



SISTEM PEMANTAUAN DAYA PADA WIRELESS SENSOR NETWORK MENGGUNAKAN ALGORITMA PRIORITY SCHEDULING

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagai persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Ihksan Sahib Latif

NIM: 135150307111011



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**

2020

PENGESAHAN

SISTEM PEMANTAUAN DAYA PADA *WIRELESS SENSOR NETWORK* MENGUNAKAN ALGORITMA *PRIORITY SCHEDULING*

SKRIPSI

Keminatan Teknik Komputer

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

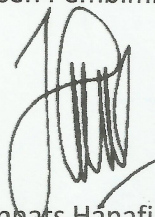
Disusun Oleh :

Muhammad Ihksan Sahib Latif
NIM: 135150307111011

Skripsi ini telah diuji dan dinyatakan lulus pada
22 Juli 2020

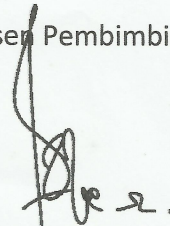
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Mochammad Hannats Hanafi Ichsan, S.ST., M. T.
NIP: 1988 122920 190310 10

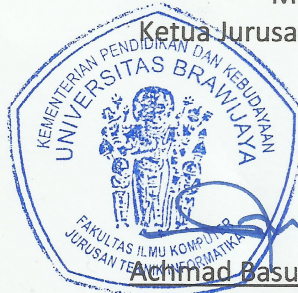
Dosen Pembimbing 2



Agung Setia Budi, S.T., M.T., Ph.D
NIK: 201304 870423 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Achmad Basuki, S.T., M.MG., Ph.D.
NIP: 197411182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 Juli 2020



Muhammad Ihksan Sahib Latif

NIM: 135150307111011

PRAKATA

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Kuasa, karena atas limpahan rahmat dan karunia-Nya maka penulis dapat menyelesaikan laporan tugas akhir pada skripsi ini. Adapun maksud penyelesaian skripsi ini adalah untuk memenuhi persyaratan dalam menempuh ujian Sarjana Fakultas Ilmu Komputer, Universitas Brawijaya Malang. Judul skripsi yang dikerjakan oleh penulis adalah “Sistem Pemantauan Daya Pada *Wireless Sensor Network* Menggunakan Algoritma *Priority Scheduling*”.

Penulis ingin mengucapkan terima kasih dalam proses pengerjaan skripsi dan penyusunan laporan skripsi ini tidak akan terwujud tanpa adanya bantuan dan dorongan dari berbagai pihak. Oleh karena itu, maka pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua serta keluarga penulis yang sudah memberikan doa dan dukungan kepada penulis.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Achmad Basuki, S.T., M.MG., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Mochammad Hannats Hanafi Ichsan, S. ST., M. T. selaku dosen pembimbing I skripsi ini.
5. Bapak Agung Setia Budi, S. T., M. T., Ph.D selaku dosen pembimbing II skripsi ini.
6. Seluruh teman-teman tercinta yang telah membantu proses penyelesaian skripsi ini yang mana tidak dapat disebutkan satu persatu.

Akhir kata, penulis berharap semoga laporan skripsi ini dapat bermanfaat bagi seluruh pihak dan dapat menjadi acuan untuk pengembangan penelitian berikutnya.

Malang, 22 Juli 2020

Muhammad Ihksan Sahib Latif
Ihksanmuhammad23@gmail.com

ABSTRAK

Muhammad Ihksan Sahib Latif, Sistem Pemantuan Daya Pada Wireless Sensor Network Menggunakan Algoritma Priority Scheduling

Pembimbing: Mochammad Hannats Hanafi Ichsan, S. ST., M. T. dan Agung Setia Budi, S. T., M. T., Ph.D

Wireless Sensor Network adalah sebuah teknologi nirkabel yang diperlukan untuk keperluan pemantauan kondisi lingkungan sekitar, yang terdiri dari beberapa *sensor node* yang dapat berkomunikasi dan memproses informasi satu sama lain. Permasalahan pada WSN mengenai konsumsi daya yang mengakibatkan daya pada node hanya menggunakan baterai untuk operasinya sehingga mengakibatkan cadangan energi daya yang terbatas. Pada saat satu node mati, maka akan mengakibatkan perubahan peformansi dan akan muncul kendala jika harus melakukan konservasi energi daya berulang-ulang dan dapat mengurangi keuntungan dari WSN. Solusi masalah tersebut penulis mengimplementasikan algoritma *Priority Scheduling* dimana algoritma tersebut akan ditempatkan pada masing – masing *node* yang akan memprioritaskan urutan daya berdasarkan daya terbesar hingga ke terkecil, fungsi dari NRF24L01 digunakan untuk mengirimkan dan menerima data. Untuk mekanisme monitoring daya pada setiap *node* tersedia *mikrokontroler* Arduino uno R3, sensor ACS712-30A, sensor tegangan, sensor DHT11, dan RTC DS1307 yang berfungsi untuk menampilkan waktu hasil data secara *real Time*. Berdasarkan hasil pengujian yang telah dilakukan, didapatkan akurasi pengukuran DHT11 sebesar 98.52% hingga 99.26%, akurasi pengukuran tegangan sebesar 99.6%, dan akurasi pengukuran arus sebesar 94.86% hingga 97.86%. Untuk peforma penerimaan data yang didapatkan membutuhkan waktu rata-rata sebesar 2 hingga 3 detik. Sedangkan pengambilan data daya menggunakan *Priority Scheduling* pada ruangan ada halangan dan tidak ada halangan memiliki persentase keberhasilan 100%. Persentase keberhasilan didapatkan dengan melihat hasil data daya yang ditampilkan pada 10 kali percobaan pada ruangan yang berbeda dan urutannya sesuai dari data daya terbesar hingga ke terkecil.

Kata kunci: *Priority Scheduling*, WSN, Sensor, NRF24L01, Monitoring Daya, Arduino Uno R3

ABSTRACT

Muhammad Ihksan Sahib Latif, Power Monitoring System on Wireless Sensor Network Using Priority Scheduling Algorithm

Supervisors: Mochammad Hannats Hanafi Ichsan, S. ST., M.T. and Agung Setia Budi, S.T., M. T., Ph.D

Wireless Sensor Network is a wireless technology that is needed to monitor the condition of the surrounding environment, which consists of several sensor nodes that can communicate and process information with each other. Problems with WSN regarding power consumption which results in power at the node using only the battery for its operation resulting in limited energy reserves. When a node dies, it will result in changes in performance and obstacles will arise if you have to do energy conservation power over and over again and can reduce the benefits of WSN. The solution to the problem is the writer implements the Priority Scheduling algorithm where the algorithm will be placed on each node which will prioritize the order of power based on the largest to the smallest power, the function of NRF24L01 is used to send and receive data. For the power monitoring mechanism at each node, there is an Arduino uno R3 microcontroller, ACS712-30A sensor, Voltage sensor, DHT11 sensor, and RTC DS1307 which functions to display the Time of data results in real-Time. Based on the results of tests that have been done, the accuracy of DHT11 measurements is 98.52% to 99.26%, the Voltage measurement accuracy is 99.6%, and the current measurement accuracy is 94.86% to 97.86%. For the performance of receiving data, it takes an average of 2 to 3 seconds. While taking power data using Priority Scheduling in the room there are obstacles and no obstacles have a 100% success rate. Percentage of success is obtained by looking at the results of the power data that is displayed on 10 trials in different rooms and order according to the largest to the smallest power data.

Keywords: *Priority Scheduling, Wireless Sensor Network, Sensor, NRF24L01, Power Monitoring, Arduino Uno R3*

DAFTAR ISI

PERSETUJUAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Algoritma <i>Priority Scheduling</i>	6
2.3 <i>Wireless Sensor Network</i>	7
2.3.1 Arsitektur <i>Wireless Sensor Network</i>	8
2.3.2 Bagian-Bagian Pada <i>Wireless Sensor Network</i>	9
2.4 <i>Power Monitoring System</i>	9
2.5 Sensor Arus ACS712-30A.....	10
2.6 Sensor Tegangan.....	10
2.7 Daya.....	11
2.7.1 Daya Aktif.....	11
2.7.2 Daya Reaktif.....	12
BAB 3 METODOLOGI.....	13
3.1 Metodologi Penelitian.....	13
3.2 Studi Literatur.....	14

3.3 Analisis Kebutuhan Sistem.....	14
3.3.1 Kebutuhan Perangkat Keras.....	14
3.3.2 Kebutuhan Perangkat Lunak.....	14
3.4 Perancangan Sistem.....	14
3.5 Implementasi Sistem.....	15
3.6 Pengujian Sistem.....	16
3.7 Kesimpulan dan Saran.....	16
BAB 4 ANALISIS KEBUTUHAN.....	17
4.1 Kebutuhan Pengguna.....	17
4.2 Kebutuhan Sistem.....	17
4.2.1 Kebutuhan Perangkat Keras.....	17
4.2.2 Kebutuhan Perangkat Lunak.....	22
4.3 Kebutuhan Fungsional.....	23
4.4 Kebutuhan Non Fungsional.....	23
4.4.1 Karakteristik Pengguna.....	24
4.4.2 Lingkungan Pengguna.....	24
4.4.3 Asumsi dan Ketergantungan.....	24
4.4.4 Batasan Desain Sistem.....	24
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	26
5.1 Perancangan Sistem.....	26
5.1.1 Gambaran Umum Sistem.....	26
5.1.2 Perancangan Modul <i>Master</i>	28
5.1.3 Perancangan Modul <i>Slave 1, Slave 2 dan Slave 3</i>	36
5.1.4 Perancangan Mekanisme Algoritma <i>Priority Scheduling</i>	41
5.1.5 Perancangan Pin Modul.....	37
5.2 Implementasi Sistem.....	43
5.2.1 Implementasi Sistem Pada Modul <i>Master Node</i>	43
5.2.2 Urutan Berjalannya Modul <i>Master Node</i>	46
5.2.3 Implementasi Sistem Pada Modul <i>Slave 1 Node, Slave 2</i> <i>dan Slave 3</i>	49
5.2.4 Urutan Berjalannya Modul <i>Slave 1 Node, Slave 2 dan</i> <i>Slave 3</i>	50



BAB 6 PENGUJIAN DAN ANALISIS.....	53
6.1 Pengujian Hasil <i>Sensing</i>	53
6.1.1 Tujuan.....	53
6.1.2 Peralatan	53
6.1.3 Langkah Pengujian	53
6.1.4 Hasil.....	57
6.1.5 Analisis.....	60
6.2 Pengujian Pengiriman Data	61
6.2.1 Tujuan.....	61
6.2.2 Peralatan.....	61
6.2.3 Langkah Pengujian	61
6.2.4 Hasil.....	62
6.2.4.1 Hasil Pengujian Ada Halangan.....	62
6.2.4.2 Hasil Pengujian Tidak Ada Halangan.....	67
6.2.5 Analisis.....	72
BAB 7 PENUTUP	74
7.1 Kesimpulan.....	74
7.2 Saran.....	75
DAFTAR REFERENSI	76



DAFTAR TABEL

Tabel 3.1 Kebutuhan Perangkat Keras.....	14
Tabel 3.2 Kebutuhan Perangkat Lunak.....	14
Tabel 4.1 Spesifikasi Arduino Uno R3.....	19
Tabel 4.2 Spesifikasi Modul RTC DS1307	20
Tabel 4.3 Spesifikasi DHT11	21
Tabel 4.4 Spesifikasi Sensor ACS712-30A.....	21
Tabel 5.1 Keterangan Pin NRF24L01 Pada <i>Master</i>	30
Tabel 5.2 Keterangan Pin Modul RTC Pada <i>Master</i>	31
Tabel 5.3 Keterangan Pin Sensor DHT11 Pada <i>Master</i>	31
Tabel 5.4 Keterangan Pin Sensor ACS712-30A Pada <i>Master</i>	32
Tabel 5.5 Keterangan Pin Sensor Tegangan Pada <i>Master</i>	32
Tabel 5.6 Keterangan Pin NRF24L01 Pada Setiap <i>Slave</i>	37
Tabel 5.7 Keterangan Pin Sensor DHT11 Pada Setiap <i>Slave</i>	38
Tabel 5.8 Keterangan Pin Sensor ACS712-30A Pada Setiap <i>Slave</i>	38
Tabel 5.9 Keterangan Pin Sensor Tegangan Pada Setiap <i>Slave</i>	39
Tabel 5.10 Potongan Kode Sumber Inisialisasi <i>Input</i> dan <i>Output</i> Modul <i>Node Master</i>	44
Tabel 5.11 Potongan Kode Sumber <i>Master</i> Mengirimkan <i>Request</i> Ke <i>Slave</i>	46
Tabel 5.12 Potongan Kode Sumber <i>Master</i> Mengecek <i>Request</i> Data	46
Tabel 5.13 Potongan Kode Sumber Baca Data <i>Slave</i>	47
Tabel 5.14 Potongan Kode Sumber Memasukkan Data <i>Master</i> Dan <i>Slave</i>	48
Tabel 5.15 Potongan Kode Sumber Mengumpulkan Semua Data.....	48
Tabel 5.16 Potongan Kode Sumber Menampilkan Data Berdasarkan Prioritas ...	49
Tabel 5.17 Potongan Kode Sumber Inisialisasi <i>Input</i> Dan <i>output</i> <i>Node Slave</i>	50
Tabel 5.18 Potongan Kode Sumber Mengecek <i>Request</i> Data <i>Slave</i>	51
Tabel 5.19 Potongan Kode Sumber <i>Request</i> Data Dan Mengambil Data	51
Tabel 5.20 Potongan Kode Sumber Mengirimkan Data Ke <i>Master</i>	52
Tabel 6.1 Hasil Pengujian Pengambilan Data Akurasi Suhu <i>Master</i>	57
Tabel 6.2 Hasil Pengujian Pengambilan Data Akurasi Suhu Setiap <i>Slave</i>	58



Tabel 6.3 Hasil Pengujian Pengambilan Data Tegangan <i>Master</i>	58
Tabel 6.4 Hasil Pengujian Pengambilan Data Tegangan Setiap <i>Slave</i>	59
Tabel 6.5 Hasil Pengujian Pengambilan Data Arus <i>Master</i>	59
Tabel 6.6 Hasil Pengujian Pengambilan Data Arus Setiap <i>Node</i>	60
Tabel 6.7 Hasil Pengujian Ke-1 dengan jarak 1 meter ada halangan.....	62
Tabel 6.8 Hasil Pengujian Ke-2 dengan jarak 2 meter ada halangan.....	59
Tabel 6.9 Hasil Pengujian Ke-3 dengan jarak 3 meter ada halangan.....	59
Tabel 6.10 Hasil Pengujian Ke-4 dengan jarak 4 meter ada halangan.....	63
Tabel 6.11 Hasil Pengujian Ke-5 dengan jarak 5 meter ada halangan.....	63
Tabel 6.12 Hasil Pengujian Ke-6 dengan jarak 6 meter ada halangan.....	63
Tabel 6.13 Hasil Pengujian Ke-7 dengan jarak 7 meter ada halangan.....	64
Tabel 6.14 Hasil Pengujian Ke-8 dengan jarak 8 meter ada halangan.....	64
Tabel 6.15 Hasil Pengujian Ke-9 dengan jarak 9 meter ada halangan.....	64
Tabel 6.16 Hasil Pengujian Ke-10 dengan jarak 10 meter ada halangan.....	64
Tabel 6.17 Pengujian Daya <i>Priority Scheduling</i> dengan jarak 1-10meter ada halangan.....	65
Tabel 6.18 Hasil Pengujian Ke-1 dengan jarak 1 meter tidak ada halangan.....	67
Tabel 6.19 Hasil Pengujian Ke-2 dengan jarak 2 meter tidak ada halangan.....	68
Tabel 6.20 Hasil Pengujian Ke-3 dengan jarak 3 meter tidak ada halangan.....	68
Tabel 6.21 Hasil Pengujian Ke-4 dengan jarak 4 meter tidak ada halangan.....	68
Tabel 6.22 Hasil Pengujian Ke-5 dengan jarak 5 meter tidak ada halangan.....	68
Tabel 6.23 Hasil Pengujian Ke-6 dengan jarak 6 meter tidak ada halangan.....	68
Tabel 6.24 Hasil Pengujian Ke-7 dengan jarak 7 meter tidak ada halangan.....	69
Tabel 6.25 Hasil Pengujian Ke-8 dengan jarak 8 meter tidak ada halangan.....	69
Tabel 6.26 Hasil Pengujian Ke-9 dengan jarak 9 meter tidak ada halangan.....	69
Tabel 6.27 Hasil Pengujian Ke-10 dengan jarak 10 meter tidak ada halangan.....	69
Tabel 6.28 Pengujian Daya <i>Priority Scheduling</i> dengan jarak 1-10meter ada halangan.....	70

DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Wireless Sensor Network</i>	7
Gambar 2.2 Arsitektur <i>Wireless Sensor Network</i> Secara Umum.....	8
Gambar 2.3 <i>Hall Effect</i> Pada ACS712-30A.....	10
Gambar 2.4 Rangkaian Sensor Tegangan.....	11
Gambar 3.1 Diagram Alir Metode Penelitian.....	13
Gambar 3.2 Gambaran Umum Perancangan.....	15
Gambar 4.1 Diagram Analisis Kebutuhan.....	17
Gambar 4.2 Laptop Acer Aspire E1-451G.....	18
Gambar 4.3 Arduino Uno R3.....	18
Gambar 4.4 NRF24L01.....	19
Gambar 4.5 RTC DS1307.....	20
Gambar 4.6 DHT11.....	20
Gambar 4.7 Sensor ACS712-30A.....	21
Gambar 4.8 Rangkaian Sensor Tegangan.....	22
Gambar 4.9 Kebutuhan Perangkat Lunak.....	22
Gambar 5.1 Diagram Blok Sistem.....	28
Gambar 5.2 Skematik Diagram Modul <i>Master</i>	29
Gambar 5.3 Flowchart Diagram Modul <i>Master</i>	34
Gambar 5.4 Skematik Diagram Modul <i>Node Slave 1, Slave 2 dan Slave 3</i>	36
Gambar 5.5 <i>Flowchart Diagram</i> Modul <i>Node Slave 1, Slave 2 dan Slave 3</i>	40
Gambar 5.6 <i>Flowchart</i> Algoritma <i>Priority Scheduling</i>	43
Gambar 5.7 Implementasi Sistem Pada Modul <i>Node Master</i>	44
Gambar 5.8 Implementasi Perangkat Keras Modul <i>Node Slave 1, Slave 2 dan Slave 3</i>	49
Gambar 6.1 <i>Verify Slave 1</i>	54
Gambar 6.2 <i>Verify Slave 2</i>	54
Gambar 6.3 <i>Verify Slave 3</i>	55
Gambar 6.4 <i>Verify Master</i>	55
Gambar 6.5 Upload Program.....	56



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Wireless Sensor Network (WSN) adalah sebuah teknologi nirkabel yang diperlukan untuk keperluan pemantauan kondisi lingkungan sekitar, yang terdiri dari beberapa *sensor node* yang dapat saling berkomunikasi dan memproses informasi satu sama lain (Nikolic, 2014). *Wireless Sensor Network* merupakan teknologi yang menggabungkan sistem tertanam dan jaringan komunikasi yang menjadikannya lebih efisien dan efektif. (Prihtiadi H, 2017). Sumber daya utama pada WSN adalah baterai yang memiliki jumlah energi terbatas dan WSN harus memiliki *lifetime* jaringan yang panjang untuk memenuhi persyaratan dalam pengukuran. Dalam beberapa kasus, dibutuhkan *lifetime* jaringan hingga beberapa bulan bahkan tahun (Anastasi dkk, 2008). Oleh karena itu, efisiensi penggunaan energi menjadi isu utama pada WSN (Chang dkk, 2012).

Dari beberapa masalah dalam implementasi WSN adalah mengenai konsumsi daya yang mengakibatkan daya pada node hanya menggunakan baterai untuk operasinya sehingga mengakibatkan cadangan energi daya yang terbatas. Pada saat satu node mati, maka akan mengakibatkan perubahan performansi dan akan muncul kendala jika harus melakukan konservasi energi daya berulang-ulang atau akan meningkatkan biaya. Konsumsi energi daya merupakan faktor penting dalam menentukan *lifetime* suatu jaringan, maka energi daya yang digunakan harus seefisien mungkin agar menghasilkan performa yang maksimum. Untuk mengatasi masalah tersebut diusulkan penelitian yang bertujuan dalam sistem pemantauan daya pada *Wireless Sensor Network* (Tama, 2010).

Penelitian sebelumnya yang dilakukan oleh Dinata (2015) telah melakukan monitoring daya untuk menggantikan sistem pengukuran energi listrik secara manual dan konvensional. Perangkat ini terdiri 4 (empat) bagian yaitu sensor, processor, display dan network. Bagian sensor terdiri dari Current Transformer dan AC to AC Power Adapter. Processor yang digunakan adalah Arduino UNO, display yang digunakan adalah Liquid Crystal Display (LCD) tipe untuk menampilkan data keluaran real time. Pada bagian terakhir yaitu network terdiri dari *Ethernet Shield*, 3G Router, 3G Modem untuk komunikasi ke *Database Server* sebagai tempat penyimpanan tetap dan pengolahan data lebih lanjut, namun dalam proses penerimaan data terdapat delay yang tidak beraturan. Dalam penelitian ini menjelaskan bahwa sistem tersebut hanya dibuat dalam lingkup terbatas yaitu dibuat hanya untuk memonitoring 1 node saja serta dalam pengiriman data masih terdapat delay yang tidak beraturan, dalam kasus ini semua tergantung dalam pembuatan minimum sistem beserta algoritma pengiriman data yang ditentukan, serta dalam pemilihan beberapa sensor harus tepat, bila dalam pembuatan sistem tidak terorganisir dengan baik maka akan berakibat fatal dan bisa memakan daya besar.

Pada penelitian berikutnya yang dilakukan oleh Karthikeyan (2014), telah mengimplementasikan algoritma *First Come First Served* untuk penjadwalan



berbagai jenis paket data, seperti waktu nyata dan tidak *real time* pada paket data di *node* sensor dengan sumber daya kendala dalam jaringan sensor nirkabel yang sebagian besar mekanisme penjadwalan paket yang ada dari *Wireless Sensor Network*.

Dalam penelitian ini algoritma yang akan diimplementasikan dalam sistem adalah algoritma *Priority Scheduling* dengan kategori *non-preemptive*, dikarenakan algoritma ini dapat mendahulukan proses data yang memiliki prioritas tertinggi. Prioritas pada proses dapat ditentukan berdasarkan tingkat kepentingan proses dan tiap-tiap proses *Priority Scheduling* dilengkapi dengan nomor prioritas. Data yang dimaksud adalah hasil sensor pada data pengujian yang telah dikerjakan di daerah berbagai tempat (Jovanovic dkk, 2011).

Multi sensor merupakan salah satu poin utama yang menjadi alasan dalam pengerjaan penelitian ini, karena pada penelitian sebelumnya hanya memiliki 1 *node*. Kebutuhan *multi sensor* bertujuan untuk kebutuhan akuisisi data, pemantauan daya dan pada penelitian ini *multi sensor* di implementasikan pada 4 *node* (*Master, Slave 1, Slave 2, dan Slave 3 Node*).

Berdasarkan masalah yang ada, pada penelitian ini akan dirancang sebuah "Sistem Pemantauan Daya Pada *Wireless Sensor Network* Menggunakan Algoritma *Priority Scheduling*". Terdapat beberapa modul yang akan di implementasikan yaitu modul NRF24L01 untuk komunikasi data secara *wireless* yang berguna untuk pengiriman data antar *slave* dan *master*. Modul berikutnya adalah Arduino Uno R3 merupakan mikrokontroler yang bekerja dengan metode *blocking* dan tidak digunakan untuk *multitask* karena *Priority Scheduling non-preemptive* dapat diset secara internal atau peneliti yang memutuskan apa saja yang diperlukan dalam memonitoring pada suatu sistem secara terjadwal. Selanjutnya terdapat modul sensor yang digunakan yaitu sensor tegangan, modul sensor arus adalah ACS712-30A. Peneliti menggunakan sensor arus dan tegangan berguna untuk mengetahui besaran daya pada masing-masing modul. Sensor DHT 11 adalah sensor kelembaban dan suhu. Modul RTC DS1307 merupakan *realtime clock module* digunakan untuk penyimpanan waktu secara nyata. USB ke *serial port* digunakan untuk pengiriman data *serial* dari *master node* menuju PC/Laptop. Selain itu peneliti akan melakukan pemantauan melalui laptop dengan perantara *serial monitor*.

