



**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN *WIRELESS SENSOR NETWORK*  
PADA PENDETEKSI DINI POTENSI KEBAKARAN  
LAHAN GAMBUT MENGGUNAKAN BANANA PI IOT**

Hendra Irawan  
NRP 2215 105008

Dosen Pembimbing  
Dr. Muhammad Rivai, S.T., M.T.  
Fajar Budiman S.T., M.,Sc

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 141599**

**DESIGN OF WIRELESS SENSOR NETWORK IN THE  
EARLY DETECTION OF POTENTIAL FIRE PEATLINE  
USING BANANA PI IOT**

Hendra Irawan  
NRP 2215 105008

Supervisors  
Dr. Muhammad Rivai S.T.,M.T  
Fajar Budiman S.T.,M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul "**Rancang Bangun *Wireless Sensor Network* pada Pendeteksi Dini Potensi Kebakaran Lahan Gambut Menggunakan Banana Pi IoT**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 31 Juli 2017



Hendra Irawan

*(halaman ini sengaja dikosongkan)*

**RANCANG BANGUN WIRELESS SENSOR NETWORK PADA  
PENDETEKSI DINI POTENSI KEBAKARAN LAHAN  
GAMBUT MENGGUNAKAN BANANA PI IOT**

**TUGAS AKHIR**

**Dilajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Elektronika  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

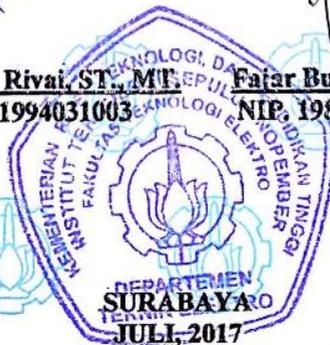
Menyetujui

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Muhammad Rivai, ST., MT.  
NIP : 196904261994031003

Fajar Budiman, S.T., M.Sc.  
NIP. 198607072014041001



*(halaman ini sengaja dikosongkan)*

RANCANG BANGUN *WIRELESS SENSOR NETWORK* PADA  
PENDETEKSI DINI POTENSI KEBAKARAN LAHAN GAMBUT  
MENGUNAKAN BANANA PI IOT

Nama : Hendra Irawan  
Dosen Pembimbing : 1. Dr. Muhammad Rivai S.T., M.T.  
2. Fajar Budiman S.T.,M.Sc.

## ABSTRAK

Lahan gambut merupakan bagian dari ekosistem yang banyak mengandung bahan organik yang tidak mudah terurai. Seringnya terjadi kebakaran lahan gambut membuat Indonesia kerap menderita kabut asap, sehingga diperlukan suatu sistem yang dapat mendeteksi potensi terjadinya kebakaran lahan gambut. Telah banyak penelitian tentang hal penanganan masalah tersebut, namun penelitian yang ditawarkan biasanya berupa pendeteksi paska kebakaran lahan gambut. Selain itu banyak teknologi yang ditawarkan mempunyai harga produksi yang mahal dan penggunaan daya yang tinggi. Sistem yang dirancang penulis adalah penerapan *Wireless Sensor Network*. Sistem tersebut menggunakan topologi *tree* pada pendeteksi dini potensi kebakaran lahan gambut yang diintegrasikan dengan *Internet of Things*. Menggunakan Arduino sebagai prosesor dari sensor node yang dibuat dan menggunakan Banana Pi sebagai server dari keseluruhan sistem. Sistem ini mengutamakan penanganan sebelum terjadinya kebakaran dengan membaca suhu lingkungan setiap node. Keluaran dari sistem yang dibuat menunjukkan keberhasilan dalam menyampaikan informasi potensi kebakaran. Daya yang dikonsumsi sensor node yang dibuat adalah 6.57W, jarak optimum peletakan antar node adalah 100m dan kesuksesan membaca GPS mencapai 80%.

Kata kunci: Banana Pi, *Internet of Things*, Lahan Gambut, *Wireless Sensor Network*.

*(halaman ini sengaja dikosongkan)*

DESIGN OF WIRELESS SENSOR NETWORK IN THE EARLY  
DETECTION OF POTENTIAL FIRE PEATLAND USING  
BANANA PI IOT

Name : Hendra Irawan  
Supervisors : 1. Dr. Muhammad Rivai S.T., M.T.  
2. Fajar Budiman S.T.,M.Sc.

***ABSTRACT***

Peatlands are part of the ecosystem which contains many organic materials that are difficult to decompose. The frequent peatland fires cause Indonesia to suffer from haze, so it needs a system that can detect the potential of peat land fires. There has been a lot of research on how to handle the problem, but the research offered is usually a post-fire peatland detector. In addition, many of the technologies are offered have expensive production prices and high power usage. So that can be interpreted the system offered ignore the losses caused by fire. The system designed by the author is the application of Wireless Sensor Network. The system uses a tree topology on early detection of potential peatland fires which is integrated with the Internet of Things. Arduino is used as the processor of the node sensor and Banana Pi is used as the server of the whole system. This system prioritizes handling prior to fire by reading the ambient temperature of each node. The output of the created system shows the success in sending information of potential fire. The power consumed by the node sensor is 6.57W, the optimum distance between the node is 100m and the success of GPS readings reached 80%.

Keywords: Banana Pi, Peatlands, Wireless Sensor Network, Internet of Things.

*(halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

### **RANCANG BANGUN *WIRELESS SENSOR NETWORK* PADA PENDETEKSI DINI POTENSI KEBAKARAN LAHAN GAMBUT MENGGUNAKAN BANANA PI IOT**

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahannya dari dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Dr. Muhammad Rivai S.T., M.T. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
2. Fajar Budiman S.T, M.Sc. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Dr. Tri Arief Sardjono, ST., MT., Astria Nur Irfansyah, ST., M.Eng. Ph.D, Rahmad Setiawan, ST., MT., dan Ir. Siti Halimah Baki, MT., selaku dosen penguji atas koreksi dan arahan selama penulis melaksanakan proses sidang.
4. Dr.Eng. Ardyono Priyadi, S.T., M.Eng. selaku Ketua Departemen Teknik Elektro ITS Surabaya.
5. Bapak, Ibu, adik serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
6. Seluruh dosen bidang studi elektronika.

7. Teman-teman Laboratorium Elektronika Industri yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Terakhir, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak. Penulis juga berharap Tugas Akhir ini dapat membantu pengembangan aplikasi *image processing*.

Surabaya, 24 Juli 2017

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN TUGAS AKHIR .....	iii
PEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	3
1.5. Metodologi .....	3
1.6. Sistematika Penulisan.....	4
1.7. Relevansi .....	5
BAB II TEORI PENUNJANG.....	7
2.1 Wireless Sensor Network (WSN) .....	8
2.1.1 Arsitektur Dasar WSN.....	8
2.1.2 Topologi yang Digunakan pada WSN .....	10
2.2 Internet of Things.....	11
2.3 Mikrokontroler.....	12
2.4 Sensor Suhu.....	14
2.5 Oprational Amplifier.....	15
2.5.1 Differential Amplifier.....	16
2.5.2 Non inverting Amplifier .....	17
2.6 Analog to Digital Converter .....	18
2.7 Radio Frequency.....	19
2.7.1 Modulasi Analog.....	20
2.7.2 Modulasi Digital.....	20
2.8 GPS (Global Potition System) .....	21
2.9 Banana Pi M2U.....	23
2.10 Python .....	24
2.11 Komunikasi Serial.....	25

2.11.1 Serial Peripheral Interface (SPI) .....	25
2.11.2 UART .....	26
2.12 Catu daya .....	27
2.13 Database .....	28
2.14 Website .....	28
<b>BAB III PERANCANGAN SISTEM.....</b>	<b>31</b>
3.1 Gambaran Umum Sistem .....	31
3.2 Perancangan Sensor Node .....	33
3.2.1 Sistem Sensor Node .....	34
3.2.2 Hardware Sensor Node .....	35
3.2.3 Software Sensor Node .....	44
3.3 Perancangan Server.....	47
3.3.1 Sistem Server .....	47
3.3.2 Hardware Server .....	48
3.3.3 Software Server .....	51
<b>BAB IV PENGUJIAN DAN ANALISA .....</b>	<b>53</b>
4.1 Pengujian Fungsional.....	54
4.1.1 Pengujian Catu Daya pada Sensor Node .....	55
4.1.2 Pengujian Sensor Suhu .....	56
4.1.3 Pengujian Komunikasi WSN .....	59
4.1.4 Pengujian GPS .....	61
4.1.5 Pengujian Server .....	63
4.1.6 Pengujian Database .....	64
4.1.7 Pengujian Website.....	65
4.2 Pengujian Internet of Things .....	67
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>69</b>
5.1 Kesimpulan .....	69
5.2 Saran .....	69
<b>DAFTAR PUSTAKA.....</b>	<b>71</b>
<b>LAMPIRAN .....</b>	<b>73</b>
<b>BIODATA PENULIS.....</b>	<b>99</b>

## DAFTAR GAMBAR

Gambar 2. 1 Kebakaran Lahan Gambut.....	7
Gambar 2. 2 Arsitektur WSN.....	9
Gambar 2. 3 WSN dengan <i>Multi Hop</i> Linear.....	9
Gambar 2. 4 WSN dengan <i>Sink Node</i> .....	10
Gambar 2. 5 Topologi Wireless Sensor Network .....	11
Gambar 2. 6 Blok Sistem IoT.....	12
Gambar 2. 7 Arduino Nano .....	13
Gambar 2. 8 Rangkaian Ekuivalen Sensor <i>Infrared</i> .....	15
Gambar 2. 9 Simbol <i>Operasional Amplifier</i> (Op-Amp).....	15
Gambar 2. 10 Differential Amplifier .....	17
Gambar 2. 11 Non Inverting Amplifier .....	18
Gambar 2. 12 Sinyal <i>Frequency Shift Keying</i> (FSK).....	21
Gambar 2. 13 Ilustrasi Satelit GPS .....	22
Gambar 2. 14 Banana Pi M2U .....	23
Gambar 2. 15 IDE Pyhton.....	24
Gambar 2. 16 Konfigurasi SIP .....	26
Gambar 2. 17 Konfigurasi TX dan RX.....	27
Gambar 2. 18 Rangkaian <i>Rectifier</i> .....	27
Gambar 2. 19 Logo Firebase .....	30
Gambar 2. 20 Pemrograman HTML Menggunakan Sublime.....	30
Gambar 2. 21 Tampilan Hasil Output pada Opera .....	30
Gambar 3. 1 Diagram Blok Sistem.....	32
Gambar 3. 2 Ilustrasi Sistem .....	33
Gambar 3. 3 Diagram Blok Sistem.....	35
Gambar 3. 4 Ilustrasi Sensor .....	35
Gambar 3. 5 Rangkaian Catu Daya 3,3V nRF24L01+ .....	37
Gambar 3. 6 Sensor Suhu TS118-3 .....	37
Gambar 3. 7 Rangkaian Operational Amplifier Thermopile.....	38
Gambar 3. 8 Rangkaian akuisisi Thermistor .....	39
Gambar 3. 9 Bentuk Modul A4988 dan Wiring Rangkaian .....	41
Gambar 3. 10 nRF24L01 .....	42

Gambar 3. 11 NEO N8M.....	43
Gambar 3. 12 Komunikasi nRF24L01 dan GPS Neo M8N .....	44
Gambar 3. 13 Diagram Blok pada Sensor Node .....	45
Gambar 3. 14 <i>Flowchart</i> Proses Pembacaan Suhu .....	46
Gambar 3. 15 Diagram Blok Server .....	47
Gambar 3. 16 Ubuntu MATE .....	50
Gambar 3. 17 Tampilan Perangkat Tambahan .....	51
Gambar 3. 18 Diagram Alir Perancangan Sistem.....	52
Gambar 4. 1 Bentuk Sistem yang Dibuat.....	53
Gambar 4. 2 Server dari Sistem yang Dibuat.....	54
Gambar 4. 3 Instalasi Pengukuran Arus .....	55
Gambar 4. 4 Pengujian Suhu dan Pembacaan ADC .....	58
Gambar 4. 5 Grafik ADC VS Suhu .....	58
Gambar 4. 6 Peta Area Percobaan.....	59
Gambar 4. 7 Hasil Serial Monitor Percobaan .....	60
Gambar 4. 8 Hasil Pembacaan <i>Receiver</i> pada Server .....	61
Gambar 4. 9 Hasil Pembacaan GPS .....	63
Gambar 4. 10 Tampilan Proses IO pada Server .....	64
Gambar 4. 11 Tampilan Awal Firebase .....	65
Gambar 4. 12 Letak Folder KlbPWFrGLW3zVOb7rOn.....	65
Gambar 4. 13 Tampilan Awal Web.....	66
Gambar 4. 14 Tampilan saat Dioperasikan .....	67
Gambar 4. 15 Tampilan dari Hasil Pembacaan Suhu Sekitar.....	68
Gambar 4. 16 Email Masuk Kepada Operator .....	68

## DAFTAR TABEL

Tabel 3. 1 Konfigurasi Penggunaan Pin I/O Arduino Sensor Node.	36
Tabel 3. 2 Konfigurasi Penggunaan Pin I/O Arduino Server .....	48
Tabel 4. 1 Hasil Pengukuran Daya Sensor Node.....	56
Tabel 4. 2 Pembacaan Suhu dan ADC pada Sensor .....	58
Tabel 4. 3 Percobaan Jarak Antar Node.....	60
Tabel 4. 4 Percobaan Pembacaan Koordinat GPS .....	62

*(halaman ini sengaja dikosongkan)*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Lahan gambut adalah bagian dari ekosistem hutan tropis dengan tanah yang lembab dan banjir berkala hingga menghalangi kayu dan daun mati dari proses pembusukan. Dengan luas lahan 26 juta Ha, Indonesia merupakan pemilik lahan gambut keempat terbesar di dunia setelah Amerika Serikat, Rusia, dan Kanada [1]. Dari segi manfaat lahan gambut memiliki peranan yang besar bagi kehidupan di sekitarnya, salah satu manfaatnya adalah lahan gambut dapat menyerap air dan melepaskannya secara perlahan, sehingga menjamin ketersediaan air sepanjang tahun. Lahan gambut juga berfungsi sebagai habitat pendukung keanekaragaman hayati yaitu sebagai lahan kehutanan, perkebunan, dan pertanian [2].

Namun, dengan potensinya yang besar, kebakaran lahan gambut menjadi hal yang sering terjadi di Indonesia. Setidaknya 2,6 juta Ha lahan gambut terbakar pada tahun 2015 yang menyebabkan kerugian hingga 221 triliun rupiah. Selain kerugian materi, terjadi juga kerugian non materi yang meliputi rusaknya habitat flora dan fauna, polusi yang mengganggu aktivitas, hingga masalah kesehatan [3].

Penyebab kebakaran gambut sangatlah beragam, mulai dari kesalahan manusia yaitu pembukaan lahan, pembuangan puntung rokok sembarangan, dan pembakaran sampah; faktor alam yaitu meliputi suhu yang tinggi, curah hujan yang sedikit, lahan yang mudah terbakar [4].

Puncak kebakaran hutan terjadi pada saat musim kemarau. Di saat kemarau lahan mulai mengalami pengeringan, kelembaban udara sangat rendah dan suhu lahan tinggi. Pada kondisi tersebut sangat mudah terpicunya kebakaran pada lahan gambut.

Beberapa solusi yang pernah ditawarkan adalah pengendalian secara manual sering dilakukan, untuk mengawasi terjadinya kebakaran lahan gambut, dibuat banyak gardu pandang untuk digunakan sebagai tempat mengawasi secara langsung kebakaran yang terjadi, lalu menanganinya bila telah melihat titik api. Namun cara ini dinilai sangat menguras tenaga dan membutuhkan banyak tenaga kerja dan tidak efisien karena tingkat fokus manusia yang terbatas, fisik yang terbatas, serta pemborosan sumberdaya manusia.

Pada zaman dewasa ini tentu kita tidak asing dengan teknologi satelit, terdapat satelit yang dapat memantau suhu permukaan bumi menggunakan citra kamera. Salah satu satelit yang dapat digunakan untuk memantau permukaan bumi adalah NOAA-AVHRR (*National Oceanic and Atmospheric Administration - Advanced Very High Resolution Radiometer*). Namun kelemahannya adalah resolusinya yang terlalu rendah yaitu 1 KM, hal tersebut membuat satelit tidak dapat mendeteksi kebakaran yang lebih kecil. Selain itu NOAA juga memiliki kelemahan berupa waktu pembacaan yang lama, diperlukan waktu satu hingga dua hari untuk memperoleh informasi yang diinginkan, sehingga bila terjadi kebakaran saat proses pembacaan baru akan diketahui satu hingga dua hari kemudian.

Ada juga teknologi yang menggunakan citra digital (*image processing*) yang menggunakan kamera inframerah namun tidak dapat meng-cover lahan yang sangat luas, lahan gambut disuatu daerah dapat mencapai ratusan hingga ribuan hektar.

Sehingga diperlukan teknologi yang mampu beroperasi di wilayah yang luas, mampu bekerja secara mandiri serta dapat memberikan informasi secara akurat dan *real time*.

## **1.2. Rumusan Masalah**

Sehubungan Dengan Latar Belakang yang telah dijelaskan sebelumnya, terdapat beberapa masalah yang akan dibahas antara lain sebagai berikut:

1. Cara membaca suhu lingkungan
2. Mengkomunikasikan data antar node sensor
3. Cara server mengirim data ke-*database*
4. Cara mengakses *database* dari perangkat yang lain
5. Cara menginformasikan terjadinya kebakaran lahan gambut

## **1.3. Batasan Masalah**

Batasan masalah yang akan dibahas dalam Tugas Akhir ini adalah:

1. Sensor yang digunakan merupakan sensor *thermopile* dengan jarak pembacaan 10 cm dari objek.
2. Perancangan Wireless Sensor Network menggunakan topologi tree.
3. Penerapan sistem dilakukan dengan pengkondisian lingkungan dikarenakan tidak adanya lahan gambut ditempat penelitian berlangsung.

4. *Cloud* yang digunakan adalah firebase.

#### 1.4. Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah

1. Membaca suhu lingkungan menggunakan sensor suhu *thermopile* dan dengan tambahan mekanik berupa motor *stepper*.
2. Menggunakan *Wireless Sensor Network* (WSN) untuk menerima dan mengirim data antyar node
3. Mengimplementasikan *Internet of Things* (IoT) pada sistem agar sistem dapat diakses dari jarak yang jauh
4. Merancang website sebagai *interface* yang dapat memberikan informasi terhadap data sensor
5. Mengirimkan pesan elektronik kepada pihak-pihak yang telah ditentukan untuk mengambil tindakan lanjutan

#### 1.5. Metodologi

Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. Studi literature  
Studi literatur tugas akhir ini bersumber pada jurnal-jurnal, buku referensi dan *datasheet* komponen yang digunakan.
2. Perancangan Hardware Wireless Sensor Network  
Sistem meliputi GPS, *Radio Frequency*, sensor dan hingga mikrokontroler. Sehingga diperlukan board yang dapat mengintegrasikan semuanya.
3. Perancangan perangkat lunak Wireless Sensor Network  
Perancangan perangkat lunak dimulai dari penginisialisasian node hingga komunikasi dengan server. Arduino Nano yang digunakan akan berkomunikasi secara serial dengan node lain maupun server.
4. Instalasi Server dan pembuatan pembuatan aplikasi *interface* antar perangkat  
Instalasi Banana Pi dengan OS hingga pembuatan aplikasi yang digunakan untuk berkomunikasi dengan node.
5. Pengimplementasian *Internet of Things* pada sistem  
Pengimplementasian IoT dengan pembuatan server secara *cloud* sehingga sistem akan bekerja secara mandiri apabila terjadi suatu permasalahan dilapangan. Tindakan yang dilakukan adalah dengan mengirimkan pesan elektronik.

6. Perancangan Website  
Perancangan website digunakan sebagai *interfice* antara sistem dengan user agar data yang diperoleh dapat mudah dimengerti dan dapat diakses dimana saja dengan koneksi internet.
7. Pengujian dan Perbaikan sistem  
Pengujian awal dilakukan per bagian sistem, yaitu rangkaian *Wireless Sensor Network*, sistem *Internet of Things* yang diimplementasikan, sistem peringatan, dan pemrograman pemetaan. Pada tahap selanjutnya, dilakukan pengujian sistem total dengan menempatkan sistem di suatu area yang luas. Lalu sistem dijalankan dan diamati hasil dari pemetaan dan dilakukan perbaikan sehingga hasil lebih optimal.
8. Penulisan laporan tugas akhir  
Tahap penulisan laporan tugas akhir dilakukan beriringan pengerjaan.
9. Penyusunan Jurnal Ilmiah POMITS  
POMITS (Publikasi Ilmiah Online Mahasiswa ITS) digunakan untuk mempublikasikan hasil Tugas Akhir. Jurnal ilmiah disusun berdasarkan template yang ada pada situs web POMITS.

## 1.6. Sistematika Penulisan

Sistematika penulisan pada tugas akhir ini dibagi menjadi beberapa bab bagian rincian:

### BAB 1 :PENDAHULUAN

Menguraikan latarbelakang, perumusan masalah, batasan masalah, tujuan dan manfaat yang berkaitan dengan pengerjaan dan penyusunan Tugas Akhir ini.

### BAB 2 :TEORI PENUNJANG

Pada bab ini dikemukakan berbagai macam dasar teori yang berhubungan dengan permasalahan yang dibahas, antara lain meliputi teori tentang pengenalan, WSN, IoT, HTML, Python, Firebase, mikrokontroler, Banana Pi, GPS, *Radio Frequency*, motor stepper, dan thermopile.

### BAB 3 :PERANCANGAN DAN RELEFANSI ALAT

Berisi tentang tahap-tahap perancangan sistem jaringan sensor, baik secara *hardware* dan *software*, komunikasi antar node dan komunikasi secara radio.

