSN.
3. Mempelajari *library* WSN pada Java.
4. Mempelajari cara menggunakan *platform* Eclipse.
5. Mengimplementasikan algoritma *Flooding Time Synchronization Protocol* yang telah dipelajari dengan menggunakan *platform* Eclipse.
6. Melakukan pengujian terhadap algoritma yang telah diimplementasikan ke dalam perangkat lunak sinkronisasi waktu pada WSN.
7. Menganalisis hasil performa algoritma *Flooding Time Synchronization Protocol* bila dibandingkan dengan algoritma *Reference Broadcast Synchronization*.

1.6 Sistematika Pembahasan

Dokumen skripsi ini terdiri atas enam bab dengan sistematika sebagai berikut:

Bab 1 membahas latar belakang diperlukannya sinkronisasi waktu dalam WSN, diikuti dengan penjelasan singkat mengenai algoritma *Flooding Time Synchronization Protocol*. Selain itu, dibahas

juga mengenai rumusan masalah, tujuan penelitian, pembatasan masalah yang akan diselesaikan, serta metode penelitian yang dilakukan.

Bab 2 menerangkan teori-teori yang akan digunakan dalam penelitian ini, seperti pengertian *Wireless Sensor Network* (WSN), node sensor, sistem operasi pada WSN, protokol *stack*, protokol komunikasi, topologi jaringan, arsitektur jaringan, jenis-jenis *delay* yang muncul, penerapan WSN dalam dunia nyata, sinkronisasi waktu pada WSN, serta penjelasan mengenai algoritma *Flooding Time Synchronization Protocol*.

Bab 3 menganalisis masalah yang ingin diselesaikan dan membahas secara umum modul-modul yang dibutuhkan oleh perangkat lunak. Analisis dimulai dengan menentukan spesifikasi WSN yang akan digunakan, lalu diikuti dengan penjelasan umum mengenai cara algoritma dalam perangkat lunak dapat mengatasi masalah tersebut. Bab ini pun akan menganalisis diagram kelas secara umum, diagram *use case*, serta skenario pengguna yang dapat terjadi.

Bab 4 berisi perancangan aplikasi sinkronisasi waktu yang dibangun. Perancangan yang dibahas terdiri atas perancangan masukan dan keluaran, perancangan format pesan sinkronisasi, perancangan interaksi antar node sensor, serta perancangan detail diagram kelas.

Bab 5 berisi hasil implementasi dan pengujian terhadap aplikasi sinkronisasi waktu yang dibangun, serta perbandingan hasil analisis performansi antara algoritma *Flooding Time Synchronization Protocol* dengan *Reference Broadcast Synchronization*. Hasil implementasi yang dibahas mencakup lingkungan implementasi dan hasil dari implementasi itu sendiri. Bagian pengujian membahas hasil dan kesimpulan yang didapat setelah melakukan pengujian fungsional dan eksperimental. Bagian perbandingan algoritma berisi hasil perbandingan analisis performansi yang dihasilkan oleh kedua algoritma tersebut.

Bab 6 berisi kesimpulan dan saran berdasarkan pengalaman membangun aplikasi sinkronisasi waktu dengan algoritma *Flooding Time Synchronization Protocol* pada skripsi ini.