1.2 Rumusan Masalah

1. Bagaimana hasil data pada masing-masing *node* sensor tegangan, sensor arus ACS712-30A, sensor DHT11 dapat menghasilkan hasil performa akurasi *sensing* yang sesuai dengan *Avometer* dan termometer?
2. Bagaimana mengimplementasikan algoritma *Priority Scheduling* pada keempat *node* yang dapat menghasilkan dan menampilkan data daya secara berurutan dari daya terbesar hingga ke terkecil?
3. Bagaimana hasil keluaran data daya pada *master node, slave 1 node, slave 2 node, dan slave 3 node*?



4. Bagaimana peforma sinkronisasi waktu transmisi data pada keempat sensor *node*?

1.3 Tujuan

Penelitian ini memiliki tujuan diantaranya:

1. WSN dapat digunakan untuk monitoring hasil data yang mampu memiliki sistem yang *minimum* sehingga dapat berguna secara optimal.
2. *Transmitter* dan *receiver* akan berjalan secara optimal dengan mengimplementasikan algoritma *Priority Scheduling* dan juga menghindari terjadinya *error* dalam pengiriman data pada setiap *node*.
3. Jika setiap *node* akan ditempatkan pada situasi yang tidak sama, berapa % keberhasilan yang akan didapatkan tingkat akurasi dan *transmisi* pengiriman data.
4. Tingkat akurasi setiap sensor dan monitoring data daya memiliki keberhasilan yang dapat diharapkan.

1.4 Manfaat

Algoritma *Priority Scheduling* berguna untuk *monitoring power* sistem yang akan digunakan secara terjadwal dan secara *real Time* dalam mengakuisisi berbagai macam data pada setiap *node* dalam ruang kerja WSN dalam menangani pemborosan daya yang digunakan setiap *node*.

1.5 Batasan Masalah

Ruang lingkup pada penelitian ini memiliki berbagai macam batasan dalam mempermudah penelitian sebagai berikut:

1. Implementasi mekanisme algoritma *Priority Scheduling* menggunakan Arduino Uno R3 dan menggunakan komunikasi nirkabel yaitu modul NRF24L01.
2. *Node* yang terdiri dari *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* merupakan *node* yang digunakan dalam implementasi.
3. Sensor yang digunakan terdiri dari beberapa modul sensor yaitu sensor tegangan, sensor arus berupa ACS712-30A, dan sensor yang dapat mendeteksi suatu kelembaban dan suhu berupa DHT11 yang akan digunakan pada setiap *slave node* dan *master node*. Akan tetapi hanya *master node* saja yang memiliki modul penyedia data waktu yaitu RTC DS1307.
4. Penerimaan dan pengiriman data dari *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* dapat terhubung secara *transmitter* dan *receiver* dengan menggunakan modul *wireless* NRF24L01.
5. Pengimplementasian algoritma *Priority Scheduling* terdapat pada *master node*.



1.6 Sistematika Pembahasan

Beberapa penelitian memiliki sub bab yang disusun untuk laporan akhir, didalam sub bab terdapat berbagai macam sistematika pembahasan yang dikelompokkan dan diuraikan sebagai berikut:

BAB 1 PENDAHULUAN

Pada pendahuluan terdiri dari beberapa anak sub bab yang terdiri dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Kumpulan penelitian terdahulu yang sesuai sebagai panutan skripsi memaparkan berbagai macam penjelasan – penjelasan secara ilmiah terdapat berbagai macam sumber, teori dan konsep dalam penyusunan laporan penelitian dapat menjadi landasan kepastakaan penulis.

BAB 3 METODOLOGI

Studi Pustaka, analisa kebutuhan, desain sistem, implementasi, pengujian, analisa, dan kesimpulan merupakan pembahasan metodologi pada penulisan dan penelitian bab ini.

BAB 4 ANALISIS KEBUTUHAN

Sistem yang akan dirancang dan Analisa sesuai dengan pembahasan didapatkan dengan melihat kembali dasar teori dan proses implementasi apa saja yang dibutuhkan untuk berjalannya penelitian ini.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Terdapat perancangan dan implementasi apa saja yang dibutuhkan dibutuhkan *node master* dan *slave* dalam menjalankan penelitian ini.

BAB 6 PENGUJIAN DAN ANALISIS

Langkah-langkah pengujian, hasil dan serta analisis pengujian pada sistem ini merupakan penjelasan pada bab ini.

BAB 7 PENUTUP

Kesimpulan dalam bab ini dapat diambil dari pertanyaan rumusan masalah dan saran apa saja untuk membenahi kelanjutan dari skripsi dan alat yang telah diciptakan.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang kajian pustaka dan dasar teori yang didapatkan dari berbagai referensi. Pada bagian kajian pustaka membahas tentang penelitian yang sudah ada dan diusulkan. Sedangkan pada dasar teori membahas teori apa saja yang diperlukan dalam penelitian.

2.1 Tinjauan Pustaka

Berikut ini dibutuhkan beberapa penelitian yang bersangkutan sebagai pengembangan dan acuan pada sistem yang akan digunakan oleh peneliti. Penelitian yang bersangkutan yaitu penelitian algoritma *Priority Scheduling* dan penelitian sensor arus dengan menggunakan sensor arus berupa sensor ACS712-30A dalam menggunakan pengiriman data secara *real Time* monitoring.

Penelitian sebelumnya yang berkaitan dan dikerjakan oleh (Arrosyid dkk, 2018), melakukan implementasi peningkatan layanan bagi penyedia energi listrik dengan menggunakan *Wireless Sensor Network* untuk monitoring parameter energi listrik. Adapun *point-point* energi listrik yang dimonitoring adalah tegangan, arus, frekuensi dan beda *phase*. Setelah melakukan tahap perancangan dan pembuatan sistem yang kemudian ditunjukkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan bahwa pengiriman data secara nirkabel menggunakan RF *modules* dapat diandalkan, karena pada pengiriman 1000 data, hanya terjadi kehilangan data kurang dari 0.7 % dari data yang dikirim. RF *modules* mempunyai kehandalan pada jarak pengiriman data. *Output* dari sensor arus, tegangan, frekuensi dan beda *phase* cukup bagus karena perubahannya cukup linier dan sebanding dengan perubahan pada sisi input. Persamaan penelitian ini dengan penelitian sebelumnya adalah sama-sama menggunakan media *Wireless Sensor Network*.

Penelitian berikutnya yang dilakukan oleh (Atefi, 2011), peneliti melakukan *real Time scheduling algorithms* untuk *Wireless Sensor Network*. Menggunakan *Wireless Sensor Network* yang artinya waktu nyata aplikasi yang diminta untuk berjalan secara *real Time* yang menggunakan desain sistem operasi sensornet (OS). (Jason, 2015) telah melakukan penelitian tentang *real Time scheduling strategy* untuk *Wireless Sensor Network* yang menggunakan jaringan sensor O.S yang membahas beberapa karakteristik dan keterbatasan desain operasi dalam jaringan sensor nirkabel.

(Karthikeyan, 2014) telah melakukan penelitian tentang *Priority based packet scheduling Aproach* untuk *Wireless Sensor Networks*. Peneliti mengimplementasikan algoritma *Priority Scheduling* untuk penjadwalan berbagai jenis paket data, seperti waktu nyata dan tidak *real Time* pada paket data di *node* sensor dengan sumber daya kendala dalam jaringan sensor nirkabel yang sebagian besar mekanisme penjadwalan paket yang ada dari Wireless Jaringan sensor yang menggunakan FCFS (*First Come First Served*), dan prioritas *non – preemptif* dan *preemptif* algoritma penjadwalan prioritas.



Sedangkan pada penelitian sekarang parameter energy listrik yang dimonitoring adalah *node* yang terdiri dari empat buah *node* antara lain *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* dengan menggunakan algoritma *Priority Scheduling* pada modul dan sensor sistem yang ada pada sistem secara *real Time*. Dimana pada sensor tersebut terdapat sistem *output* dan *input* dengan fase yang berbeda. Perbedaan penelitian ini dengan penelitian terdahulu adalah penelitian yang dilakukan saat ini perancangan sistem *input* dan *output*nya menggunakan penghitungan waktu secara *real Time*, dapat membaca arus, daya, tegangan, suhu dan kelembaban pada jarak atau situasi yang berbeda. Penelitian ini dapat menghemat energi dalam lingkup *Wireless Sensor Network*, akuisisi data berupa besaran suhu dan kelembaban. Pengiriman dan penerimaan data antara *master node* dan masing-masing *slave* menggunakan *wireless NRF24L01*. Dalam dunia *Wireless Sensor Network* monitoring dilakukan dengan cara menggunakan lebih dari satu unit *wireless sensor node* didalam suatu area. Dengan cara seperti itu pemantauan dalam penelitian membutuhkan sumber daya yang memadai agar seluruh *node* dapat bekerja dengan baik. Sehingga dalam rangka penghematan daya secara *hardware* dengan menambahkan fitur dengan membuat mode *low power* dengan membuat Arduino Uno R3 dalam mode *sleep* selama 2 detik.

2.2 Algoritme *Priority Scheduling*

Menurut (Munir, 2011) Algoritma yaitu urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Langkah-langkah logis berarti kebenarannya harus dapat ditentukan benar atau salah. Sedangkan menurut (Cormen, 2001) Algoritma ialah setiap prosedur komputasi yang terdefinisi dengan baik yang mengambil beberapa nilai, atau seperangkat nilai-nilai, sebagai masukan dan menghasilkan beberapa nilai atau seperangkat nilai-nilai sebagai *output*. Kita juga dapat melihat sebuah algoritma sebagai alat bantu untuk memecahkan masalah.

Algoritma ialah bersifat *programming language independent*. Sebuah algoritma dapat diimplementasikan dengan berbagai bahasa pemrograman, tapi penulisnya tidak tergantung pada bahasa pemrograman tertentu. Algoritma dapat disajikan dengan dua teknik yaitu dengan tulisan dan gambar, penyajian algoritma dalam bentuk tulisan biasanya menggunakan metode seperti Bahasa Indonesia Terstruktur (BIT), *pseudocode*, *spark*, *Structured English*. Sedangkan penyajian algoritma dalam bentuk gambar biasanya menggunakan metode seperti *Flowchart*, *hirarcy plus input-process-output (HIPO) chart*, *structured chart* dan *nassi schneiderman chart*. (Thomason (2004)

Sedangkan berdasarkan KBBI, algoritma merupakan prosedur sistematis untuk memecahkan masalah matematis dalam langkah-langkah terbatas. Masalah merupakan sesuatu yang harus diselesaikan, dan masalah menentukan secara umum yang diinginkan pada hubungan *input/output*, dan algoritma menggambarkan prosedur komputasi tertentu untuk mencapai yang hubungan *input/output*.



Pada ilmu komputer yang sering dipertanyakan adalah bukan bagaimana menyelesaikan suatu persoalan (*problem*), melainkan bagaimana menyelesaikan persoalan dengan baik, sebagai contoh adalah bagaimana mengatasi permasalahan manajemen *bandwidth* internet sehingga tidak terjadi *overload* atau *client* saling berebut *bandwidth*. Ada beberapa teknik yang dilakukan untuk menangani hal tersebut diatas, namun yang menjadi permasalahan bukanlah menemukan bagaimana cara mengefisiensi *bandwidth* yang sudah ada melainkan menambah kapasitasnya untuk memenuhi kebutuhan user.

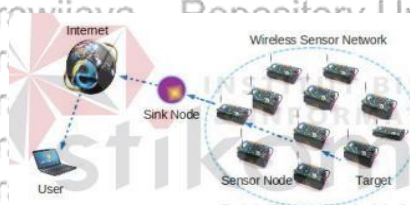
Priority Scheduling memiliki pengertian penjadwalan algoritma yang mendahulukan proses yang memiliki prioritas tertinggi. Prioritas pada satu proses dapat ditentukan melalui *Time limit*, *Memory requirement*, Akses file, Perbandingan antara *burst M/K* dengan *CPU burst*, dan tingkat kepentingan proses. *Priority Scheduling* pada tiap-tiap proses dilengkapi dengan nomor prioritas. *Priority Scheduling* memiliki dua skema yaitu non-preemptive dan preemptive.

Dengan skema jika P1 yang datang pada saat P0 sedang berjalan, maka yang akan dilihat adalah prioritas P1. Tapi seandainya prioritas P1 lebih besar dibandingkan prioritas P0, maka:

1. Pada non – preemptive, algoritma tetap akan menyelesaikan P0 sampai habis CPU burst-nya dan meletakkan P1 pada posisi head queue.
2. Sedangkan pada preemptive, P0 akan dihentikan dulu, dan CPU ganti dialokasikan untuk P1. Misalnya terdapat lima proses P1, P2, P3, P4 dan P5 yang datang secara berurutan dengan CPU burst dalam milidetik.

2.3 Wireless Sensor Network

Menurut, Bathia, 2010 pengertian *Wireless Sensor Network* (WSN) atau jaringan sensor nirkabel adalah suatu jaringan nirkabel yang terdiri dari beberapa sensor *node* yang bersifat *individu* yang diletakkan ditempat-tempat yang berbeda untuk memonitoring kondisi suatu tempat dan dapat berinteraksi dengan lingkungannya dengan cara *sensing*, *controlling* dan *communication* terhadap parameter-parameter fisiknya. Arsitektur WSN dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur *Wireless Sensor Network*

Sumber: (<http://virtual-labs.ac.in>)

Wireless Sensor Network adalah sebuah jaringan yang terdiri dari sekumpulan *node-node* sensor yang memiliki kemampuan untuk mendeteksi suatu keadaan di sekitarnya, memproses data yang diperoleh, dan saling berkomunikasi satu sama lain untuk mengirimkan data ke *node* pusat (Sen, 2009). Pada umumnya WSN

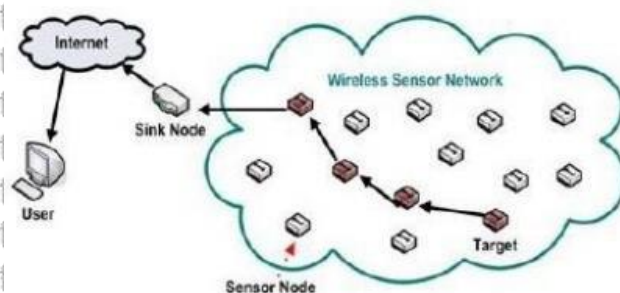


digunakan untuk memantau keadaan suatu lingkungan, seperti suhu, suara, tekanan, dll. WSN mulai banyak diterapkan karena beberapa keunggulannya, antara lain yaitu rendah daya atau energy, skalabilitas yang baik, mudah diimplementasikan, ukuran yang kecil, dan sebagainya. Tetapi WSN juga memiliki masalah pada keamanan dan rentan terhadap serangan yang dilakukan pada jaringan (Lewis, 2004). Untuk menangani masalah tersebut dapat dilakukan dengan mengimplementasikan protocol keamanan. Namun karena ukurannya yang kecil WSN memiliki keterbatasan *resource* atau sumber daya, sehingga untuk melakukan proses komputasi yang kompleks akan membutuhkan energi yang besar pula. Hal ini lah yang menjadi permasalahan utama dalam WSN. Untuk itu protokol keamanan dituntut untuk dapat memberikan keamanan yang baik namun dengan konsumsi energi yang rendah (Putra, 2010).

Seiring dengan berkembangnya teknologi dewasa ini semua berkembang ke ranah konektivitas lingkungan fisik. Kebanyakan observasi yang dilakukan di lapangan melibatkan banyak faktor dan parameter-parameter untuk mendapatkan hasil yang maksimal dan akurat. Dengan adanya teknologi WSN, memungkinkan peneliti untuk mendapat informasi yang maksimal tanpa harus berada di area sensor. Informasi dapat diakses dari jarak jauh melalui gadget seperti laptop, *remote device*, *server* dan sebagainya.

2.3.1 Arsitektur *Wireless Sensor Network*

Wireless Sensor Network dapat diatur dengan *node* sensor disebar dengan tujuan untuk menangkap adanya gejala atau fenomena yang hendak diteliti. Jumlah *node* yang disebar dapat ditentukan sesuai kebutuhan dan tergantung beberapa faktor misalnya luas area, kemampuan sensing *node*, dan sebagainya. Tiap *node* dalam WSN dapat melakukan pemantauan lingkungan terbuka secara langsung dengan memanfaatkan beberapa macam sensor. (Mittal, 2010). Arsitektur WSN secara umum dapat ditunjukkan pada Gambar 2.2.



Gambar 2.2 Arsitektur *Wireless Sensor Network* secara umum

Sumber: (telekom.ee.uui.ac.id)

Pada Gambar 2.2 dapat dilihat, *node* sensor yang berukuran kecil disebar dalam suatu area sensor. *Node* sensor tersebut memiliki kemampuan untuk meneruskan data yang dikumpulkan ke *node* lain yang berdekatan. Data



dikirimkan melalui transmisi radio akan diteruskan menuju *base station* yang merupakan penghubung antara *node* sensor dan *user*.

2.3.2 Bagian-Bagian Pada *Wireless Sensor Network*

Wireless Sensor Network (WSN) terbagi atas lima bagian di antaranya adalah sebagai berikut (Culler, 2004).

1. *Transceiver*, berfungsi untuk menerima atau mengirim data dengan menggunakan protokol IEEE 802.15.4 atau IEEE 802.11b/g kepada *device* lain.
2. Mikrokontroler, berfungsi untuk melakukan fungsi perhitungan, mengontrol dan memproses *device* yang terhubung dengan mikrokontroler.
3. *Power source*, berfungsi sebagai sumber energi bagi sistem *wireless sensor* secara keseluruhan.
4. *External memory*, berfungsi sebagai tambahan *memory* bagi sistem *wireless sensor*, pada dasarnya sebuah unit mikrokontroler memiliki unit *memory* sendiri.
5. Sensor, berfungsi untuk men-sensing besaran-besaran fisis yang hendak diukur. Sensor adalah suatu alat yang mampu untuk mengubah suatu bentuk energi ke bentuk energi lain, dalam hal ini adalah mengubah dari energi besaran yang diukur menjadi energi listrik yang kemudian diubah oleh ADC menjadi deretan pulsa terkuantisasi yang kemudian bisa dibaca oleh mikrokontroler.

2.4 *Power Monitoring System*

Power Monitoring System adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. Monitoring menyediakan data dasar untuk menjawab permasalahan. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses suatu objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis tindakan untuk mempertahankan manajemen yang sedang berjalan (Pambudi, 2014). Secara umum monitoring bertujuan mendapatkan umpan balik bagi kebutuhan program proses pembelajaran yang sedang berjalan, dengan mengetahui kebutuhan ini pelaksanaan program akan segera mempersiapkan kebutuhan dalam pembelajaran tersebut. Kebutuhan bisa berupa biaya, waktu, personel, dan alat. Pelaksanaan program akan mengetahui berapa biaya yang dibutuhkan, berapa lama waktu yang tersedia untuk kegiatan tersebut. Dengan demikian akan diketahui pula berapa jumlah tenaga yang dibutuhkan, serta alat apa yang harus disediakan untuk melaksanakan program tersebut (Insnowaty, 2016).

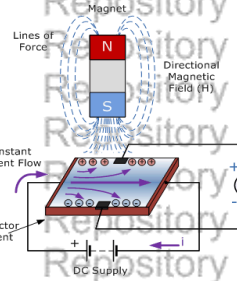
2.5 Sensor Arus ACS712-30A

Sensor arus ACS712-30A merupakan suatu IC terpaket yang berguna sebagai sensor arus menggantikan *transformator* arus yang relatif besar dalam hal ukuran. Pada prinsipnya ACS712-30A sama dengan sensor *Hall Effect* lainnya yaitu dengan memanfaatkan medan magnetik disekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus (Allegro, 2017).

Sensor arus ACS712 memiliki beberapa fitur kelebihan antara lain:

1. Rendah *noise*.
2. 5 mikrodetik keluaran dalam menanggapi arus masukan.
3. *Bandwith* 80 kHz.
4. Total *output error* 1,5% pada $T_A=25^{\circ}\text{C}$
5. Tampak Kecil, *low-profile* paket SOIC8.
6. 1,2 MW resistansi konduktor internal.
7. Isolasi tegangan 2,1 kVRMS minimum dari pin 1-4 ke pin 5-8.
8. 66-185 mV / A sensitivitas keluaran.

Hall Effect adalah fenomena fisika dimana aliran listrik/elektron dalam pelat konduktor terpengaruh oleh paparan medan magnet, lengkapnya dapat dibaca merujuk pada jurnal nasional tentang *Hall Effect* yang ditulis oleh (Novi Tri Nugraheni dari Universitas Airlangga Surabaya). Secara sederhana, pemanfaatan *Hall Effect* pada IC ACS712-30A ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Hall Effect pada ACS712-30A

Sensor arus ACS712-30A ada beberapa macam di jual secara umum yaitu dari 5A, 20A, dan 30A , akan tetapi yang digunakan pada penelitian ini adalah ACS712-30A, ACS712-30A memiliki besar arus maksimum yang dapat dideteksi sebesar 30A, di mana tegangan pada pin keluaran akan berubah secara linear mulai dari 2,5V ($\frac{1}{2} \times V_{CC}$, tegangan catu daya $V_{CC}=5\text{V}$) untuk kondisi tidak ada arus hingga 4,5V. Pada arus sebesar +30A atau 0,5V dan pada arus sebesar -30A (positif/negatif tergantung polaritas, nilai di bawah 0,5V atau di atas 4,5V dapat dianggap lebih dari batas maksimum). Perubahan tingkat tegangan berkorelasi linear terhadap besar arus sebesar 83,333 mV/Ampere.

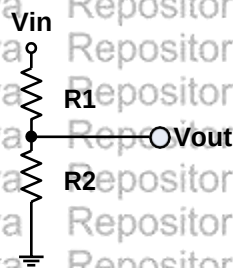
2.6 Sensor Tegangan

Rangkaian pembagi tegangan disebut juga sebagai rangkaian potensial. *Input* ke sebuah rangkaian pembagi tegangan adalah tegangan V_{in} . Tegangan V_{in}



tersebut menggerakkan arus I untuk mengalir melewati kedua resistor. Karena kedua resistor terhubung secara seri, maka arus yang sama mengalir melewati tiap-tiap resistor. Tahanan efektif dari kedua resistor seri adalah $R_1 + R_2$. Jatuh tegangan pada gabungan kedua resistor ini adalah V_{in} (Ahmad, 2007).

Prinsip sensor tegangan yang digunakan untuk akuisisi data tegangan pada sistem yang digunakan adalah menggunakan rangkaian pembagi tegangan yaitu dengan 2 buah resistor dengan besar masing – masing yaitu $1M\Omega$ dan $100k\Omega$. Digunakannya rangkaian pembagi tegangan adalah bertujuan agar *output* yang dikeluarkan rangkaian pembagi tegangan (V_{out}) lebih kecil dari pada tegangan masukan (V_{in}). Berikut merupakan gambar *sensor* tegangan terdapat pada gambar, *sensor* ini menggunakan rangkaian pembagi tegangan dan perhitungannya. Rangkaian Sensor tegangan dapat dilihat di Gambar 2.4.



Gambar 2.4 Rangkaian Sensor Tegangan
Sumber: (Kho, 2016)

$$\text{Rumus: } V_{out} = V_{in} * \left(\frac{R_2}{R_1 + R_2} \right)$$

Diketahui:

$$V_{in} = 5V$$

$$R_1 = 1M?$$

$$R_2 = 100K?$$

Ditanya: Berapa V_{out} pada saat V_{in} 5V?

Jawab:

$$V_{out} = 5V * \left(\frac{100K}{(1M + 100K)} \right) = 0.45V$$

Jadi, nilai V_{out} adalah sebesar 0.45V

2.7 Daya

Daya adalah energi yang dikeluarkan untuk melakukan usaha. Dalam sistem tenaga listrik, daya merupakan jumlah energi yang digunakan untuk melakukan kerja atau usaha. Daya memiliki satuan *Watt*, yang merupakan perkalian dari tegangan (*Volt*) dan arus (*amphere*). (Hudaya, 2014)

Daya dinyatakan dalam P , Tegangan dinyatakan dalam V dan arus dinyatakan dalam I , sehingga besarnya daya dapat dinyatakan sebagai berikut.

$$P = V \times I \quad (2.1)$$

$$P = \text{Volt} \times \text{mAmphere} \quad (2.2)$$



$$P = mWatt$$

(2.3)

2.7.1 Daya Aktif

Daya aktif (*active power*) adalah daya yang terpakai untuk melakukan energi sebenarnya. Satuan daya aktif adalah *watt*. Adapun persamaan dalam daya aktif sebagai berikut.