#### BAB 4 :PENGUJIAN DAN ANALISIS SISTEM

Bab ini membahas mengenai pengujian dari sistem yang telah diimplementasikan pada sistem jaringan sensor dan analisa data berdasarkan parameter yang ditetapkan.

#### BAB 5 :PENUTUP

Berisi tentang kesimpulan dan saran yang diperoleh dalam Tugas Akhir ini.

##### **1.7. Relevansi**

Manfaat dari tugas akhir ini adalah mengetahui kerja sistem jaringan sensor yang dihubungkan dengan penyimpanan database yang dilengkapi dengan mikrokontroler Radio Frequency, thermopile, GPS, dan Banana Pi.

*(halaman ini sengaja dikosongkan)*

## BAB II

### TEORI PENUNJANG

Pada umumnya bukan merupakan fenomena asing lagi di beberapa wilayah Indonesia, terutama Sumatera, Kalimantan dan Papua. Fenomena alam ini berkembang menjadi suatu bentuk bencana alam yang mempunyai dampak terhadap aspek-aspek kehidupan masyarakat. Gambut adalah jenis tanah yang terbentuk dari akumulasi sisa-sisa tumbuhan yang setengah membusuk; oleh sebab itu, kandungan bahan organiknya tinggi. Sisa-sisa tumbuhan yang mengering membuat lahan gambut menjadi mudah terbakar[5]. Untuk mencegah terjadinya kebakaran diperlukan sistem pendeteksi di potensi kebakaran. Dalam hal ini kami mengajukan suatu jaringan sensor nirkabel untuk mengatasi permasalahan yang telah di ungkapkan[6].

Kebakaran di lahan gambut biasanya diawali dengan penyulutan api di atas permukaan tanah. Api akan bergerak ke segala arah, termasuk ke bawah permukaan. Penjalaran api ke bawah permukaan yang membakar lapisan gambut dipengaruhi oleh kadar air lapisan gambut dan tidak dipengaruhi angin sebagai kebakaran bawah (*ground fire*). Api akan bergerak ke atas permukaan dipengaruhi oleh kecepatan dan arah angin sebagai kebakaran permukaan (*surface fire*) dan bila mencapai tajuk pohon akan menjadi kebakaran tajuk (*crown fire*). Dari ulasan tersebut, dirasa perlu adanya pembuatan sistem untuk mendeteksi dini dan pencegahan pada potensi kebakaran hutan[7]. Kebakaran lahan gambut seperti pada Gambar 2.1.



**Gambar 2. 1** Kebakaran Lahan Gambut

## 2.1 Wireless Sensor Network (WSN)

*Wireless Sensor Network* atau sering disingkat dengan WSN adalah suatu peralatan sistem *embedded* yang berkomunikasi tanpa kabel yang di dalamnya terdapat satu atau lebih sensor dan dilengkapi dengan peralatan sistem komunikasi. Sensor di sini digunakan untuk menangkap informasi sesuai dengan karakteristik dan penyajian informasi melalui komunikasi internet [8].

Pengembangan *Wireless Sensor Network* (WSN) kini sangat pesat, di mana pada saat ini WSN banyak digunakan dalam *smart city*, *smart home*, dan banyak contoh lain. Keringkasan perangkat dan biaya yang murah menjadi faktor mengapa WSN sekarang menjadi berkembang pesat dan sering digunakan.

### 2.1.1 Arsitektur Dasar WSN

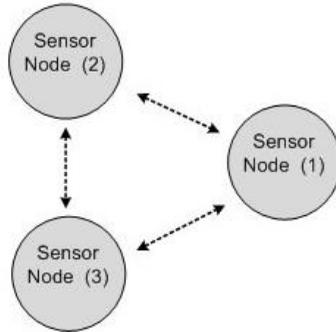
WSN memiliki arsitektur dasar yang wajib dimiliki. Penting untuk mengetahui apa yang dimaksud dengan RFD dan FFD pada jenis sensor serta masing-masing fungsi, prinsip dasar arsitektur, proses *Multi Hop* di dalam WSN, Hubungan WSN dengan *gateway*, serta hubungan WSN dengan *node*.

#### 1. Tipe RFD dan FFD pada Arsitektur WSN

Dalam pengimplementasiannya dilapangan jaringan sensor *node-node* sensor saling berkomunikasi satu dengan yang lain dan saling bertukar data, sehingga sampai ke *node* tujuan. Kemudian *Gateway node* mengirimkan data-data tersebut ke server. Namun dalam jaringan sensor terdapat pilihan jenis *node* sensor yaitu RFD (*Reduced Function Device*) dan FFD (*Full Function Device*). RFD berfungsi keperluan *node* yang berfokus pada pengambilan data saja untuk menghemat daya. Sedangkan FFD selain berfungsi sebagai sensor, FFD juga berfungsi meneruskan data dari RFD maupun FFD yang lain.

#### 2. Prinsip Dasar Arsitektur WSN

Arsitektur dasar WSN prinsipnya terdiri dari beberapa *node* sensor yang dapat saling berkomunikasi satu sama lain dengan bertujuan transfer data atau fungsi kontrol yang lain. Perangkat sensor terhubung secara *Ad Hoc* (dapat berkomunikasi langsung tanpa perantara jaringan lain) dan mendukung *Multi Hop*. Gambar 2.2 merupakan contoh arsitektur WSN.



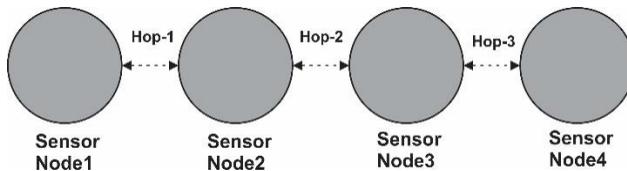
**Gambar 2. 2** Arsitektur WSN

3. Kemampuan *Multi Hop* pda WSN

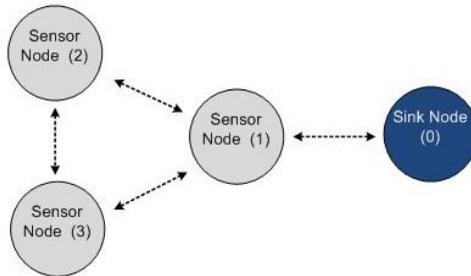
*Wireless Sensor Network* mempunyai kemampuan *Multi Hop*, yang memungkinkan setiap node dapat berkomunikasi dengan sensor yang lain dengan menggunakan sensor yang di tengahnya sebagai perantara (berperan sebagai *Router*). Sebagai ilustrasi dapat dilihat pada Gambar 2.3.

4. WSN dan Sink Node

Sebuah WSN dapat terdiri dari banyak sensor node dan sebuah *Sink Node*. *Sink node* dapat berupa pengumpul data (*Data Collector*) atau sebagai *Gateway* (pintu gerbang) ke jaringan internet. *Sink node* dapat dikatakan node paling ujung, ilustrasi dapat dilihat pada gambar dibawah



**Gambar 2. 3** WSN dengan Multi Hop Linear



**Gambar 2. 4** WSN dengan *Sink Node*

## 5. WSN dan *Node*

Dengan demikian, sebuah WSN dapat disimpulkan sebagai kumpulan dari beberapa *node* dengan peran dan fungsi masing-masing. Fungsi masing-masing *node* dibedakan berdasarkan kebutuhan yang ada dilapangan seperti berikut:

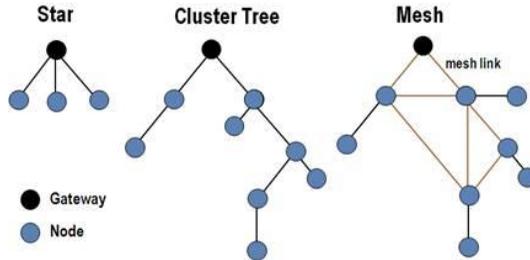
- a. *Sensor node* digunakan sebagai membaca data-data yang ada dilingkungan sesuai objek yang dipantau.
- b. *Route node* berfungsi untuk meneruskan paket data ke *node* yang lain. *Node* ini berfungsi sebagai komunikasi *Multi Hop*. *Sensor node* juga dapat berfungsi sebagai *router node*.
- c. *Sink node* atau *gateway node* berfungsi sebagai menyimpan data yang dikirimkan dari *sensor node*, kemudian meneruskannya ke jaringan internet.

### 2.1.2 Topologi yang Digunakan pada WSN

Dalam penggunaannya ada beberapa jenis topologi yang sering digunakan dalam WSN. Pemilihan topologi bertujuan menyesuaikan kebutuhan dan medan penempatan jaringan. Adapun beberapa topologi jaringan yang umum digunakan dalam pembangunan sebuah sistem WSN adalah sebagai berikut:

#### 1. Topologi *Star*

Topologi ini merupakan topologi paling dasar, dimana setiap *node* mempertahankan suatu jalur komunikasi langsung dengan *gateway*. Topologi ini sederhana namun membatasi jarak *node* yang dapat dicapai.



**Gambar 2. 5** Topologi *Wireless Sensor Network* [9]

## 2. Topologi *Cluster/Tree*

Arsitektur topologi *cluster* lebih kompleks dibanding dengan topologi *star*. Setiap *node* masih mempertahankan satu jalur komunikasi untuk *gateway*. Pengiriman data dapat dilakukan melalui *node-node router*, namun masih dalam satu jalur tersebut.

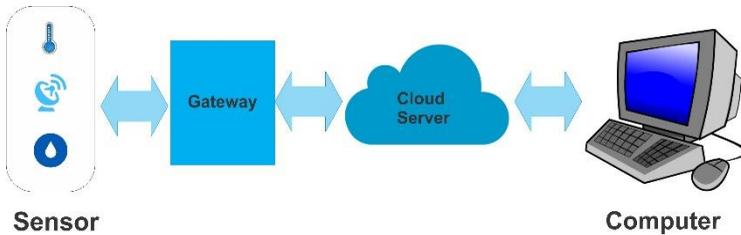
## 3. Topologi *Mash*

Topologi ini merupakan topologi yang lebih sempurna dibanding topologi-topologi sebelumnya. Dengan menggunakan jalur komunikasi yang lebih banyak untuk meningkatkan kehandalan sistem. Dalam sebuah jaringan *mash*, *node* akan mencari jalan untuk menuju ke *gateway* [10]. Adapun bentuk topologi dari WSN dapat dilihat pada Gambar 2.2.

## 2.2 *Internet of Things*

Secara umum *Internet of Things* (IoT) adalah suatu konsep menghubungkan perangkat pintar dengan perangkat pintar yang lain baik digunakan untuk bertukar data maupun untuk keperluan pengendalian melalui internet. Diperjalananya kini, konsep IoT sudah banyak digunakan pada objek di sekitar kita. Diperkirakan pada tahun 2020 dala 50 miliar objek yang terhubung dengan Internet [11].

Meski telah digunakan pada banyak perangkat untuk menunjang kehidupan sehari-hari, *Internet of Things* belum memiliki definisi yang baku. Namun secara sederhana konsep *Internet of Things* dapat dilihat pada Gambar 2.6.



**Gambar 2. 6** Blok Sistem IoT

Dalam perkembangannya IoT erat dengan komunikasi *machine-to-machine* (M2M) dalam bidang pengoperasiannya. *Machine-to-machine* adalah komunikasi yang terjadi antara mesin dengan mesin tanpa memerlukan komunikasi manusia ke manusia atau manusia ke mesin [12].

### 2.3 Mikrokontroler

Mikrokontroler sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (market need) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara massal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor). Sebagai kebutuhan pasar, mikrokontroler hadir untuk memenuhi selera industri dan para konsumen akan kebutuhan dan keinginan alat-alat bantu bahkan mainan yang lebih baik dan canggih.

Tidak seperti sistem komputer, yang mampu menanganiberbagai macam program aplikasi (misalnya pengolah kata, pengolah angkadan lain sebagainya), mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa disimpan). Perbedaan lainnya terletak pada perbandingan RAM dan ROM. Pada sistem komputer perbandingan RAM dan ROM-nya besar, artinya program-program pengguna disimpan dalam ruang RAM yang relatif besar, sedangkan rutin-rutin antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada Mikrokontroler, perbandingan ROM dan RAM-nya yang besar, artinya program kontrol disimpan dalam ROM (bisa Masked ROM atau Flash PEROM) yang ukurannya relatif lebih besar, sedangkan

RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan.

Arduino Nano seperti pada Gambar 2.7 adalah papan pengembangan (*development board*) mikrokontroler yang berbasis *chip* ATmega328P dengan bentuk yang sangat kecil. Arduino Nano memiliki 14 buah digital pin yang dapat digunakan sebagai *input* atau *output*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki resistor pull-up sekitar 20-50k *ohm* (secara default dalam posisi *disconnect*). Nilai maksimal adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroler. Arduino sendiri didukung dengan IDE pemrogramannya sendiri dengan bahasa pemrograman C. IDE pada Arduino dapat sangat membantu dari proses pemrograman karena sangat mudah dalam pengoprasianya.

Beberapa pin memiliki fungsi khusus :

1. Serial, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
2. *External Interrupts*, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interups*.
3. PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit  
SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK)  
mendukung komunikasi SPI dengan menggunakan SPI Library
4. LED : Pin 13. Pada pin 13 terhubung *built-in* led yang dikendalikan oleh digital pin nomor 13.



**Gambar 2. 7** Arduino Nano [17]

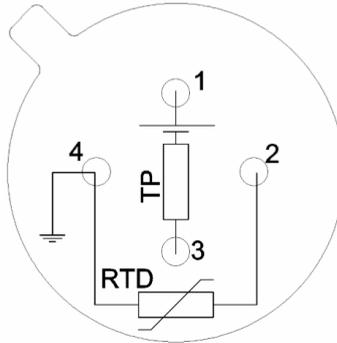
## 2.4 Sensor Suhu

Termometer adalah alat yang digunakan untuk mengukur suhu (temperatur), ataupun perubahan suhu. Istilah termometer berasal dari bahasa Latin *thermos* yang berarti bahang dan *meter* yang berarti untuk mengukur. Pengukuran suhu dapat dilakukan menggunakan sensor. Sensor yang digunakan untuk mengukur suhu terbagi dua yakni sensor kontak dan sensor non-kontak. Beberapa sensor kontak adalah termokopel, termistor, dan RTDs. Salah satu sensor non-kontak adalah termometer infra merah. Alat ini mengukur panas (energi infra merah) dari objek dengan memfokuskan energi ini melalui sistem optik menggunakan detektor. Signal dari detektor kemudian disajikan dalam suhu setelah melalui serangkaian proses.

*Infrared Thermometer* menawarkan keuntungan, yakni mampu melakukan pengukuran suhu yang cepat dan akurat dengan objek dari kejauhan dalam menentukan temperatur objek tersebut tanpa kontak fisik. Sehingga sistem pengukurannya tidak terkontaminasi oleh objek, dan rusak. Banyak penggunaan teknologi ini yang berhubungan dengan industri yakni memberikan keuntungan pada pencatat temperatur dalam situasi dimana objek tidak bisa dicapai atau bergerak dengan cepat, dimana kontak tidak mungkin dilakukan karena temperatur terlalu tinggi atau dibawah pengaruh listrik.

*Infrared Thermometer* terdiri dari lensa pemfokus energi infra merah pada detector, dapat mengubah energi menjadi sinyal elektrik yang bisa ditunjukkan dalam unit temperatur setelah disesuaikan dengan variasi temperatur lingkungan. Konfigurasi fasilitas pengukur suhu ini bekerja dari jarak jauh tanpa menyentuh objek. Dengan demikian, termometer infra merah berguna mengukur suhu pada keadaan dimana termokopel atau sensor tipe lainnya tidak dapat digunakan atau tidak menghasilkan suhu yang akurat untuk beberapa keperluan pengukuran.

Pada Gambar 2.8 merupakan gambar dari rangkaian ekuivalen dari *Infrared Thermometer*. Dari gambar tersebut dapat dilihat bahwa kaki 1 dan 3 adalah keluran dari sinyal *Infrared Thermometer* yang rangkaian ekuivalensinya berupa hambatan dan sumber tegangan. Sedangkan pada kaki 2 dan 4 merupakan keluaran dari sensor RTD yang rangkaian ekuivalen nya berupa hambatan yang akan berubah seiring perubahan suhu pada sensor.

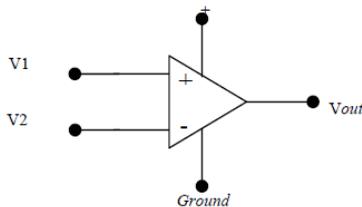


**Gambar 2. 8** Rangkaian Ekuivalen Sensor Infrared

## 2.5 *Operational Amplifier*

*Operational amplifier* (Op-Amp) adalah suatu penguat berpenguatan tinggi yang terintegrasi dalam sebuah *chip* IC yang memiliki dua *input inverting* dan *noninverting* dengan sebuah terminal *output*, dimana rangkaian umpan balik dapat ditambahkan untuk mengendalikan karakteristik tanggapan keseluruhan pada *operational amplifier* (Op-Amp). Pada dasarnya *operational amplifier* (Op-Amp) merupakan suatu penguat diferensial yang memiliki 2 input dan 1 *output*.

Op-amp ini digunakan untuk membentuk fungsi-fungsi linier yang bermacam-macam atau dapat juga digunakan untuk operasi-operasi tak linier, dan seringkali disebut sebagai rangkaian terpadu linier dasar. Penguat operasional (Op-Amp) merupakan komponen elektronika analog yang berfungsi sebagai amplifier multiguna dalam bentuk IC dan memiliki simbol seperti pada Gambar 2.9.



**Gambar 2. 9** Simbol Operasional Amplifier (Op-Amp)

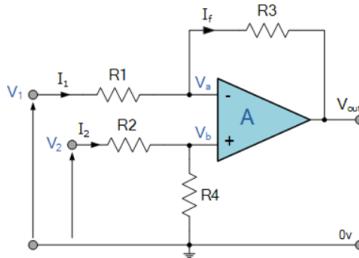
Prinsip kerja sebuah operasional Amplifier (Op-Amp) adalah membandingkan nilai kedua input (input *inverting* dan input non-*inverting*), apabila kedua input bernilai sama maka output Op-amp tidak ada (nol) dan apabila terdapat perbedaan nilai input keduanya maka output Op-amp akan memberikan tegangan output. Operasional amplifier (Op-Amp) dibuat dari penguat diferensial dengan 2 input. Sebagai penguat operasional ideal, operasional amplifier (Op-Amp) memiliki karakteristik sebagai berikut :

1. Impedansi Input ( $Z_i$ ) besar =  $\infty$
2. Impedansi Output ( $Z_o$ ) kecil = 0
3. Penguatan Tegangan ( $A_v$ ) tinggi =  $\infty$
4. Band Width respon frekuensi lebar =  $\infty$
5.  $V_o = 0$  apabila  $V_1 = V_2$  dan tidak tergantung pada besarnya  $V_1$ .

### **2.5.1 Differential Amplifier**

Penguat diferensial dalam suatu penguat operasional (Op-Amp) dibuat menggunakan kopling langsung (DC kopling) yang bertujuan untuk menghilangkan efek yang ditimbulkan akibat penambahan atau pemasangan kapasitor *bypass* maupun kapasitor kopling. Penggunaan kopling DC pada penguat diferensial ini bertujuan untuk menghindari permasalahan perlambatan yang terjadi akibat pengisian muatan pada kapasitor-kapasitor kopling (penggandeng) oleh tegangan sumber DC, dengan demikian titik kerja DC untuk mencapai titik stabil diperlukan juga waktu tunda (time constant). Sehingga mengakibatkan terjadinya efek kenaikan batas frekuensi bawah (fL) karena adanya kenaikan waktu untuk mencapai stabil (*time constant*) yang lebih lambat.

Karena penguat pasangan differensial didalamnya terdiri dari dua buah transistor, maka untuk mendapatkan titik kerja DC yang simetris, diperlukan dua buah transistor yang mempunyai konfigurasi bentuk phisis dengan karakteristik yang sama. Sedangkan untuk menghindari akibat pengaruh adanya perubahan temperatur yang berbeda pada kedua transistor tersebut, sebaiknya cara pemasangan kedua transistor adalah dibuat sedemikian rupa agar sedapat mungkin berpasangan-berhimpit satu sama lainnya.



Gambar 2. 10 *Differential Amplifier*

$$V_{out} = -\frac{R_2}{R_3} \cdot V_1 + \left(1 + \frac{R_2}{R_4}\right) \cdot \left(\frac{R_4}{R_2 + R_4}\right) \cdot V_2 \quad (2.1)$$

Ketika nilai resistansi  $R_1=R_2$  dan  $R_3 = R_4$  maka:

$$V_{out} = \frac{R_1}{R_3} \cdot (V_1 - V_2) \quad (2.2)$$

Bila

$$A = \frac{V_{out}}{V_{input}} = \frac{V_{out}}{(V_1 - V_2)} \quad (2.3)$$

Maka :

$$A = \frac{R_1}{R_3} \quad (2.4)$$

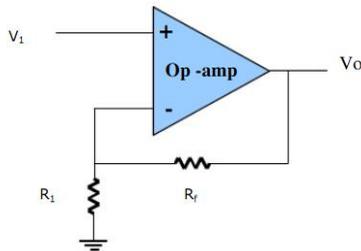
Pada Gambar 2.10 merupakan rangkaian dari penguat deferensial. Penguatan yang dihasilkan adalah seperti pada Persamaan 2.1. dengan kondisi tertentu sehingga persamaan rumus yang digunakan untuk mendapat besaran penguatan adalah seperti Persamaan 2.4.

### 2.5.2 Non inverting Amplifier

Non-Inverting Amplifier merupakan penguat sinyal dengan karakteristik dasat sinyal output yang dikuatkan memiliki fasa yang sama dengan sinyal input. Penguat non inverting amplifler dapat dibangun menggunakan penguat operasional, karena penguat operasional memang didesain untuk penguat sinyal baik membalik

ataupun tak membalik. Rangkaian penguat tak membalik ini dapat digunakan untuk memperkuat isyarat AC maupun DC dengan keluaran yang tetap sefase dengan sinyal input. Impedansi masukan dari rangkaian penguat tak membalik (non inverting amplifier) berharga sangat tinggi dengan nilai impedansi sekitar 100 Mohm.