Untuk satuan phasa:

$$P = V \times I \times \cos f$$

(2.4)

Untuk tiga phasa:

$$P = 3 \times V \times I \times \cos f$$

(2.5)

Daya ini digunakan secara umum oleh konsumen dan dikonversikan dalam bentuk kerja.

2.7.2 Daya Reaktif

Terjadinya pembetulan pada medan magnet untuk jumlah daya yang diperlukan adalah pengertian dari daya reaktif. *Fluks* medan magnet merupakan hasil dari medan magnet yang terbentuk. Berikut merupakan contoh daya yang menghasilkan daya reaktif yaitu transformator, motor, dan lain – lain.

Var merupakan satuan daya reaktif.

Untuk satuan phasa:

$$Q = V \times I \times \sin f$$

(2.6)

Untuk tiga phasa:

$$Q = 3 \times V \times I \times \sin f$$

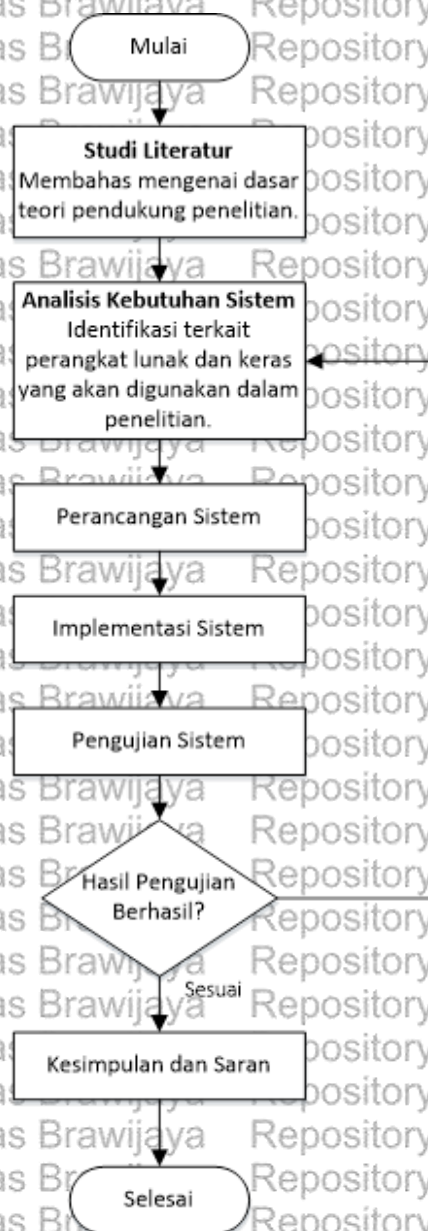
(2.7)



BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Metodologi Penelitian pada bab ini membutuhkan penerapan agar skripsi yang telah dikerjakan ini berjalan dengan baik. Studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, kesimpulan dan saran merupakan proses urutan metodologi penelitian yang akan dilakukan oleh penulis. Alur metodologi penelitian dapat dilihat secara lengkap pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian



3.2 Studi Literatur

Studi literatur menjelaskan apa saja yang dibutuhkan penulis dalam menyelesaikan masalah yang terdapat pada penelitian yang telah dikerjakan dan sistem dapat bekerja sesuai tanpa adanya kendala. Studi literatur harus dan dibaca oleh penulis dapat dicari di jurnal, buku, *e-book*, dokumentasi dan penelitian yang membahas tentang algoritma *Priority Scheduling*.

3.3 Analisis Kebutuhan Sistem

Dalam pekerjaan penelitian ini dibutuhkan analisis sistem apa saja alat-alat yang harus digunakan dalam pengerjaan penelitian ini. Kebutuhan perangkat keras merupakan alat apa saja yang harus digunakan penulis dalam penelitian ini dan kebutuhan perangkat lunak merupakan aplikasi dan *library* apa saja yang dibutuhkan pada penelitian ini.

3.3.1 Kebutuhan Perangkat Keras

Pada pengerjaan kebutuhan perangkat keras dibutuhkan berbagai macam alat, berikut merupakan nama-nama alat dan fungsi alat yang akan digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan Perangkat Keras

Nama Alat	Fungsi
Arduino Uno R3	Bekerja sebagai mikrokontroler setiap <i>node</i>
NRF24L01	Bekerja untuk komunikasi <i>node wireless</i>
RTC DS1307	Bekerja untuk menampilkan data secara <i>real Time</i>
DHT11	Bekerja untuk mendeteksi besaran suhu dan kelembaban pada berbagai macam kondisi alam
ACS712-30A	Bekerja untuk menghitung arus yang digunakan pada suatu <i>node</i>
Sensor Tegangan	Bekerja untuk menghitung tegangan yang digunakan pada suatu <i>node</i>
USB	Digunakan sebagai komunikasi Arduino Uno R3 dengan PC, saklar atau powerbank

3.3.2 Kebutuhan Perangkat Lunak

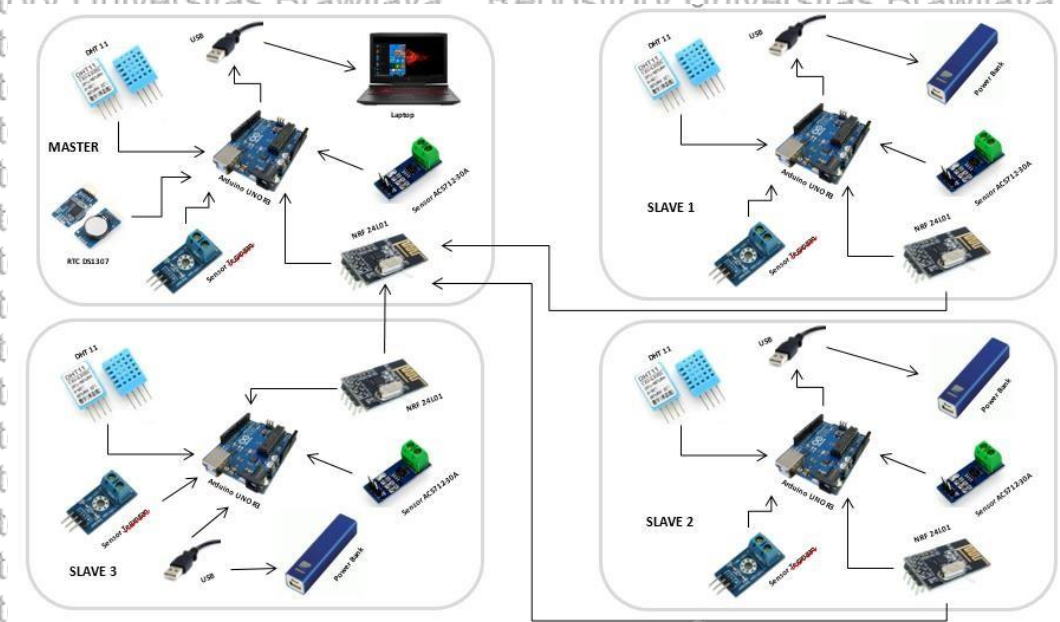
Aplikasi yang akan digunakan penulis dalam melakukan penelitian ini adalah Arduino IDE dalam pembuatan codingan, *eagle* sebagai pembuatan rancangan PCB dan *library* pada Arduino IDE yang akan digunakan. Penjelasan fungsi dari kebutuhan perangkat lunak dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kebutuhan Perangkat Lunak

<i>Library</i>	<i>Fungsi</i>
SPI	komunikasi dengan NRF24L01
NRF24L01	Sebagai komunikasi menerima dan mengirim data
NRF24L01 Network	Sebagai komunikasi menerima dan mengirim data pada 4 <i>node</i>
Wire.h	Sebagai penunjang penggunaan <i>library</i> RTCLib pada segi komunikasi I2C
RTCLib	Untuk memudahkan penggunaan alat RTC
One Wire	Berkomunikasi menerima dan mengirim data DHT11

3.4 Perancangan Sistem

Pada tahap pengerjaan perancangan sistem akan dilakukan agar sistem dapat bekerja dengan baik tanpa kendala sehingga dapat memudahkan peneliti dalam mengerjakan penelitian ini. Pada Gambar 3.2 akan dijelaskan apa saja alat yang akan digunakan peneliti dalam merancang dan merencanakan penelitian ini.



Gambar 3.2 Gambaran Umum Perancangan

Rancangan sistem pada Gambar 3.2 memiliki berbagai macam alat, alat tersebut memiliki 4 buah *node* yang akan digunakan. *Slave node* dan *master node* mempunyai berbagai macam alat yang akan dirancang, alat tersebut meliputi Arduino uno R3 sebagai mikrokontroler, USB pada *node* yang disambungkan ke *power bank*, saklar, dan laptop memberikan daya untuk menyalakan *node-node*, NRF24L01 sebagai pengiriman dan penerimaan semua data pada *node*, DHT11 sebagai menghitung suatu suhu dan kelembaban pada suatu tempat, ACS712-30A sebagai menghitung keluaran arus masing-masing *node* yang akan digunakan, sensor tegangan yang memberikan tegangan sebesar 5v pada *node*. Akan tetapi



hanya *node master* saja yang memiliki RTC DS1307 yang akan digunakan sebagai perhitungan waktu pengiriman dan penerimaan data *node* secara *real Time*.

3.5 Implementasi Sistem

Pada penelitian ini mengimplementasikan sistem yang digunakan berupa *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node*. Algoritma *Priority Scheduling* akan mengirimkan dan menerima setiap data *node* menggunakan NRF24L01 sebagai *wireless*. *Master node* sebagai penerima data dari *slave 1 node*, *slave 2 node* dan *slave 3 node* dimana penerimaan data yang diprioritaskan adalah pengiriman data yang diurutkan berdasarkan daya yang terbesar hingga ke terkecil pada kelompok *node*. *Slave node* akan mengecek terlebih dahulu apakah adakah radio komunikasi dari *master* masuk atau tidak, jika ada *request* data dari *master* maka *slave* akan melakukan pembacaan data pada masing-masing *node*, data tersebut berupa data suhu dan kelembaban yang didapatkan dari DHT11, data arus yang didapatkan dari sensor ACS, dan menghitung waktu penyelesaian satu siklus pengambilan data pada kelompok *slave node*. Selanjutnya semua data akan dikirimkan ke *master*, akan tetapi terdapat pengecekan no id, jika sesuai maka kelompok *slave* akan mengirimkan data ke *master node*. Setelah semua data terkumpul pada *master node* maka selanjutnya *master* akan menghitung daya yang masuk dari kelompok *slave* dan kemudian *master* akan melakukan pengurutan *Priority Scheduling* daya pada *master* dan masing-masing *slave* dari yang terbesar hingga ke terkecil. Selanjutnya terjadilah proses menampilkan semua data berdasarkan *Prioritas*.

3.6 Pengujian Sistem

Pada langkah ini akan dilakukan pengujian sistem sehingga proses berjalannya sistem tidak ada kendala sama sekali, diantaranya:

1. Pada *master node* dan masing-masing *slave node* akan melakukan pengujian akurasi hasil data *sensing* dan melakukan perbandingan hasil *sensing* dengan *Avometer* dan termometer.
2. Pada *master node* dan masing-masing *slave node* akan melakukan pengujian hasil data daya yang dihasilkan pada setiap *node*.
3. Algoritma *Priority Scheduling* akan melakukan pengujian pengiriman data pada jarak 1-10meter pada ruangan ada halangan dan tanpa halangan, apakah data yang dikirimkan akan menghasilkan performa yang diinginkan jika jarak yang dilakukan pada 1-10meter dan melihat persentase keberhasilan pengiriman data daya pada masing-masing *node* menggunakan *Priority Scheduling* berhasil atau tidak.

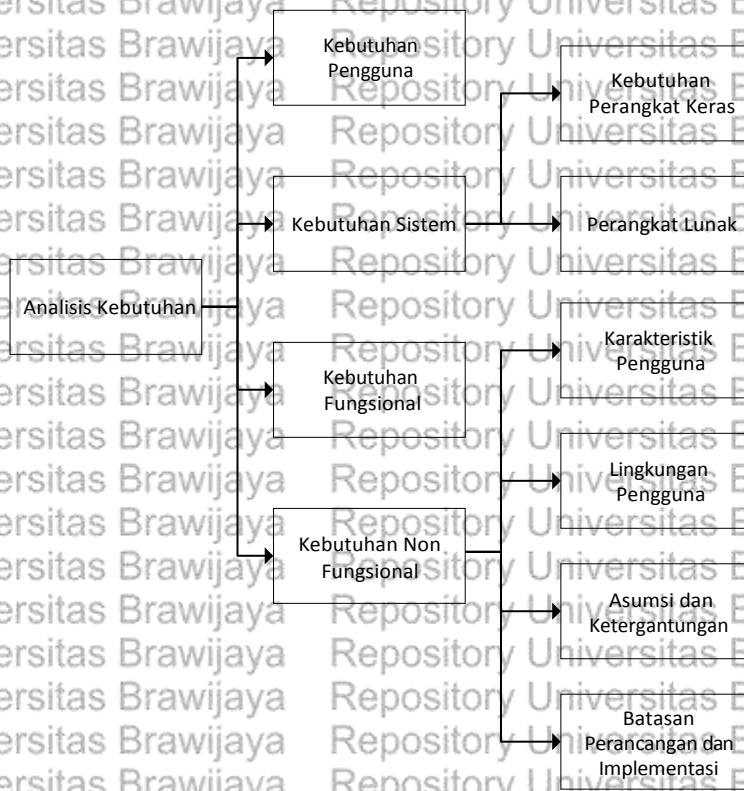
3.7 Kesimpulan dan Saran

Pada tahap ini pada rumusan masalah yang dapat dilihat pada bab 1 merupakan jawaban dari kesimpulan yang akan dilakukan penulis dan saran bertujuan melihat jika ada berbagai macam kekurangan apa saja pada skripsi ini maka disarankan untuk melakukan pembenahan dan dikembangkan apa saja yang kurang dalam skripsi ini.



BAB 4 ANALISIS KEBUTUHAN

Kebutuhan yang dibutuhkan apa saja dalam pengerjaan analisis yaitu kebutuhan pengguna, kebutuhan sistem yang terdiri dari dua kebutuhan, yang pertama kebutuhan perangkat keras dan yang kedua perangkat lunak, selanjutnya kebutuhan fungsional, dan yang terakhir kebutuhan non fungsional yang terbagi menjadi 4 kebutuhan, kebutuhan yang pertama merupakan karakteristik pengguna, kedua lingkungan pengguna, ketiga asumsi dan ketergantungan, dan keempat merupakan batasan perancangan dan implementasi dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Analisis Kebutuhan

4.1 Kebutuhan Pengguna

Dengan melihat hasil semua data pada *serial monitor*, penulis dapat melihat semua hasil data pada setiap *node* dan memonitoring hasil data *sensing* yang telah didapatkan pada hasil masing-masing *node*.

4.2 Kebutuhan Sistem

4.2.1 Kebutuhan Perangkat Keras

Pada proses ini akan menjelaskan mengapa penulis menggunakan perangkat keras ini, penjelasannya adalah seperti ini.

1. Peneliti menjalankan Aplikasi Arduino IDE dan menampilkan hasil data dari keempat *node* ke *serial monitor*. Laptop Acer Aspire E1-451G merupakan laptop peneliti yang digunakan karena dari spesifikasi *laptop* tersebut sudah sangat mumpuni untuk menjalankan aplikasi Arduino Uno R3 dan menampilkan data dari keempat *node*. Peneliti menggunakan laptop pada Gambar 4.2.



Gambar 4.2 Laptop Acer Aspire E1-451G

Penjelasan dari gambar laptop:

Laptop : Acer Aspire E1-451G
 Prosesor : AMD A10-5750M APU with Radeon™ HD Graphics
 Memori RAM : 8 GB DDR3
 Slot Memori : Double Channel
 Harddisk : 500GB

2. Mikrokontroler *master node*, *slave 1 node*, *slave 2 node* dan *slave 3 node* pada pengujian ini adalah Arduino Uno R3. Mikrokontroler yang digunakan pada *master node* dan setiap *slave node* sebagai pusat mengendalikan sistem yang mengerjakan data dan menyimpan *interrupt code*. Peneliti menggunakan Arduino Uno R3 karena *Board* ini memiliki 14 digital *input / output* pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 *input analog*, 16 MHz osilator kristal, koneksi USB, *jack* listrik tombol *reset*. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya. Peneliti juga menggunakan Arduino Uno R3 karena Arduino Uno R3 tidak bisa untuk *multitask*, akan tetapi Arduino Uno R3 bekerja dengan metode *blocking* yang mana sangat cocok untuk menggunakan metode *Priority Scheduling non-preemptive*. Bentuk fisik dari mikrokontroler Arduino Uno R3 yang digunakan akan dijelaskan secara spesifikasi dan diperlihatkan pada Gambar 4.3 dan Tabel 4.1.

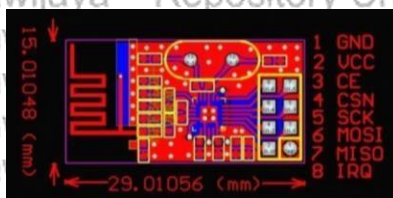


Gambar 4.3 Arduino Uno R3

Tabel 4.1 Spesifikasi Arduino Uno R3

Mikrokontroler	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital i/o pins	14 (of which 6 provide PWM output)
Pwm digital i/o pins	6
Analog input pins	6
Dc current per i/o pin	20 mA
Dc current for 3.3v pin	50 mA
Flash memory	32 KB (ATmega328P) Of witch 0.5 KB used by bootloader
Sram	2 KB (ATmega328P)
Eeprom	1 KB (ATmega328P)
Clock speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3. NRF24L01 pada perangkat keras digunakan penulis terdapat beberapa *node* yaitu, *master node*, *slave 1 node*, *slave 2 node* dan *slave 3 node* yang merupakan sebuah modul *transceiver* komunikasi nirkabel yang memanfaatkan gelombang radio pada frekuensi 2,4GHz untuk berkomunikasi dan mengirimkan data pada *receiver sensor node* dengan keunggulan pengiriman data dengan gelombang radio ialah jangkauannya yang lebih luas dari *Bluetooth*. Peneliti memilih NRF24L01 karena memiliki perangkat keras berupa *baseband logic Enhanced ShockBurst* dan *protocol accelerator* yang memungkinkan untuk berkomunikasi dalam kecepatan tinggi. Modul ini dapat mendukung antarmuka SPI (*Serial Peripheral Interface*). Selain itu modul ini mempunyai fitur *true ULP solution*, yang berfungsi sebagai penghemat konsumsi daya sehingga hemat energi. NRF24L01 memiliki 8 buah pin, di antaranya: VCC, GND, IRQ, SCK, MISO, MOSI, CE, dan CSN seperti yang akan ditunjukkan pada Gambar 4.4.



Gambar 4.4 NRF24L01

4. RTC DS1307 pada perangkat keras hanya terdapat pada *master node* saja. RTC DS1307 digunakan untuk menyimpan data waktu yang diperlukan oleh mikrokontroler dan menempatkan data waktu yang sesuai ditentukan oleh sensor yang dipakai untuk monitoring. Peneliti menggunakan RTC DS1307



karena dapat memberikan informasi waktu data secara akurat dan dapat menjaga atau menyimpan data yang kirim dan diterima secara real Time yang dinilai cukup akurat karena menggunakan isolator Kristal eksternal, sehingga saat perangkat mikrokontroler yang difungsikan untuk mengakses RTC sebagai sumber data waktu akan dimatikan dan data waktu tidak akan hilang selama baterai yang terhubung pada RTC tersebut tidak mati. RTC DS 1307 berkomunikasi menggunakan *protocol* I2C dan konsumsi daya yang digunakan sebesar 3,3-5,5V. Bentuk fisik dari RTC DS1307 yang digunakan akan dijelaskan secara spesifikasi dan diperlihatkan pada Gambar 4.5 dan Tabel 4.2.



Gambar 4.5 RTC DS1307

Tabel 4.2 Spesifikasi Modul RTC DS1307

Size:	38mm x 22mm x 14mm
Operating Voltage:	3.3V – 5.5 V
Clock Accuracy:	0-40 range, the accuracy 2ppm,
Memory chips:	AT24C32 (storage capacity 4KB)

5. Sensor suhu DHT11 pada perangkat keras terdapat pada *master node*, *slave 1 node*, *slave 2 node* dan *slave 3 node*. DHT11 merupakan sebuah sensor yang dapat mengkalibrasi data suhu menjadi sinyal keluaran digital yang kemudian diproses oleh mikrokontroler. Peneliti menggunakan DHT11 karena DHT11 memiliki tingkat stabilitas yang sangat baik serta memiliki fitur kalibrasi yang sangat akurat. DHT11 termasuk sensor yang memiliki kualitas yang sangat baik, dinilai dari respon dan pembacaan data yang cepat dan akurat. Transmisi sinyal DHT11 bisa mencapai 20meter dan sangat mendukung untuk penelitian ini yang menguji jarak minimum dan jarak maksimal data. Bentuk fisik dari DHT11 akan dijelaskan secara spesifikasi dan diperlihatkan pada Gambar 4.6 dan Tabel 4.3.

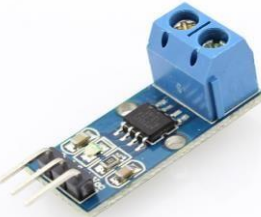


Gambar 4.6 DHT11

Tabel 4.3 Spesifikasi DHT11

Size:	38mm x 22mm x 14mm
Operating Voltage:	3.3V – 5.5 V
Clock Accuracy:	0-40 range, the accuracy 2ppm,
Memory chips:	AT24C32 (storage capacity 4KB)

6. Sensor ACS712-30A pada perangkat keras terdapat pada *master node*, *slave 1 node*, *slave 2 node* dan *slave 3 node*. Sensor ACS712-30A merupakan sebuah sensor yang dapat mengkalibrasi suatu arus yang terpakai dalam sistem. Peneliti menggunakan ACS712-30A karena sensor ini memiliki pembacaan dengan kecepatan yang tinggi, karena didalam rangkaiannya terdapat *low-offset linear hall* dengan satu lintasan yang terbuat dari tembaga. Ketelitian dalam pembacaan sensor dioptimalkan dengan cara pemasangan komponen yang ada didalamnya antara penghantar yang akan menghasilkan medan magnet dengan *hall transducer* secara berdekatan. Sensor ACS712-30A memiliki ketebalan penghantar arus didalam sensor sebesar 3x kondisi *overcurrent*. Bentuk fisik dari sensor ACS712-30A akan dijelaskan secara spesifikasi dan diperlihatkan pada Gambar 4.7 dan tabel 4.4.

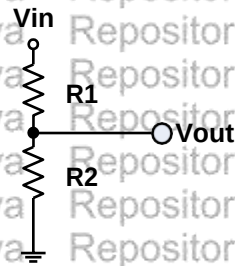


Gambar 4.7 Sensor ACS712-30A

Tabel 4.4 Spesifikasi Sensor ACS712-30A

PCB board size:	31mm x 13mm
Sensitivity:	66mV per Amp
Power:	5V DC
Measures:	30Amps AC or DC

7. Sensor tegangan pada perangkat keras terdapat pada *master node*, *slave 1 node*, *slave 2 node* dan *slave 3 node*. Peneliti menggunakan sensor tegangan karena untuk akuisisi data tegangan pada sistem yang digunakan adalah menggunakan rangkaian pembagi tegangan yaitu dengan 2 buah resistor dengan besar masing – masing yaitu 1MΩ dan 100kΩ. Digunakannya rangkaian pembagi tegangan bertujuan untuk output yang dikeluarkan rangkaian pembagi tegangan (Vout) lebih kecil dari pada tegangan masukan (Vin). Berikut merupakan gambar rangkaian sensor tegangan terdapat pada Gambar 4.8.



Gambar 4.8 Rangkaian Sensor Tegangan

4.2.2 Kebutuhan Perangkat Lunak

Pada kebutuhan perangkat lunak terdapat *transmitter* dan *receiver* yang terdiri dari beberapa *library*, gambaran keseluruhan akan ditunjukkan pada Gambar 4.9 dibawah ini:



Gambar 4.9 Kebutuhan Perangkat Lunak

Pada *transmitter* sensor masing-masing *node*:

- Arduino IDE adalah program yang digunakan mikrokontroler yang digunakan untuk menulis, mengecek apakah program sudah benar apa tidak, meng-*upload* program jika sudah benar.
- SPI berfungsi untuk menentukan pin komunikasi SPI (*Serial Peripheral Interface*) antara NRF24L01 agar dapat digunakan untuk pengiriman data dengan menjalankan algoritma *Priority Scheduling*. Konfigurasi SPI di *node slave 1, slave 2 dan slave 3* memiliki no id, untuk *slave 1* no idnya "01", untuk *slave 2* no idnya "02" dan untuk *slave 3* no idnya "03" agar data yang terkirim sesuai prioritas dan data yang terkirim tidak tertukar.
- ADC digunakan untuk menangani pembacaan analog dari data sensor, arus dan sensor tegangan.
- One Wire merupakan program yang digunakan untuk pembacaan sensor DHT11.