Gambar 2.11 merupakan rangkaian dari penguatan non inverting. Persamaan yang dihasilkan dari rangkaian tersebut adalah seperti pada Persamaan (2.6).



**Gambar 2. 11** *Non Inverting Amplifier*

$$V_i = \frac{R_1}{R_1 + R_f} \cdot V_o \quad (2.5)$$

$$A = \frac{R_1 + R_f}{R_1} = 1 + \frac{R_f}{R_1} \quad (2.6)$$

## 2.6 Analog to Digital Converter

*Analog to digital converter* (ADC) adalah suatu piranti yang dirancang untuk mengubah sinyal - sinyal analog menjadi bentuk sinyal digital. Atau dapat pula disimpulkan ADC dapat merubah nilai suatu masukan yang berupa tegangan listrik dalam bentuk voltase atau sinyal analog lainnya menjadi keluaran berupa nilai digital. Dalam penggunaannya ADC adalah tegangan maksimum yang dikonversikan dari rangkaian pengkondisian sinyal, resolusi, ketepatan dan waktu konversinya.

Resolusi pada ADC berantung pada kerapatan bit pencuplik analog yang digunakan. Pada Arduino, bit pencuplik yang digunakan adalah 10 bit, dengan 5V DC sebagai  $V_{ref}$ . Sehingga dengan Persamaan 2.7 didapatkan resolusi pembacaan perbit sebesar 4,8mV.

$$Q = \frac{V_{ref}}{2^n} \quad (2.7)$$

$$Q = \frac{5}{2^{10}} = 4,88 \times 10^{-3} V = 4,88 \text{ mV}$$

Q = Resolusi (V)

$V_{ref}$  = Tegangan reverensi (V)

N = bit pencuplik

## 2.7 *Radio Frequency*

RF (Radio Frekuensi) adalah sebuah perangkat yang menerima frekuensi radio dalam kisaran tertentu. RF berfungsi sebagai teknologi komunikasi nirkabel pada aktifitas transfer data. Pada prakteknya, RF terdiri dari pemancar (*Transmitter*) dan penerima (*Receiver*) yang ditanam bersama pada suatu sirkuit. Untuk berkomunikasi, masing-masing radio harus memiliki rasio sinyal yang sama.

Untuk memperoleh frekuensi yang digunakan sebagai komunikasi pada perangkat maka diperlukan modulasi sinyal. Modulasi adalah proses perubahan (*varying*) suatu gelombang periodik sehingga menjadikan suatu sinyal mampu membawa suatu informasi. Dengan proses modulasi, suatu informasi (biasanya berfrekuensi rendah) bisa dimasukkan ke dalam suatu gelombang pembawa, biasanya berupa gelombang sinus berfrekuensi tinggi. Terdapat tiga parameter kunci pada suatu gelombang sinusioidal yaitu : amplitudo, fase dan frekuensi. Ketiga parameter tersebut dapat dimodifikasi sesuai dengan sinyal informasi (berfrekuensi rendah) untuk membentuk sinyal yang termodulasi.

Peralatan untuk melaksanakan proses modulasi disebut modulator, sedangkan peralatan untuk memperoleh informasi informasi awal (kebalikan dari dari proses modulasi) disebut

demodulator dan peralatan yang melaksanakan kedua proses tersebut disebut modem.

Modem berasal dari singkatan MODulator DEModulator. Perangkat keras ini digunakan untuk mengubah sinyal digital menjadi sinyal analog sehingga data dari komputer bisa dikirimkan melalui saluran telepon atau saluran lainnya.

Data dari komputer yang berbentuk sinyal digital diberikan kepada modem untuk diubah menjadi sinyal analog. Sinyal analog tersebut dapat dikirimkan melalui beberapa media telekomunikasi seperti telepon dan radio.

Setibanya di modem tujuan, sinyal analog tersebut diubah menjadi sinyal digital kembali dan dikirimkan kepada mikrokontroler. Informasi yang dikirim bisa berupa data analog maupun digital sehingga terdapat dua jenis modulasi yaitu

- modulasi analog
- modulasi digital [13]

### 2.7.1 Modulasi Analog

Dalam modulasi analog, proses modulasi merupakan respon atas informasi sinyal analog.

Teknik umum yang dipakai dalam modulasi analog :

1. Modulasi berdasarkan sudut
  - a. Modulasi Fase (*Phase Modulation* - PM)
  - b. Modulasi Frekuensi (*Frequency Modulation* - FM)
2. Modulasi Amplitudo (*Amplitudo Modulation* - AM)
  - a. *Double-sideband modulation with unsuppressed carrier (used on the radio AM band)*
  - b. *Double-sideband suppressed-carrier transmission (DSB-SC)*
  - c. *Double-sideband reduced carrier transmission (DSB-RC)*
  - d. *Single-sideband modulation (SSB, or SSB-AM), very similar to single-sideband suppressed carrier modulation (SSB-SC)*
  - e. *Vestigial-sideband modulation (VSB, or VSB-AM)*
  - f. *Quadrature amplitude modulation (QAM)*

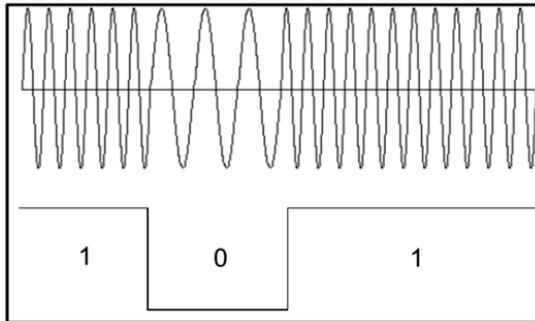
### 2.7.2 Modulasi Digital

Dalam modulasi digital, suatu sinyal analog di-modulasi berdasarkan aliran data digital.

Perubahan sinyal pembawa dipilih dari jumlah terbatas simbol alternatif. Teknik yang umum dipakai adalah :

- a. *Phase Shift Keying* (PSK), digunakan suatu jumlah terbatas berdasarkan fase.
- b. *Frekuensi Shift Keying* (FSK), digunakan suatu jumlah terbatas berdasarkan frekuensi.
- c. *Amplitudo Shift Keying* (ASK), digunakan suatu jumlah terbatas amplitudo.

Modulasi digital *Frequency Shift Keying* (FSK) seperti pada Gambar 2.12 merupakan sejenis *Frequency Modulation* (FM), dimana sinyal pemodulasinya (sinyal digital) menggeser *output*-nya antara dua frekuensi yang telah ditentukan sebelumnya, yang biasa diistilahkan frekuensi *mark* dan *space*. Modulasi digital dengan FSK juga menggeser frekuensi carrier menjadi beberapa frekuensi yang berbeda didalam *band*-nya sesuai dengan keadaan digit yang dilewatkannya. Jenis modulasi ini tidak mengubah amplitudo dari signal carrier yang berubah hanya frekuensi.

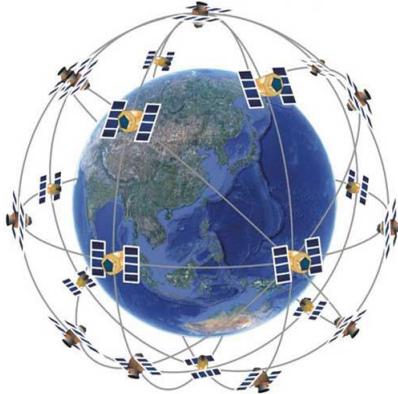


**Gambar 2. 12** Sinyal *Frequency Shift Keying* (FSK).

## 2.8 GPS (Global Potition System)

GPS (*Global Position System*) adalah sistem navigasi yang menggunakan satelit. Didesain dapat memberikan informasi posisi secara mudah, dengan posisi dapat dikenali hampir semua di permukaan bumi. Sistem navigasi merupakan bagian terpenting, terutama untuk mengetahui posisi suatu objek tanpa harus memerhatikan lingkungan sekitar. GPS memberikan informasi lokasi

dengan koordinat lintang (*latitude*) dan bujur (*longitude*) berdasarkan koordinat bumi.



**Gambar 2. 13** Ilustrasi Satelit GPS

Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberinama GPS reciever yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik. Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Uniknya, walau satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis. Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau area *coverage* yang lebih luas. Satelit-satelit ini akan selalu berada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya *blank spot* (area yang tidak terjangkau oleh satelit). Gambar 2.13 merupakan ilustrasi dari banyaknya satelit yang beredar dibumi.

Setiap mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi Anda di atas permukaan bumi. GPS reciever sendiri berisi beberapa integrated circuit (IC) sehingga murah dan teknologinya mudah untuk di gunakan oleh semua orang. GPS dapat digunakan

untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan di integrasikan dengan komputer maupun laptop [14].

## 2.9 Banana Pi M2U

Banana Pi M2U adalah *Board* berukuran kecil yang memiliki konsumsi daya rendah. Dengan menggunakan prosesor *Quad core* dan RAM 2 GB membuat Banana Pi M2U sangat bagus digunakan untuk pemrosesan data yang berat. Selain itu, Banana Pi M2U dilengkapi juga dengan WiFi yang mempermudah melakukan koneksi ke internet. OS Banana Pi kompatibel dengan *Android KitKat*, *Resbian*, *Ubuntu*, dan lain-lain.

Banana Pi M2U dilengkapi dengan *General-Purpose Input/Output* (GPIO) yang mempermudah dalam pemrosesan *input-output* pada *Banana Pi*. Pada GPIO terdapat *port-port* unik yang dapat digunakan, mulai dari pin I/O, pin ADC, pin serial dan pin Power. Bentuk *Banana Pi* dapat dilihat pada Gambar 2.14 [15].



**Gambar 2. 14** Banana Pi M2U

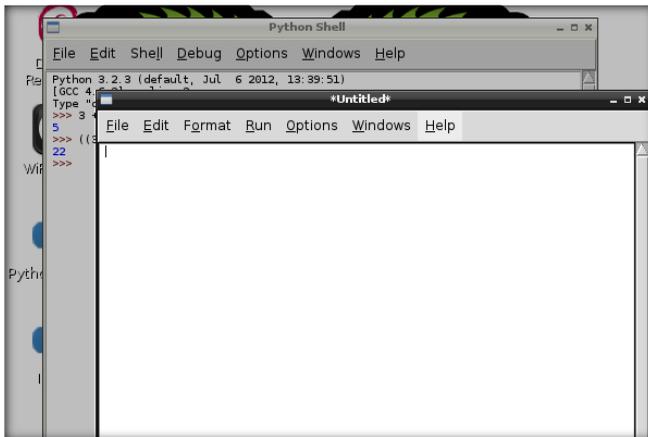
Adapun spesifikasi *Banana Pi M2U* adalah sebagai berikut:

1. *SoC – Allwinner R40 quad Core ARM Cortex A7 processor with ARM Mali-400MP2 GPU*
2. *System Memory – 2G DDR3 SDRAM*
3. *Storage – 8GB eMMC flash (16, 32 or 64GB as options), SATA interface, micro SD slot up to 256 GB*

4. *Connectivity* – 1x Gigabit Ethernet port, 802.11 b/g/n WiFi and Bluetooth 4.0 (AP6212 module)
5. *Video Output* – HDMI 1.4 port up to 1080p60, 4-lane MIPI DSI display connector
6. *Audio I/O* – HDMI, 3.5mm headphone jack, built-in microphone
7. *USB* – 2x USB 2.0 host ports, 1x micro USB OTG port
8. *Camera* – CSI camera connector
9. *Expansion* – 40-pin Raspberry Pi compatible header with GPIOs, I2C, SPI, UART, ID EEPROM, 5V, 3.3V, GND signals.
10. *Debugging* – 3-pin UART for serial console
11. *Misc* – Reset, power, and u-boot buttons; IR receiver
12. *Power Supply* – 5V via barrel connector, or 3.7V Lithium battery via battery connector on the back of the board. AXP221s PMIC
13. *Dimensions* – 92 x 60 mm

## 2.10 Python

*Python* adalah sebuah interpreter, yang memberikan kemudahan kepada penggunanya ketika membutuhkan sebuah interaksi. Ketika menjalankan *script* program pertama kali, *Python* akan melakukan proses pembuatan *byte code*, sehingga saat menjalankan program kedua kali dan seterusnya, eksekusi program akan berlangsung lebih cepat.



**Gambar 2. 15** IDE Python

Sejak kehadiran Raspberry Pi dan Banana Pi, yang mengusung bahasa *Python* sebagai bahasa yang disarankan, popularitas *Python* semakin melesat. Juga dukungan Google dalam implementasi komputasi awan menggunakan rujukan bahasa *Python* [16].

*Python* dimanfaatkan untuk pemrosesan *Graphic user interface* (GUI). GUI dibuat dengan tujuan menghubungkan data-data menjadi suatu grafik dan pemetaan sehingga mempermudah informasi dimengerti oleh operator dilapangan. Bentuk tampilan *Python* dapat dilihat pada Gambar 2.15.

## 2.11 Komunikasi Serial

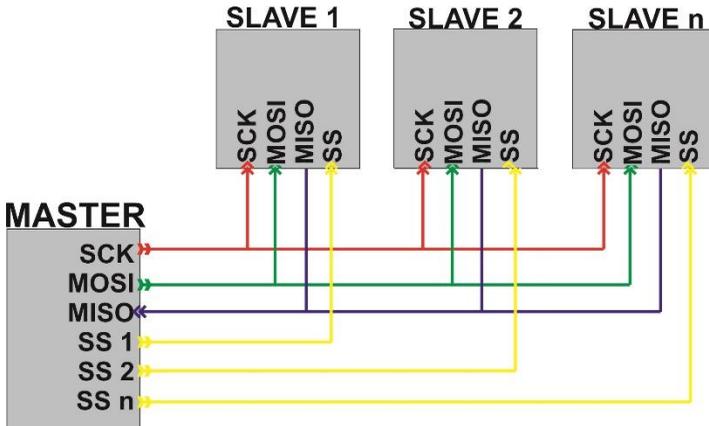
Komunikasi serial adalah salah satu metode komunikasi data dimana pengiriman data hanya satu bit yang dikirimkan melalui satu bit jalur saja. Komunikasi serial ada dua macam, yaitu *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan *clock* dan mengirimkan *clock* tersebut bersama-sama dengan data. *Serial Peripheral Interface* (SPI) merupakan salah satu contoh dari komunikasi serial *synchronous*. *Asynchronous serial* adalah komunikasi diterima kedua pihak menghasilkan *clock* data yang di transmisikan tanpa *clock*. Untuk menyamakan persepsi data yang dikirim dengan data yang diterima, maka frekuensi pembacaannya harus disamakan. Contoh komunikasi *asynchronous serial* adalah *Universal Asynchronous Receiver Transmitter* (UART) yang terdapat pada serial *port* COM komputer.

### 2.11.1 Serial Peripheral Interface (SPI)

SPI merupakan *full-duplex* (komunikasi dua arah). Dengan demikian, dalam situasi tertentu dapat mengirim dan menerima data secara bersamaan. Pada komunikasi SPI biasanya digunakan pin *Serial Clock* (SCK) untuk menyamakan frekuensi pembacaan, sedangkan untuk mengirim data digunakan pin *Master Output Slave Input* (MOSI) dan untuk menerima data digunakan pin *Master Input Slave Output* (MISO).

Dalam komunikasi SPI selalu setidaknya ada satu *master* dan dapat terdapat beberapa *slave*. *Slave* bertugas melayani *master* dalam bentuk pemberian informasi atau sebagai eksekusi *output*. Konfigurasi pin yang digunakan untuk memilih *slave* mana yang aktif

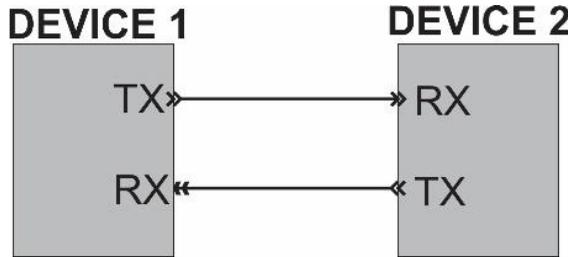
digunakan pin *Select Slave* (SS). Pada Gambar 2.11 merupakan bentuk konfigurasi pin pada komunikasi SPI.



Gambar 2. 16 Konfigurasi SIP

### 2.11.2 UART

Pada komunikasi serial *Universal Asynchronous Receiver Transmitter* (UART) biasa digunakan konfigurasi pin TX dan RX. Dimana pin TX adalah yang berfungsi sebagai pengirim (*Trasmmitter*) dan pin RX adalah sebagai penerima (*Receiver*). Untuk menggunakan komunikasi ini biasanya harus disamakan terlebih dahulu berapa kecepatan transmisi yang akan digunakan pada awal pemrograman. Kecepatan yang dimaksud biasa disebut *baut rate* dengan satuan bit per menit. Pada Gambar 2.12 merupakan bentuk konfigurasi dari komunikasi menggunakan TX dan RX.

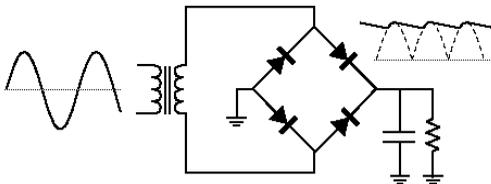


**Gambar 2. 17** Konfigurasi TX dan RX

## 2.12 Catu daya

*Power Supply* merupakan pemberi sumber daya bagi perangkat elektronika. Perangkat elektronika mestinya dicatu oleh *power supply* arus searah DC (*direct current*) yang stabil agar dapat dengan baik. Baterai atau *accu* adalah sumber catu daya DC yang paling baik. Namun untuk aplikasi yang membutuhkan catu daya lebih besar, sumber dari baterai tidak cukup. Sumber catu daya yang besar adalah sumber bolak-balik AC (*alternating current*) dari pembangkit tenaga listrik. Untuk itu diperlukan suatu perangkat catu daya yang dapat mengubah arus AC menjadi DC.

Perangkat yang dapat digunakan untuk menyesuaikan tegangan dengan perangkat elektronika adalah transformator. Sedangkan untuk menyearahkan gelombang AC dapat digunakan rangkaian *diode bridge*. Sedangkan untuk mengurangi *ripple* yang dihasilkan rangkaian dapat menggunakan filter kapasitor. Gambar 2.18 merupakan rangkaian catu daya penyearah gelombang penuh dengan filter kapasitor. Perhitungan  $V_{DC}$  dapat dilihat pada Persamaan 2.8. Sedangkan persamaan  $V_{ripple}$  dapat dilihat pada Persamaan 2.9.



**Gambar 2. 18** Rangkaian Rectifier

$$V_{DC} = \frac{2V_{max}}{\pi} \quad (2.8)$$

$$V_{Ripple} = \frac{I_{Beban}}{F \cdot C} = 1 + \frac{R_f}{R_1} \quad (2.9)$$

### 2.13 Database

Basis data atau *database* adalah kumpulan informasi yang tersimpan dalam komputer secara sistematis sehingga diperlukan suatu aplikasi untuk memperoleh informasi pada *database* tersebut. Biasanya database digunakan untuk menyimpan informasi yang bersifat kontiniu dan otomatis dalam penginputanya.

*Database* pada konsep yang sekarang ini lebih diterapkan pada penyimpanan awan (*cloud*). Tujuan menggunakan *cloud* pada konsep database. Karena sistem ini akan mempermudah pengaksesan data dari sistem yang lebih luas yang tidak terbatas pada satu ruanglingkup Komputer saja. Banyak sistem penyimpanan *cloud* yang dapat digunakan pada sekarang ini yaitu seperti, MySQL, Firebase, Thingspeak dan lain-lain.

Firebase merupakan *platform* pengembang aplikasi *mobile* dan aplikasi web. Firebase menyediakan produk penyimpanan *database* yang bersifat *realtime*. *Platform* firebase memiliki banyak *library* yang memungkinkan untuk mengaksesnya melalui layanan Android, IOS, Javacript, Java, dan masih banyak lagi.

Untuk menyingkronkan data yang disimpan, *firebase* menyediakan *Application Programing Interface* (API) terdiri dari beberapa fitur yang dapat dikonfigurasi dan disesuaikan dengan masing-masing kebutuhan. API pada firebase menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push* notifikasi dari server. Firebase merupakan *platform* dibawah perusahaan Google. Gambar 2.19 merupakan bentuk logo dari *firebase*.

### 2.14 Website

Website adalah suatu halaman web yang saling berhubungan yang umumnya berada pada server yang berisikan kumpulan

informasi yang disediakan secara perorangan, kelompok atau organisasi. Website yang ditempatkan pada server biasanya dapat diakses menggunakan jaringan internet dengan alamat internet atau dikenal sebagai URL (*Uniform Resource Locator*).

Sebuah halaman web merupakan file yang ditulis *sebagai plain text* yang diatur dan dikombinasikan sedemikian rupa dengan intruksi berbasis HTML atau XHTML. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditambahkan seperti layaknya sebuah halaman pada monitor.

HTML5 (*HyperText Markup Language 5*) adalah sebuah bahasa *markup* yang digunakan untuk menstrukturkan dan menampilkan isi *Word Wide Web*, yang merupakan inti dari internet. HTML5 merupakan seri kelima dari HTML (yang pertama kali diciptakan pada tahun 1990).