Pada *receiver sensor node master*:

- Arduino IDE adalah program yang digunakan mikrokontroler yang digunakan untuk menulis, mengecek apakah program sudah benar apa tidak, meng-*upload* program jika sudah benar.
- SPI berfungsi untuk menentukan pin komunikasi SPI (*Serial Peripheral Interface*) antara NRF24L01 agar dapat digunakan untuk *receiver* dan menerima hasil pengiriman data dari *transmitter* dengan menjalankan



algoritma *Priority Scheduling*. Konfigurasi SPI di *node master* memiliki no id "00"

- I2C AVR merupakan program yang digunakan untuk menangani komunikasi RTC DS1307. I2C digunakan untuk membaca dan menyimpan data waktu yang digunakan sistem.
- ADC digunakan untuk menangani pembacaan analog dari data sensor, arus dan sensor tegangan.
- Serial AVR merupakan program yang digunakan untuk mengirim data serial monitor ke laptop.
- One wire merupakan program yang digunakan untuk pembacaan sensor DHT11.

4.3 Kebutuhan Fungsional

Fungsi dari kebutuhan fungsional ialah digunakan untuk melihat apakah semua sistem pada setiap *node* dapat memberikan hasil yang benar dan sesuai kebutuhan penulis dan tujuan apa saja yang dapat dibentuk dari sistem tersebut. Pada penelitian ini, kebutuhan fungsional yang diperlukan adalah:

1. DHT11 memiliki fungsi pembacaan data

Fungsi dari pembacaan data sensor DHT11 adalah untuk keperluan melihat hasil data suhu dan kelembaban yang diambil pada jarak yang berbeda-beda dan untuk menghitung kelembaban pada jarak yang berbeda-beda modul sensor akan mendapatkan tegangan sebanyak 5V dan akan melakukan konversi pada besaran suhu dan *output* yang dihasilkan adalah kelembaban. Sebelum hasil data disimpan pada mikrokontroler, data tersebut harus diterima dahulu jika tidak maka data yang dihasilkan tidak sesuai.

2. Sensor ACS712-30A memiliki fungsi pembacaan data

Fungsi dari pembacaan data sensor ACS712-30A adalah untuk keperluan melihat hasil data arus yang diambil pada sistem yang digunakan. *Input* hasil data didapatkan dari besaran tegangan yang digunakan pada masing-masing *node* dan *output* didapatkan dengan mengkonversi besaran arus yang digunakan. Sebelum data disimpan pada mikrokontroler, data tersebut diterima terlebih dahulu di mikrokontroler.

3. Sensor tegangan memiliki fungsi pengambilan data

Fungsi dari pengambilan data pada sensor tegangan dibutuhkan untuk mengetahui hasil daya yang digunakan pada masing-masing *node*. Besaran input yang dihasilkan sebanyak 5V akan dikonversikan menjadi *output* pada besaran tegangan. Sebelum data disimpan pada mikrokontroler, data tersebut akan diterima terlebih dahulu di mikrokontroler.

4. Modul NRF24L01 *receiver* memiliki fungsi penerimaan data

Fungsi dari penerimaan data pada modul NRF24L01 dibutuhkan untuk menerima hasil data dari masing-masing *node* dengan menggunakan modul NRF24L01 yang berfungsi sebagai komunikasi *wireless*. Hasil data akan dikirimkan menggunakan *transmitter* NRF24L01 pada masing-masing *node* dan semua hasil data dapat dilihat di *serial monitor*.



5. Modul NRF24L01 *transmitter* memiliki fungsi pengiriman data. Fungsi dari pengiriman data pada modul NRF24L01 dibutuhkan untuk mengirimkan hasil data yang dihasilkan pada masing-masing *node*. Hasil data dikirimkan ke *receiver* modul NRF24L01 pada masing-masing *node* dan semua hasil data dapat dilihat di *serial monitor*.

6. Data Sensing memiliki fungsi pengolahan. Fungsi dari pengolahan data sensing dibutuhkan untuk menghasilkan data sensing yang baik dan monitoring akan berjalan secara *low power*. Data sensing yang digunakan adalah rumus yang terdapat pada *library* sehingga dapat menghasilkan keluaran sensor rumus. Keluaran hasil data yang dimaksud besaran suhu berupa $^{\circ}\text{C}$, kelembaban berupa %, nilai pada arus berupa mA, nilai tegangan berupa V, dan daya berupa mW.

4.4 Kebutuhan Non Fungsional

4.4.1 Karakteristik Pengguna

Pada bab ini peneliti melakukan monitoring data yang dihasilkan pada masing-masing *node* berupa data *sensing* melalui *serial monitor*. Jika terjadi kesalahan atau ingin mengganti *code program* peneliti disarankan untuk meng-*upload* ulang *code program* pada aplikasi Arduino IDE dan karakteristik pengguna pada sistem ini yaitu pasif.

4.4.2 Lingkungan Pengguna

Pengujian akan dilakukan pada sebuah ruangan tertutup dan ruangan terdapat halangan. Pada kedua pengujian masing masing *node* akan diletakkan pada jarak tertentu yang berguna untuk melihat nilai minimum dan maksimal pengiriman data yang akan dilakukan.

4.4.3 Asumsi dan Ketergantungan

Asumsi dan ketergantungan pada modul dan sensor yang akan digunakan pada sistem adalah:

1. Pemasangan alat-alat pada setiap board *node* tidak boleh ada penyimpangan.
2. Hasil akuisisi diharapkan mendapatkan hasil yang tepat, sensor yang digunakan untuk akuisisi data adalah DHT11, ACS712-30A, dan sensor tegangan.
3. Dapat menghitung hasil *sensing* keluaran setiap data *node* secara *real Time* menggunakan RTC DS1307.
4. *Priority Scheduling* harus berhasil mengirim dan menerima hasil setiap data daya *node* dari data daya terbesar hingga ke terkecil menggunakan NRF24L01.

4.4.4 Batasan Desain Sistem

Tujuan yang diharapkan peneliti dalam implementasi sistem dalam beberapa Batasan pada desain sistem adalah:

1. Peneliti menggunakan Arduino uno R3 sebagai mikrokontroler.



2. Peneliti menggunakan berbagai macam sensor, sensor yang pertama yaitu sensor ACS712-30A, sensor kedua yaitu sensor tegangan, dan yang terakhir yaitu sensor DHT11.
3. RTC DS1307 digunakan sebagai pengaturan *Time system* untuk waktu pengiriman dan penerimaan data.
4. Besaran tegangan yang dibutuhkan *node master* minimal sebesar 5V dan kabel USB berfungsi untuk menyambungkan *node master* ke laptop.
5. Besaran tegangan yang dibutuhkan masing-masing *node slave* minimal sebesar 5V dan kabel USB berfungsi untuk menyambungkan ke *powerbank* atau saklar.
6. NRF24L01 digunakan untuk mengirimkan dan menerima hasil data dari semua *node* dengan menggunakan *Priority Scheduling*.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Perancangan pada sistem terdiri dari beberapa penjelasan pengerjaan rekayasa kebutuhan. Proses yang akan dijelaskan pada bab ini adalah proses perancangan sistem secara *hardware* dan *software*, kemudian dilanjutkan dengan implementasi sistem pada penelitian ini. Selanjutnya penjelasan potongan kode sumber dan algoritma *Priority Scheduling* secara mekanisme.

5.1.1 Gambaran Umum Sistem

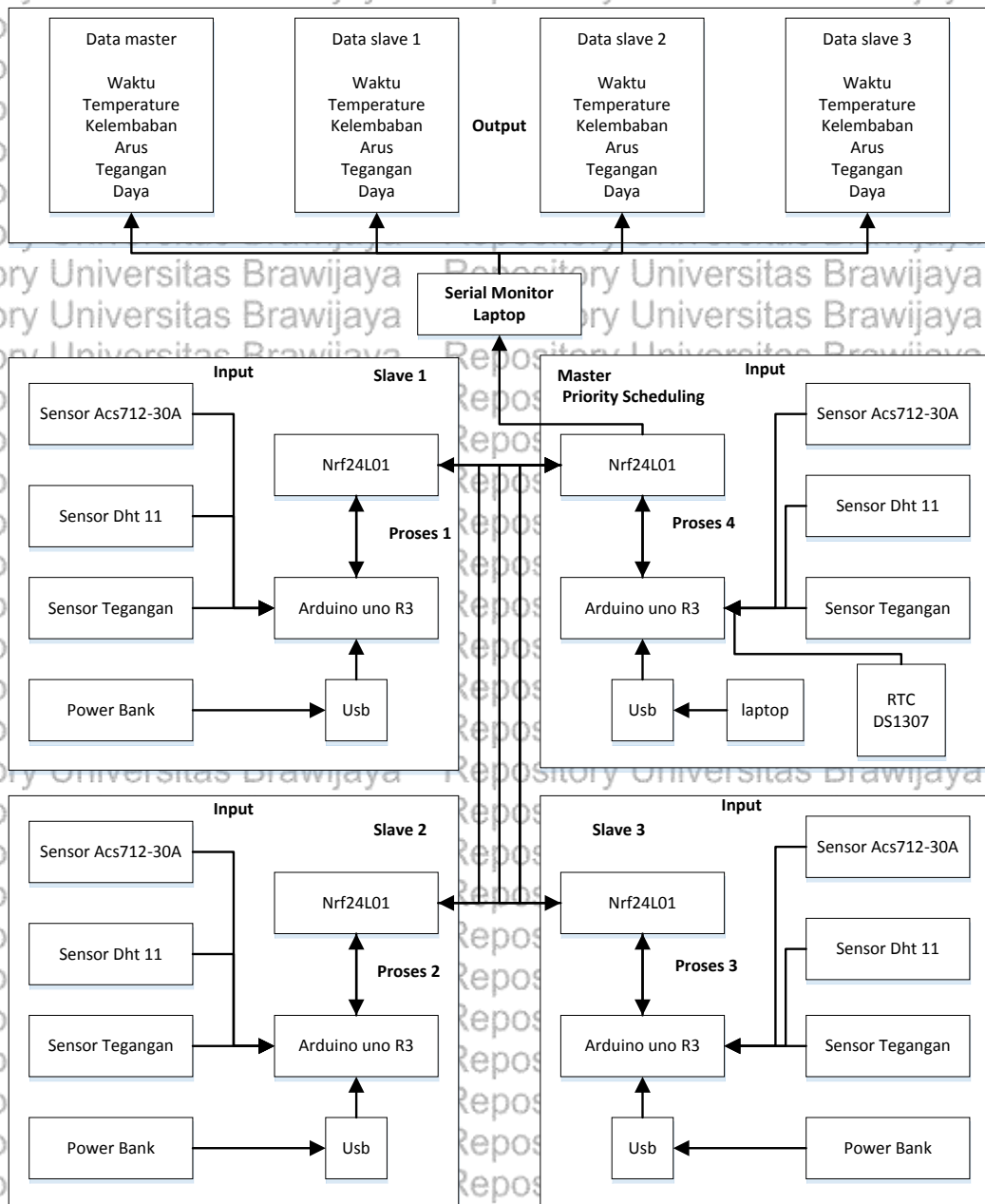
Perancangan umum pada pembuatan sistem terdapat 4 buah modul *node*, yang pertama *master node*, *slave 1 node*, *slave 2 node*, *slave 3 node*. Pada keempat modul *node* terdapat tiga fase yaitu *input*, proses, dan *output*. Fase input terdapat pada keempat modul *node* yang disebutkan terdapat berbagai macam komponen alat yang digunakan. Komponen pada modul *master node* dan *slave* berupa sensor DHT11, tegangan, ACS172-30A, *radio wireless* NRF24L01, USB, *power bank* atau saklar. Pada modul *master node* terdapat RTC DS1307 dan modul *slave node* tidak memiliki modul ini. Dari beberapa komponen alat yang sudah disebutkan terdapat beberapa fungsi yang akan diimplementasikan pada modul *master node* dan modul *slave node*. Pada sensor DHT11 digunakan untuk pengambilan data suhu dan kelembaban, sensor tegangan memiliki tegangan 5V yang digunakan untuk pengambilan data tegangan pada sistem modul *master node*, modul *slave node* dan berfungsi sebagai mencari data daya yang akan digunakan pada masing-masing modul *node*, pada sensor arus peneliti menggunakan sensor ACS712-30A yang akan digunakan untuk menghitung data arus pada setiap modul *node*. Hasil *sensing* yang didapatkan dari perhitungan sensor akan dibaca datanya dan akan disimpan datanya pada mikrokontroler Arduino uno R3 yang berfungsi sebagai *transmitter* dan *receiver* dalam menyimpan data. Selanjutnya modul *node master* mempunyai RTC DS1307 yang akan digunakan untuk menampilkan data *Time completion* yang merupakan hasil waktu untuk menyelesaikan waktu satu siklus percobaan pada semua *node*, *Time arrival* merupakan hasil waktu datang setiap *slave*, dan untuk menampilkan waktu data secara *real Time* pada masing-masing modul *node*. USB terdapat pada masing-masing modul *node* berfungsi untuk menyambungkan masing-masing modul *node* ke saklar atau *powerbank* yang berguna untuk menyalakan atau memberikan arus pada modul *node*.

Pada fase selanjutnya, yaitu fase proses merupakan pengiriman data pada setiap modul *node*. Pengiriman data menggunakan algoritma *Priority Scheduling* yang artinya pada penelitian ini data yang akan ditampilkan secara berurutan berdasarkan hasil dari besaran daya yang digunakan pada setiap modul *node*. Data akan ditampilkan secara berurutan dari data daya terbesar hingga ke terkecil pada setiap modul *node*. Suatu *node* memiliki daya yang paling besar disitu juga biasanya harus dicek apakah ada masalah pada modul tersebut atau tidak, sehingga tidak merusak sensor-sensor yang lainnya jika kelebihan daya. Pada setiap modul *node* terdapat *no id* yang digunakan untuk mengirimkan dan



menerima data secara benar dan tidak tertukar dalam pengiriman datanya. *Node master* memiliki *no id* 00, *slave 1 node* memiliki *no id* 01, *slave 2 node* memiliki *no id* 02, dan *slave 3 node* memiliki *no id* 03.

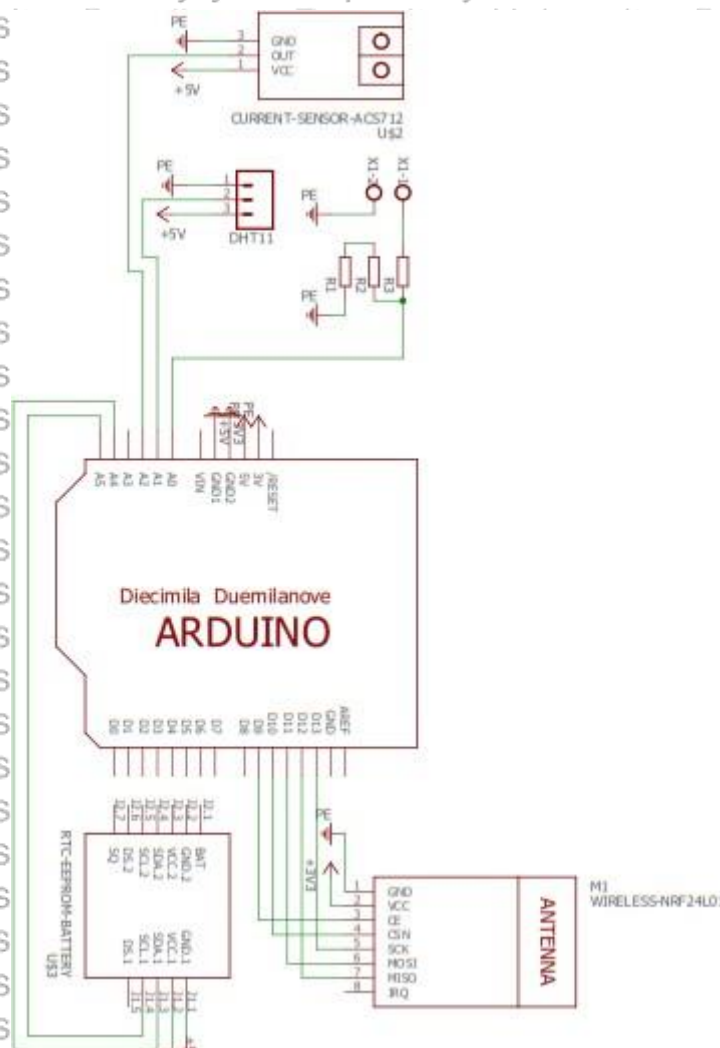
Pada fase hasil *output* modul *master node* akan membaca dan mengumpulkan semua data dari modul *slave node* terlebih dahulu dan modul *master node* akan membaca data sendiri. Setelah semua data modul *node* terkumpul, selanjutnya *master* akan membaca hasil data daya yang telah digunakan oleh setiap modul *node*. Setelah itu *master* akan menampilkan hasil data secara berurutan berdasarkan besarnya data daya hingga data daya ke terkecil. Dibawah ini merupakan Diagram blok yang merupakan gambaran umum terkait rancangan sistem yang dibuat pada Gambar 5.1.



Gambar 5.1 Diagram Blok Sistem

5.1.2 Perancangan Modul Master

Tujuan dirangkainya beberapa modul suatu sistem yang dapat bekerja untuk keperluan perancangan pada modul *master node* dapat dilihat pada Gambar 5.2 yang menunjukkan Gambar skematik diagram pada rangkaian modul *master node*.



Gambar 5.2 Skematik Diagram Modul Master

Mikrokontroler yang akan digunakan *master node* adalah Arduino uno R3 memiliki 20 pin input dan output dan akan terhubung pada masing-masing modul *node*. Berikut merupakan penjelasan konfigurasi antar pin komponen-komponen:

1. Pada rangkaian modul *master node* terdapat modul yang bertugas mendapatkan data hasil sensing dan mengirimkannya dari *slave* ke *master* atau sebaliknya dengan menggunakan media komunikasi nirkabel adalah modul NRF24L01. Besaran tegangan yang dibutuhkan dalam perancangan ini sebesar 3,3 V, tegangan 3,3V didapatkan dari tegangan pada saklar atau *powerbank* dengan perantara *supply* tegangan menggunakan USB. Berikut konfigurasi pin modul *master node* yang terdapat pada Tabel 5.1.



Tabel 5.1 Keterangan Pin NRF24L01 dan Pin Arduino Uno R3 Pada Master

Pin Modul NRF24L01	Pin Arduino Uno R3
CSN	D10
CE	D9
MOSI	D11
MISO	D12
SCK	D13
GND	GND
VCC	3V3

Berikut adalah deskripsi dari pin NRF24L01 dan pin Arduino uno R3 yang terdapat pada tabel *master node*:

- CSN (*Chip Select Not*) adalah pin *low* aktif dan biasanya tetap tinggi. Ketika pin ini menjadi rendah, NRF24L01 mulai mendengarkan pada port spi untuk data dan memprosesnya. D10 merupakan pin CS Arduino Uno R3.
- CE (*Chip Enable*) adalah pin aktif-tinggi. Ketika dipilih, NRF24L01 akan mengirim atau menerima yang digunakan untuk mengaktifkan komunikasi Arduino Uno R3 dengan NRF24L01.
- MOSI digunakan untuk pengiriman data dari NRF ke Arduino Uno R3, melalui pin 11. D11 sebagai penerima data dari Arduino Uno R3.
- MISO digunakan untuk penerimaan hasil data pada modul NRF24L01 menuju modul Arduino uno R3 melalui pin D12. Pin D12 sebagai pengirim data dari Arduino Uno R3.
- SCK digunakan sebagai clock untuk komunikasi antara Arduino Uno R3 dan NRF24L01.
- GND (*Ground*) merupakan pin yang digunakan untuk menghubungkan jalur ground antara NRF dan Arduino Uno R3.
- VCC berfungsi untuk menerima/mengalirkan arus listrik. Pin VCC NRF24L01 pada *master* merupakan input *Voltase* yang digunakan ke Arduino Uno R3 dengan besaran 3v3.

- Pada rangkaian modul *master node* mempunyai modul RTC DS1307 yang berfungsi dalam menampilkan hasil pengiriman waktu data secara *real Time* pada *master node*. Modul RTC membutuhkan penambahan *library* RTCLib pada program Arduino dalam proses pengerjaan. Pin SCL, SDA, GND, dan VCC merupakan konfigurasi pin RTC yang akan digunakan dan dapat dilihat pada Tabel 5.2.



Tabel 5.2 Keterangan Pin Modul RTCDS1307 dan Pin Arduino Uno R3 Pada Master

Pin Modul RTC1307	Pin Arduino Uno R3
SCL	A5
SDA	A4
GND	GND
VCC	VCC

Penjelasan pada Tabel 5.2 yang terdiri dari pin modul RTC DS1307 dan Arduino uno R3 pada *node master* adalah:

- Pada modul RTC DS1307 terdapat pin SCL yang digunakan mengirimkan dan menerima hasil data dari mikrokontroler menuju RTC DS1307 dengan menggunakan saluran *clock*. *Serial clock* yang digunakan untuk Arduino Uno R3 menuju RTC DS1307 adalah *serial clock line*.
- Pin SDA dalam modul RTC DS1307 digunakan untuk menyalurkan data yang digunakan untuk menghubungkan data dengan mikrokontroler menuju RTC DS1307. Pin A4 pada Arduino Uno R3 digunakan sebagai jalur pengiriman data waktu dari RTC DS1307.
- Ground rtc* ke *ground* Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.
- VCC merupakan sumber energy listrik utama. Pada modul rtc digunakan untuk sumber daya sebesar 5v.

- Pada rangkaian modul *master node* terdapat modul yang bertugas mendapatkan data hasil *sensing* yaitu data hasil kelembaban dan suhu dengan menggunakan sensor DHT11. Sensor DHT11 akan melakukan perhitungan *sensing* pada *master node* dan hasil datanya akan dikirimkan ke mikrokontroler. Mikrokontroler memiliki pin DATA yang berfungsi sebagai penerimaan hasil data input menuju modul sensor DHT11, kemudian program Arduino Uno R3 akan mengolah data input yang didapatkan dari sensor DHT11 menjadi data suhu dengan pengujian yang dilakukan berbagi macam jarak. Berikut konfigurasi pin modul *master node* yang terdapat pada Tabel 5.3.

Tabel 5.3 Keterangan Pin Sensor DHT11 Pada Master

Pin Sensor DHT11	Pin Arduino Uno R3
DATA	A1
GND	GND
VCC	VCC

Penjelasan pada Tabel 5.3 yang terdiri dari pin modul DHT11 dan Arduino uno R3 pada *master node* adalah:

- Data merupakan jalur komunikasi data antara Arduino Uno R3 dan DHT11.



- b. *Ground* DHT11 ke *ground* Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.
- c. VCC merupakan sumber listrik utama. Pada modul DHT11 digunakan untuk sumber daya sebesar 5v.
4. Pada rangkaian modul *master node* terdapat modul yang bertugas menghitung suatu arus pada perancangan setiap *node* yang telah diterapkan dengan menggunakan modul sensor ACS712-30A. Terdapat beberapa pin yang digunakan pada modul ini adalah pin Sensor memiliki pin VCC, GND, *INPUT*, dan *OUTPUT*. Sensor ini memiliki kemampuan menghitung suatu arus sebanyak 30A. Berikut konfigurasi pin modul *master node* yang terdapat pada Tabel 5.4.