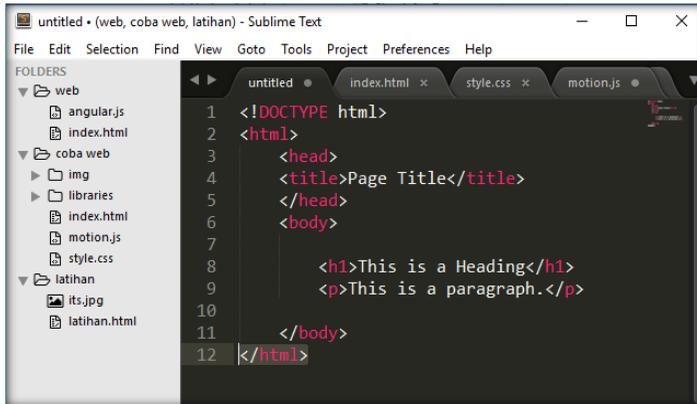
Pemrograman HTML sangatlah mudah dan tidak perlu menggunakan IDE khusus dalam pemrogramannya. Pemrograman HTML dapat dilakukan dengan menggunakan aplikasi *Notepad*, *Notepad++*, *Microsoft Word*, *Sublime* dan lain-lain. Dan untuk pengujiannya dapat dibuka langsung pada *browser* yang telah *support* dengan HTML5 seperti *Opera*, *Google Chrome*, *Internet Explore* dan lain-lain. Adapun tampilan HTML dapat dilihat pada Gambar 2.20, sedangkan untuk tampilannya pada web dapat dilihat pada Gambar 2.21.

Biasanya penggunaan HTML untuk membuat tampilan web statis. Kekurangan aplikasi HTML terletak pada keharusan *update* program terus-menerus untuk mengikuti perubahan yang terjadi. Namun pada pemrograman HTML yang ditonjolkan adalah sisi tampilan yang banyak mengandung grafis. Namun untuk mengatasi kekurangan pada web statis, biasanya digunakan web dinamis sebagai solusi.

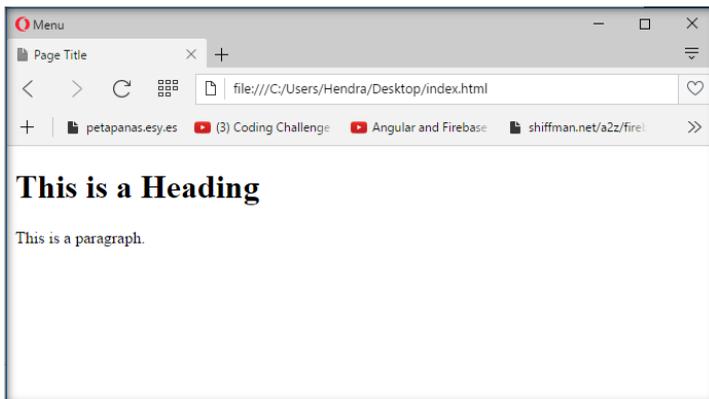
Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan. Perubahan informasi dalam halaman-halaman web dapat ditangani dengan perubahan data, bukan dengan perubahan program. Sebagai implementasinya aplikasi web dapat dikoneksikan dengan *database*. Dengan demikian perubahan informasi dapat dilakukan dengan penambahan atau perubahan data pada *database*.



**Gambar 2. 19** Logo Firebase



**Gambar 2. 20** Pemrograman HTML Menggunakan Sublime



**Gambar 2. 21** Tampilan Hasil Output pada Opera

## **BAB III**

### **PERANCANGAN SISTEM**

Pada bab ini akan dijelaskan perancangan sistem mulai dari gambaran umum sistem, perancangan perangkat keras (*hardware*), perancangan perangkat lunak (*software*). Didalam bab ini juga dijelaskan bagaimana komunikasi antar perangkat baik dari sesama *node*, *node* dengan server, server dengan *cloud*, hingga akuisisi data pada *cloud*.

*Hardware* atau perangkat keras merupakan bentuk fisik sistem secara keseluruhan yang berfungsi sebagai sensor, penerima data, pengirim data, atau pemroses data. Perancangan perangkat keras disesuaikan dengan kebutuhan dari sistem.

*Software* atau perangkat lunak berfungsi sebagai pengendali proses kerja seluruh sistem. Perangkat lunak memberi intruksi kepada perangkat keras untuk melakukan tugas tertentu agar sistem sistem dapat merespon masalah yang timbul pada lingkungan.

#### **3.1 Gambaran Umum Sistem**

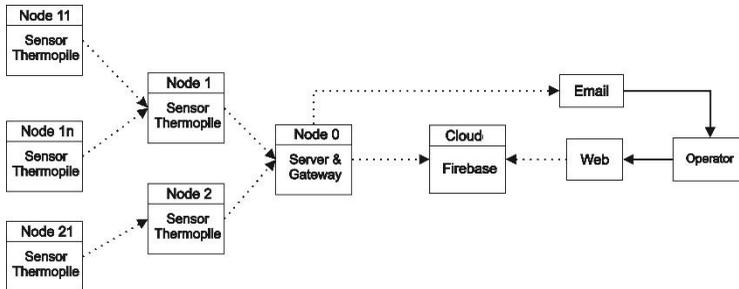
Pada tahapan ini adalah tahapan pengenalan terhadap sistem yang dirancang mulai dari masukan hingga keluaran dari sistem yang dirancang. Diagram blok terdiri dari beberapa blok fungsional yang menggambarkan keseluruhan sistem. Sistem yang akan dijelaskan dibagi menjadi tiga yaitu sistem perangkat secara keseluruhan sistem sensor dan sistem pada server.

Perancangan kelompok blok pada sistem untuk menghasilkan sistem *Wireless Sensor Network* dan sistem *Internet of Things* dapat dilihat pada Gambar 3.1. Adapun blok-blok yang digunakan adalah sebagai berikut:

1. *Node 0* sebagai *server* dan *gateway* berfungsi sebagai penyimpan data secara *offline* maupun *online* dan mengirim pesan kepada operator apabila data yang diterima menginformasikan terjadinya potensi kebakaran lahan gambut.
2. *Node n* berfungsi sebagai sensor pembaca suhu sekitar *node* yang kemudian dikirim menuju *node* atas agar menuju *server*.

3. *Cloud* sebagai penyimpanan secara *online*, selain menghindari *losing* data pada penyimpanan *offline*, juga agar data dapat diakses dari jarak jauh melalui internet.
4. Operator adalah objek tujuan dari email yang dikirim oleh server, serta objek yang mengakses halaman web untuk memantau sistem secara *online*.

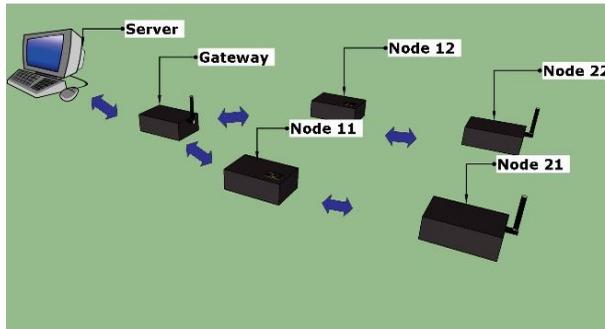
Diagram blok dari sistem ini ditunjukkan pada Gambar 3.1. Secara garis besar dibagi menjadi empat sub sistem yaitu akuisisi data sensor pada *node*, *Wireless Sensor Network*, *Internet of Things*, dan akuisisi data pada *coloude* agar dapat ditampilkan pada halaman web. Bagian *node* dikendalikan menggunakan Arduino nano. Sedangkan untuk mengupload data dari *gateway* ke *cloud* menggunakan *Banana Pi*. Untuk mengirimkan data mengkomunikasikan data antar sensor hingga sampai ke *gateway* menggunakan komunikasi *Wireless Sensor Network*. *Node* akan mengirimkan data *array* pembacaan suhu sekitar *node*.



**Gambar 3. 1** Diagram Blok Sistem

Selain itu, tugas akhir ini juga menerapkan teknologi *Wireless Sensor Network* (WSN) dan *Internet of Things* (IoT) yang digunakan untuk mengkomunikasikan data antar sensor atau *node* agar data dapat diterima oleh server dan kemudian di simpan pada *cloud* lalu dapat diakses melalui internet. Dengan teknologi WSN dan IoT diharapkan sensor dapat disebar pada area yang luas untuk memberikan informasi tentang potensi kebakaran lahan gambut dengan *range* area yang luas pula. Pada Gambar 3.2 merupakan

bentuk ilustrasi dari sistem yang akan dibuat. Seperti yang dilihat pada gambar tersebut, sensor akan diletakkan saling berjauhan dengan yang lain.



**Gambar 3. 2** Ilustrasi Sistem

Pada perancangannya sensor ini direncanakan untuk diletakkan pada hamparan lahan gambut yang luas untuk melakukan monitoring potensi terjadinya kebakaran. Peletakan *platform* ini bersifat statis, namun dapat dipindahkan dengan ketentuan masih didalam *range* komunikasi dari jaringan sensor. Sensor ini juga dapat *upgrade* dengan menambah sensor yang beroperasi untuk memperluas area pengawasan. Sistem ini diharapkan dapat beroperasi secara mandiri, sehingga tidak diperlukan operator untuk mengawasi sistem secara terus menerus. Saat sistem ini dioperasikan, maka masing-masing sensor akan membaca suhu disekitarnya, kemudian akan mengirimkan data menuju server melalui jaringan *wireless sensor network*. Kemudian *server* akan mengakuisisi data lalu menyimpan data secara *offline* maupun secara *online* kedalam *database*. Jika terjadi potensi kebakaran, maka sistem secara otomatis akan mengirimkan email pada list sistem yang telah ditentukan.

### 3.2 Perancangan Sensor *Node*

Pada perancangan sensor *node* bertujuan untuk melakukan pengumpulan data panas yang berada pada lahan gambut yang luas. Evaluasi yang akan dilakukan sistem ini adalah pembacaan suhu lingkungan, lokasi koordinat *node*, serta komunikasi antar *node*. Suhu

yang telah dibaca kemudian diteruskan ke *node* yang lebih tinggi hingga guna pemrosesan lebih lanjut pada server.

Pembahasan pada subbab ini akan membahas tentang perancangan *hardware* hingga perancangan *software* dari sistem *node*. Tujuan perancangan adalah untuk menghasilkan suatu keluaran sistem sesuai spesifikasi yang diinginkan.

### 3.2.1 Sistem Sensor *Node*

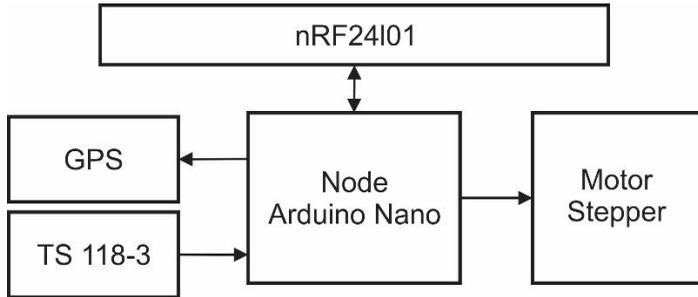
Penyusunan blok-blok fungsional pada sistem untuk menghasilkan sistem sensor *node* dapat direpresentasikan pada Gambar 3.3. Adapun blok fungsional yang digunakan adalah sebagai berikut:

1. Menggunakan mikrokontroler berupa Arduino Nano sebagai *node* berfungsi sebagai mengatur seluruh proses kerja sensor.
2. Menggunakan radio frekuensi dengan tipe nRF24101 berfungsi sebagai sarana komunikasi *wireless* dengan *node* lain atau dengan server. Informasi yang dikomunikasikan berupa data pembacaan sensor masing-masing *node* yang ditujukan kepada server.
3. GPS Neo N8M sebagai penentu lokasi dari sensor *node* yang dipasang.
4. Sensor inframerah TS 118-3 berfungsi sebagai sensor suhu untuk menangkap suhu sekitar *node*.
5. Motor *Stepper* berfungsi untuk memutar sensor suhu agar dapat membaca suhu dengan area yang lebih luas.

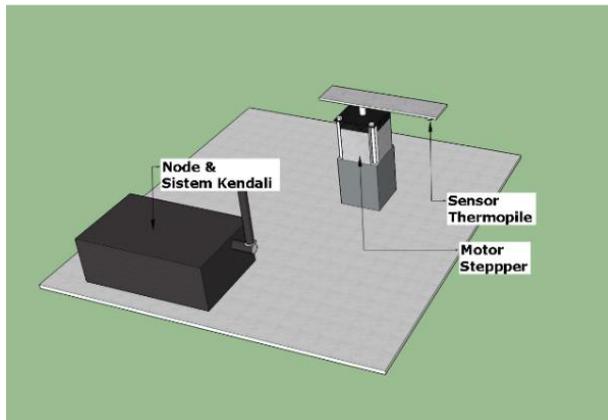
Pembacaan satu bit akan dibuat dapat membaca suhu berupa luasan wilayah sehingga keluarannya diharapkan dapat membentuk peta panas area. Pada Gambar 3.4 merupakan ilustrasi dari bentuk sistem sensor yang dibuat. Seperti yang terlihat pada gambar tersebut, sensor akan berada di sumbu dari sebuah motor *stepper* yang akan diputar. Motor *stepper* akan diputar 360° secara penuh agar sensor *thermopile* dapat membaca suhu dari keliling sensor.

Pada pengimplementasiannya nanti akan dibuat tiga buah sensor *node dummy*. Tujuan dari penggunaan sensor *node dummy* adalah untuk menguji apakah sistem telah dapat bekerja secara benar pada konsep *Wireless Sensor Network*. Sistem kerja dari sensor *node*

*dummy* adalah memberikan data secara acak, dan meneruskan data sensor dari sensor yang dinaunginya.



**Gambar 3. 3** Diagram Blok Sistem



**Gambar 3. 4** Ilustrasi Sensor

### 3.2.2 Hardware Sensor Node

Mikrokontroler yang digunakan untuk mengatur sensor *node* adalah Arduino Nano. Pemilihan mikrokontroler berdasarkan kebutuhan *input output* yang kecil sehingga Arduino nano dirasa cukup untuk memenuhi kebutuhan sistem.

Seluruh unit masukan dan keluaran yang digunakan oleh sensor *node* dapat dilihat pada Tabel 3.1. Pin Arduino yang tidak digunakan akan dibiarkan dalam kondisi *reset (reset state)* atau tidak berfungsi.

**Tabel 3. 1** Konfigurasi Penggunaan Pin I/O Arduino Sensor *Node*

No	Pin Arduino	Komponen	Fungsi
1	A0	TS188-3	Pembacaan Thermopile
2	A1	TS188-3	Pembacaan Thermistor
3	A3	AD4988	<i>Step</i>
4	A4	AD4988	<i>Direction</i>
5	D3	GPS NEO M8N	RX
6	D4	GPS NEO M8N	TX
7	D9	nRF24L01	CE
8	D10	nRF24L01	CNS
11	D11	nRF24L01	MISO
12	D12	nRF24L01	MOSI
13	D13	nRF24L01	SCK

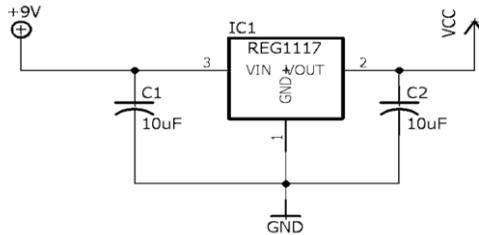
### 3.2.2.1 Perancangan Catu Daya *Node*

Catu daya merupakan hal yang pertama kali harus diperhatikan, catu daya yang baik akan menghasilkan kinerja sistem yang baik pula. Catu daya harus mampu mendistribusikan daya sesuai dengan karakteristiknya.

Catu daya bersumber dari baterai 12 V dan dua buah baterai 9 V. Baterai 12V akan disambungkan langsung melalui pin Vin. Pada pin Vin akan Arduino Nano. Tegangan akan diubah menjadi 5 V dan 3,3 V. Tegangan 5 V digunakan untuk suplai catu daya untuk GPS Neo M8N, untuk 3,3 V digunakan untuk catudaya radio komunikasi nRF24L01, 12 V digunakan untuk catu daya Arduino serta motor *stepper*, sedangkan dua buah baterai 9 V digunakan untuk catu daya *amplifier* sensor *Thermopile*.

Namun dikarenakan keluaran tegangan 3,3 V Arduino Nano tidak mampu untuk mengoperasikan radio frekuensi nRF24L01+(antena luar), sehingga digunakan catudaya tambahan

berupa rangkaian regulator tegangan 3,3 V menggunakan IC 1117 seperti pada Gambar 3.5.



**Gambar 3. 5** Rangkaian Catu Daya 3,3V nRF24L01+

### 3.2.2.2 Perancangan Sensor Suhu

Perancangan sensor suhu pada sistem ini menggunakan *Infrared sensor* TS188-3 Thermopile. Pada tipe TS188-3 memiliki suhu kerja  $-20^{\circ}\text{C}$  sampai dengan  $100^{\circ}\text{C}$  dengan karakteristik output  $-0,6\text{mV}$  sampai dengan  $4,4\text{mV}$ , dengan resistansi sensor  $1000\text{ ohm}$  pada suhu  $0^{\circ}\text{C}$ . TS188-3 memiliki empat buah pin yang mana pin 1 merupakan TP+, pin 2 merupakan Ni-RTD, pin 3 merupakan TP-, dan pin 4 merupakan *ground*. Gambar 3.6 merupakan bentuk dari thermopile TS188-3 [18].

Parameter	Typical	Condition
Package	TO-18	
Absorber Area	$0.7 \times 0.7\text{ mm}^2$	
Resistance of Thermopile	$43.8\text{ k}\Omega$	$+25^{\circ}\text{C}$
TC of Resistance	$-0.06 \pm 0.04\text{ \%}/\text{K}$	$+25^{\circ}\text{C} \rightarrow +75^{\circ}\text{C}$
Thermopile Voltage	$4.4 \pm 1.1\text{ mV}$	$+25^{\circ}\text{C}$ , BB $+100^{\circ}\text{C}$ , DC totally filled field of view
TC of sensitivity	$-0.45 \pm 0.08\text{ \%}/\text{K}$	$+25^{\circ}\text{C} \rightarrow +75^{\circ}\text{C}$
Noise Equivalent Voltage	$30\text{ nV}/\text{Hz}^{1/2}$	$+25^{\circ}\text{C}$
Rise Time	$20 \pm 5\text{ ms}$	$t_{90}$
Field of View	$120^{\circ}$	
Filter	$8\text{--}14\text{ }\mu\text{m}$	band pass
Operation Temperature	$-20 \dots +85^{\circ}\text{C}$	
Operation Temperature	$-20 \dots +100^{\circ}\text{C}$	non permanent
Ambient Temperature Sensor	Ni-RTD	
Resistance	$1000 \pm 4\text{ }\Omega$	$0^{\circ}\text{C}$
TC of Resistance	$6178 \pm 150\text{ ppm}/\text{K}$	$0^{\circ}\text{C} \rightarrow +100^{\circ}\text{C}$
Connections		
Pin 1	TP +	
Pin 2	Ni-RTD	
Pin 3	TP -	
Pin 4	GND	

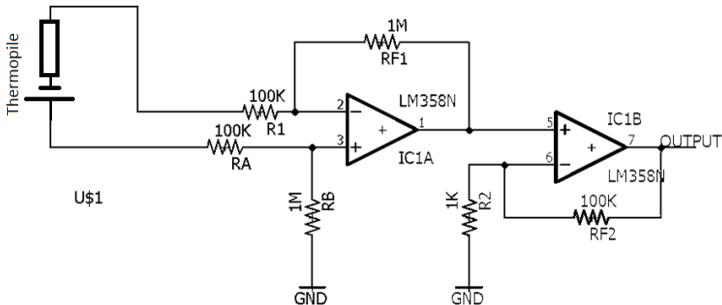


**Gambar 3. 6** Sensor Suhu TS118-3

Pemilihan sensor thermopile dikarenakan untuk membaca suhu permukaan yang luas diperlukan sensor suhu yang bertipe contactless agar dapat membaca suhu dari jarak jauh dan sensor dapat diarahkan sudut pembacaanya.

Thermopile berfungsi sebagai pembaca suhu yang bersifat *contactless* adalah membaca suhu lingkungan *node* berada. Suhu yang dapat dibaca sensor TS188-3 adalah  $-20^{\circ}\text{C}$  hingga  $100^{\circ}\text{C}$  dengan tegangan output  $-0.6\text{ mV}$  hingga  $4.4\text{ mV}$ .

Untuk mengkonversi data analog dengan *range*  $5\text{ mV}$  ( $-0.6\text{ mV}$  hingga  $4.4\text{ mV}$ ) ke *range* pembacaan pin ADC pada Arduino Nano dapat dilakukan dengan menggunakan *Operational Amplifier* bertingkat dengan *gain* penguatan sebesar  $\pm 1000$  kali penguatan. Adapun rangkaian yang digunakan seperti ditunjukkan pada Gambar 3.7. Dari rangkaian tersebut, maka dapat dihitung penguatan rangkaian dengan menggunakan Persamaan (2.4) dan persamaan (2.6). Output dapat langsung dihubungkan pada kaki Analog 0 pada Arduino Nano.



**Gambar 3. 7** Rangkaian *Operational Amplifier* Thermopile

Diketahui:

$$V_{\text{sensor}} = -0.4 - 4.6\text{ mV (Range = 5mV)}$$

$$V_{\text{in micro}} = 0 - 5\text{ V (Range =5V)}$$

$$R_1 = R_A = 100\text{ KOhm}$$

$$R_{F1} = R_B = 1\text{ MOhm}$$

$$R_2 = 1\text{ KOhm}$$

$$R_{F2} = 100\text{ KOhm}$$

Sehingga:

Penguatan yang dibutuhkan =  $\pm 1000$

Perhitungan:

Opamp 1 merupakan rangkaian penguat deferensial. Dimana apabila  $R1 = RA$  dan  $RF1 = RB$ . Maka persamaan yang digunakan adalah sebagai berikut.

Maka:

$$A_1 = \frac{RF_1}{R_1} = \frac{10^6}{10^5} = 10x$$

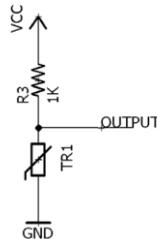
Opamp 2 merupakan rangkaian penguat non inverting. Persamaan yang digunakan adalah sebagai berikut.