Tabel 5.4 Keterangan Pin Sensor ACS712-30A Pada *Master*

Pin ACS712-30A	Pin Arduino Uno R3
<i>OUTPUT</i>	A2
GND	GND
VCC	VCC

Penjelasan pada Tabel 5.4 yang terdiri dari pin sensor ACS712-30A dan Arduino uno R3 pada *master node* adalah:

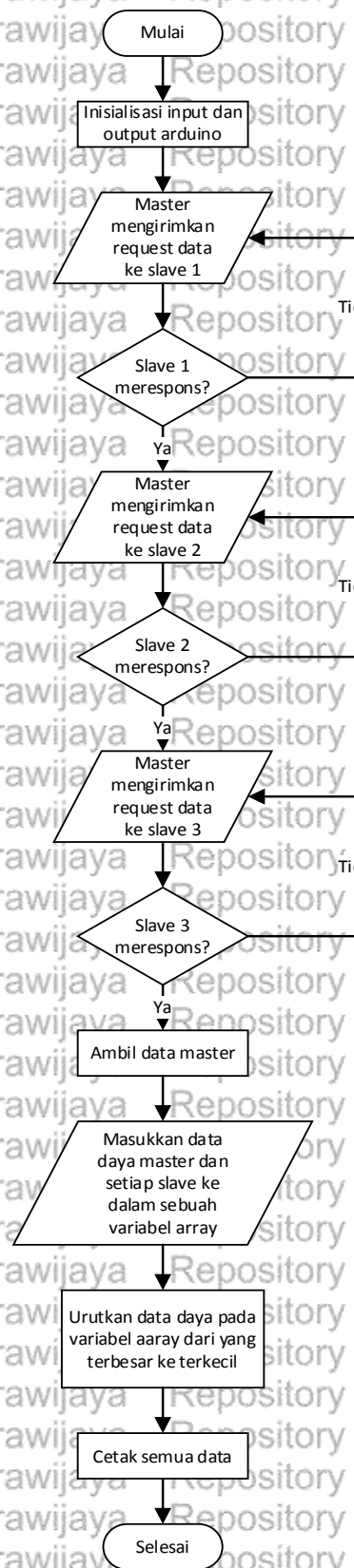
- a. *Output* pin ACS pada *master* digunakan sebagai *ouput* tegangan yang masuk ke pin ADC yaitu pin A2.
- b. *Ground* ACS712-30A ke *ground* Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.
- c. VCC pin ACS712-30A ke Vcc Arduino Uno R3 berfungsi sebagai mengukur arus dan digunakan sebagai sumber daya.
5. Pada rangkaian modul *master node* terdapat modul yang bertugas menghitung tegangan data yang digunakan pada setiap *node* dengan menggunakan sensor tegangan. Sensor tegangan memiliki 3 buah resistor yang diimplementasikan untuk menampilkan hasil nilai yang rendah jika bedakan dengan nilai masukkan dan sensor ini juga bisa disebut dengan rentengan pembagi tegangan. Berikut konfigurasi pin modul *master node* yang terdapat pada Tabel 5.5

Tabel 5.5 Keterangan Pin Sensor Tegangan Pada *Master*

Pin Sensor Tegangan	Pin Arduino Uno R3
VCC	VCC
<i>OUTPUT</i>	A0
GND	GND

Penjelasan pada Tabel 5.5 yang terdiri dari pin sensor tegangan dan Arduino uno R3 pada *master node* adalah:

- a. VCC pada pin sensor tegangan ke pin Arduino Uno R3 memberikan tegangan pada Arduino Uno R3.
- b. *Output* merupakan *output* tegangan yang masuk ke pin ADC (A0).
- c. Ground pada pin sensor tegangan ke ground pin Arduino Uno R3 uno berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.



Gambar 5.3 Flowchart Diagram Modul Master



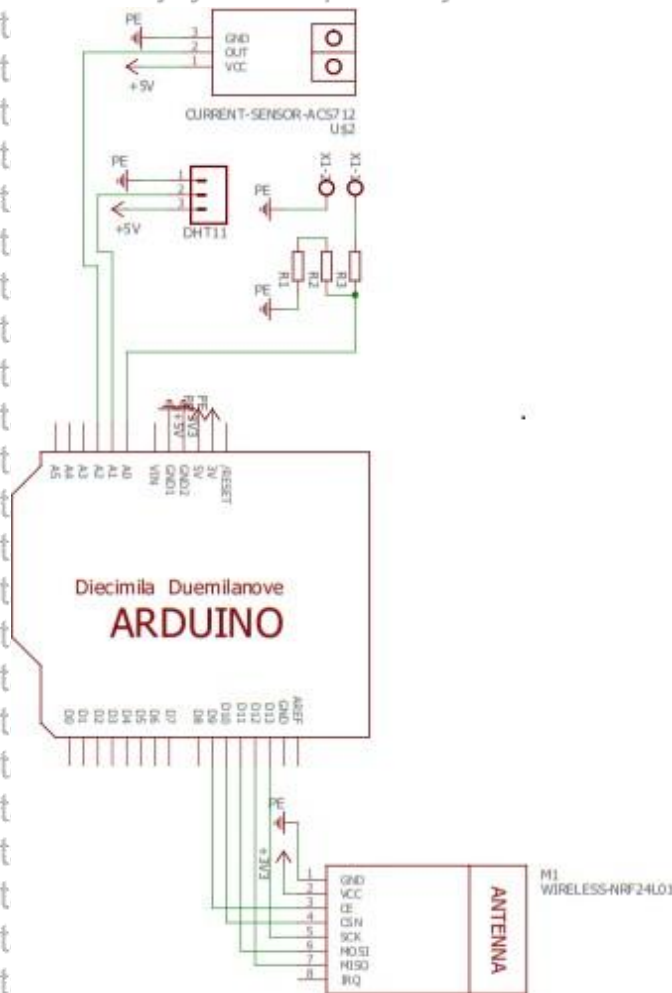
Pada modul *master node* terdapat bentuk dan alur berjalannya modul *master node* yang dapat dilihat pada hasil *flowchart diagram* pada Gambar 5.3.

- a. Ketika Perangkat modul *master node* diaktifkan, *master node* akan melakukan *inisialisasi input dan output* pada Arduino Uno R3. Deklarasi beberapa *library* yang digunakan dalam pembuatan program. *Library* SPI adalah *library* yang digunakan sebagai *library* untuk komunikasi secara *Serial Peripheral Interface*, dimana NRF24L01 berkomunikasi secara SPI dengan Arduino Uno R3, *library* Wire adalah *library* yang digunakan sebagai komunikasi secara I2C oleh modul RTC DS1307 dengan Arduino Uno R3. *Library* NRF24L01 dan RF24Network merupakan *library* yang digunakan untuk melakukan perintah dan penerimaan data dengan modul NRF24L01. *Library* RTCLib merupakan *library* yang digunakan untuk meakukan perintah dan menerima data dengan modul RTC DS1307. *Library* Dht merupakan *library* yang digunakan untuk berkomunikasi dengan modul DHT11 dan *library* LowPower merupakan *library* yang digunakan untuk membuat Arduino Uno R3 bekerja dengan mode low power. Selanjutnya mendefinisikan pin analog A1 sebagai pin untuk dht dan pembuatan objek RF24, RF24Network, DHT, RTC. Setelah pembuatan objek dan definisi pin, selanjutnya deklarasi *variable* array untuk memasukkan nama hari secara *real Time*. Deklarasi *variable* alamat terdiri dari *slave* 1 dengan *no id* 01, *slave* 2 dengan *no id* 02, *slave* 3 dengan *no id* 03, dan *master* dengan *no id* 00. Deklarasi selanjutnya adalah deklarasi *variable* untuk penyimpanan data *millisecond* sebelum eksekusi dilakukan, *variable* penyimpanan waktu setiap *slave* dan *variable* untuk alamat. Deklarasi tipe data *struct* pada *master* digunakan untuk menyimpan beberapa *variable* data *slave*. Selanjutnya pembuatan deklarasi data *struct dataClient* untuk menyimpan *variable-variable* *dataZero* atau data *master*, *dataOne* atau data *slave* 1, *dataTwo* atau data *slave* 2, *dataThree* atau data *slave* 3. Setelah itu pembuatan deklarasi *array* untuk menyimpan nilai daya dari masing-masing *slave* dan menyimpan sementara data-data daya pada waktu proses pengurutan *Priority Scheduling*.
- b. Setelah modul *master* selesai melakukan insialiasi *input dan output* pada Arduino Uno R3, modul *master* akan mengirimkan *request* ke masing-masing *slave* 1, 2, 3. Jika tidak ada balasan maka modul *master* akan terus mengirimkan *request* ke *slave* 1, 2, 3.
- c. Modul *master* akan mengecek apakah semua *request* sudah terkirim ke masing-masing *node*. Program dimulai dengan melakukan pengecekan nomor id yang masuk. Jika id yang masuk adalah 0 maka data *master* ditampilkan, jika id yang masuk adalah 1 maka data dari *slave* 1 ditampilkan, jika id yang masuk adalah 2 maka data yang ditampilkan adalah *slave* 2, dan jika id yang masuk adalah 3 maka data yang ditampilkan adalah data *slave* 3.
- d. Jika ada jawaban dari *slave* maka *master* akan membaca data Arus, Tegangan, Suhu, Kelembaban pada masing-masing *slave*.
- e. Setelah pembacaan data *slave* selesai maka *master* akan membaca data sendiri menjadi sebuah data array. Pada saat semua data masuk, maka akan dilakukan pengecekan terhadap data daya dalam *variable array* tersebut. Data daya dalam *variable array* tersebut akan di urutkan dari yang terbesar dan kemudian

akan ditampilkan atau mencetak semua data sesuai dengan prioritas daya dari yang terbesar terlebih dahulu hingga terkecil.

5.1.3 Perancangan Modul *Slave 1*, *Slave 2* dan *Slave 3*

Perancangan pada modul *slave 1 node*, *slave 2* dan *slave 3* tidak jauh berbeda dengan modul *master node* terdapat beberapa sensor yang digunakan yaitu sensor arus, sensor tegangan, sensor DHT11, USB, dan wireless modul NRF24L01. Perbedaan antara modul *slave 1 node*, *slave 2*, *slave 3* dan *master node* adalah tidak adanya modul RTC pada setiap modul *slave node*. RTC pada modul *master* digunakan untuk menghitung hasil waktu penerimaan dan pengiriman data ke setiap *slave node*. Power yang digunakan pada setiap *slave node* untuk menyalakan board yaitu power bank, sedangkan pada *master node* menggunakan kabel usb yang dihubungkan ke laptop untuk menyalakan *board master*. Skematik diagram rangkaian pada modul *slave 1 node*, *slave 2* dan *slave 3* dapat dilihat di Gambar 5.4.



Gambar 5.4 Skematik Diagram Modul *Slave 1 Node*, *Slave 2* dan *Slave 3*



Mikrokontroler yang akan digunakan *slave 1 node*, *slave 2 node*, dan *slave 3 node* adalah Arduino uno R3 memiliki 20 pin input dan output dan akan terhubung pada masing-masing modul *node*. Berikut merupakan penjelasan konfigurasi antar pin komponen-komponen:

1. Pada rangkaian modul masing-masing *slave node* terdapat modul yang bertugas mendapatkan data hasil sensing dan mengirimkannya dari *slave* ke *master* atau sebaliknya dengan menggunakan media komunikasi nirkabel adalah modul NRF24L01. Besaran tegangan yang dibutuhkan dalam perancangan ini sebesar 3,3 V, tegangan 3,3V didapatkan dari tegangan pada saklar atau *powerbank* dengan perantara *supply* tegangan menggunakan USB. Berikut konfigurasi pin modul setiap *slave node* yang terdapat pada Tabel 5.6.

Tabel 5.6 Keterangan Pin NRF24L01 dan Pin Arduino Uno R3 Setiap *Slave*

Pin Modul NRF24L01	Pin Arduino Uno R3
CSN	D10
CE	D9
MOSI	D11
MISO	D12
SCK	D13
GND	GND
VCC	3V3

Berikut adalah deskripsi dari pin NRF24L01 dan pin Arduino uno R3 yang terdapat pada tabel setiap *slave node*:

- CSN (*Chip Select Not*) adalah pin *low* aktif dan biasanya tetap tinggi. Ketika pin ini menjadi rendah, NRF24L01 mulai mendengarkan pada port spi untuk data dan memprosesnya. D10 merupakan pin CS Arduino Uno R3.
- CE (*Chip Enable*) adalah pin aktif-tinggi. Ketika dipilih, NRF24L01 akan mengirim atau menerima yang digunakan untuk mengaktifkan komunikasi Arduino Uno R3 dengan NRF24L01.
- MOSI digunakan untuk pengiriman data dari NRF ke Arduino Uno R3 melalui pin 11. D11 sebagai penerima data dari Arduino Uno R3.
- MISO digunakan untuk penerimaan hasil data pada modul NRF24L01 menuju modul Arduino uno R3 melalui pin D12. Pin D12 sebagai pengirim data dari Arduino Uno R3.
- SCK digunakan sebagai clock untuk komunikasi antara Arduino Uno R3 dan NRF24L01.
- GND (*Ground*) merupakan pin yang digunakan untuk menghubungkan jalur ground antara nrf dan Arduino Uno R3.
- VCC berfungsi untuk menerima/mengalirkan arus listrik. Pin Vcc NRF24L01 pada setiap *slave node* merupakan input *Voltase* yang digunakan ke Arduino Uno R3 dengan besaran 3v3.



2. Pada rangkaian modul setiap *slave node* terdapat modul yang bertugas mendapatkan data hasil *sensing* yaitu data hasil kelembaban dan suhu dengan menggunakan sensor DHT11. Sensor DHT11 akan melakukan perhitungan *sensing* pada setiap *slave node* dan hasil datanya akan dikirimkan ke mikrokontroler. Mikrokontroler memiliki pin DATA yang berfungsi sebagai penerimaan hasil data input menuju modul sensor DHT11, kemudian program Arduino Uno R3 akan mengolah data input yang didapatkan dari sensor DHT11 menjadi data suhu dengan pengujian yang dilakukan berbagi macam jarak. Berikut konfigurasi pin modul setiap *slave node* yang terdapat pada Tabel 5.7.

Tabel 5.7 Keterangan Pin Sensor DHT11 Pada Setiap *Slave*

Pin Sensor DHT11	Pin Arduino Uno R3
DATA	A1
GND	GND
VCC	VCC

Penjelasan pada Tabel 5.7 yang terdiri dari pin modul DHT11 dan Arduino uno R3 pada setiap *slave node* adalah:

- Data merupakan jalur komunikasi data antara Arduino Uno R3 dan DHT11.
 - Ground DHT11 ke ground Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.
 - VCC merupakan sumber listrik utama. Pada modul DHT11 digunakan untuk sumber daya sebesar 5v.
3. Pada rangkaian modul setiap *slave node* terdapat modul yang bertugas menghitung suatu arus pada perancangan setiap *slave node* yang telah diterapkan dengan menggunakan modul sensor ACS712-30A. Terdapat beberapa pin yang digunakan pada modul ini adalah pin Sensor memiliki pin VCC, GND, *INPUT*, dan *OUTPUT*. Sensor ini memiliki kemampuan menghitung suatu arus sebanyak 30A. Berikut konfigurasi pin modul setiap *slave node* yang terdapat pada Tabel 5.8.

Tabel 5.8 Keterangan Pin Sensor ACS712-30A Pada Setiap *Slave*

Pin ACS712-30A	Pin Arduino Uno R3
<i>OUTPUT</i>	A2
GND	GND
VCC	VCC

Penjelasan pada Tabel 5.8 yang terdiri dari pin sensor ACS712-30A dan Arduino uno R3 pada setiap *slave node* adalah:

- Output* pin ACS setiap *slave node* digunakan sebagai *ouput* tegangan yang masuk ke pin ADC yaitu pin A2.



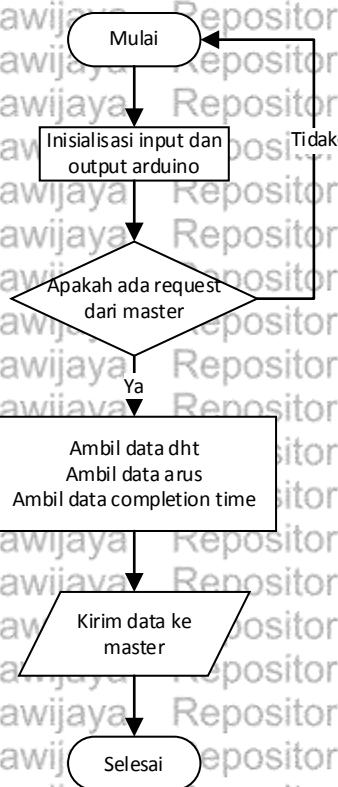
- b. Ground ACS712-30A ke *ground* Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.
- c. VCC pin ACS712-30A ke VCC Arduino Uno R3 berfungsi sebagai mengukur arus dan digunakan sebagai sumber daya.
4. Pada rangkaian setiap *slave node* terdapat modul yang bertugas menghitung tegangan data yang digunakan pada setiap *node* dengan menggunakan sensor tegangan. Sensor tegangan memiliki 3 buah resistor yang diimplementasikan untuk menampilkan hasil nilai yang rendah jika bedakan dengan nilai masukan dan sensor ini juga bisa disebut dengan rentengan pembagi tegangan. Berikut konfigurasi pin modul setiap *slave node* yang terdapat pada Tabel 5.9.

Tabel 5.9 Keterangan Pin Sensor Tegangan Pada Setiap *Slave*

Pin Sensor Tegangan	Pin Arduino Uno R3
VCC	VCC
OUTPUT	A0
GND	GND

Penjelasan pada Tabel 5.9 yang terdiri dari pin sensor tegangan dan Arduino uno R3 pada setiap *slave node* adalah:

- a. VCC pada pin sensor tegangan ke pin Arduino Uno R3 memberikan tegangan pada Arduino Uno R3.
- b. *Output* merupakan *output* tegangan yang masuk ke pin adc (a0).
- c. Ground pada pin sensor tegangan ke ground pin Arduino Uno R3 berfungsi sebagai menyeimbangkan beda potensial sehingga tidak terjadi kebocoran pada tegangan dan arus.



Gambar 5.5 Flowchart Diagram Modul Slave 1 Node, Slave 2 dan Slave 3

Dapat dilihat pada Gambar 5.5 yang terdiri dari beberapa alat pada rangkaian pada *slave 1 node*, *slave 2 node*, dan *slave 3 node* beserta kegunaannya adalah:

- Ketika perangkat setiap *slave node* dijalankan, setiap *slave node* akan melakukan inisialisasi *input* dan *ouput* pada Arduino Uno R3 dilanjutkan dengan deklarasi beberapa *library* yang digunakan dalam pembuatan program. *Library* SPI adalah *library* yang digunakan sebagai *library* untuk komunikasi secara Serial Peripheral Interface, dimana NRF24L01 berkomunikasi secara SPI dengan Arduino Uno R3, *library* Wire adalah *library* yang digunakan sebagai komunikasi secara I2C oleh modul RTC DS1307 dengan Arduino Uno R3. *Library* NRF24L01 dan RF24Network merupakan *library* yang digunakan untuk melakukan perintah dan penerimaan data dengan modul NRF24L01. *Library* RTCLib merupakan *library* yang digunakan untuk melakukan perintah dan menerima data dengan modul RTC DS1307. *Library* Dht merupakan *library* yang digunakan untuk berkomunikasi dengan modul DHT11 dan *library* LowPower merupakan *library* yang digunakan untuk membuat Arduino Uno R3 bekerja dengan mode *low power*. Setelah deklarasi beberapa *library*, selanjutnya pembuatan object RF24, pembuatan RF24Network. Setelah melakukan deklarasi beberapa *library*, kemudian dilanjutkan dengan pembuatan *object* RF24, RF24Network, dan DHT. Kemudian dilanjutkan dengan pembuatan deklarasi *variable* alamat *master* dan alamat masing-masing *slave*. Setiap *slave* memiliki no id, *slave 1* memiliki no id 01, *slave 2* memiliki no id 02, *slave 3* memiliki no id 03, dan *master* memiliki no id 00. Deklarasi selanjutnya adalah deklarasi *variable* untuk penyimpanan data



millisecond sebelum eksekusi dilakukan, *variable* penyimpanan waktu setiap *slave*, dan *variable* untuk alamat. Deklarasi tipe data *struct* pada masing-masing *slave* digunakan untuk menyimpan beberapa *variable* data *slave* atau data *master*.

- b. Setelah *slave* melakukan inisialisasi *input* dan *output* pada Arduino Uno R3, setiap *slave* akan mengecek apakah ada *request* untuk permintaan data dari *master*, jika tidak ada maka akan melakukan pengulangan ke inisialisasi *input* dan *output* sampai cek radio komunikasi masuk atau diterima.
- c. Jika ada *request* setiap *slave* akan mengambil data DHT, data arus yang dibutuhkan masing - masing *slave*, data *Time completion* yang merupakan hasil waktu untuk menyelesaikan waktu satu siklus percobaan pada semua *node*, *Time arrival* merupakan hasil waktu datang setiap *slave* menuju client dan dibutuhkan arus yang digunakan masing-masing *slave*.
- d. Setelah masing-masing *slave* mengambil data DHT, data arus, data *Time completion*, dan data *Time arrival* selanjutnya masing-masing *slave* mengirimkan data ke *master*. Setiap *slave* memiliki no id, no id *slave node* 1 adalah 1, *slave node* 2 adalah 2, *slave node* 3 adalah 3.

5.1.4 Perancangan Mekanisme Algoritma *Priority Scheduling*

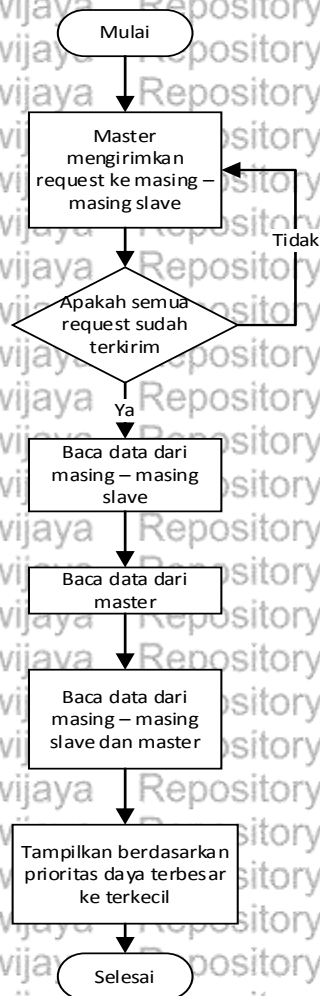
Priority Scheduling memiliki pengertian penjadwalan algoritma yang mendahulukan proses yang memiliki prioritas tertinggi. *Priority Scheduling* pada tiap-tiap proses dilengkapi dengan nomor prioritas, *Priority Scheduling* memiliki skema yaitu non-preemptive. Pada sistem yang telah dirancang hasil data yang didahulukan atau diprioritaskan adalah kiriman data berdasarkan dari daya terbesar hingga ke terkecil pada modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node*. Selanjutnya akan diurutkan dan ditampilkan dari data daya terbesar hingga ke terkecil. Kiriman data antara keempat *node* tersebut tidak ada bedanya, yang membedakan adalah id yang sudah dibuat di keempat *node* tersebut. Alur berjalannya perancangan mekanisme algoritma *Priority Scheduling* dilihat hasil *flowchart diagram* dan bentuknya pada Gambar 5.6.

- a. *Master Node* mengirimkan *request* ke masing-masing *slave*. Jika tidak ada balasan maka modul *master* akan terus mengirimkan *request* ke masing-masing *slave*.
- b. Modul *master* akan mengecek apakah semua *request* sudah terkirim ke masing-masing *slave*. Program dimulai dengan melakukan pengecekan nomor id yang masuk. Jika id yang masuk adalah 00 maka data *master* ditampilkan, jika id yang masuk adalah 01 maka data dari *slave 1* ditampilkan, jika id yang masuk adalah 02 maka data yang ditampilkan adalah *slave 2*, dan jika id yang masuk adalah 03 maka data yang ditampilkan adalah data *slave 3*.
- c. Setelah semua *request* sudah terkirim, *master* akan membaca data dari setiap *slave node*. Hasil data yang telah *master node* baca dari setiap *slave node* yaitu data arus, tegangan, suhu, kelembaban, *Time completion*, dan *Time arrival*.
- d. Setelah selesai membaca data dari setiap *slave*, maka *master* akan membaca data arus, tegangan, suhu, kelembaban, *Time completion*, dan *Time arrival* menjadi sebuah data array.



e. Pada saat semua data masuk, program dimulai dengan membuat Arduino Uno R3 dalam mode low power dengan membuat Arduino Uno R3 dalam mode sleep selama 2 detik dan membuat BOD dan ADC Off. Kemudian perintah `network.update()` adalah untuk memeriksa apakah ada data masuk dari masing-masing *slave*. Jika ada data masuk maka lihat dulu id *nodenya*, jika id = 1 maka masukkan data akan menuju ke dalam *variable struct* `dataOne` dan memasukkan data daya ke dalam variabel array `dataPower`, jika id = 2 maka masukkan data akan menuju ke dalam *variable struct* `dataTwo` dan memasukkan data daya ke dalam *variable* array `dataPower`. Jika id = 3 maka masukkan data akan menuju ke dalam *variable struct* `dataThree` dan memasukkan data daya ke dalam *variable* array `dataPower`. Kemudian program berlanjut dengan memanggil *slave* 1, *slave* 2, dan *slave* 3 yang dimana jika ada jawaban dari *slave* tadi maka hasil akan ditangkap oleh perintah `network.update()` diatas. Kemudian *master* akan membaca data suhu dan kelembaban dan juga data arus dan data daya dimasukkan ke *variable* array data daya.

f. Setelah semua data masuk, maka akan dilakukan pengecekan terhadap data daya dalam *variable* array tersebut. Data daya dalam *variable* array akan diurutkan datanya dengan menggunakan *Priority Scheduling* dan menghasilkan tampilan data berdasarkan prioritas hasil data daya terbesar hingga ke terkecil yang akan ditampilkan.