Maka

$$A_2 = 1 + \frac{RF_2}{R_2} = 1 + \frac{10^5}{10^3} = 101x$$

Sehingga penguatan keseluruhan rangkaian dapat dihitung:

$$A = A_1 * A_2 = 10 * 101 = \mathbf{1010x}$$



**Gambar 3. 8** Rangkaian akuisisi *Thermistor*

Selain *Thermopile*, TS188-3 juga memiliki *output* berupa *thermistor* yang berfungsi untuk mengukur suhu dari sensor itu sendiri. Pengukuran suhu pada sensor diperlukan dikarenakan keluaran suhu *thermopile* sangat tergantung dengan suhu sensor. Keluaran dari *thermistor* berupa resistansi dimana resistansi bernilai

10 K *Ohm* pada suhu 25°C serta akan meningkat jika suhu semakin rendah dan semakin rendah nilainya apabila suhu semakin tinggi.

Pada Gambar 3.8 merupakan perancangan akuisisi data pada output thermistor menggunakan pembagi tegangan. Hal tersebut dimaksudkan untuk merubah besaran resistansi yang dihasilkan thermistor menjadi tegangan. Perubahan besaran output diperlukan Karena masukan analog pada Arduino berupa masukan tegangan antara 0 V hingga 5 V.

Persamaan (3.1) merupakan perhitungan akuisisi data dari *thermistor*. Tegangan *output* akan semakin besar bila suhu semakin rendah dan tegangan *output* akan semakin rendah apabila suhu sensor semakin tinggi. *Output* dari rangkaian ini dihubungkan pada kaki A0 pada Arduino.

$$V_o = \frac{TR}{TR + R} 5V \quad (3.1)$$

### 3.2.2.3 Perancangan *Driver Motor Stepper*

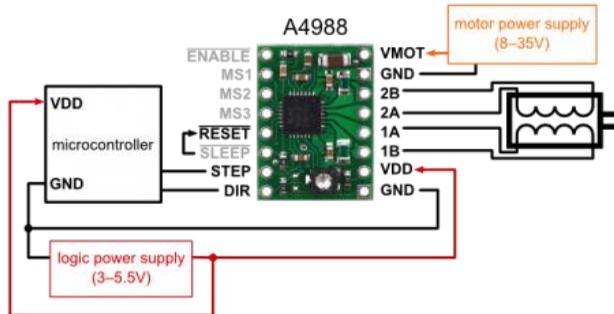
Perancangan *driver motor stepper* menggunakan rangkaian modul A4988. Pada modul A4988 memiliki 5 buah kaki yang dihubungkan ke Arduino yang digunakan untuk mengendalikan motor *stepper*, yaitu *step*, *direction*, VCC 12 V, VCC 5 V, *Ground*. Lalu 4 buah kaki 1A, 1B, 2A, dan 2B dihubungkan ke kaki motor *stepper*.

Pada Gambar 3.9 merupakan bentuk rangkaian yang akan digunakan untuk mengontrol motor *stepper* menggunakan Arduino. Terlihat terdapat dua buah catu daya yang digunakan untuk mengoperasikan sistem ini. Catu daya yang digunakan yaitu 8 V sampai 35 V digunakan untuk suplay motor *stepper*, sedangkan 5 V digunakan untuk suplay IC kontrolernya.

Penggunaan A4988 dapat dilakukan dengan sebagai berikut:

1. Memberi pulsa pada kaki *step*, setiap pulsa menghasilkan pergerakan satu step putaran pada motor *stepper*.
2. Memberi logika high pada kaki *direction* untuk membuat arah putaran motor *clockwise*. Sedangkan logika *low* untuk memutar motor *anticlockwise*.

3. Kaki *sleep* dan *reset* di-*short*.
4. Kaki 1A, 1B, 2A, dan 2B dihubungkan dengan *stepper 4 wire*.
5. Kaki *power*  $V_{MOT}$  diberi tegangan untuk *suplay* motor berkisar tegangan 8V hingga 35V.
6. Senangkan kaki *power*  $V_{DD}$  yang berfungsi sebagai *suplay* rangkaian driver diberi tegangan *suplay* sebesar 5V.



**Gambar 3. 9** Bentuk Modul A4988 dan *Wiring* Rangkaian

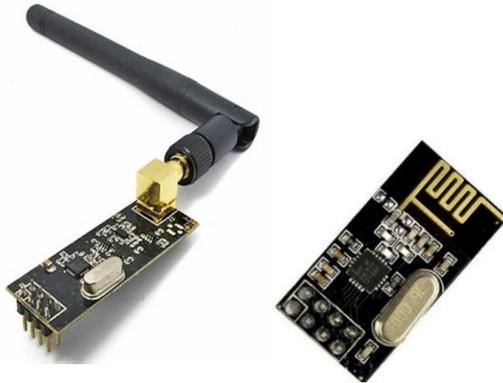
### 3.2.2.4 Perancangan Komunikasi Antar *Node*

Modul yang digunakan untuk berkomunikasi adalah nRF24L01. Modul Radio Frekuensi nRF24L01 adalah sebuah modul komunikasi jarak jauh yang memanfaatkan pita gelombang RF 2.4 GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan antar muka SPI untuk berkomunikasi. Tegangan modul ini adalah 5V DC.

Modul nRF24L01 seperti Gambar 3.10 memiliki *baseband logic Enhanced ShockBurst* dengan *hardware* protokol accelerator yang support “*high-speed SPI interface for the application controller*”. nRF24L01 memiliki *true UTP solution*, yang memungkinkan daya baterai bertahan hingga berbulan-bulan hingga bertahun-tahun. Adapun spesifikasi dari nRF24L01.

1. Beroperasi pada pita frekuensi 2.4 GHz
2. Data rate hingga 2Mbps
3. *Ultra low power*
4. Penanganan paket data otomatis.

## 5. Penanganan paket otomatis



**Gambar 3. 10** nRF24L01[19]

Modul tersebut merupakan perangkat yang memiliki metode komunikasi SPI. Adapun konfigurasi pin yang digunakan ke Arduino yaitu VCC ke 3V, GND ke GND, CSN ke pin D10, CE ke pin D9, MOSI ke pin 11, SCK ke pin 13, dan MISO ke pin 12. Bentuk rangkaian yang digunakan dapat dilihat pada Gambar 3.12.

### 3.2.2.5 Perancangan GPS

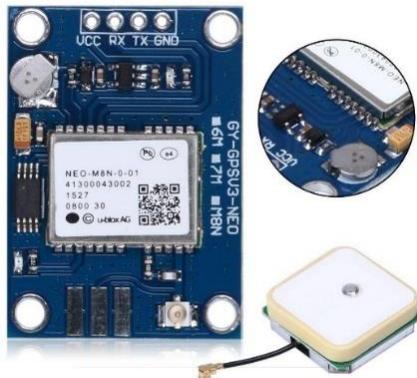
Untuk mengambil data GPS digunakan modul GPS NEO M8N. Modul NEO-M8N seperti pada Gambar 3.11 adalah produk dari *u-blox* yang telah menerapkan teknologi GNSS (*Global Navigation Satellite System*) yang digunakan untuk menentukan posisi objek menggunakan referensi satelit. Satelit yang dapat dikoneksikan dengan banyak jenis satelit. Jenis satelit yang dapat diakses diantaranya GPS L1C/A, GLONASS L1OF dan BeiDou B1.

NEO-M8N melacak dan menerima sinyal GPP L1C/A pada frekuensi 1575.42 MHz dengan durasi pembacaan data kurang dari 1 detik, namun untuk optimumnya dibutuhkan waktu  $\pm 2$  detik. Pada modul ini menggunakan komunikasi serial untuk akuisisi data.

atanya. Berikut spesifikasi modul NEO-M8N:

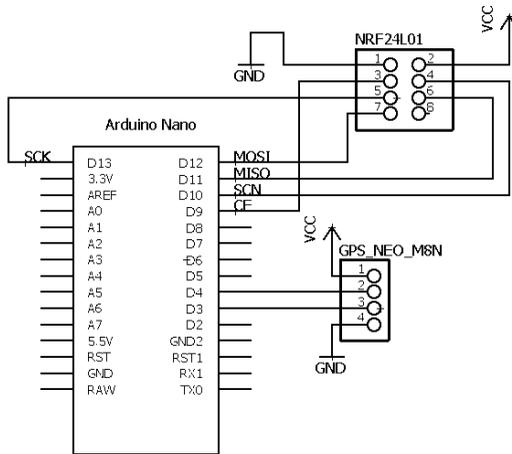
1. 72 –Chennel *u-blox M8 engine*.

2. Kompatibel pada jaringan GPS L1C/A, BeiDou B1, GLONASS L1OF, Galileo E1B/C<sup>2</sup>.
3. Sensitifitas tracking dan navigasi -167dBm, pada saat pembacaan terekstrim bisa hanya -156dBm.
4. Frekuensi *update* data  $\pm 5$  Hz.
5. Akurasi *velocity* 0,05 m/s.
6. Akurasi pada sumbu horizontal 2,5m untuk autonomus dan 2m untuk SBAS.
7. Limit operasional pada *altitude* 50.000m dan *velocity* 500 m/s.
8. Komunikasi data serial UART.
9. Terdapat konektor antena.
10. Tegangan *suplay* 3,3V-5V.



**Gambar 3. 11** NEO N8M[20]

Modul tersebut merupakan modul dengan komunikasi serial dengan *baut rate* 9800. Nilai yang dihasilkan dari pembacaan lokasi dikirimkan secara terus menerus sehingga koordinat lokasi selalu update. Konfigurasi pin yang digunakan yaitu TX ke D4, dan RX ke pin D3. Adapun rangkaian yang digunakan seperti pada Gambar 3.12.

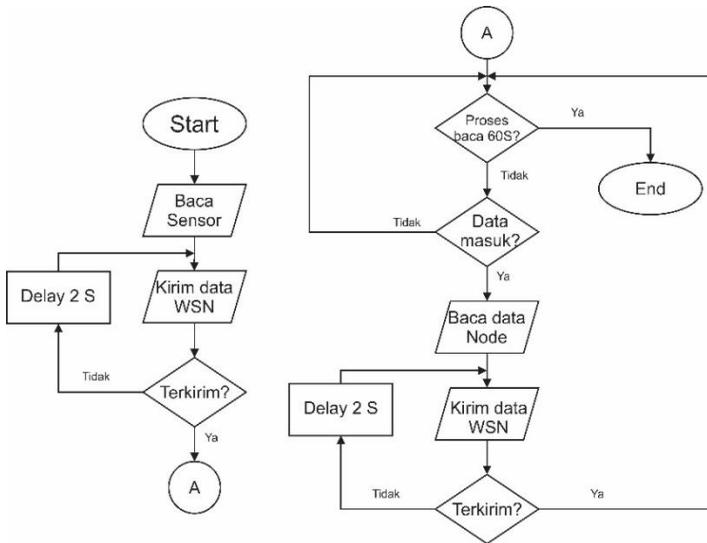


**Gambar 3. 12** Komunikasi nRF24L01 dan GPS Neo M8N

### 3.2.3 Software Sensor Node

Pada Gambar 3.13 merupakan diagram alir dari sistem sensor *node* yang dibuat. Adapun algoritma untuk menjalankan sistem ini adalah sebagai berikut:

1. *Node* membaca suhu permukaan lahan menggunakan sensor *thermopile*, data yang didapat berupa data *array* pembacaan suhu sekeliling sensor.
2. *Node* mengirimkan data ke server atau ke *node* atasnya untuk diteruskan ke server. Apabila tidak terkirim akan diulangi pengiriman dengan didahului delay selama 2 detik
3. Sistem akan menunggu 60 detik sebelum sistem mengakhiri program.
4. Selama menunggu waktu 60 detik, sensor *node* melakukan pembacaan data dari sensor *node* yang lain untuk kemudian dikirimkan ke *node* yang lebih atau diteruskan ke server.
5. Setelah waktu mencapai 60 detik maka sistem akan diakhiri.



**Gambar 3. 13** Diagram Blok pada Sensor Node

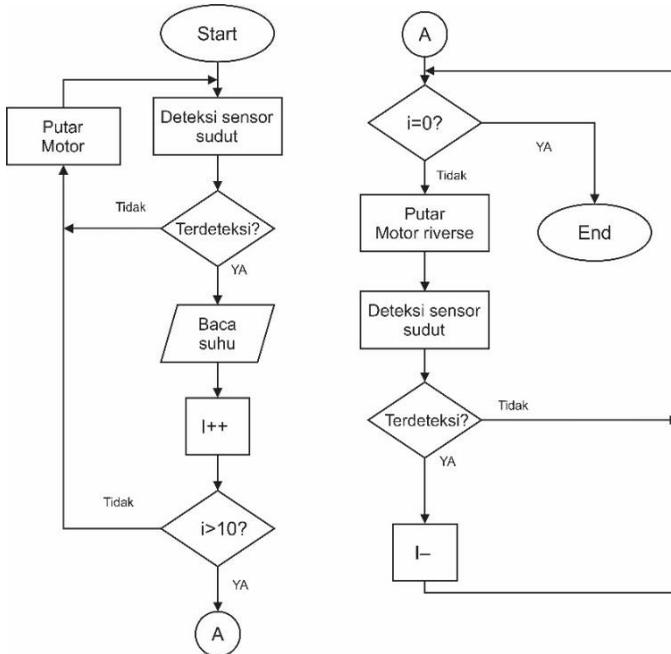
Sistem ini akan terintegrasi dengan sistem dengan mendefinisikan alamat *node* yang digunakan serta *channel* yang digunakan. Karena sistem yang digunakan adalah *Wireless Sensor Network* dengan topologi *tree* sehingga perlu untuk mengetahui alamat yang hendak dituju. Pendefinisian alamat dan algoritma alamat yang dituju dideklarasikan di awal pemrograman dari sistem.

*Wireless Sensor Network* akan diterapkan pada Arduino yang berfungsi sebagai prosesor dari sensor node. Penggunaan Arduino dapat mempermudah perancangan sistem software dengan memanfaatkan *library* yang tersedia. *Library* yang digunakan pada sistem untuk menerapkan *Wireless Sensor Network* adalah *NRF24Network.h* [21].

Pada pembacaan suhu sekitar sensor *node* menggunakan *thermopile* dan dibantu dengan gerakan mekanik dari motor *stepper*. Sedangkan untuk menentukan sudut pembacaan menggunakan bantuan sensor optokopler dan bantuan cakram yang digunakan sebagai penunjuk sudut pembacaan. Adapun algoritma yang digunakan adalah seperti Gambar 3.14.

Algoritma yang digunakan untuk pembacaan suhu lingkungan sekitar dari *node* sensor adalah sebagai berikut:

1. Sensor akan mendeteksi sudut pembacaan.
2. Apabila berum berada pada sudut yang tepat, maka motor *stepper* akan terus bergerak hingga menemukan sudut pembacaan.
3. Setelah menemukan sudut pembacaan maka sensor suhu akan membaca suhu pada titik itu.
4. Proses ini diulang sebanyak berapa kali pembacaan akan dilakukan dan apabila sudut pembacaan telah mencapai 360°.
5. Apabila telah membaca sensor maka motor akan kembali ketitik semula.



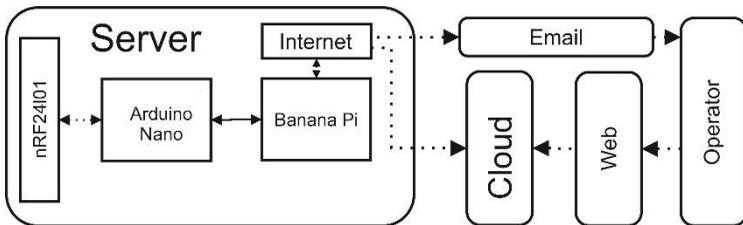
**Gambar 3. 14** Flowchard Proses Pembacaan Suhu

### 3.3 Perancangan Server

Server berfungsi sebagai pengumpul data dari *node-node*, yang kemudian data tersebut disimpan dan diproses. Proses yang dilakukan saat data berada di server adalah proses identifikasi dan penyimpanan data secara *online* maupun *offline*. Pada saat pemrosesan diperlukan beberapa perangkat utama yaitu Banana Pi, Arduino Nano, dan nRF24L01.

#### 3.3.1 Sistem Server

Blok-blok fungsional yang disusun untuk membuat sistem server yang diperlihatkan pada Gambar 3.15. berikut merupakan blok fungsional yang digunakan untuk membentuk sistem server:



**Gambar 3. 15** Diagram Blok Server

1. nRF24L01 berfungsi sebagai *receiver* data dari node yang dinaungi server.
2. Arduino nano berfungsi sebagai perantara atau *interface* antara nRF24L01 dengan *Banana Pi*.
3. Bananapi berfungsi sebagai pengatur proses pada server, baik dari penerimaan data sensor, penyimpanan data *offline*, penyimpanan data secara *online* hingga pemberian peringatan terhadap operator dengan cara mengirimkan *Email* kepada operator.
4. *Cloud* digunakan untuk penyimpanan secara *online* yang kemudian data yang disimpan pada sistem ini dapat dibaca pada jarak jauh dari Web yang telah ditunjuk.
5. Web berfungsi sebagai interface antara sistem dengan *user* atau operator. Web mengambil data yang tersimpan pada cloud yang

kemudian data di *interface* kan dengan user yang ingin memantau sistem dilapangan.

### 3.3.2 Hardware Server

Perancangan *Hardware* pada server didasarkan pada kebutuhan beberapa perangkat yang dibutuhkan oleh sistem. Diperlukan Arduino Nano untuk pengoperasian radio frekuensi nRF24L01+. Lalu diperlukan pula Banana Pi yang digunakan sebagai server, bertugas menyimpan data, mengolah data, dan mengirimkan data dari server ke *cloud*.

Seluruh unit masukan dan keluaran yang digunakan oleh Arduino dapat dilihat pada Tabel 3.2. Pin Arduino yang tidak digunakan akan dibiarkan dalam kondisi reset (*reset state*) atau tidak berfungsi.

**Tabel 3. 2** Konfigurasi Penggunaan Pin I/O Arduino Server

No	Pin Arduino	Komponen	Fungsi
1	USB	Banana Pi	Komunikasi Serial
2	D9	nRF24L01	CE
3	D10	nRF24L01	CNS
4	D11	nRF24L01	MISO
5	D12	nRF24L01	MOSI
6	D13	nRF24L01	SCK

#### 3.3.2.1 Perancangan Catu Daya Server

Catu daya merupakan suatu hal yang sangat penting dalam perancangan sistem server, dikarenakan dengan adanya catudaya yang baik, maka akan menghasilkan menghasilkan *output* yang optimal. Perancangan catu daya didasarkan dengan kebutuhan masing-masing komponen. Adapun kebutuhan dan catu daya yang digunakan masing-masing komponen sebagai berikut:

1. *Buck converter* membutuhkan suplai dari adaptor dengan masukan 12 V sehingga dapat mengeluarkan *output* tegangan 5V.
2. Banana Pi membutuhkan catu daya berupa tegangan 5 V dengan arus 2 A yang dihubungkan dengan Mikro USB. Untuk

memenuhi kebutuhan suplay menggunakan rangkaian *buck converter* dengan keluaran 5 V.

3. Arduino Nano membutuhkan suplay 7 V sampai 20 V yang dihubungkan pada *port* Vin. Untuk memenuhi kebutuhan suplai digunakan Adaptor 12 V yang sama dengan masukan *buck converter*.
4. nRF24L01+ membutuhkan catu daya berupa tegangan 3,3 V dengan konsumsi daya 20 dBm. Untuk memenuhi kebutuhan suplay digunakan catu daya dari regulator AMS1117 3,3 V yang inпитan regulator berasal dari baterai yang sama dengan baterai yang digunakan oleh Arduino. Gambar 3.5 rangkaian catu daya untuk nRF24L01+.

### 3.3.2.2 Komunikasi pada Server

Komunikasi juga salah satu komponen yang harus diperhatikan pada sistem server, dengan berbagai macam perangkat yang digunakan sehingga untuk menghubungkan semua perangkat diperlukan komunikasi yang sesuai dengan masing-masing perangkat. Berikut merupakan hubungan serta komunikasi yang digunakan antar modul pada server:

1. Komunikasi yang digunkakan antara Arduino dengan nRF24L01 pada server sama dengan komunikasi yang dilakukan antara Arduino dengan nRF24L01+ pada *node*, untuk bentuk rangkaian dapat dilihat pada Gambar 3.9 bagian *wiring* nRF24L01+.
2. Komunikasi antara Arduino dengan Banana Pi digunakan komunikasi UART dengan frekuensi sampling 9600. Adanya TTL CH340G sebagai antarmuka UART pada Arduino Nano membuat komunikasi antara Arduino dan Banana Pi dapat dilakukan dengan menggunakan USB. Hal tersebut mempermudah komunikasi antara Arduino dengan Banana Pi.
3. Komunikasi antara Arduino dengan cloud menggunakan jaringan internet. Adanya fitur WiFi pada Banana Pi membuat mempermudah dalam penyediaan jaringan internet. WiFi yang tersedia dapat dikoneksikan dengan perangkat penyedia jaringan internet.

### 3.3.2.3 Sistem Operasi

*Banana Pi* sebagai server memerlukan sistem operasi agar dapat digunakan. Sistem operasi yang digunakan pada sistem ini adalah Ubuntu MATE yang dapat menjalankan pemrograman python dan *browser* yang mendukung HTML5. Dengan adanya spesifikasi ini dapat mempermudah dalam proses akuisisi data serta pemrosesan lainnya. Gambar 3.16 merupakan tampilan dari sistem operasi Ubuntu MATE.



**Gambar 3. 16** Ubuntu MATE

#### 3.3.1 Perangkat Pendukung

*Banana Pi* selain sebagai server juga berfungsi sebagai interface dengan operator, sehingga dibutuhkan *display* serta inputan agar dapat mengendalikan sistem.

Sebagai *display* digunakan monitor dengan koneksi menggunakan HDMI serta sebagai inputan menggunakan *mouse* serta *keyboard*. Perangkat tersebut sangat menunjang dalam sistem pengawasan dan kendali. Pada Gambar 3.17 merupakan tampilan dari perangkat tambahan yang digunakan.



**Gambar 3. 17** Tampilan Perangkat Tambahan

### **3.3.3 Software Server**

Server merupakan perangkat yang akan digunakan sebagai kontrol data-data yang dikirim oleh *node* melalui jaringan WSN. Perangkat ini akan mengirimkan data dari sensor *node* menuju *cloud* dan mengirimkan pesan berupa *email* pada operator bila terjadi gangguan. Diagram alir sistem server dapat dilihat pada Gambar 3.18.

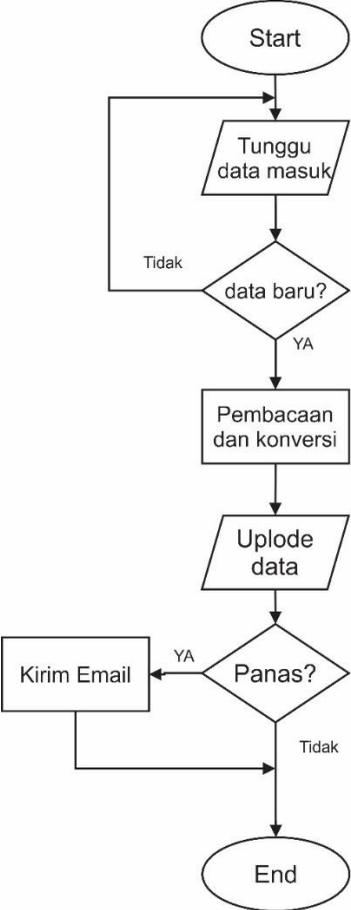
Algoritma yang digunakan pada sistem server adalah sebagai berikut:

1. Server akan menunggu data masuk
2. Bila terdapat data masuk, maka akan dilakukan prosesan data berupa mempaket data kemudian menyimpan secara offline serta secara online.
3. Apabila terdeteksi suhu pembacaan tidak normal, maka sistem akan mengirimkan pesan berupa email kepada operator.
4. Server kembali ke mode menunggu data masuk.

Pada perancangan sistem pada server perlu diperhatikan adalah jaringan internet yang digunakan untuk mengirimkan data ke sistem cloud serta mengirim email pada saat terdeteksi terjadinya panas pada titik tertentu.

Sistem yang terdiri dari beberapa perangkat utama dan beberapa perangkat pendukung selain digunakan untuk proses diatas, juga digunakan sebagai interface antara sistem dengan operator. Untuk interface yang dirancang akan ditampilkan pada halaman

website sehingga tidak memerlukan banyak pembuatan interface pada sistem. Website akan diakses menggunakan browser yang mendukung HTML5. Salah satu yang mendukung HTML5 adalah Mozilla Firefox.

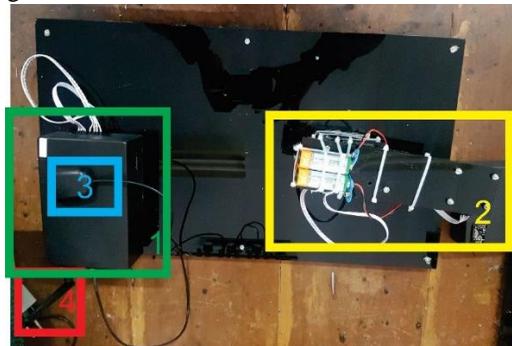


**Gambar 3. 18** Diagram Alir Perancangan Sistem

## BAB IV PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai pengujian dan analisa dari rancang bangun *Wireless Sensor Network* pada pendeteksi dini potensi kebakaran lahan gambut menggunakan Banana Pi IoT. Pengujian dan Analisa ini bertujuan untuk mengetahui kesesuaian kinerja alat dengan perancangan yang telah dibuat. Hasil dai pengamatan dan Analisa data dari pengujian digunakan untuk mengevaluasi sistem dan dapat digunakan sebagai referensi pengembangan lebih lanjut.

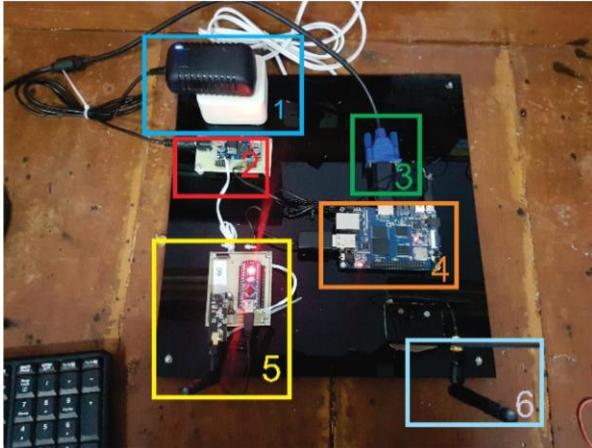
Pengujian ini dilakukan dengan melalui tahapan pengujian blok fungsional dan pengujian seluruh sistem. Tahapan pengujian blok fungsional dilakukan untuk mengetahui kinerja dari setiap komponen yang menyusun setiap sistem sehingga data dapat disajikan secara terperinci. Tahapan pengujian pengujian sistem dengan menggabungkan masing-masing blok fungsional menjadi sebuah sistem yang dapat menjalankan tugas khusus. Pengujian pada bab ini terdiri dari pengujian respon sensor *thermopile* terhadap lingkungan, pengujian jarak transmisi data antar node, pengujian web server, dan perancangan sistem secara keseluruhan. Pada Gambar 4.1 merupakan bentuk dari sistem *Sensor Node* yang telah dibuat. Sedangkan pada Gambar 4.2 merupakan bentuk dari Server pada sistem yang dibuat.



**Gambar 4. 1** Bentuk Sistem yang Dibuat

Keterangan dari Gambar 4.1 adalah sebagai berikut:

1. Kotak kontrol *Node*
2. Sensor Suhu serta pengkondisi sudut pembacaan suhu
3. Antena GPS
4. Antena nRF24L01



**Gambar 4. 2** Server dari Sistem yang Dibuat

Keterangan dari gambar 4.2 adalah sebagai berikut:

1. Catudaya berupa adaptor 12 V dan 2 A
2. Buck Converter untuk menurunkan tegangan dari 12 V ke 5 V. Digunakan untuk catudaya Banana Pi
3. Konektor HDMI untuk antarmuka Banana Pi dengan operator
4. Bananapi sebagai server dari sistem yang dibuat
5. Perangkat receiver WSN berupa Arduino dengan nRF24L101
6. Antena WiFi dari Banana Pi

#### 4.1 Pengujian Fungsional

Pengujian fungsional merupakan pengujian yang dilakukan pada masing-masing blok. Pengujian tersebut bertujuan untuk mengetahui apakah sistem bekerja sesuai yang diinginkan atau tidak,

selain itu untuk mengetahui karakteristik dari keluaran dan masukan masing-masing blok.

#### 4.1.1 Pengujian Catu Daya pada Sensor Node

Pengujian catu daya dilakukan untuk mengetahui konsumsi daya pada sistem sensor node. Untuk mengetahui daya yang dikonsumsi pada sensor node, maka sistem akan di-*suplay* dengan tegangan yang tetap dan kemudian dilakukan pembacaan arus yang dikonsumsi oleh sistem sensor node.

Catu daya yang digunakan untuk memberi tegangan pada saat pengukuran adalah output dari catu daya GW Instek. Output di set dengan tegangan 11,3 V. Sedangkan untuk pembacaan arus dilakukan dengan melakukan hubungan seri antara catu daya dengan beban.

Pengujian ini dilakukan seperti pada Gambar 4.3. untuk pengambilan data dilakukan saat dua kondisi yaitu saat kondisi sensor node aktif mengambil data suhu dan saat kondisi sensor node standby.



**Gambar 4. 3** Instalasi Pengukuran Arus

Pada Tabel 4.1 adalah hasil dari pembacaan pengujian arus dari sensor node. Pengukuran yang dilakukan adalah pengukuran arus yang mengalir menuju beban dengan tegangan yang ditentukan yaitu 11,3V. Untuk memperoleh besaran daya baik daya saat aktif maupun saat *standby* menggunakan Persamaan 4.1, Sedangkan untuk memperoleh besaran daya permenit menggunakan Persamaan 4.2.

$$P = V * I \quad (4.1)$$

$$P_{Rata} = \frac{(P_S * 40) + (P_A * 20)}{60} \quad (4.2)$$

Pada Persamaan 4.2 nilai 20 dan 40 merupakan durasi masing-masing kondisi pada sensor node pada satu menit. sedangkan 60 merupakan factor pembagi 60 detik atau 1 menit.

**Tabel 4. 1** Hasil Pengukuran Daya Sensor Node

No	Saat Standby			Saat Aktif			P <sub>rata</sub> (W)
	I (A)	V (V)	P <sub>s</sub> (W)	I (A)	V (V)	P <sub>A</sub> (W)	
1	0.57	11.3	6.4	0.64	11.3	7.2	6.57
2	0.56	11.3	6.3	0.65	11.3	7.3	6.49
3	0.59	11.3	6.6	0.66	11.3	7.4	6.79
4	0.55	11.3	6.2	0.64	11.3	7.2	6.38
5	0.57	11.3	6.4	0.64	11.3	7.2	6.57
6	0.58	11.3	6.5	0.64	11.3	7.2	6.67
Daya Rata-Rata							6.57

Keterangan : I : Arus pembacaan (A)

V : Tegangan pembacaan (V)

P<sub>s</sub> : Daya saat standby (W)

P<sub>A</sub> : Daya aktif (W)

P<sub>Rata</sub> : Daya rata-rata saat satu kali siklus (W)

Pada Tabel 4.1 hasil pengukuran daya rata-rata diperoleh sebesar 6.57 watt. Daya didapat dengan waktu stanby 40 detik, sehingga dapat dioptimalkan lagi apabila waktu *standby* diperlama.

#### 4.1.2 Pengujian Sensor Suhu

Pengujian ini dilakukan seperti pada Gambar 4.4, untuk menyamakan nilai yang terbaca pada output sensor dengan nilai yang

sesungguhnya dengan proses kalibrasi. Semakin bagus posisi kalibrasi, maka semakin akurat data yang dihasilkan. Gambar 4.4 menunjukkan metode kalibrasi dari sensor. Program yang digunakan pada pengujian ini adalah sebagai berikut.

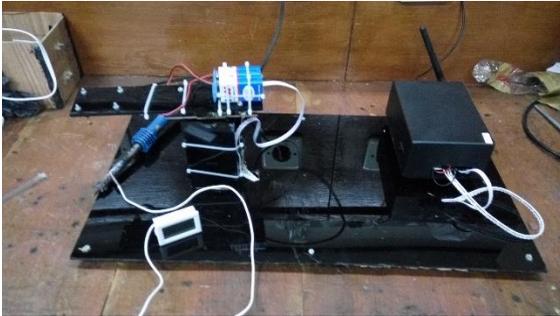
```
void setup() {
  Serial.begin(9600);
}

void loop() {
  uint16_t a,b,i, temp;
  a = analogRead(A4);
  b = analogRead(A5);
  Serial.print(a);
  Serial.print("    ");
  Serial.print(b);
  Serial.print("    ");
  delay(500);
}
```

Pada pengujian kalibrasi diperoleh data pada Tabel 4.2. Setelah mendapatkan hasil pembacaan maka akan dilakukan perancangan persamaan matematika dari hasil yang didapat. Persamaan yang telah didapat kemudian digunakan pada pemrograman pada Arduino. Pembacaan yang dilakukan dengan dua buah masukan yaitu pembacaan sensor *thermistor* dan pembacaan *thermopile*.

Hasil pembacaan yang telah didapat maka didapat persamaan (4.3). Persamaan yang telah didapat lalu diimplementasikan pada pemrograman Arduino, sehingga Arduino dapat digunakan untuk pembacaan suhu pada sensor node.

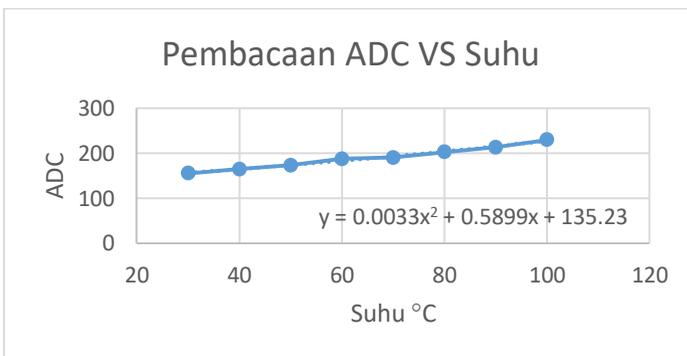
$$y = 0.0033x^2 + 0.5899x + 135.23 \quad (4.3)$$



**Gambar 4. 4** Pengujian Suhu dan Pembacaan ADC

**Tabel 4. 2** Pembacaan Suhu dan ADC pada Sensor

ADC		Jarak Objek ke Sensor (cm)	Suhu Terbaca (°C)
Thermopile	Thermistor		
155	559	10	30
164	559	10	40
173	559	10	50
187	559	10	60
190	559	10	70
202	599	10	80
213	599	10	90
229	599	10	100



**Gambar 4. 5** Grafik ADC VS Suhu

### 4.1.3 Pengujian Komunikasi WSN

Pada pengujian ini dilakukan untuk mengetahui apakah sistem dapat bekerja dengan baik dalam ruang lingkup jaringan. Pengujian dilakukan dengan dua tahap, pertama pengujian jarak komunikasi antar *node*, kedua adalah menghidupkan semua sensor *node* untuk mengetahui jangkauan terjauh dari sensor tersebut.

#### 4.1.3.1 Pengujian Titik Terjauh Antar *node*

Pengujian dilakukan untuk mencari titik terjauh dari sensor dengan transmisi tanpa halangan fisik antar *node*. Pada pengujian ini, *node* diletakkan pada wilayah yang luas dan tidak memiliki halangan saat peletakan *nodenya*. Area yang digunakan untuk percobaan adalah seperti pada Gambar 4.6. Dapat dilihat pada gambar tersebut jarak antar sensor diletakkan secara garis lurus dan tanpa ada halangan antar *nodenya*. Program yang digunakan untuk menguji sistem ini merupakan program utama dari sistem yang terdapat pada lampiran.

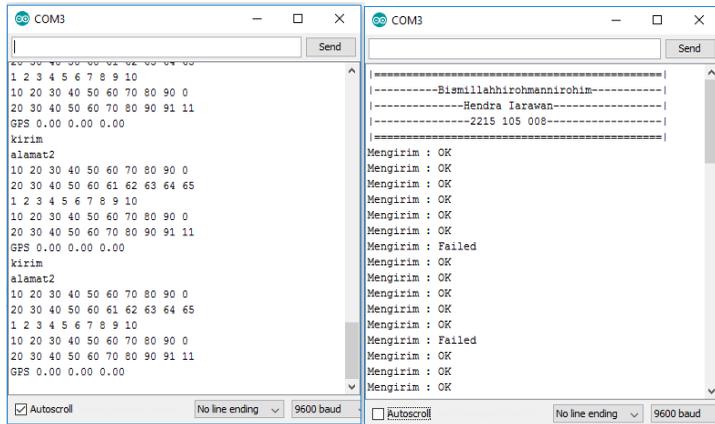
Pada Tabel 4.3 merupakan hasil dari pengukuran jarak antar *node*. Dapat diketahui biasanya komunikasi terjauh yang dapat dilakukan adalah 150 m. Pada percobaan ini masih terdapat beberapa kali *error* data namun pengiriman tetap dapat dilakukan pada saat pengiriman kedua atau ketiga. Gambar 4.7 merupakan hasil dari pembacaan serial monitor percobaan.



Gambar 4. 6 Peta Area Percobaan

**Tabel 4. 3** Percobaan Jarak Antar Node

Jarak (m)	Respon		
	Percobaan 1	Percobaan 2	Percobaan 3
10	Bisa	Bisa	Bisa
20	Bisa	Tidak Bisa	Bisa
30	Bisa	Bisa	Bisa
40	Tidak Bisa	Bisa	Bisa
50	Bisa	Bisa	Bisa
60	Bisa	Tidak Bisa	Bisa
70	Bisa	Bisa	Bisa
100	Bisa	Bisa	Bisa
150	Tidak Bisa	Tidak Bisa	Tidak Bisa



**Gambar 4. 7** Hasil Serial Monitor Percobaan

#### 4.1.3.2 Pengujian Keseluruhan Wireless Sensor Network

Pada sistem *Wireless Sensor Network* menggunakan topologi *tree*, seluruh sensor dan *receiver transmitter* diaktifkan. Pada uji coba dilakukan dengan jarak antar *node* 10 meter. Gambar 4.8 merupakan lokasi uji coba sensor node.

Hasil yang diperoleh pada *receiver server* dapat dilihat pada Gambar 4.5. Dari data yang diperoleh dapat disimpulkan bahwa *wireless sensor network* dapat bekerja dengan baik dan dapat menyalurkan data antar *node* dengan baik pula.

```
kirim
alamat2
10 20 30 40 50 60 70 80 90 0
20 30 40 50 60 61 62 63 64 65
1 2 3 4 5 6 7 8 9 10
10 20 30 40 50 60 70 80 90 0
20 30 40 50 60 70 80 90 91 11
GPS 0.00 0.00 0.00
kirim
alamat2
10 20 30 40 50 60 70 80 90 0
20 30 40 50 60 61 62 63 64 65
1 2 3 4 5 6 7 8 9 10
10 20 30 40 50 60 70 80 90 0
20 30 40 50 60 70 80 90 91 11
GPS 0.00 0.00 0.00
```

**Gambar 4. 8** Hasil Pembacaan Receiver pada Server

#### 4.1.4 Pengujian GPS

Pengujian GPS dilakukan dengan tujuan untuk mengetahui keakurasian dari sensor untuk menunjukkan lokasi sensor *node*. Indikator dari keberhasilan subbbab ini adalah hasil dari pembacaan titik koordinat lokasi.

Pengujian dilakukan pada ruangan Laboraturium Elektronika Industri B402 dengan menggunakan program dibawah dengan hasil output terdapat pada Tabel 4.4. Hasil dari pengujian dapat dilihat pada Gambar 4.9. Terdapat beberapa kali sensor *node* tidak dapat mengakses koordinat lokasi. Hal tersebut dikarenakan lokasi pengujian dilakukan pada ruangan tertutup, sehingga menghalangi dari pembacaan koordinat sensor yang dipasang. Persenstase keberhasilan pembacaan GPS adalah 80%.

```
void get_gps() {
    double home_lat;
    double home_lon;
    double home_alt;
    while (ss.available()>0) {
        gps.encode(ss.read());
    }
    if (gps.location.isUpdated()){
        home_lat = gps.location.lat();
```

```

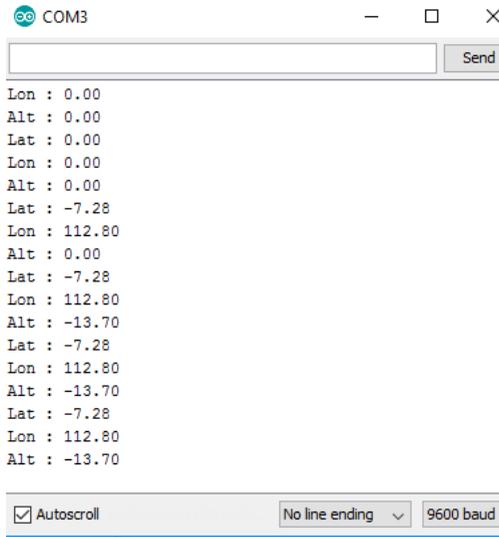
delay(10);
home_lon = gps.location.lng();
delay(10);
home_alt = gps.altitude.meters();
delay(10);
}

Serial.print("Lat : ");
Serial.println(home_lat);
Serial.print("Lon : ");
Serial.println(home_lon);
Serial.print("Alt : ");
Serial.println(home_alt);
}

```

**Tabel 4. 4** Percobaan Pembacaan Koordinat GPS

No	Latitude (Lintang)	Longitude (Bujur)	Meters (MDPL)
1	0	0	0
2	0	0	0
3	-7.28	112.8	0
4	-7.28	112.8	-13.7
5	-7.28	112.8	-13.7
6	-7.28	112.8	-13.5
7	-7.28	112.8	-13.7
8	-7.28	112.8	-13.5
9	-7.28	112.8	-13.6
10	-7.28	112.8	-13.7
Persentase Keberhasilan = 80%			



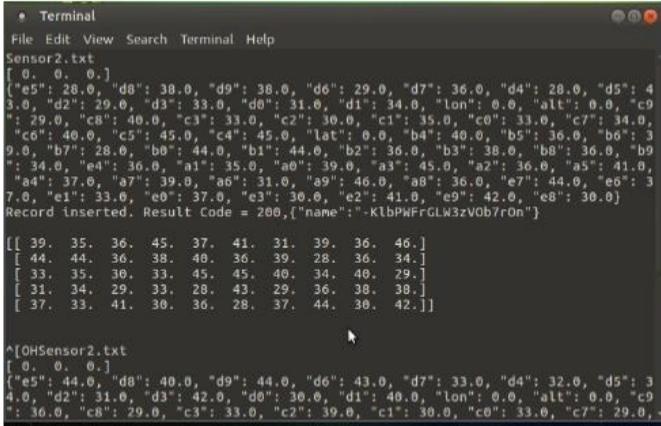
**Gambar 4. 9** Hasil Pembacaan GPS

#### **4.1.5 Pengujian Server**

Pengujian pada sever deilakukan dengan cara pembacaan data yang berasal dari jaringan WSN, data kemudian di dianalisa apakah data yang diterima terdapat data yang menunjukkan terjadinya potensi kebakaran lahan gambut, apabila terjadi potensi kebakaran lahan gambut, maka server akan secara mandiri mengirimkan pesan berupa email ke operator, lalu data akan di simpan secara *offline* dan *online*. Penyimpanan *offline*, dilakukan dengan menyimpan file dalam bentuk TXT. Sedangkan penyimpanan *online*, dilakukan dengan menyimpan file pada penyimpanan *coude* firebase. Pada Gambar 4.10 merupakan bentuk dari penyimpanan *online* pada server.