Gambar 5.6 Flowchart Algoritma Priority Scheduling

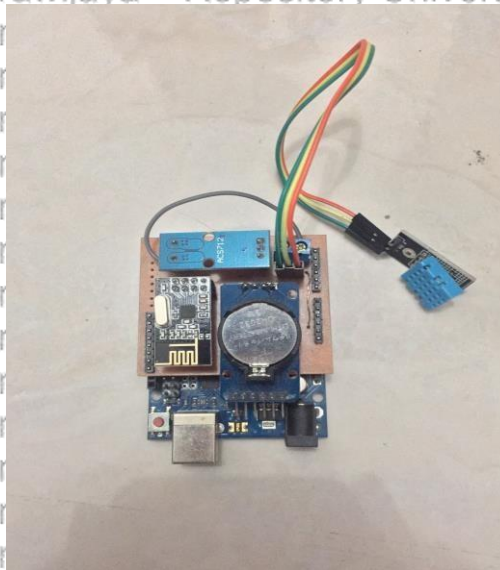
5.2 Implementasi Sistem

Jika perancangan sistem sudah selesai dikerjakan selanjutnya implementasi sistem akan dapat dikerjakan. Tahapan-tahapan implementasi sistem pada modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* merupakan acuan tahapan perancangan yang akan dikerjakan dalam sistem ini.

5.2.1 Implementasi Sistem Pada Modul *Node Master*

Perancangan sistem dan pembahasan akan dikerjakan sesuai dengan implementasi sistem pada modul *master node*. Terdapat beberapa alat pada modul *master node* yang akan digunakan dalam perancangan pada sistem ini. Perancangan alat yang digunakan adalah Arduino uno R3 untuk mikrokontroler, Kabel USB untuk pemberi arus dan daya pada setiap alat *master node*, modul NRF24L01 untuk komunikasi nirkabel, RTC DS1307 yang akan digunakan untuk menampilkan data *Time completion* yang merupakan hasil waktu untuk menyelesaikan waktu satu siklus percobaan pada semua *node*, *Time arrival* merupakan hasil waktu datang setiap *slave*, dan untuk menampilkan waktu data secara *real Time* pada masing-masing modul *node*, ACS712-30A untuk sensor arus

dan sensor tegangan untuk menghitung tegangan yang dibutuhkan dan mencari hasil daya pada *master node*. Hasil implementasi sistem pada modul *master node* dapat dilihat pada Gambar 5.7.



Gambar 5.7 Implementasi Sistem Pada Modul *Master Node*

Dalam implementasi sistem pada perangkat lunak dibutuhkan Arduino IDE yang digunakan untuk pengerjaan bahasa pemrograman C dan juga diperlukannya beberapa komposisi dalam keperluan menjalankan fungsi pada setiap modul sensor dan komunikasi yang digunakan di modul *master node*.

Tabel 5.10 Potongan Kode Sumber Inisialisasi Input dan *Output* Pada Arduino

No	Potongan Kode Sumber
1	#include <SPI.h>
2	#include <Wire.h>
3	#include <RTClib.h>
4	#include <RF24Network.h>
5	#include <RF24Network_config.h>
6	#include <Sync.h>
7	#include <RF24L01.h>
8	#include <printf.h>
9	#include <RF24.h>
10	#include <RF24_config.h>
11	#include <LowPower.h>
12	#include <dht.h>
13	#define dht_pin A1
14	RF24 radio(9,10);
15	RF24Network network(radio);
16	dht DHT;
17	RTC_DS1307 rtc;



18	char daysOfTheWeek[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu"};
19	const uint16_t writingAddressOne=01;
20	const uint16_t writingAddressTwo=02;
21	const uint16_t writingAddressThree=03;
22	const uint16_t thisAddress = 00;
23	unsigned long firstTime;
24	unsigned long TimeTravel1, TimeTravel2, TimeTravel3;
25	uint8_t add;
26	struct dataClient{
27	float id;
28	float temp;
29	float hum;
30	float current;
31	unsigned long TimeArrival;
32	unsigned long TimeCompletion;
33	};
34	dataClient dataZero;
35	dataClient dataOne;
36	dataClient dataTwo;
37	dataClient dataThree;
38	dataClient data;
39	float dataPower[4], tempPower;
40	byte temp, buf[4];

Potongan kode sumber pada Tabel 5.10 memiliki pernyataan sebagai berikut:

- Potongan kode sumber No.1-12 adalah inisialisasi *library* yang diimplementasikan pada modul *master node*. Program *library* yang digunakan berupa *library* SPI karena dapat berkomunikasi secara Serial Peripheral Interface, dimana NRF24L01 berkomunikasi secara SPI dengan Arduino Uno R3. *Library* NRF24L01 dan RF24Network merupakan *library* yang digunakan untuk melakukan perintah dan penerimaan data dengan modul NRF24L01. *Library* RTCLib merupakan *library* yang digunakan untuk melakukan perintah dan menerima data dengan modul RTC1307 DS Selain itu *library* dht merupakan *library* yang digunakan untuk berkomunikasi dengan modul DHT11. Selanjutnya potongan kode sumber No.13 adalah definisi pin untuk DHT11 yaitu menggunakan pin Analog A1.
- Potongan kode sumber No.14-16 adalah pembuatan objek RF24, RF24Network, DHT, dan RTC.
- Potongan kode sumber No.18 merupakan deklarasi *variable* array untuk memasukkan nama hari berdasarkan Bahasa Indonesia dan nama hari secara real *Time*.
- Potongan kode sumber No.19-22 adalah Deklarasi *variable* alamat terdiri dari *slave* 1 dengan no id 01, *slave* 2 dengan no id 02, *slave* 3 dengan no id 03, dan *master* dengan no id 00.



- e. Potongan kode sumber No.23-24 adalah deklarasi *variable* yang berfungsi menyimpan hasil keluaran data berupa mili second sebelum eksekusi dilakukan, *variable* penyimpanan waktu setiap *slave* dan *variable* untuk alamat.
- f. Potongan kode sumber No.25-33 adalah deklarasi bertipe data *struct* pada *master* digunakan untuk menyimpan beberapa *variable* data *slave*.
- g. Potongan kode sumber No.34-38 adalah deklarasi data *struct* dataClient untuk menyimpan *variable-variable* dataZero atau data *master*, dataOne atau data *slave 1 node*, dataTwo atau data *slave 2 node*, dataThree atau data *slave 3 node*.
- h. Potongan kode sumber No.39-40 adalah deklarasi array berfungsi menyimpan hasil bilangan daya dari masing-masing *slave* dan menyimpan sementara data-data daya pada waktu proses pengurutan *Priority Scheduling*.

5.2.2 Urutan Berjalannya Modul Master Node

Sesudah melakukan inisialisasi *input* dan *output* pada *master node*, dilanjutkan dengan mengerjakan codingan dan dapat dilihat pada Tabel 5.11.

Tabel 5.11 Potongan Kode Sumber Master Mengirimkan Request Ke Masing-Masing Slave

No	Potongan Kode Sumber
1	adds = 1;
2	RF24NetworkHeader header2(writingAddressOne);
3	bool ok = network.write(header2, &adds, sizeof(adds));
4	adds = 2;
5	RF24NetworkHeader header3(writingAddressTwo);
6	bool ok2 = network.write(header3, &adds, sizeof(adds));
7	adds = 3;
8	RF24NetworkHeader header4(writingAddressThree);
9	bool ok3 = network.write(header4, &adds, sizeof(adds));

Potongan kode sumber pada Tabel 5.11 memiliki pernyataan sebagai berikut:

- a. Potongan kode sumber No.1-3 adalah *variable* adds pada *slave 1* adalah 1 dan pembuatan object RF24NetworkHeader untuk *slave 1*. Selanjutnya deklarasi *variable* ok dengan tipe data Boolean untuk melihat apakah pengiriman data berhasil atau tidak. Jika ok maka cetak pernyataan dibawah pada serial monitor.
- b. Potongan kode sumber No.4-6 merupakan *variable* adds pada *slave 2* adalah 2 dan pembuatan object RF24NetworkHeader untuk *slave 2*. Selanjutnya deklarasi *variable* ok dengan tipe data Boolean untuk melihat apakah pengiriman data berhasil atau tidak. Jika ok maka cetak pernyataan dibawah pada serial monitor.
- c. Potongan kode sumber No.7-9 adalah *variable* adds pada *slave 3* adalah 3 dan pembuatan object RF24NetworkHeader untuk *slave 3*. Selanjutnya deklarasi *variable* ok dengan tipe data Boolean untuk melihat apakah pengiriman data berhasil atau tidak. Jika ok maka cetak pernyataan dibawah pada serial monitor.



Tabel 5.12 Potongan Kode Sumber Master Mengecek Apakah Semua Request Data Sudah Terkirim Ke Setiap Slave

No	Potongan Kode Sumber
1	<code>bool ok = network.write(header2, &adds, sizeof(adds));</code>
2	<code>bool ok2 = network.write(header3, &adds, sizeof(adds));</code>
3	<code>bool ok3 = network.write(header4, &adds, sizeof(adds));</code>

Potongan kode sumber pada Tabel 5.12 memiliki pernyataan sebagai berikut:

- Potongan kode sumber No.1-3 adalah pengecekan apakah semua data *request* sudah terkirim ke *slave*. Jika *variable* *adds* sudah ok maka id *slave* 1 *node* adalah 1, *slave* 2 *node* adalah 2, *slave* 3 *node* adalah 3 maka akan mencetak pernyataan dibawah serial monitor.

Tabel 5.13 Potongan Kode Sumber Baca Data Arus, Tegangan, Suhu, Kelembaban Pada Setiap Slave

No	Potongan Kode Sumber
1	<code>switch(header.from_node){</code>
2	<code>case 1:</code>
3	<code>dataOne = data;</code>
4	<code>dataPower[1]=(5.0*dataOne.current);</code>
5	<code>Serial.print("Client One-2 :");</code>
6	<code>break;</code>
7	<code>case 2:</code>
8	<code>dataTwo = data;</code>
9	<code>dataPower[2] = (5.0*dataTwo.current);</code>
10	<code>Serial.println("Client Two-2 :");</code>
11	<code>break;</code>
12	<code>default:</code>
13	<code>dataThree = data;</code>
14	<code>dataPower[3] = (5.0*dataThree.current);</code>
15	<code>Serial.println("Client Three-2 :");</code>
16	<code>break;</code>

Potongan kode sumber pada Tabel 5.13 memiliki pernyataan sebagai berikut:

- Potongan kode sumber No.1 adalah sebuah *statement* pengadaian dengan metode *switch case* dengan data *header node* sebagai parameter.
- Potongan kode sumber No.2-6 adalah jika data header adalah 1 maka akan memasukkan data power dari *slave* 1 kedalam *variable* data array *dataPower* ke-2 atau index-1 dan keluar dari statemen *switch case*.
- Potongan kode sumber No.7-11 adalah jika data header adalah 2 maka akan memasukkan data power dari *slave* 2 kedalam *variable* data array *dataPower* ke-3 atau index-2 dan keluar dari statemen *switch case*.
- Potongan kode sumber No.12-16 adalah jika data header adalah 3 maka akan memasukkan data power dari *slave* 3 kedalam *variable* data array *dataPower* ke-4 atau index-3 dan keluar dari statemen *switch case*.



Tabel 5.14 Potongan Kode Sumber Memasukkan Data *Master* Dan Setiap *Slave* Kedalam *Variable Array*

No	Potongan Kode Sumber
1	firstTime = millis();
2	DHT.read11(dht_pin);
3	float h = DHT.Humidity;
4	float t = DHT.temperature;
5	dataZero.hum = h;
6	dataZero.temp = t;
7	dataZero.id = 0;
8	dataZero.current = random(6023,6545)/100.0;
9	dataZero.TimeCompletion = millis()-firstTime;

Potongan kode sumber pada Tabel 5.14 memiliki pernyataan sebagai berikut:

- Potongan kode sumber No.1-9 adalah eksekusi proses pengumutan hasil data pada *master*. Pemungutan yang diambil berupa hasil data arus, tegangan, suhu, kelembaban, data id pada *master* adalah 0, dan data *Time completion* menjadi sebuah data array.

Tabel 5.15 Potongan Kode Sumber Mengumpulkan Semua Data Dari *Slave* Dan *Master* Menjadi Sebuah Data Array

No	Potongan Kode Sumber
1	dataPower[0] = (5.0 * dataZero.current);
2	buf[0]=0;
3	buf[1]=1;
4	buf[2]=2;
5	buf[3]=3;
6	for(int i=0;i<3;i++){
7	for(int j=i+1;j<4;j++){
8	if(dataPower[i] < dataPower[j]){
9	temp = buf[i];
10	buf[i] = buf[j];
11	buf[j] = temp;
12	tempPower = dataPower[i];
13	dataPower[i] = dataPower[j];
14	dataPower[j] = tempPower;
15	}
16	}
17	}

Potongan kode sumber pada Tabel 5.15 memiliki pernyataan sebagai berikut:

- Potongan kode sumber No.1-5 adalah perhitungan daya dengan rumus $P = V \times I$. Daya yang dihitung adalah daya dari *master* dengan no id 0, *slave* 1 dengan no id 1, *slave* 2 dengan no id 2, dan *slave* 3 dengan no id 3.

- b. Potongan kode sumber No.6-17 adalah algoritma yang digunakan sebagai pengurutan *Priority Scheduling* apakah akan menghasilkan sebuah data daya pada *master* dan masing-masing *slave* dari yang terbesar hingga ke terkecil sebagai prioritas.

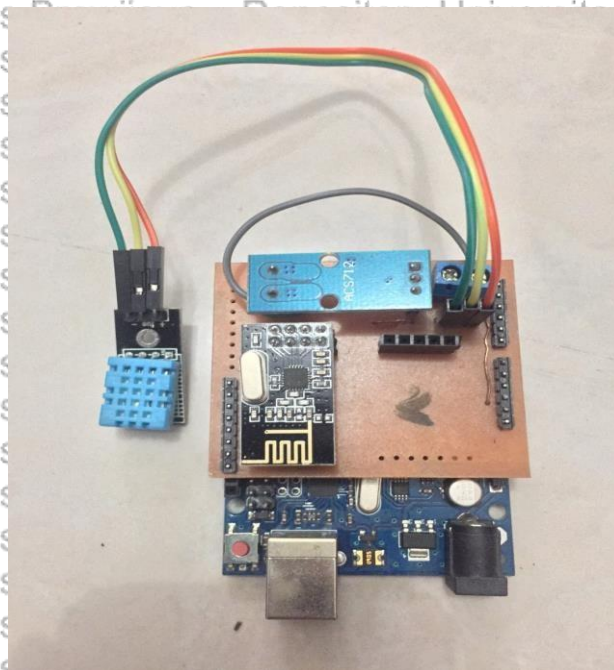
Tabel 5.16 Potongan Kode Sumber Menampilkan Data Berdasarkan Prioritas Data Daya Dari Yang Terbesar Ke Terkecil

No	Potongan Kode Sumber
1	for(int i=0;i<4;i++){
2	displayData(buf[i], i);
3	delay(2000);
4	}
5	Serial.println("End of The Loop ...");
6	delay(1000);
7	}

Potongan kode sumber pada Tabel 5.16 memiliki pernyataan sebagai berikut:

- a. Potongan kode sumber no.1-7 adalah proses menampilkan semua data berdasarkan prioritas.

5.2.3 Implementasi Sistem Pada Modul *Slave 1 Node*, *Slave 2* dan *Slave 3*



Gambar 5.8 Implementasi Perangkat Keras Modul *Slave 1*, *Slave 2* dan *Slave 3*

Perancangan sistem dan pembahasan akan dikerjakan sesuai dengan implementasi sistem pada modul *slave 1 node*, *slave 2 node*, dan *slave 3 node*. Terdapat beberapa alat pada setiap modul *slave node* yang akan digunakan dalam perancangan pada sistem ini. Perancangan alat yang digunakan adalah Arduino uno R3 untuk mikrokontroler, Kabel USB untuk pemberi arus dan daya pada setiap alat



master node, dan modul NRF24L01 untuk komunikasi nirkabel. Hasil implementasi sistem pada setiap modul *slave node* dapat dilihat pada Gambar 5.8.

5.2.4 Urutan Berjalannya Modul *Node Slave 1*, *Slave 2* dan *Slave 3*

Tabel 5.17 Potongan Kode Sumber Inisialisasi *Input* dan *Output* Setiap *Slave*

No	Potongan Kode Sumber
1	#include <SPI.h>
2	#include <RF24Network.h>
3	#include <RF24Network_config.h>
4	#include <Sync.h>
5	#include <dht.h>
6	#include <nRF24L01.h>
7	#include <printf.h>
8	#include <RF24.h>
9	#include <RF24_config.h>
10	#define dht_pin A1
11	RF24 radio(9,10);
12	RF24Network network(radio);
13	dht DHT;
14	const uint16_t writingAddress = 00;
15	const uint16_t thisAddress = 01;
16	unsigned long firstTime;
17	struct dataClient{
18	float id;
19	float temp;
20	float hum;
21	float current;
22	unsigned long TimeCompletion;
23	};
24	dataClient data;

Potongan kode sumber pada Tabel 5.17 memiliki pernyataan sebagai berikut:

- Potongan kode sumber no.1-10 adalah inisialisasi *library* yang diimplementasikan pada setiap modul *slave node*. Program *library* yang digunakan berupa *library* SPI yang dapat berkomunikasi secara Serial Peripheral Interface, dimana NRF24L01 berkomunikasi secara SPI dengan Arduino Uno R3. *Library* NRF24L01 dan RF24Network merupakan *library* yang digunakan untuk melakukan perintah dan penerimaan data dengan modul NRF24L01. Selain itu *library* dht merupakan *library* yang digunakan untuk berkomunikasi dengan modul DHT11.
- Potongan kode sumber no.11-13 adalah pembuatan *object* RF24, RF24Network, dan DHT.
- Potongan kode sumber no.14-15 adalah deklarasi *variable* alamat *master* dan alamat setiap *slave*. Setiap *slave* memiliki no id, *slave 1* memiliki no id 01, *slave*



2 memiliki no id 02, dan *slave* 3 memiliki no id 03. Potongan kode sumber no.16 adalah deklarasi *variable* yang menunjukkan waktu sebelum proses pengambilan data.

- d. Potongan kode sumber no.17-23 adalah deklarasi tipe data *struct* untuk menyimpan beberapa *variable slave*. Data yang tersimpan berupa id *slave*, suhu, kelembaban, dan waktu secara real *Time*. Baris ke-24 merupakan deklarasi data *struct* dataSlave untuk menyimpan *variable-variable* data *slave* atau data *master*.

Tabel 5.18 Potongan Kode Sumber Mengecek Apakah Ada Request Dari Slave Untuk Permintaan Data, Jika Tidak Ada Maka Balik Ke Start

No	Potongan Kode Sumber
1	network.upDate();
2	while(network.available()){
3	RF24NetworkHeader header;
4	byte adds;
5	network.read(header, &adds, sizeof(adds));

Potongan kode sumber pada Tabel 5.18 memiliki pernyataan sebagai berikut:

- Potongan kode sumber no.1 adalah upDate network yang berfungsi mengecek apakah ada radio komunikasi yang masuk atau tidak.
- Potongan kode sumber no.2 adalah merupakan blok atau bagian penerimaan data jika ada hasil yang akan diterima.
- Potongan kode sumber no.3-4 adalah deklarasi object RF24NetworkHeader ke header dan *variable* adds bertipe byte.
- Potongan kode sumber no.5 adalah membaca apakah ada data yang masuk didalam network.

Tabel 5.19 Potongan Kode Sumber Jika Ada Request Maka Ambil Data Dht11, Data Arus Dan Data Time completion

No	Potongan Kode Sumber
1	firstTime = millis();
2	DHT.read11(dht_pin);
3	float h = DHT.Humidity;
4	float t = DHT.temperature;
5	data.hum = h;
6	data.temp = t;
7	data.id = 1,2,3;
8	data.current = random(6023,6545)/100.0;
9	data.TimeCompletion = millis()-firstTime;

Potongan kode sumber pada Tabel 5.19 memiliki pernyataan sebagai berikut:

- Potongan kode sumber no.1-7 adalah memasukkan nilai millis ke dalam *variable* firstTime. Selain itu jika ada *request* maka akan ambil data suhu, kelembaban, data arus, data id setiap *slave*.



- b. Potongan kode sumber no.8-9 adalah memasukkan data *Time completion* yang merupakan hasil waktu untuk menyelesaikan waktu satu siklus percobaan pada semua *node*, *Time arrival* merupakan hasil waktu datang setiap *slave*.

Tabel 5.20 Potongan Kode Sumber Mengirimkan Data Ke Master

No	Potongan Kode Sumber
1	RF24NetworkHeader header2(writingAddress);
2	bool ok = network.write(header2, &data, sizeof(data));
3	}
4	}

Potongan kode sumber pada Tabel 5.20 memiliki pernyataan sebagai berikut:

- a. Potongan kode sumber no.1-4 adalah pengiriman setiap data *slave* ke *master*. Jika no id sesuai maka setiap data *slave* akan mengirimkan data ke *master*.



BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis yang dilakukan pada bab ini akan dikerjakan menurut pengujian yang bertahap dan untuk mendapatkan hasil jawaban dari rumusan masalah pada penelitian skripsi.

6.1 Pengujian Hasil Sensing

Pengujian hasil *sensing* data yang dilakukan peneliti terdiri dari beberapa modul sensor yang terdapat pada modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* yang akan digunakan untuk mengambil data *sensing* arus, tegangan, dan suhu.

6.1.1 Tujuan

Dari beberapa pengujian apakah hasil *sensing* pada setiap modul sensor *node* akan menunjukkan performa yang sama atau berbeda jika pengujian dilakukan di berbagai macam tempat.