Pada penyimpanan *firebase*, data disimpan dalam bentuk *library*, data akan diberi alamat khusus oleh server, lalu server akan mengirimkan data tersebut ke sistem *firebase*, kemudian *firebase* memberitahukan apakah data telah berhasil disimpan atau belum. Dari Gamabr 4.8 dapat terlihat bahwa server berhasil menyimpan data pada *firebase* dengan kode penyimpanan “KlbPWFrGLW3zVOb7rOn”. Data yang telah disimpan dalam

bentuk *library*, dapat dipanggil kemudian dengan cara memanggil folder penyimpanan kemudian memanggil *library* dari data yang telah diberikan oleh server.



```
Terminal
File Edit View Search Terminal Help
Sensor2.txt
[ 0. 0. 0.]
{"e5": 28.0, "d8": 38.0, "d9": 38.0, "d6": 29.0, "d7": 36.0, "d4": 28.0, "d5": 43.0, "d2": 29.0, "d3": 33.0, "d0": 31.0, "d1": 34.0, "lon": 0.0, "alt": 0.0, "c9": 29.0, "c8": 40.0, "c3": 33.0, "c2": 30.0, "c1": 35.0, "c0": 33.0, "c7": 34.0, "c6": 40.0, "c5": 45.0, "c4": 45.0, "lat": 0.0, "b4": 40.0, "b5": 36.0, "b6": 39.0, "b7": 28.0, "b0": 44.0, "b1": 44.0, "b2": 36.0, "b3": 38.0, "b8": 36.0, "b9": 34.0, "e4": 36.0, "a1": 35.0, "a0": 39.0, "a3": 45.0, "a2": 36.0, "a5": 41.0, "a4": 37.0, "a7": 39.0, "a6": 31.0, "a9": 46.0, "a8": 36.0, "e7": 44.0, "e6": 37.0, "e1": 33.0, "e0": 37.0, "e3": 30.0, "e2": 41.0, "e9": 42.0, "e8": 30.0}
Record inserted. Result Code = 200, {"name": "-KlbPWFrGLW3zVOb7rOn"}

[[ 39. 35. 36. 45. 37. 41. 31. 39. 36. 46.]
 [ 44. 44. 36. 38. 40. 36. 39. 28. 36. 34.]
 [ 33. 35. 30. 33. 45. 45. 40. 34. 40. 29.]
 [ 31. 34. 29. 33. 28. 43. 29. 36. 38. 38.]
 [ 37. 33. 41. 30. 36. 28. 37. 44. 30. 42.]]

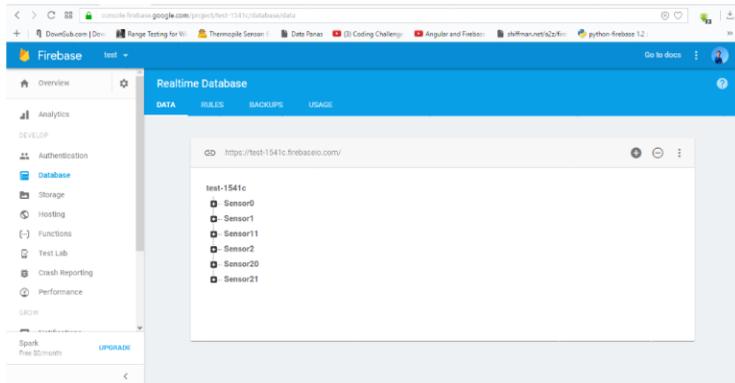
^[[OHSensor2.txt
[ 0. 0. 0.]
{"e5": 44.0, "d8": 40.0, "d9": 44.0, "d6": 43.0, "d7": 33.0, "d4": 32.0, "d5": 34.0, "d2": 31.0, "d3": 42.0, "d0": 30.0, "d1": 40.0, "lon": 0.0, "alt": 0.0, "c9": 36.0, "c8": 29.0, "c3": 33.0, "c2": 39.0, "c1": 30.0, "c0": 33.0, "c7": 29.0, -
```

Gambar 4. 10 Tampilan Proses IO pada Server

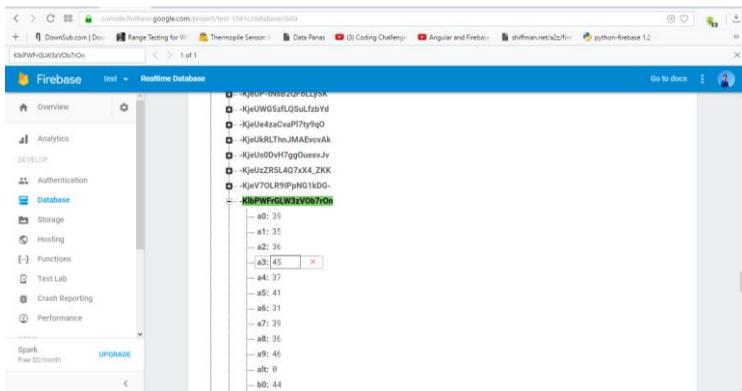
#### 4.1.6 Pengujian Database

Pengujian *database* merupakan pengujian yang dilakukan dengan melihat data yang telah di-*upload* oleh server kedalam *database*, data dapat dilihat dengan cara membuka halaman *database* di <https://firebase.google.com> atau dapat mengakses halaman <https://console.firebase.google.com/project/test-1541c/database/data>. Data yang telah disimpan diberinama test-1541c, pengaksesan data bersifat terbuka, sehingga dapat diakses meski bukan pemilik akun penyimpanan *database* tersebut. Gambar 4.11 merupakan tampilan awal dari penyimpanan *database*, terdapat beberapa folder yang berisi data sensor dari beberapa *node*.

Pada Gambar 4.12 merupakan penyimpanan *database* yang dimaksudkan sebelumnya. Dapat dilihat pada gambar tersebut terdapat folder dengan nama penyimpanan “KlbPWFrGLW3zVOb7rOn” yang merupakan data yang telah dibahas pada subbab sebelumnya. Dari hasil pengamatan ini dapat disimpulkan bawasanya server berhasil menyimpan data sensor.



Gambar 4. 11 Tampilan Awal Firebase



Gambar 4. 12 Letak Folder KlbPWFrGLW3zVOB7rOn

#### 4.1.7 Pengujian Website

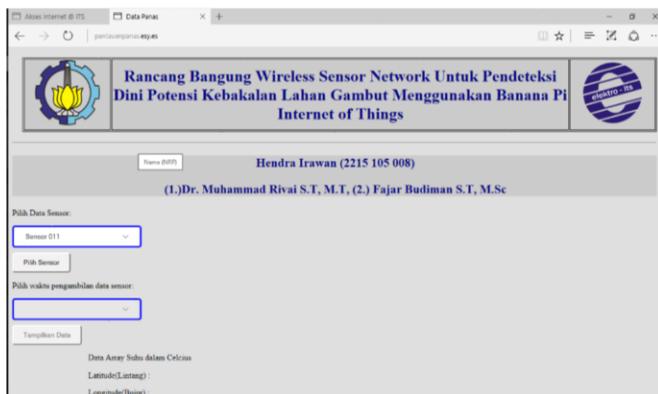
Website didesain sebagai interface antara sistem dengan *user* atau operator menggunakan HTML5. Sehingga diperlukan *browser* yang memiliki fitur yang dapat mengakses HTML5. Penggunaan HTML dalam pendesainan *interface* dikarenakan HTML memiliki akses pada *firebase*. Hal tersebut dikarenakan *firebase* memiliki *library* yang dapat digunakan pada HTML.

Pemublikasian web menggunakan *publizer* milik Idhoustinger yang merupakan penyedia *free hosting*. Lama aktif *hosting* tersebut selama satu tahun. Pada penerapannya untuk meng-*upload* data digunakan aplikasi tambahan yaitu File Zilia.

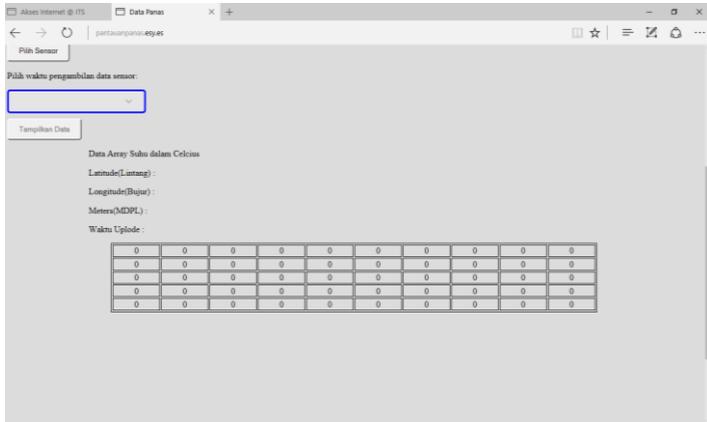
Desain yang telah dibuat lalu dipublikasikan dengan alamat <http://pantauanpanas.esy.es..> Pada desain website ditampilkan berupa fitur-fitur *interface* dapat dipahami dan diakses dengan mudah. Bentuk tampilan awal website dapat dilihat pada Gambar 4.13.

Fitur-fitur yang ditampilkan pada website seperti pada Gambar 4.13 dan 4.14 adalah sebagai berikut:

- *Header* berupa tampilan Judul tugas akhir beserta logo ITS dan Elektro ITS.
- Nama penyusun tugas akhir.
- Pilihan *node* sensor.
- Pilihan pembacaan data.
- Tampilan koordinat lokasi sensor berupa koordinat GPS.
- Tampilan data pembacaan sensor berupa tabel.
- Tampilan ilustrasi pembacaan suhu.



**Gambar 4. 13** Tampilan Awal Web



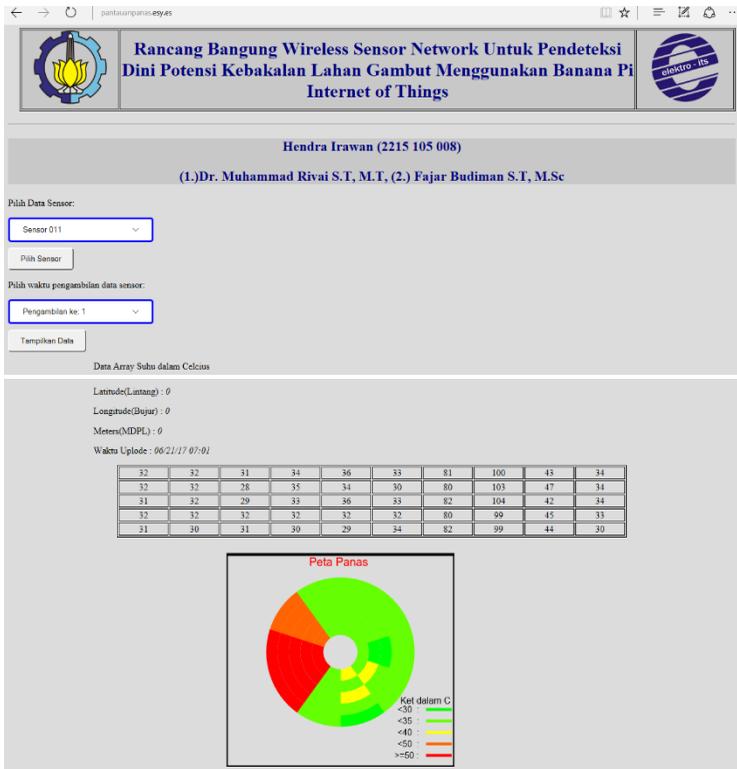
**Gambar 4. 14** Tampilan saat Dioperasikan

## 4.2 Pengujian *Internet of Things*

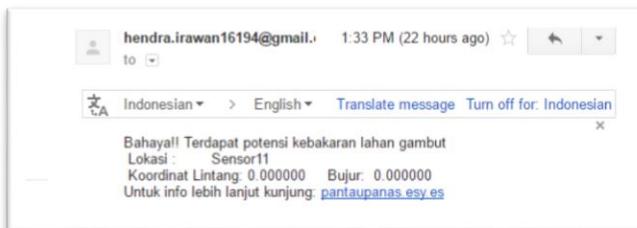
Sistem *Internet of Things* merupakan induk dari semua sistem yang dibangun. Pengujian *Internet of Things* dapat dilakukan dengan cara mengaktifkan seluruh sistem yang ada dan melihat respon keluaran dari sistem keseluruhan yang keluarannya akan dipublikasikan melalui internet.

Hasil dari sistem keseluruhan dapat dilihat dengan mengunjungi <http://pantauanpanas.esy.es/>. Dapat dilihat dari tampilan halaman web terdapat panel-panel yang terhubung dengan database sehingga dapat menampilkan isi dari *database*. Hasil dari sistem dapat dilihat pada Gambar 4.15. Selain itu, keluaran sistem akan menghasilkan keluaran berupa email kepada operator yang telah ditentukan.

Gambar 4.16 merupakan bentuk email yang dikirim. Pengiriman email dialamatkan kepada [hendrainawan160194@gmail.com](mailto:hendrainawan160194@gmail.com). Pengiriman terbukti berhasil saat terjadi pembacaan potensi kebakaran.



**Gambar 4. 15** Tampilan dari Hasil Pembacaan Suhu Sekitar.



**Gambar 4. 16** Email Masuk Kepada Operator

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Kesimpulan yang dapat diambil oleh penulis dari tugas akhir ini adalah sistem yang dibuat dengan Arduino sebagai prosesor sensor node dan Banana Pi sebagai server dapat dioperasikan sebagai peralatan *Wireless Sensor Network* dengan topologi *tree* dengan baik. Konsumsi daya rata-rata pada peralatan sensor sebesar 6,57W. Pencuplik data analog sensor suhu dengan kepadatan data 10bit.

Jarak terjauh antar node yang dapat dilakukan sistem adalah 100 m tanpa halangan. Jarak pantau dapat diperjauh dengan menambah rantai node. Penyimpanan data pada server dapat dilakukan dengan cara *online* maupun *offline*. Sebagai interface agar pengguna mudah memahami data pembacaan dapat mengunjungi <http://pantauanpanas.esy.es/>. Pemberitahuan bahaya terhadap potensi kebakaran lahan gambut dapat dilaksanakan dengan pengiriman email kepada operator. Pengujian Internet of Things pada sistem dilakukan dengan mengaktifkan seluruh sistem yang ada.

#### **5.2 Saran**

Beberapa saran yang penulis bisa berikan untuk pengembangan tugas akhir adalah:

1. Perancangan sensor yang perlu lebih ditingkatkan adalah penambahan sensor-sensor pendukung dalam mendeteksi potensi kebakaran lahan gambut.
2. Penerapan sistem dirasa perlu untuk memberi panel surya agar penyipadan daya selalu tersedia.
3. Perlu adanya penelitian lebih lanjut tentang penggunaan *Wireless Sensor Network* berbasis nRF24L01 agar dapat menggunakan topologi yang lebih bagus dibandingkan dengan topologi yang digunakan pada saat sekarang ini.

*(halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] Anonim, “Gambut Indonesia Luas Tersebar dan Mudah Terbakar,” *GEOMAGZ*. [Online]. Available: <http://geomagz.geologi.esdm.go.id/gambut-indonesia-luas-tersebar-dan-mudah-terbakar/>. [Accessed: 12-Dec-2016].
- [2] Kelompok Kerja Pengelolaan LAhan Gambut Nasional, *Pengelolaan Lahan Gambut*, no. September. 2006.
- [3] E. Mileva *et al.*, “Kerugian dari Kebakaran Hutan,” *world bank group*, p. 12, 2016.
- [4] H. Donasi *et al.*, “2 Penyebab Mengapa Hutan Riau Kembali Terbakar,” p. 2015, 2016.
- [5] U. Chokkalingam, *Kebakaran di Lahan Rawa/Gambut di Sumatera: Masalah dan Solusi*. Jakarta: Center for International Forestry Research, 2003.
- [6] T. Maya, “Kebakaran Hutan dan Lahan Gambut di Kalimantan Tengah Butuh Penanganan Serius,” *WFF*, 2012. [Online]. Available: <http://www.wwf.or.id/?11040/Kebakaran-Hutan-dan-Lahan-Gambut-di-Kalimantan-Tengah-Butuh-Penanganan-Serius>.
- [7] A. Akbar and S. Faidil, “Kebakaran hutan dan lahan rawa gambut: penyebab faktor pendukung dan alternatif pengelolaannya,” pp. 1–18, 2013.
- [8] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks*. 2007.
- [9] “What Is a Wireless Sensor Network?,” *National Instruments*. [Online]. Available: <http://www.ni.com/white-paper/7142/en/>.
- [10] I. P. A. E. Pratama and S. Suakanto, *Wireless Sensor Network*. Bandung: INFORMATIKA, 2015.
- [11] E. D. Meutia, J. Teknik, E. Universitas, and S. Kuala, “Internet of Things – Keamanan dan Privasi,” *Semin. Nas. dan Expo Tek. Elektro 2015*, pp. 85–89, 2015.
- [12] Y. Yudhanto, “Apa itu Internet of Things?,” *Jurnalkomputer*, vol. 20, no. 3, pp. 1–7, 2007.
- [13] Y. Oktarina and P. Risma, “Aplikasi Sensor Radio Frekuensi ( Rf ),” *J. Teknol. DAN Inform. (TEKNOMATIKA)Aplikasi*

- Sens. Radio Frekuensi*, vol. 5, no. 2, pp. 106–119, 2015.
- [14] A. El-Rabany, *GPS The Global Positioning System*. London: Artech House, 2002.
  - [15] Banana Pi, “What is Banana Pi M2 Ultra?,” *Banana Pi*. [Online]. Available: <http://www.banana-pi.org/m2u.html>. [Accessed: 05-Dec-2016].
  - [16] R. L. Halterman, “Fundamentals of Python Programming,” Chattanooga: Southern Adventist University, 2016.
  - [17] J. Melorose, R. Perroy, and S. Careas, *Arduino Nano*, vol. 1. 2015.
  - [18] Measurement Specialties, “Thermopile Sensor TS118-3,” no. June, pp. 1–2, 2007.
  - [19] Nordic Semiconductor, “nRF24L01 Single Chip 2 . 4GHz Transceiver Product Specification,” no. July, 2007.
  - [20] u-Blox, “NEO-M8 u-blox M8 concurrent GNSS modules: Data Sheet,” 2015.
  - [21] “Library nRF24Network.h.” [Online]. Available: <http://maniacbug.github.io/RF24Network/index.html>.

## LAMPIRAN

### 1. Program Sensor Node

Program sensor node digunakan untuk mengatur semua kinerja dari sensor node. Adapun program yang digunakan adalah sebagai berikut:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

//Instalasi Radio
RF24 radio(9,10); //sambungan pin CE dan CS
RF24Network network(radio);

const uint16_t mynode = 011; //alamat node
const uint16_t nodeatas = 01; // alamat node
atas

const unsigned long interval =1000; //ms -
>interval pengiriman data ke node
selanjutnya
const unsigned long waktubaca =60000;

//Instalasi GPS
static const int RXPin = 4, TXPin =
3;//komunikasi TX RX untuk GPS
static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial ss(RXPin,TXPin);
```

```

double buff_gps[3];

//double uav_lat = -7.27679;
//double uav_lon = 112.785446;
//double uav_alt = 50.00;

//waktu akses
unsigned long sebelum;

// nilai step akses
int x = 1;

// membuat paket data
struct paket{
    uint16_t alamat; //alamat asala paket
    //unsigned long waktu; //waktu dibutuhkan
    dalam pengiriman
    uint16_t data[5][10]; //data yang
    berhasil disimpan
    double data_gps[3];
};

paket paketku;

// Instalasi Stepper
const int stepPin = A1;
const int dirPin = A0;
const int delayStep = 4;

void setup() {
    Serial.begin(9600);
    //delay(2000);
    ss.begin(GPSBaud);
    pinMode(stepPin, OUTPUT);

```

```

pinMode(dirPin,OUTPUT);
delay(2000);

Serial.println("|=====
=====|");
Serial.println("|-----
Bismillahirohmannirohim-----|");
Serial.println("|-----Hendra
Iarawan-----|");
Serial.println("|-----2215 105
008-----|");

Serial.println("|=====
=====|");

SPI.begin();
radio.begin();
network.begin(/*chenel*/ 84, /*alamat
node*/ mynode); //untuk 01 dan 02 chenel
80 dan chenel atas 81, untuk 11 dan 22
chenel 79 dan chenel atas 80,

}
void data(){
int a, b;
for(int i=0;i<10;i++){
for (int j=0;j<5;j++){
a = analogRead(A4);
b = analogRead(A5);
if (b<568){
paketku.data[j][i] =
map(a,230,470,20,110);
}else{
paketku.data[j][i] =
map(a,220,350,20,100);
}
}
}
digitalWrite(dirPin,HIGH);
for(int j=0;j<20;j++){

```

```

        digitalWrite(stepPin, HIGH);
        delay(delayStep);
        digitalWrite(stepPin, LOW);
        delay(delayStep);
    }

void get_gps() {
    double home_lat;
    double home_lon;
    double home_alt;
    while (ss.available() > 0) {
        gps.encode(ss.read());
    }
    if (gps.location.isUpdated()) {
        home_lat = gps.location.lat();
        delay(10);
        home_lon = gps.location.lng();
        delay(10);
        home_alt = gps.altitude.meters();
        delay(10);
    }
    if (home_lat != 0) {
        buff_gps[0] = home_lat;
        buff_gps[1] = home_lon;
        buff_gps[2] = home_alt;
    } else {
        home_lat = buff_gps[0];
        home_lon = buff_gps[1];
        home_alt = buff_gps[2];
    }

    paketku.data_gps[0] = home_lat;
    paketku.data_gps[1] = home_lon;
    paketku.data_gps[2] = home_alt;
}

void kirim(paket mypaket) {
    unsigned long waktu = millis();

```

```

    network.begin(/*chenel*/ 82, /*alamat
node*/ mynode);//chenel 01 dan 02 =81,
chenel 011 dan 021=80
    network.update();
    if (waktu - sebelum>=interval);{
        sebelum=waktu;
        Serial.print("Mengirim : ");
        RF24NetworkHeader header(/*node tujuan*/
01);
        bool ok= network.write(header, &mypaket,
sizeof (mypaket));
        if(ok){
            Serial.println("OK");
            network.begin(/*chenel*/ 84, /*alamat
node*/ mynode);//chenel 01 dan 02 =80,
chenel 011 dan 021=79
            if (x==1)
                step();
        }else{
            Serial.println("Failed");
            sebelum=waktu;
            kirim(mypaket);
            delay(500);
        }
        delay(1000);
    }
}

void step(){
    x++;
    if (x>2)
        x=1;
}

void terima(){
    unsigned long waktu = millis();
    paket dapat;
    for( sebelum=waktu ; waktu-
sebelum<=waktubaca; waktu=millis()){

```

```

network.update();
int i;
while(network.available()){
    RF24NetworkHeader header;
    bool ok = network.read(header, &dapat,
sizeof (dapat));
    if(ok){
        Serial.println("recieved packet #");
        kirim(dapat);
    }else
        //Serial.println("belum ada data");
    }
}
step();
}

void loop() {
//  get_gps();
//while (1){
    switch (x){
        case 1:
            data();
            break;
        case 2:
            terima();
            break;

    }

}
}

```

## 2. Program Server

Program pada server ditujukan untuk menerima data dari node, menyimpan data pada file, menyimpan data pada firebase, serta mengirimkan preringatan kepada operator apabila terdeteksi potensi kebakaran lahan gambut.

Program akan dibagi menjadi dua yaitu pada Arduino untuk berkomunikasi dengan node, dan Banana Pi digunakan sebagai pemroses pada server.

#### a. Arduino

```
#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

RF24 radio(9,10);
RF24Network network(radio);

const uint16_t mynode = 00; //alamat node

struct paket{
    uint16_t alamat; //alamat asala paket
    //unsigned long waktu; //waktu
    dibutuhkan dalam pengiriman
    uint16_t data[5][10]; //data yang
    berhasil disimpan
    double data_gps[3];
};
paket dapat;

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

    SPI.begin();
    radio.begin();
    network.begin(/*chenel*/ 81, /*alamat
node*/ mynode);
}

void loop() {
    network.update();
    while(network.available()){
        //Serial.println("cek");
        RF24NetworkHeader header;
```

```

        bool ok = network.read(header,
&dapat, sizeof (dapat));
        if(ok) {
            Serial.println("kirim");
            Serial.print ("alamat");
            Serial.println(dapat.alamat,OCT);
            for(int i=0;i<5;i++){
                //Serial.print("[i]");
                for(int j=0;j<10;j++){

Serial.print(dapat.data[i][j]);
                Serial.print(" ");
                }
                Serial.println("");
            }
            Serial.print("GPS ");
            for(int i=0;i<3;i++){
                Serial.print(dapat.data_gps[i]);
                Serial.print(" ");
            }
            Serial.println("");
        }else
            Serial.println("-");
    }
}

```

## b. Banana Pi

```

import serial #library serial
import numpy as np #library array
import requests
import json
import datetime
import smtplib

url = 'https://test-1541c.firebaseio.com/'

port = serial.Serial("/dev/ttyUSB0", 9600)
#definisi serial
data_str="" #deklarasi string

```

```

nilai=np.ones((5,10))
 kirim=np.ones(53)
 dict=[]
 gps=np.ones(3)
 bahaya=0

while True:
    a = port.readline()
    b = ' kirim' in a
    a = port.readline()
    if 'alamat' in a:
        x = len(a)-2
        nama= "Sensor"+a[6:x]
        nama_file = nama+".txt"
        print nama_file
        fo = open(nama_file,"w")
        while b:
            a = port.readline()
            #print a
            if 'GPS' in a:
                fo.close()
                fo = open("GPS "+nama_file, "w")
                fo.write(a[3:])
                n = a[4:]
                x = n.find(' ')
                m=float(n[:x])
                gps[0]=float(m)
                n=n[x+1:]
                x = n.find(' ')
                m=float(n[:x])
                gps[1]=m
                n=n[x+1:]
                x = n.find(' ')
                m=float(n[:x])
                gps[2]=m
                fo.close()
                print gps
                now = datetime.datetime.now()

```

```

    waktu=str( now.strftime("%D
%H:%M"))
    print waktu
    b = False
else:

    #print a
    fo.write(a)
    data_str= data_str+a
#print data_str
i=0

while i<5:
    if("\n" in data_str):
        x=data_str.find("\n")
    else:
        x=len(data_str)
    a=data_str[:x]
    data_str= data_str[x+1:]
    #print data_str
    k=0
    j=0
    while j<10:
        if(" " in a):
            k=a.find(" ")

        else:
            k=len(a)

        y=a[:k]
        nilai[i][j]= int (y)
        if (nilai[i][j]>50):
            bahaya = 1
        dict1 = {str(i)+' '+str(j):
int(y) }
        dict.append(dict1)
        a=a[k+1:]
        j=j+1
    i=i+1

data={'a0':nilai[0][0], 'a1':nilai[0][1],

```

```

'a2':nilai[0][2], 'a3':nilai[0][3], 'a4':nilai
[0][4], 'a5':nilai[0][5], 'a6':nilai[0][6], 'a7
':nilai[0][7], 'a8':nilai[0][8], 'a9':nilai[0]
[9],

'b0':nilai[1][0], 'b1':nilai[1][1], 'b2':nilai
[1][2], 'b3':nilai[1][3], 'b4':nilai[1][4], 'b5
':nilai[1][5], 'b6':nilai[1][6], 'b7':nilai[1]
[7], 'b8':nilai[1][8], 'b9':nilai[1][9],

'c0':nilai[2][0], 'c1':nilai[2][1], 'c2':nilai
[2][2], 'c3':nilai[2][3], 'c4':nilai[2][4], 'c5
':nilai[2][5], 'c6':nilai[2][6], 'c7':nilai[2]
[7], 'c8':nilai[2][8], 'c9':nilai[2][9],

'd0':nilai[3][0], 'd1':nilai[3][1], 'd2':nilai
[3][2], 'd3':nilai[3][3], 'd4':nilai[3][4], 'd5
':nilai[3][5], 'd6':nilai[3][6], 'd7':nilai[3]
[7], 'd8':nilai[3][8], 'd9':nilai[3][9],

'e0':nilai[4][0], 'e1':nilai[4][1], 'e2':nilai
[4][2], 'e3':nilai[4][3], 'e4':nilai[4][4], 'e5
':nilai[4][5], 'e6':nilai[4][6], 'e7':nilai[4]
[7], 'e8':nilai[4][8], 'e9':nilai[4][9],
        'lat':gps[0], 'lon': gps[1],
'alt': gps[2], 'time' : waktu}
    try:
        data=json.dumps(data)
        print data
        result =
requests.post(url+nama+'.json',data)
        print 'Record inserted. Result Code
= '+
str(result.status_code)+' ,'+result.text+'\n'
    except IOError:
        print 'error! something wrong'
        print nilai
        print '\n'
        if (bahaya==1):

```

```

        try:
            fromaddr =
'hendra.irawan16194@gmail.com'
            toaddrs =
'hendrairawan160194@gmail.com'
            msg = "Bahaya!! Terdapat potensi
kebakaran lahan gambut \n Lokasi : \t %s \n
Koordinat Lintang: %f \t Bujur: %f \n Untuk
info lebih lanjut kunjungi:
pantaupanas.esy.es" % (nama, gps[0], gps[1])
            username =
'hendra.irawan16194@gmail.com'
            password = 'xndrairawan'
            server =
smtpplib.SMTP('smtp.gmail.com:587')
            server.starttls()
            server.login(username,password)
            server.sendmail(fromaddr,
toaddrs, msg)
            server.quit()
            print "success"
        except:
            print "send mail faild"
            bahaya=0

    else:
        print "data salah"

```

### 3. Program Web

Program Web digunakan untuk mengambil data dari database, menampilkannya pada halaman web untuk memberikan infoemasi kepada operator mengenai kondisi lapangan. Pemrograman web dibagi menjadi tiga tahap pemrograman yaitu pemrograman HTML, pemrograman, CSS, dan Java Split.

#### a. HTML

```

<!DOCTYPE html>
<html>

```

```

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet"
type="text/css" href="style.css">
  <script language="javascript"
type="text/javascript"
src="libraries/p5.js"></script>
  <!--<script language ="javascript"
type="text/javascript"
src="libraries/p5.dom.js"></script>-->
  <script
src="https://www.gstatic.com/firebasejs/3.9.
0/firebase.js"></script>
  <script src = "motion.js"></script>
  <title>Data Panas</title>

</head>
<body >
  <h1 >
    <table style="width: 100%">
      <tr><td ></td>
      <td
style="width: 900px"><b>Rancang Bangun
Wireless Sensor Network Untuk Pendeteksi
Dini Potensi Kebakalan Lahan Gambut
Menggunakan Banana Pi Internet of
Things</b></td>
      <td ></td></tr>

    </table>
  </h1>

```

```

        <!--<h1 style="text-align:center;">Database Wireless Sensor Network</h1-->
        <hr>
        <h2 title="Nama (NRP)"><p><b>Hendra Irawan (2215 105 008)</b></p>
        <p><b>(1.)Dr. Muhammad Rivai S.T, M.T, (2.) Fajar Budiman S.T, M.Sc</b></p>

        </h2>

        <form>
            <p>Pilih Data Sensor: </p>
            <select id="namaSensor" >
                <option>-Pilih Sensor-</option>
                <option>Sensor 01</option>
                <option>Sensor 02</option>
                <option>Sensor 011</option>
                <option>Sensor 021</option>
            </select>
        </form>
        <button type="Button" onclick="pilihSensor()">Pilih Sensor</button>
        <form>
            <p>Pilih waktu pengambilan data sensor: </p>
            <select id="waktuSensor" >
                </select>
        </form>
        <button id = "tampil" type="Button" onclick="tampilData()">Tampilkan Data</button>

```

```

    <p style="margin-left: 150px">Data
Array Suhu dalam Celcius</p>
    <p style="margin-left:
150px">Latitude (Lintang) : <var
id='Lat'></var></p>
    <p style="margin-left:
150px">Longitude (Bujur) : <var
id='Lon'></var></p>
    <p style="margin-left:
150px">Meters (MDPL) : <var
id='Alt'></var></p>
    <p style="margin-left: 150px">Waktu
Uplode : <var id='Time'></var></p>
    <table>
        <tr> <td id='00'>0</td><td
id='01'>0</td><td id='02'>0</td><td
id='03'>0</td><td id='04'>0</td><td
id='05'>0</td><td id='06'>0</td><td
id='07'>0</td><td id='08'>0</td><td
id='09'>0</td> </tr>
        <tr> <td id='10'>0</td><td
id='11'>0</td><td id='12'>0</td><td
id='13'>0</td><td id='14'>0</td><td
id='15'>0</td><td id='16'>0</td><td
id='17'>0</td><td id='18'>0</td><td
id='19'>0</td> </tr>
        <tr> <td id='20'>0</td><td
id='21'>0</td><td id='22'>0</td><td
id='23'>0</td><td id='24'>0</td><td
id='25'>0</td><td id='26'>0</td><td
id='27'>0</td><td id='28'>0</td><td
id='29'>0</td> </tr>
        <tr> <td id='30'>0</td><td
id='31'>0</td><td id='32'>0</td><td
id='33'>0</td><td id='34'>0</td><td
id='35'>0</td><td id='36'>0</td><td
id='37'>0</td><td id='38'>0</td><td
id='39'>0</td> </tr>

```

```

        <tr> <td id='40'>0</td><td
id='41'>0</td><td id='42'>0</td><td
id='43'>0</td><td id='44'>0</td><td
id='45'>0</td><td id='46'>0</td><td
id='47'>0</td><td id='48'>0</td><td
id='49'>0</td> </tr>
    </table>
    <canvas id="myCanvas" width="400"
height="400" >
        Your browser does not support the
HTML5 canvas tag.
    </canvas>
</body>
</html>

```

## b. CSS

```

h1{
    text-align:center;
    background-color: rgb(200,200,200);
    color: navy;
    margin-left: 20px;
}

h2 {
    background-color: rgb(200,200,200);
    color: navy;
    text-align:center;
}

table,tr,td{
    width: 70%;
    text-align: center;
    table-layout: fixed;
    margin: auto;
    border: 1px solid black;
}

select {

```

```

width: 20%;
padding: 10px 20px;
border: blue solid;
border-radius: 5px;
}

input{
margin-top: 10px;
padding: 10px 20px;
border-radius: 5px;
}
canvas{
border: none;
margin-left:30%;
}
button{
margin-top: 10px;
padding: 10px 20px;
border-radius: 5px;
}
body{

background-color: rgb(220,220,220);
color: rgb(255,255,255);
}

```

### c. Java Split

```

var simpan = create2dArray(5);
var sensor = create2dArray(5);
var keys = [];
var gps = [];

function setup(){
var config = {
apiKey:
"AIzaSyBZjSoskSRXe3sc2NVMAhXWUWNMdpWRWVU",

```

```

        authDomain: "test-
1541c.firebaseio.com",
        databaseURL: "https://test-
1541c.firebaseio.com",
        projectId: "test-1541c",
        storageBucket: "test-
1541c.appspot.com",
        messagingSenderId: "56321235078"
    };
    firebase.initializeApp(config);
    //console.log(firebase);
    database = firebase.database();

document.getElementById("waktuSensor").disab
led = true;

document.getElementById("tampil").disabled =
true;

    /*var ref =database.ref('Sensor1');
    ref.on('value',getData, errData);*/
}

function pilihSensor() {
    document.getElementById("waktuSensor").d
isabled = false;
    document.getElementById("tampil").disabl
ed = false;
    var x =
document.getElementById("waktuSensor").lengt
h;
    console.log(x);
    for(var i=0;i<x;i++){
        var y =
document.getElementById("waktuSensor");
        y.remove(0);
    }
}

```

```

    var x =
document.getElementById("namaSensor").select
edIndex;
    if(x==0){
        alert("Anda Belum memilih Sensor");

        document.getElementById("waktuSensor").d
isabled = true;

        document.getElementById("tampil").disabl
ed = true;
    }
    else if(x==1){
        var ref =database.ref('Sensor1');
        ref.on('value',getData, errData);
    }
    else if(x==2){
        var ref=database.ref('Sensor2');
        ref.on('value',getData, errData);
    }
    else if(x==3){
        var ref=database.ref('Sensor11');
        ref.on('value',getData, errData);
    }
    else if(x==4){
        var ref=database.ref('Sensor21');
        ref.on('value',getData, errData);
    }
}

function getData(data){
    console.log(data.val());
    sensor = data.val();

    var x =
document.getElementById("waktuSensor").lengt
h;
    console.log(x);

```

```

    for(var i=0;i<x;i++){
        var y =
document.getElementById("waktuSensor");
        y.remove(0);
    }
    keys = Object.keys(sensor);

    console.log(keys);

    for(var i = 0;i<keys.length;i++){
        var x =
document.getElementById("waktuSensor");
        var option =
document.createElement("option");
        var j=i+1;
        option.text="Pengambilan ke: " +
j.toString();
        x.add(option,x[i])
    }
}
function tampilData(){
    var i =
document.getElementById("waktuSensor").selectedIndex;
    var k = keys[i];
    var x=sensor[k].a0;
    simpan[0][0]= x;
    x=sensor[k].a1;
    simpan[0][1]= x;
    x=sensor[k].a2;
    simpan[0][2]= x;
    x=sensor[k].a3;
    simpan[0][3]= x;
    x=sensor[k].a4;
    simpan[0][4]= x;
    x=sensor[k].a5;
    simpan[0][5]= x;
    x=sensor[k].a6;
    simpan[0][6]= x;

```

```
x=sensor[k].a7;  
simpan[0][7]= x;  
x=sensor[k].a8;  
simpan[0][8]= x;  
x=sensor[k].a9;  
simpan[0][9]= x;
```

```
... ..
```

```
x=sensor[k].e0;  
simpan[4][0]= x;  
x=sensor[k].e1;  
simpan[4][1]= x;  
x=sensor[k].e2;  
simpan[4][2]= x;  
x=sensor[k].e3;  
simpan[4][3]= x;  
x=sensor[k].e4;  
simpan[4][4]= x;  
x=sensor[k].e5;  
simpan[4][5]= x;  
x=sensor[k].e6;  
simpan[4][6]= x;  
x=sensor[k].e7;  
simpan[4][7]= x;  
x=sensor[k].e8;  
simpan[4][8]= x;  
x=sensor[k].e9;  
simpan[4][9]= x;
```

```
x=sensor[k].lat;  
gps[0]=x;  
x=sensor[k].lon;  
gps[1]=x;  
x=sensor[k].alt;  
gps[2]=x;  
x=sensor[k].time;  
var time=x;
```

```

    document.getElementById('Lat').innerHTML
= gps[0];
    document.getElementById('Lon').innerHTML
= gps[1];
    document.getElementById('Alt').innerHTML
= gps[2];
    document.getElementById('Time').innerHTM
L = time;
    //console.log(simpan);

    for(var i=0;i<5;i++){
        for(var j=0;j<10;j++){
            var alamat =
i.toString()+j.toString();

            document.getElementById(alamat).innerHTM
L = simpan[i][j].toString();
        }
    }
    gambar();

}

function create2dArray(rows) {
    var arr=[];
    for (var i=0; i< rows; i++){
        arr[i]=[];
    }
    return arr;
}

function errData(err) {
    console.log('Error!');
    console.log(err);
}

function warna(suhu) {

```

```

    if (suhu<30) return "#00FF00";
    else if(suhu<35) return "#64FF00";
    else if(suhu<40) return "#FFFF00";
    else if(suhu<50) return "#FF6400";
    else if(suhu>=50) return "#FF0000";
}

function gambar(){
    var c =
document.getElementById("myCanvas");
    var a = c.getContext("2d");
    var i = 0;
    var w = 20;
    a.lineWidth="5";

    a.beginPath();
    a.moveTo(0, 30);
    a.lineTo(0, 400);
    a.lineTo(400,400);
    a.lineTo(400,30);
    a.lineTo(0,30);
    a.strokeStyle="#000000";
    a.stroke();

    a.font = "20px Arial";
    a.fillStyle = "red";
    //a.textAlign = "center";
    a.fillText("Peta Panas", 145, 50);

    a.font = "16px Arial";
    a.fillStyle = "black";
    a.fillText("Ket dalam C", 305, 290);

    a.beginPath();
    a.moveTo(350, 300);
    a.lineTo(395, 300);
    a.strokeStyle="#00FF00";
    a.stroke();

```

```
a.font = "15px Arial";
a.fillStyle = "black";
a.fillText("<30 :", 300, 305);

a.beginPath();
a.moveTo(350, 320);
a.lineTo(395, 320);
a.strokeStyle="#64FF00";
a.stroke();

a.font = "15px Arial";
a.fillStyle = "black";
a.fillText("<35 :", 300, 325);

a.beginPath();
a.moveTo(350, 340);
a.lineTo(395, 340);
a.strokeStyle="#FFFF00";
a.stroke();

a.font = "15px Arial";
a.fillStyle = "black";
a.fillText("<40 :", 300, 345);

a.beginPath();
a.moveTo(350, 360);
a.lineTo(395, 360);
a.strokeStyle="#FF6400";
a.stroke();

a.font = "15px Arial";
a.fillStyle = "black";
a.fillText("<50 :", 300, 365);

a.beginPath();
a.moveTo(350, 380);
a.lineTo(395, 380);
a.strokeStyle="#FF0000";
a.stroke();
```

```

    a.font = "15px Arial";
    a.fillStyle = "black";
    a.fillText(">=50 :", 295, 385);

    a.lineWidth="20";

    while(i<5){
        a.beginPath();

        a.arc(200,200,40+w*i,15/10*Math.PI,17/10
*Math.PI);
        a.strokeStyle= warna(simpan[i][0]);
        a.stroke();

        a.beginPath();

        a.arc(200,200,40+w*i,17/10*Math.PI,19/10
*Math.PI);
        a.strokeStyle= warna(simpan[i][1]);
        a.stroke();

        a.beginPath();

        ... ..

        a.arc(200,200,40+w*i,13/10*Math.PI,15/10
*Math.PI);
        a.strokeStyle= warna(simpan[i][9]);
        a.stroke();

        i=i+1;
    }
}

```



## BIODATA PENULIS



Hendra Irawan lahir di Merauke pada 16 Januari 1994 yang merupakan anak pertama dari dua bersaudara dari pasangan Miswanto dan Mahmudah. Penulis menyelesaikan Pendidikan dasar di SDN 1 Pengambengan dan melanjutkan Pendidikan menengah di SMPK Yos Sudarso dan SMAN 1 Tualang. Setelah menyelesaikan Pendidikan menengah, penulis mengenyam pendidikan tinggi di Diploma III Tehnik Elektro Universitas Gadjah Mada. Setelah lulus Pendidikan Diploma III pada tahun 2015, penulis melanjutkan pendidikan di Institut Teknologi Sepuluh Nopember (ITS) Surabaya sebagai mahasiswa lintas jalur.

Email:

[Hendrainrawan160194@gmail.com](mailto:Hendrainrawan160194@gmail.com)

*(sengaja dikosongkan)*