6.1.2 Peralatan

Peralatan yang dibutuhkan dalam pengujian ini berupa beberapa alat perangkat keras dan lunak yang akan digunakan adalah:

1. Modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node*
2. USB
3. Laptop
4. Termometer ruangan
5. AVO meter
6. Arduino Uno R3

6.1.3 Langkah Pengujian

Untuk mendapatkan hasil pengujian sesuai yang diinginkan peneliti, maka diperlukan langkah-langkah yang harus dilakukan sebagai berikut:

1. Kabel USB terhubung pada masing-masing modul *node* dan laptop.
2. Membuka aplikasi Arduino IDE.
3. Membuka Arduino Uno R3 file .ino, yaitu file *master*, *slave 1*, *slave 2*, dan *slave 3*
4. Setelah membuka file lalu melakukan verify atau meng *compile* apakah ada kesalahan pada program atau tidak. Jika tidak ada kesalahan, maka dilanjutkan dengan upload masing-masing program



```
client_1 | Arduino 1.8.10
File Edit Sketch Tools Help
client_1
#include <SPI.h>
#include <RF24Network.h>
#include <RF24Network_config.h>
#include <Sync.h>
#include <dht.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#define dht_pin A1

RF24 radio(9,10);
RF24Network network(radio);
dht DHT;
const uint16_t writingAddress = 00;
const uint16_t thisAddress = 01;
unsigned long firstTime;

struct dataClient{
  float id;
  float temp;
  float hum;
  float current;
  unsigned long timeCompletion;
};

dataClient data;

void setup() {
  SPI.begin();
  Serial.begin(9600);
}

Done compiling
Sketch uses 8582 bytes (26%) of program storage space. Maximum is 32256 bytes.
Global variables use 630 bytes (30%) of dynamic memory, leaving 1418 bytes for local variables.
```

Gambar 6.1 Verify Slave 1

```
client_2 | Arduino 1.8.10
File Edit Sketch Tools Help
client_2
#include <SPI.h>
#include <RF24Network.h>
#include <RF24Network_config.h>
#include <Sync.h>
#include <dht.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#define dht_pin A1

RF24 radio(9,10);
RF24Network network(radio);
dht DHT;
const uint16_t writingAddress = 00;
const uint16_t thisAddress = 02;
unsigned long firstTime;

struct dataClient{
  float id;
  float temp;
  float hum;
  float current;
  unsigned long timeCompletion;
};

dataClient data;

void setup() {
  SPI.begin();
}

Done compiling
Sketch uses 8584 bytes (26%) of program storage space. Maximum is 32256 bytes.
Global variables use 630 bytes (30%) of dynamic memory, leaving 1418 bytes for local variables.
```

Gambar 6.2 Verify Slave 2



```

client_3

#include <SPI.h>

#include <RF24Network.h>
#include <RF24Network_config.h>
#include <Sync.h>
#include <dht.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#define dht_pin A1

RF24 radio(9,10);
RF24Network network(radio);
dht DHT;
const uint16_t writingAddress = 00;
const uint16_t thisAddress = 03;
unsigned long firstTime;

struct dataClient{
  float id;
  float temp;
  float hum;
  float current;
  unsigned long timeCompletion;
};

dataClient data;

void setup() {
  SPI.begin();
}

Done compiling.
Sketch uses 8584 bytes (26%) of program storage space. Maximum is 32256 bytes.
Global variables use 630 bytes (30%) of dynamic memory, leaving 1418 bytes for local variables
  
```

Gambar 6.3 Verify Slave 3

```

server

// Memasukkan library-library yang di perlukan
#include <SPI.h>
#include <Wire.h>
#include <RTClib.h>
#include <RF24Network.h>
#include <RF24Network_config.h>
#include <Sync.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#include <LowPower.h>
#include <dht.h>
// Akhir Blok Librari

// Mendefinisikan pin analog A1 sebagai pin untuk dht
#define dht_pin A1

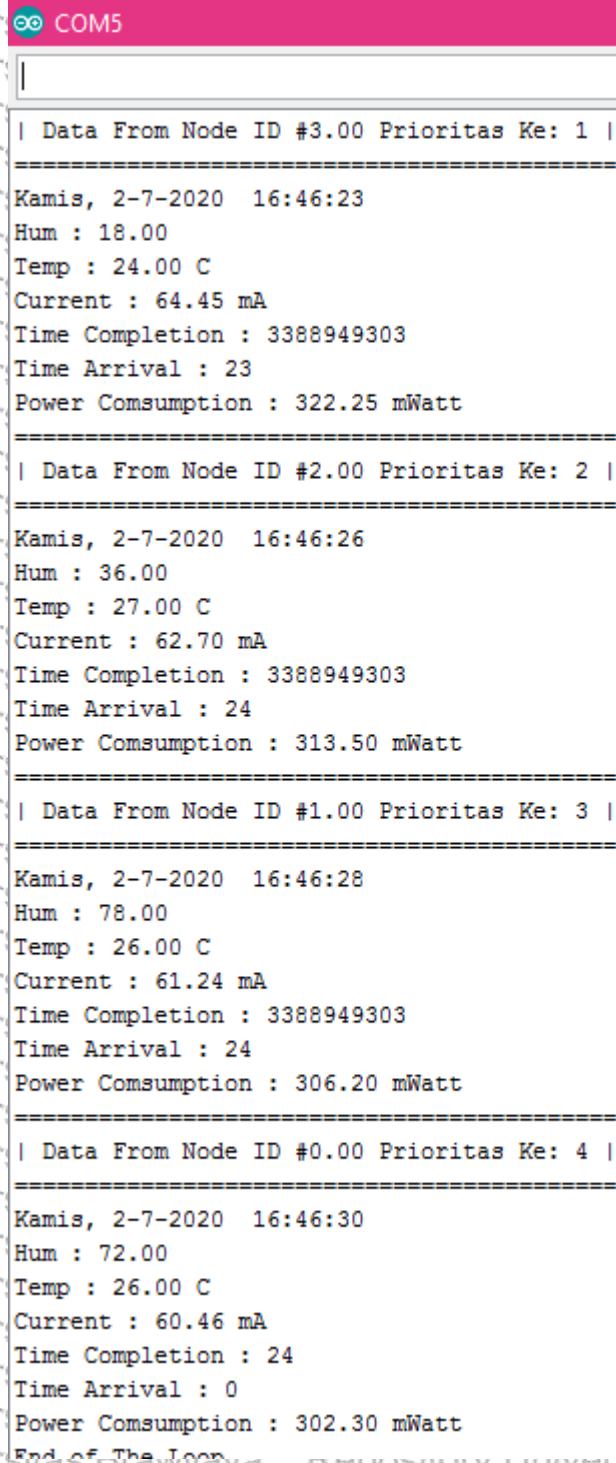
// Pembuatan object RF24
RF24 radio(9,10);
// Pembuatan object RF24Network
RF24Network network(radio);
// Pembuatan object DHT
dht DHT;
// Pembuatan object RTC
RTC_DS3231 rtc;

// Deklarasi variabel array untuk memasukkan nama hari berdasarkan bahasa indonesia
char daysOfTheWeek[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu"};
// Deklarasi variabel alamat client pertama
<

Done compiling.
Sketch uses 14064 bytes (43%) of program storage space. Maximum is 32256 bytes.
Global variables use 1384 bytes (67%) of dynamic memory, leaving 664 bytes for local variables
  
```

Gambar 6.4 Verify Master

- Setelah selesai melakukan upload program maka selanjutnya “click” serial monitor pada *master* dan semua data ditampilkan.



```

COM5

| Data From Node ID #3.00 Prioritas Ke: 1 |
=====
Kamis, 2-7-2020 16:46:23
Hum : 18.00
Temp : 24.00 C
Current : 64.45 mA
Time Completion : 3388949303
Time Arrival : 23
Power Consumption : 322.25 mWatt
=====

| Data From Node ID #2.00 Prioritas Ke: 2 |
=====
Kamis, 2-7-2020 16:46:26
Hum : 36.00
Temp : 27.00 C
Current : 62.70 mA
Time Completion : 3388949303
Time Arrival : 24
Power Consumption : 313.50 mWatt
=====

| Data From Node ID #1.00 Prioritas Ke: 3 |
=====
Kamis, 2-7-2020 16:46:28
Hum : 78.00
Temp : 26.00 C
Current : 61.24 mA
Time Completion : 3388949303
Time Arrival : 24
Power Consumption : 306.20 mWatt
=====

| Data From Node ID #0.00 Prioritas Ke: 4 |
=====
Kamis, 2-7-2020 16:46:30
Hum : 72.00
Temp : 26.00 C
Current : 60.46 mA
Time Completion : 24
Time Arrival : 0
Power Consumption : 302.30 mWatt
End of The Loop
  
```

Gambar 6.5 Upload program

- Pengujian modul sensor DHT11 mengeluarkan hasil data suhu dan menggunakan termometer sebagai penghitung data suhu pada ruangan, maka selanjutnya akan dilakukan hasil perhitungan akurasi *sensing* data suhu dengan rumus seperti dibawah ini.



Akurasi DHT11 menggunakan rumus perhitungan:

$$\text{Akurasi} = 100\% - \left(\sum \frac{\text{nilai dht11} - \text{termometer}}{\text{nilai dht11}} \times 100\% \right) \quad (6.1)$$

7. Pengujian sensor tegangan mengeluarkan hasil setiap data tegangan pada masing-masing *node* dan selanjutnya menggunakan alat AVO meter untuk melakukan perbandingan tegangan yang akan digunakan.

Akurasi tegangan menggunakan rumus perhitungan:

$$\text{Akurasi} = 100\% - \left(\sum \frac{\text{nilai tegangan} - \text{avometer}}{\text{nilai tegangan}} \times 100\% \right) \quad (6.2)$$

8. Pengujian sensor arus mengeluarkan hasil setiap data arus pada masing-masing *node* dan selanjutnya menggunakan alat AVO meter untuk melakukan perbandingan arus yang akan digunakan.

Akurasi arus menggunakan rumus perhitungan:

$$\text{Akurasi} = 100\% - \left(\sum \frac{\text{nilai arus} - \text{avometer}}{\text{nilai arus}} \times 100\% \right) \quad (6.3)$$

6.1.4 Hasil

Pengujian akan dikerjakan dengan menggunakan modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* untuk melihat hasil akuisisi performa yang akan dihasilkan pada *node* tersebut. Pada pengujian tersebut, hasil performa pada sensor DHT11 dapat dilihat pada Tabel 6.1, kemudian hasil performa pada sensor tegangan dapat dilihat pada Tabel 6.2, dan hasil performa pada sensor arus dapat dilihat pada Tabel 6.3.

Tabel 6.1 Hasil Pengujian Pengambilan Data Akurasi Suhu *Master*

No.	DHT11 <i>master</i> (°C)	Termometer (°C)	Akurasi <i>master</i> (%)
1	26	26	100
2	26	26	100
3	27	26	96.3
4	26	26	100
5	27	27	100
6	27	27	100
7	27	27	100
8	27	27	100
9	26	26	100
10	27	26	96.3
Rata-rata akurasi (%)			99.26



Tabel 6.2 Hasil Pengujian Pengambilan Data Akurasi Suhu *Slave 1, Slave 2, Slave 3*

No.	DHT11 <i>slave 1</i> (°C)	DHT11 <i>slave 2</i> (°C)	DHT11 <i>slave 3</i> (°C)	Termometer (°C)	Akurasi <i>slave 1</i> (%)	Akurasi <i>slave 2</i> (%)	Akurasi <i>slave 3</i> (%)
1	27	26	26	26	96.3	100	100
2	27	26	26	26	96.3	100	100
3	27	27	26	26	96.3	96.3	100
4	27	27	26	26	96.3	96.3	100
5	27	27	27	27	100	100	100
6	27	27	27	27	100	100	100
7	27	27	27	27	100	100	100
8	27	27	27	27	100	100	100
9	26	27	27	26	100	96.3	96.3
10	26	27	27	26	100	96.3	96.3
Rata-rata akurasi (%)					98.52	98.52	99.26

Tabel 6.3 Hasil Pengujian Pengambilan Akurasi Data Tegangan *Master*

No.	Tegangan <i>master</i> (V)	Avometer (V)	Akurasi <i>master</i> (%)
1	5	5.02	99.6
2	5	5.02	99.6
3	5	5.02	99.6
4	5	5.02	99.6
5	5	5.02	99.6
6	5	5.02	99.6
7	5	5.02	99.6
8	5	5.02	99.6
9	5	5.02	99.6
10	5	5.02	99.6
Rata-rata akurasi (%)			99.6

Tabel 6.4 Hasil Pengujian Pengambilan Akurasi Data Tegangan *Slave 1, Slave 2, Slave 3*

No.	Tegangan (V)			Avometer (V)	Akurasi (%)		
	<i>Slave 1</i>	<i>Slave 2</i>	<i>Slave 3</i>		<i>Slave 1</i>	<i>Slave 2</i>	<i>Slave 3</i>
1	5	5	5	5.02	99.6	99.6	99.6
2	5	5	5	5.02	99.6	99.6	99.6
3	5	5	5	5.02	99.6	99.6	99.6
4	5	5	5	5.02	99.6	99.6	99.6
5	5	5	5	5.02	99.6	99.6	99.6
6	5	5	5	5.02	99.6	99.6	99.6
7	5	5	5	5.02	99.6	99.6	99.6
8	5	5	5	5.02	99.6	99.6	99.6
9	5	5	5	5.02	99.6	99.6	99.6
10	5	5	5	5.02	99.6	99.6	99.6
Rata-rata akurasi (%)					99.6	99.6	99.6

Tabel 6.5 Hasil Pengujian Pengambilan Akurasi Data Arus *Master*

No.	Arus <i>master</i> (mA)	Avometer <i>master</i> (mA)	Akurasi <i>master</i> (%)
1	61.92	62.84	98.52
2	60.46	62.84	96.07
3	61.51	62.84	97.84
4	63.15	62.84	99.51
5	64.81	62.84	96.97
6	62.09	62.84	98.80
7	64.27	62.84	97.78
8	64.00	62.84	98.19
9	65.28	62.84	96.27
10	63.65	62.84	98.73
Rata-rata akurasi (%)			97.86

Tabel 6.6 Hasil Pengujian Pengambilan Akurasi Data Arus *Slave 1*, *Slave 2*, *Slave 3*

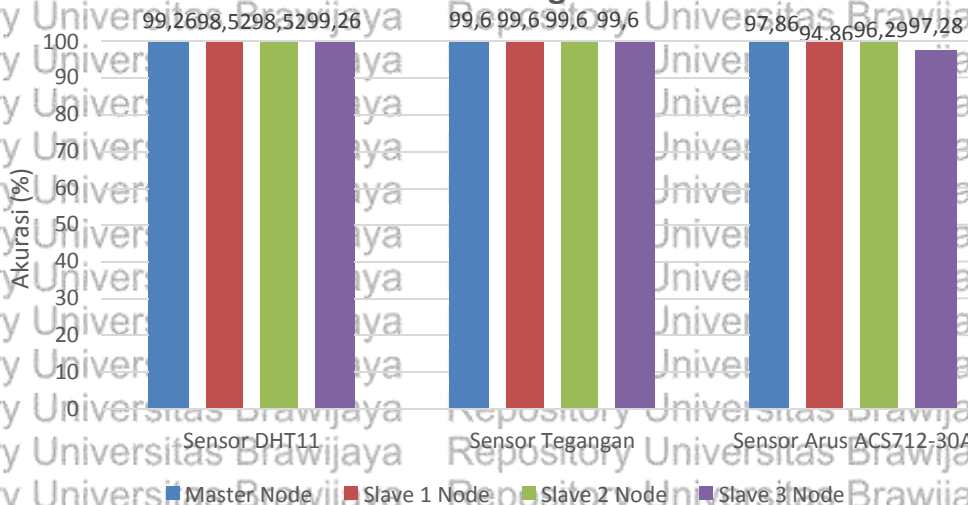
No.	Arus (mA)			Avometer (mA)			Akurasi (%)		
	<i>Slave 1</i>	<i>Slave 2</i>	<i>Slave 3</i>	<i>Slave 1</i>	<i>Slave 2</i>	<i>Slave 3</i>	<i>Slave 1</i>	<i>Slave 2</i>	<i>Slave 3</i>
1	61.89	62.62	64.29	60.23	65.45	64.88	97.44	95.49	99.09
2	63.95	61.51	61.73	60.23	65.45	64.88	94.20	93.60	94.90
3	61.63	61.08	64.64	60.23	65.45	64.88	97.73	92.85	99.63
4	64.80	65.40	64.80	60.23	65.45	64.88	92.95	99.93	99.88
5	61.04	62.82	62.55	60.23	65.45	64.88	95.88	95.85	96.28
6	64.75	62.01	63.32	60.23	65.45	64.88	93.01	94.46	97.54
7	64.32	63.04	62.41	60.23	65.45	64.88	93.64	96.18	96.05
8	60.31	64.86	62.95	60.23	65.45	64.88	99.86	99.10	96.94
9	65.37	65.14	65.11	60.23	65.45	64.88	92.13	99.53	99.65
10	63.57	62.88	60.54	60.23	65.45	64.88	91.85	95.92	92.84
Rata-rata akurasi (%)							94.86	96.29	97.28

6.1.5 Analisis

Pengujian yang dilakukan dengan memakai modul sensor DHT11 dengan melakukan percobaan *sensing* suhu menghasilkan akurasi *sensing* suhu sejumlah 98.52% hingga 99.26%. Pengujian memakai modul sensor tegangan dengan melakukan percobaan perhitungan data tegangan menghasilkan akurasi *sensing* tegangan dengan rata-rata hasil persentase 99.6%. Selanjutnya pengujian memakai modul sensor arus ACS712-30A dengan melakukan percobaan persentase arus menghasilkan akurasi *sensing* arus dengan rata-rata hasil persentase 97.86%. Setiap percobaan yang dilakukan pada setiap *node* mempunyai *transmitter* sensor yang berfungsi dalam percobaan mengetahui hasil data pada setiap modul sensor, jika *transmitter* sensor tidak dapat melakukan percobaan mengetahui hasil data akan berdampak pada proses sistem yang akan digunakan.



Akurasi Sensing Sensor



6.2 Pengujian Pengiriman Data

Pengujian hasil pengiriman data yang dilakukan peneliti terdiri dari beberapa modul yang terdapat pada modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* yang akan digunakan untuk pengiriman dan penerimaan data menggunakan *wireless* dengan melakukan percobaan algoritma *Priority Scheduling*.

6.2.1 Tujuan

Dari beberapa pengujian apakah menghasilkan performa daya yg sama atau berbeda, pengiriman waktu secara *real Time*, batas optimal sensor *node* jika pengujian dilakukan di berbagai macam tempat.

6.2.2 Peralatan

Peralatan yang dibutuhkan dalam pengujian ini berupa beberapa alat perangkat keras dan lunak yang akan digunakan adalah:

1. *Node master*
2. *Node Slave 1*
3. *Node Slave 2*
4. *Node Slave 3*
5. Kabel USB
6. Laptop
7. Program Arduino IDE

6.2.3 Langkah Pengujian

Untuk mendapatkan hasil pengujian sesuai yang diinginkan peneliti, maka diperlukan langkah-langkah yang harus dilakukan sebagai berikut:



1. Kabel USB terhubung pada masing-masing modul *node* dan laptop. Masing-masing *node* dapat digunakan diberbagai jarak dengan menggunakan saklar atau powerbank untuk menyalakan modul tersebut.
2. Menjalankan aplikasi Arduino Uno R3.
3. Membuka file pada Arduino Uno R3. File yang digunakan yaitu file *master.ino*, *slave1.ino*, *slave2.ino*, dan *slave3.ino*.
4. Setelah membuka setiap file pada masing-masing *node*, lakukan verify atau meng compile pada codingan apakah ada kesalahan pada program atau tidak. Jika tidak ada kesalahan, maka dilanjutkan dengan upload masing-masing pada program. Gambar verify atau meng compile codingan dapa dilihat pada Gambar 6.1, Gambar 6.2, Gambar 6.3, dan Gambar 6.4.
5. Selanjutnya dilakukan penguploadan pada masing-masing modul *node*. Setelah masing-masing *node* selesai melakukan upload program, Langkah selanjutnya klik serial monitor pada *node master* dan semua data akan ditampilkan. Gambar menampilkan data dapat dilihat pada Gambar 6.5.

Peneliti akan melaksanakan pengujian pada 2 tempat yang berbeda. Pengujian pengiriman dan penerimaan akan menghasilkan data sebanyak 10 kali dengan jarak pengujian 1-10meter. Pengujian akan dilakukan dengan jangka yang tidak sama, jangka yang dimaksud adalah:

- a. Tanpa ada halangan yang dimaksud adalah pengujian yang akan dikerjakan pada sebuah tanah lapang.
- b. Ada halangan yang dimaksud adalah pengujian yang akan dikerjakan pada sebuah ruangan dengan halangan tembok dan lain-lainnya.

6.2.4 Hasil

Pengujian yang akan dilakukan pada jarak tanpa ada halangan dan ada halangan dapatkah menunjukkan hasil dari peforma pengujian pengambilan dan pengiriman data.

6.2.4.1 Hasil Pengujian Ada Halangan

Percobaan yang dilakukan pada jarak ada halangan yang dikerjakan sejumlah 10 kali percobaan dengan jarak 1-10meter pada setiap modul *master node*, *slave 1*, *slave 2*, dan *slave 3*. Pengujian dilakukan pada jarak minimum sampai optimal pada pengiriman data dan daya pada masing-masing *node*. Pengujian pada jarak 1-10meter ada halangan selanjutnya dapat dilihat pada Tabel 6.7 - Tabel 6.17.

Tabel 6.7 Pengujian Ke-1 dengan Jarak 1meter Ada Halangan

No	Urutan <i>Node</i>	<i>Time</i>	Suhu (°C)	<i>Humidity</i>	Arus (mA)	Volt	Daya (mWatt)
1	<i>Slave 1</i>	21:08:58	26	89	64.70	5	323.50
2	<i>Slave 2</i>	21:09:00	27	46	63.74	5	318.70
3	<i>Master</i>	21:09:02	26	92	61.15	5	305.75
4	<i>Slave 3</i>	21:09:05	27	19	60.89	5	304.45


Tabel 6.8 Pengujian Ke-2 dengan Jarak 2meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 1	21:09:10	26	88	64.08	5	320.40
2	Slave 2	21:09:12	27	46	62.49	5	312.45
3	Slave 3	21:09:14	27	19	62.28	5	311.40
4	Master	21:09:17	27	91	62.17	5	310.85

Tabel 6.9 Pengujian Ke-3 dengan Jarak 3meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:09:22	26	91	64.38	5	321.90
2	Slave 3	21:09:24	26	19	63.55	5	317.75
3	Slave 1	21:09:27	27	88	62.26	5	311.30
4	Slave 2	21:09:29	27	46	60.70	5	303.50

Tabel 6.10 Pengujian Ke-4 dengan Jarak 4meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:09:34	26	88	63.40	5	317.00
2	Slave 1	21:09:37	26	19	63.01	5	315.05
3	Master	21:09:39	26	91	62.26	5	311.30
4	Slave 2	21:09:41	27	46	60.70	5	303.50

Tabel 6.11 Pengujian Ke-5 dengan Jarak 5meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:09:44	26	92	65.07	5	325.35
2	Slave 2	21:09:42	27	19	65.01	5	325.05
3	Slave 1	21:09:46	27	91	64.79	5	323.95
4	Slave 3	21:09:49	27	19	64.36	5	321.80

Tabel 6.12 Pengujian Ke-6 dengan jarak 6meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:09:52	26	92	64.06	5	320.30
2	Slave 1	21:09:54	27	87	63.40	5	317.00
3	Slave 3	21:09:56	27	19	62.01	5	310.05
4	Slave 2	21:09:59	27	45	61.26	5	306.30

**Tabel 6.13 Pengujian Ke-7 dengan Jarak 7meter Ada Halangan**

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:10:02	27	19	65.26	5	326.30
2	Slave 1	21:10:04	27	88	62.26	5	311.30
3	Master	21:10:06	27	91	61.76	5	308.80
4	Slave 2	21:10:08	26	46	60.70	5	303.50

Tabel 6.14 Pengujian Ke-8 dengan Jarak 8meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 2	21:10:59	27	46	65.16	5	325.80
2	Slave 3	21:11:01	26	19	64.91	5	324.55
3	Master	21:11:03	26	91	63.12	5	315.60
4	Slave 1	21:11:05	27	88	62.26	5	311.30

Tabel 6.15 Pengujian Ke-9 dengan Jarak 9meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:12:11	26	19	65.01	5	325.05
2	Master	21:12:13	27	91	62.58	5	312.90
3	Slave 2	21:12:15	27	45	62.48	5	312.40
4	Slave 1	21:12:17	27	88	62.26	5	311.30

Tabel 6.16 Pengujian ke-10 dengan Jarak 10meter Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:14:23	26	91	64.20	5	321.00
2	Slave 2	21:14:25	27	45	63.20	5	316.00
3	Slave 1	21:14:27	27	88	62.48	5	312.40
4	Slave 3	21:14:30	27	19	60.25	5	301.25



Tabel 6.17 Pengujian Daya Priority Scheduling dengan Jarak 1-10meter ada halangan

No	Jarak	Urutan node	Waktu	Arus (mA)	Volt	Daya (mWatt)
1	1meter	Slave 1	21:08:58	64.70	5	323.50
		Slave 2	21:09:00	63.74	5	318.70
		Master	21:09:02	61.15	5	305.75
		Slave 3	21:09:05	60.89	5	304.45
2	2meter	Slave 1	21:09:10	64.08	5	320.40
		Slave 2	21:09:12	62.49	5	312.45
		Slave 3	21:09:12	62.28	5	311.40
		Master	21:09:14	62.17	5	310.85
3	3meter	Master	21:09:22	64.38	5	321.90
		Slave 3	21:09:24	63.55	5	317.75
		Slave 1	21:09:27	62.26	5	311.30
		Slave 2	21:09:29	60.70	5	303.50
4	4meter	Slave 3	21:09:34	63.40	5	317.00
		Slave 1	21:09:37	63.01	5	315.05
		Master	21:09:39	62.26	5	311.30
		Slave 2	21:09:41	60.70	5	303.50
5	5meter	Master	21:09:42	65.07	5	325.35
		Slave 2	21:09:44	65.01	5	325.05

		Slave 1	21:09:46	64.79	5	323.95
		Slave 3	21:09:49	64.36	5	321.80
		Master	21:09:52	64.06	5	320.30
6	6meter	Slave 1	21:09:54	63.40	5	317.00
		Slave 2	21:09:56	62.01	5	310.05
		Slave 3	21:09:58	61.26	5	306.30
7	7meter	Slave 3	21:10:02	65.26	5	326.30
		Slave 1	21:10:04	62.26	5	311.30
		Master	21:10:06	61.76	5	308.80
		Slave 2	21:10:08	60.70	5	303.50
		Slave 2	21:10:59	65.16	5	325.80
		Slave 3	21:11:01	64.91	5	324.55
8	8meter	Master	21:11:03	63.12	5	315.60
		Slave 1	21:11:05	62.26	5	311.30
		Slave 3	21:12:11	65.01	5	325.05
9	9meter	Master	21:12:13	62.58	5	312.90
		Slave 2	21:12:15	62.48	5	312.40
		Slave 1	21:12:17	62.26	5	311.30
10	10meter	Master	21:14:23	64.20	5	321.00
		Slave 2	21:14:25	63.20	5	316.00



	<i>Slave 1</i>	21:14:27	62.48	5	312.40
	<i>Slave 3</i>	21:14:30	60.25	5	301.25

Dari Pengujian diatas, Tabel 6.7-Tabel 6.17 merupakan total pengujian pengambilan hasil data pada kejauhan minimum sampai maksimal adalah 1-10meter dan percobaan dikerjakan diruangan yang ada halangan seperti dinding dan lain-lainnya. Jika pengujian dilakukan lebih dari 10 meter, maka pengujian pengambilan data tidak terlalu efektif jika melewati batas jarak optimal. Dapat dilihat dari 10 kali percobaan delay menampilkan data ke setiap *node* berdasarkan urutan hanya memakan rata-rata waktu 2-3 *second*. Hasil setiap data *node* yang ditunjukkan pada pengujian memiliki hasil yang berbeda antara satu dan lainnya, diakibatkan oleh pengujian pada jarak yang berbeda-beda dan keadaan suatu ruangan atau tempat. Hasil data daya yang didapatkan pada pengujian didapatkan dari hasil sensor arus dikali sensor tegangan. *Priority Scheduling* pada 10 kali percobaan pun cukup berhasil, karena dapat menampilkan urutan *node Priority Scheduling* berdasarkan daya terbesar hingga ke terkecil pada masing-masing *node*.

Berdasarkan pengujian diatas dapat diketahui untuk pengujian penerimaan dan pengiriman hasil data setiap *node* pada kejauhan 1-10meter tidak terlalu berpengaruh pada *Priority Scheduling* yang mengutamakan menampilkan data daya berdasarkan daya terbesar hingga ke terkecil di setiap modul *node*.

6.2.4.2 Hasil Pengujian Tidak Ada Halangan

Pada pengujian selanjutnya, pengujian tidak ada halangan dikerjakan sejumlah 10 kali pengujian pada jarak 1-10meter di masing-masing modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3node*. Pengujian dilakukan pada jarak minimum sampai optimal pada pengiriman data dan daya pada masing-masing *node*. Pengujian pada jarak 1-10meter tidak ada halangan selanjutnya dapat dilihat pada Tabel 6.18-Tabel 6.28.

Tabel 6.18 Pengujian Ke-1 dengan Jarak 1meter Tidak Ada Halangan

No	Urutan <i>Node</i>	<i>Time</i>	Suhu (°C)	<i>Humidity</i>	Arus (mA)	<i>Volt</i>	Daya (mWatt)
1	<i>Slave 2</i>	21:17:23	27	45	64.47	5	322.35
2	<i>Slave 3</i>	21:17:25	27	18	61.75	5	308.75
3	<i>Slave 1</i>	21:17:27	26	88	61.63	5	308.15
4	<i>Master</i>	21:17:30	26	81	60.46	5	302.30

**Tabel 6.19 Pengujian Ke-2 dengan Jarak 2meter Tidak Ada Halangan**

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:18:23	27	18	63.98	5	319.90
2	Slave 1	21:18:25	26	88	63.68	5	318.40
3	Master	21:18:27	26	80	61.51	5	307.55
4	Slave 2	21:18:30	27	45	60.98	5	304.90

Tabel 6.20 Pengujian Ke-3 dengan Jarak 3meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:20:30	26	18	65.26	5	326.30
2	Slave 2	21:20:32	27	45	64.42	5	322.10
3	Slave 1	21:20:34	27	45	63.98	5	319.90
4	Master	21:20:37	26	80	63.15	5	315.75

Table 6.21 Pengujian Ke-4 dengan Jarak 4meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:22:33	27	19	62.81	5	314.05
2	Slave 1	21:22:35	26	89	61.69	5	308.45
3	Master	21:22:37	27	83	61.51	5	307.55
4	Slave 2	21:22:40	26	48	60.25	5	301.25

Tabel 6.22 Pengujian Ke-5 dengan Jarak 5meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 3	21:27:43	27	19	63.32	5	316.60
2	Master	21:27:45	27	83	63.15	5	315.75
3	Slave 2	21:27:47	26	48	63.14	5	315.70
4	Slave 1	21:27:49	26	89	62.02	5	310.10

Tabel 6.23 Pengujian Ke-6 dengan Jarak 6meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:30:13	27	83	64.81	5	324.05
2	Slave 2	21:30:15	26	48	64.11	5	320.55
3	Slave 3	21:30:17	26	19	61.26	5	306.30
4	Slave 1	21:30:20	26	89	60.67	5	303.35

**Tabel 6.24 Pengujian Ke-7 dengan Jarak 7meter Tidak Ada Halangan**

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:34:02	26	91	63.20	5	316.00
2	Slave 2	21:34:04	27	45	62.48	5	312.40
3	Slave 1	21:34:06	26	88	62.26	5	311.30
4	Slave 3	21:34:09	26	19	60.25	5	301.25

Tabel 6.25 Pengujian Ke-8 dengan Jarak 8meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Slave 2	21:40:20	26	47	65.01	5	325.05
2	Slave 1	21:40:22	26	87	64.53	5	322.65
3	Slave 3	21:40:24	27	19	63.46	5	317.30
4	Master	21:40:26	26	86	61.64	5	308.05

Tabel 6.26 Pengujian Ke-9 dengan Jarak 9meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:40:20	26	85	64.53	5	322.65
2	Slave 1	21:40:22	27	87	62.48	5	312.40
3	Slave 2	21:40:24	26	47	62.26	5	311.30
4	Slave 3	21:40:26	27	19	60.25	5	301.25

Tabel 6.27 Pengujian Ke-10 dengan Jarak 10meter Tidak Ada Halangan

No	Urutan Node	Time	Suhu (°C)	Humidity	Arus (mA)	Volt	Daya (mWatt)
1	Master	21:42:23	27	85	64.53	5	322.65
2	Slave 1	21:42:25	27	87	64.00	5	320.00
3	Slave 2	21:42:27	26	47	63.46	5	317.30
4	Slave 3	21:42:41	26	19	61.51	5	307.55



Tabel 6.28 Pengujian Daya *Priority Scheduling* dengan Jarak 1-10meter Tidak Ada Halangan

No	Jarak	Urutan node	Waktu	Arus (mA)	Volt	Daya (mWatt)
1	1meter	<i>Slave 2</i>	21:17:23	64.47	5	322.35
		<i>Slave 3</i>	21:17:25	61.75	5	308.75
		<i>Slave 1</i>	21:17:27	61.61	5	308.15
		<i>Master</i>	21:17:30	60.46	5	302.30
2	2meter	<i>Slave 3</i>	21:18:23	63.98	5	319.90
		<i>Slave 1</i>	21:18:25	63.68	5	318.40
		<i>Master</i>	21:18:27	61.51	5	307.55
		<i>Slave 2</i>	21:18:30	60.98	5	304.90
3	3meter	<i>Slave 3</i>	21:20:30	65.26	5	326.30
		<i>Slave 2</i>	21:20:32	64.42	5	322.10
		<i>Slave 1</i>	21:20:34	63.98	5	319.90
		<i>Master</i>	21:20:37	63.15	5	315.75
4	4meter	<i>Slave 3</i>	21:22:33	62.81	5	314.05
		<i>Slave 1</i>	21:22:35	61.69	5	308.45
		<i>Master</i>	21:22:37	61.51	5	307.55
		<i>Slave 2</i>	21:22:40	60.25	5	301.25
5	5meter	<i>Slave 3</i>	21:27:43	63.32	5	316.60
		<i>Master</i>	21:27:45	63.15	5	315.75

6	6meter	Slave 2	21:27:47	63.14	5	315.70
		Slave 1	21:27:49	62.02	5	310.10
		Master	21:30:13	64.81	5	324.05
		Slave 2	21:30:15	64.11	5	320.55
		Slave 3	21:30:17	61.26	5	306.30
7	7meter	Slave 1	21:30:20	60.67	5	303.35
		Master	21:34:02	63.20	5	316.00
		Slave 2	21:34:04	62.48	5	312.40
		Slave 1	21:34:06	62.26	5	311.30
		Slave 3	21:34:09	60.25	5	301.25
8	8meter	Slave 2	21:40:20	65.01	5	325.05
		Slave 1	21:40:22	64.53	5	322.65
		Slave 3	21:40:24	63.46	5	317.30
		Master	21:40:26	61.61	5	308.05
		Master	21:42:23	64.53	5	322.65
9	9meter	Slave 1	21:42:25	62.48	5	312.40
		Slave 2	21:42:27	62.26	5	311.30
		Slave 3	21:42:31	60.25	5	301.25
		Master	21:48:12	64.53	5	322.65
		Slave 1	21:48:14	64.00	5	320.00



	<i>Slave 2</i>	21:48:16	63.46	5	317.30
	<i>Slave 3</i>	21:48:18	61.51	5	307.55

Pada percobaan diatas, Tabel 6.18-Tabel 6.28 merupakan percobaan hasil pengamatan, hasil pengamatan yang dimaksud merupakan percobaan yang dilakukan dengan kejauhan minimum sampai maksimal adalah 1-10meter pada ruangan tidak ada halangan. NRF24L01 pada setiap *node* sangat optimal digunakan diruang terbuka atau tidak ada halangan walaupun terjadi angin kencang, akan tetapi jika jarak nya terlalu jauh maka akan berpengaruh dalam kualitas pengiriman data dan waktu. Dapat dilihat dalam 10 kali percobaan yang dilakukan, jangka waktu yang telah diambil secara bersamaan dan hasil dari penerimaan dan pengambilan data pada setiap *node* sensor menghasilkan data yang berbeda dikarenakan kejauhan pada setiap pengujian berbeda-beda. Dapat dilihat pada 10 kali percobaan delay menampilkan data ke setiap *node* berdasarkan urutan hanya memakan waktu 2-3 *second*. Data hasil sensor arus akan dikali sensor tegangan yang akan menghasilkan hasil daya pada setiap pengerjaan *node*. *Priority Scheduling* pada 10 kali percobaan pun cukup berhasil, karena dapat menampilkan urutan *node Priority Scheduling* berdasarkan daya terbesar hingga ke terkecil pada masing-masing *node*.

Berdasarkan pengujian diatas diketahui dengan pengujian penerimaan dan pengiriman data setiap *node* pada kejauhan 1-10 meter tidak terlalu berpengaruh pada *Priority Scheduling* yang mengutamakan menampilkan data daya berdasarkan daya terbesar hingga ke terkecil pada masing-masing modul *node*.

6.2.5 Analisis

Pada percobaan yang dilakukan pada pengiriman data ada halangan dikerjakan di sebuah ruangan yang terdapat penghalang menghasilkan kejauhan maksimal pengiriman dan penerimaan hasil data setiap *node* berjarak 10meter didapatkan rata-ratanya sebanyak 2-3 *second*, dalam pengiriman dan penerimaan data yang memiliki persentase keberhasilan 100%. Hasil yang ditunjukkan pada pengambilan data yang telah dilakukan pada pengujian ada halangan berbeda-beda dikarenakan pengaruh jarak dan masing-masing tempat pengujian. Pengujian dengan menggunakan metode *Priority Scheduling* pada 10 kali percobaan dengan jarak 1 meter-10meter pun berhasil, karena dapat menampilkan urutan *node* berdasarkan daya terbesar hingga ke terkecil pada masing-masing *node*. Sehingga ditarik kesimpulan bahwa pengujian pengiriman data *node* menggunakan *Priority Scheduling* adalah pengiriman data pada setiap *node* pada jarak 1 meter-10meter tidak berpengaruh pada *Priority Scheduling* yang mengutamakan menampilkan data daya pada masing-masing *node* berdasarkan daya terbesar hingga ke terkecil secara berurutan pada masing-masing *node*, pengambilan data daya menggunakan *Priority Scheduling* memiliki persentase keberhasilan 100%.



Pada pengujian ruang bebas atau tanpa halangan dilakukan pengujian sebanyak 10 kali percobaan diketahui pengiriman dan penerimaan data jarak maksimalnya sejauh 10 meter didapatkan rata-rata sebanyak 2-3 *second* yang memiliki persentase keberhasilan 100%. Hasil yang ditunjukkan pada pengambilan data yang telah dilakukan pada pengujian pada ruangan bebas atau tanpa halangan menunjukkan hasil data yang berbeda-beda dikarenakan pengaruh jarak dan masing-masing pada tempat pengujian. Pengujian dengan menggunakan *Priority Scheduling* pada 10 kali percobaan dengan jarak 1 meter-10 meter pun berhasil, karena dapat menampilkan urutan *node* berdasarkan daya terbesar hingga ke terkecil pada masing-masing *node*. Sehingga dapat ditarik kesimpulan bahwa pengujian pengiriman data menggunakan metode *Priority Scheduling* pada ruangan terbuka atau tanpa halangan pada masing-masing *node* dengan jarak 1 meter – 10 meter tidak mempengaruhi pengiriman data, dikarenakan *Priority Scheduling* mengutamakan menampilkan data daya pada masing-masing *node* berdasarkan daya terbesar hingga ke terkecil secara berurutan pada masing-masing *node*, pengambilan data daya menggunakan *Priority Scheduling* pada ruangan terbuka atau tanpa halangan memiliki persentase keberhasilan 100%.

Pada penerapan modul *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* dalam penempatan kejauhan pada setiap *node* tidak dianjurkan untuk meletakkan pada kejauhan maksimal *transmisi* data sehingga mampu mengurangi keberhasilan pengiriman hasil data.



BAB 7 PENUTUP

Pada bab ini akan dibahas kesimpulan apa saja yang didapatkan dari penelitian, kemudian akan membahas saran yang berguna dalam peningkatan penelitian selanjutnya.

7.1 Kesimpulan

Peneliti akan mengambil kesimpulan menurut rumusan masalah, hasil pembuatan alat, dan pengujian dan implementasi apa saja yang sudah dilakukan sebagai berikut:

1. Pada pengujian *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* terdapat modul sensor DHT 11 yang digunakan untuk pengujian hasil data sensing suhu yang menghasilkan persentase akurasi *sensing* senilai 99.26% sampai 98.50%. Rata-rata akurasi sensing pada pengujian data tegangan menunjukan hasil sebanyak 99.60%. Selanjutnya rata-rata akurasi sensing pada pengujian data arus menunjukan hasil sebanyak 94.86% sampai 97.86%. Performa pada sistem akan menghasilkan keluaran data dengan baik jika *transmitter* dan *receiver* pada masing-masing *node* diletakkan pada jarak optimal pada modul sensor *node*. Dibutuhkan modul sensor yang baik untuk dapat mengakuisisi data yang akan dihasilkan dan berjalannya pada suatu pengujian yang akan dikerjakan.
2. Pada percobaan mengirimkan hasil data dengan mengimplementasikan algoritma *Priority Scheduling node* yang didahulukan/diprioritaskan adalah kiriman data yang memiliki daya terbesar. Dari ke-4 *node* hasil dari data akan ditampilkan secara berurutan dari daya terbesar hingga ke terkecil. Program metode *Priority Scheduling* diletakkan pada *node master*. *Node master* akan mengirimkan *request* ke masing-masing *slave 1*, *slave 2*, dan *slave 3*. Jika tidak ada balasan maka *node master* akan terus mengirimkan *request* ke masing-masing *slave*. Setelah itu, *node master* akan mengecek apakah semua *request* sudah diterima ke masing-masing *node*. Pengecekan dilakukan dengan melihat no id yang masuk, *master* memiliki no id 0, *slave 1* memiliki no id 1, *slave 3* memiliki no id 3 dan jika sudah melakukan pengecekan atau ada jawaban dari setiap *node slave* maka *node master* akan membaca data pada sensor arus, tegangan, suhu, kelembaban pada masing-masing *slave node*. Setelah membaca data dari *slave 1 node*, *slave 2 node*, dan *slave 3 node* maka *master node* akan membaca data sendiri dan data diubah menjadi data array. Pada saat semua data masuk, maka akan dilakukan pengecekan terhadap data daya dalam *variable array*. Data daya dalam *variable array* tersebut kan diurutkan berdasarkan dari data daya terbesar hingga ke terkecil. Setelah itu *node master* akan mencetak dan menampilkan hasil data ke *serial monitor* berdasarkan urutan data daya *node* dari yang terbesar hingga ke terkecil secara berurutan.
3. Pada pengujian *master node*, *slave 1 node*, *slave 2 node*, dan *slave 3 node* dengan memakai algoritma *Priority Scheduling* diketahui hasil data pada setiap



node dari data suhu, kelembaban, arus, tegangan, dan daya memperlihatkan hasil data tidak sama karena kejauhan dan keadaan lokasi alat ditempatkan dapat mengakibatkan hasil penerimaan dan pengiriman data. *Node master* memiliki rata-rata daya yang didapatkan pada pengujian ada halangan pada jarak 1 – 10meter adalah 315.72 mW, pada *slave* 1 rata-rata daya yang dihasilkan 315.83 mW, pada *slave* 2 rata-rata daya yang dihasilkan 312.36 mW, dan pada *slave* 3 rata-rata daya yang dihasilkan adalah 315.39 mW. Sedangkan pada pengujian ruangan bebas atau tidak ada halangan rata-rata daya yang dihasilkan dari *node master* adalah 314.18 mW, pada *slave* 1 rata-rata daya yang dihasilkan 313.47 mW, pada *slave* 2 rata-rata daya yang dihasilkan 315.43 mW, dan pada *slave* 3 rata-rata daya yang dihasilkan 311.92 mW.

4. Hasil pengujian pada kondisi ruangan ada halangan dan tanpa ada halangan pada kejauhan 1 – 10meter dengan rata-rata percobaan penghitungan waktu pada saat mengirimkan dan menampilkan data pada setiap *node* waktu *transmisi* 2-3 *second* memiliki persentase kesuksesan menerima dan menampilkan data 100%. Performa mengirimkan dan menerima pada *transmisi* data di setiap sensor *node* ke *receiver* sensor *node* dipengaruhi oleh kejauhan, jika pengiriman data dilakukan melebihi batas optimal, maka performa transmisi data akan buruk. Pengiriman gelombang dengan menggunakan *transmitter* dan *receiver* wajib sama, sebab bakal mempengaruhi hasil dari mengirimkan dan menerima data sehingga dikirimkan secara baik.

7.2 Saran

Berikut merupakan beberapa saran dari peneliti tunjukan untuk peningkatan sistem selanjutnya adalah:

1. Dibutuhkan penelitian selanjutnya dalam pemilihan alat-alat apa saja yang digunakan pada setiap modul *node*, apakah mikrokontroler yang digunakan dapat dioptimalkan sebaik mungkin dalam penelitian, pengecekan algoritma yang dipilih juga apakah dapat berfungsi atau tidak, lebih memilih dan meneliti lagi dalam pemilihan modul sensor yang akan dicoba dalam pengujian, apakah modul tersebut bisa membuat konsumsi daya pada setiap *node* lebih menghemat penggunaan daya atau tidak dan dapat melihat apakah akurasi penerimaan dan pengiriman data bisa berjalan dengan baik atau tidak.
2. Diperlukan implementasi pengembangan berikutnya yang dapat digunakan dalam penerimaan dan pengiriman data serta memakai alat komunikasi yang yang selalu *update*.



DAFTAR REFERENSI

- Abad, M.F.K. & Jamali, M.A.J. 2011. *Modify LEACH Algorithm for Wireless Sensor Network*. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1
- Anastasi, G., Conti, M., Francesco, M.D. & Passarella, A. 2008. *Energy Conservation in Wireless Sensor Networks : A Survey*.
- Arduino, 2016, Arduino Uno Board, <https://www.arduino.cc/en/Main/ArduinoBoardUno>, diakses 11 Februari 2016.
- Arrosyid, Moch Harun etc. 2018 . *Implementasi Wireless Sensor Network Untuk Monitoring Parameter Energi Listrik Sebagai Peningkatan Layanan Bagi Penyedia Energi Listrik*. Politeknik Elektronika Negeri Surabaya(PENS). Institut Teknologi Sepuluh Nopember(ITS), Surabaya, Indonesia.
- Atefi, Kayvan etc. 2011. *Real-Time Scheduling Strategy For Wireless Sensor Networks* O.S. International Journal of Distributed and Parallel Systems (IJDPS) Vol.2, No.6, November 2011
- Balai pustaka, Departemen Pendidikan Nasional, edisi ketiga tahun 2007
- Banzi, Massimo. 2008. *Getting started with Arduino*. O'reilly
- Chang, J.Y., & Ju, P.H. 2012. *An efficient cluster-based power saving scheme for wireless sensor networks*. EURASIP Journal on Wireless Communications and Networking.
- Cormen, Thomas H. 2001. *Introduction to Algorithms, 2nd Edition*. The MIT Press, Cambridge.
- D. E. Culler, H. Wei, "Wireless Sensor Networks", Communications of the ACM, 2004, 47(6) : 30-33a
- D. E. Culler, H. Wei, "Wireless Sensor Networks", Communications of the ACM, 2004, 47(6) : 30-33.
- Erwanda, A. N. (2016). *Implementasi Time Synchronization Pada Wsn Untuk Metode Tdma Menggunakan Algoritma Tpsn*. Malang, Indonesia: Universitas Brawijaya.
- Fahri, Ridwan. (2009). *Studi Kinerja Protokol Routing On-Demand pada Jom FTEKNIK Volume 1 No. 2 Oktober 2014 9 Wireless Ad-Hoc Network Menggunakan Network Simulator-2*. Department of Electrical Engineering, Institut Teknologi Bandung
- Harkut, D.G. etc. 2014. *Real-Time Scheduling Algorithms For Wireless Sensor Network*. Circuits and Systems: An International Journal (CSIJ), Vol. 1, No. 1, January 2014
- Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H. (2000). *Energyefficient communication protocol for wireless microsensor networks*. Proc 33rd Hawaii International Conference on System Sciences (pp. 1-10).



Hidayat, M. Fatikh , Henryranu. P, Barlian , Maulana, Rizal, 2018, *Implementasi Low Power Multi Sensor Node pada Wireless*.

Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. 2000. *System Architechire Directions for Networked*. United State of America.

J. Sen.2009. "A Survey on Wireless Sensor Network Security," International Journal of Communication Networks and Information Security (IJCNIS), vol. 1, no. 2, pp. 55-78.

Jason L.H. 2003. *System Architecture for Wireless Sensor Networks*. Dissertation. University of California Berkeley. United State of America.

Jovanovic, M. D. Djordjevic, G. L. Nikolic, G. S. Petrovic, D. 2011. *Multi-Channel Media Access Control for Wireless Sensor Networks: a Survey*

Juniarahmatunisa, Astri Dwi. 2014. *Sistem Penjadwalan Penayangan Iklan dengan Menggunakan Algoritma Priority Scheduling di PT. Bandung Media Visual (I Channel TV)*. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)

Karthikeyan, R. etc. 2014. *Priority based Packet Scheduling Approach for Wireless Sensor Networks*. An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014.

Lewis, F.L. 2004. *Wireless Sensor Network*. University of Texas.

Mao Ye, Chengfa Li, Guihai Chen, & Jie Wu. 2005. *EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks*

Mittal, R., Bathia, M.P.S. 2010. *Wireless Sensor Networks for Monitoring the Environmental Activities*. IEEE Journals.

Munir, Rinaldi. 2011. *Algoritma dan pemrograman : Dalam bahasa pascal dan C, Informatika*, Bandung.

Nikolic, G., Stojcev, M., Stamenkovic, z. 2014. *Wireless Sensor node With Low-Power Sensing*. *Electronics and Energetics* Vol. 27.

Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta : Penerbit Andi

Sohraby, k., Minoli, D. & Znati, T., 2007. *Wireless Sensor Networks Technology, Protocols, and Applications*. New Jersey: John Wiley & Sons, Inc.

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

Tama, Harka Putra. 2010. *Perancangan dan Impelementasi Wireless Sensor Network (WSN) Berbasis IEEE 802.15.4/ZigBee*. Universitas Komputer Indonesia, Bandung, Indonesia.

Wulandari S, dkk, ,2016, *Implementasi Protocol Rounting Gossiping pada Wireless Sensor Network Menggunakan Komunikasi RF*, Jurnal Mahasiswa PTIK UB, Vol 8.