IMPLEMENTASI SISTEM PENDETEKSI KETINGGIAN AIR DENGAN MENGGUNAKAN WIRELESS SENSOR NETWORK NODE POINT TO POINT

SKRIPSI
KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Moch Rizki Cahyadi
NIM: 135150301111041

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

PENGESAHAN

IMPLEMENTASI SISTEM PENDETEKSI KETINGGIAN AIR DENGAN MENGGUNAKAN WIRELESS SENSOR NETWORK NODE POINT TO POINT

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Moch. Rizki Cahyadi
NIM: 135150301111041

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Sabriansyah Rizqika A., S.T,M.Eng	Edita Rosana W. , S.T., M.T., M.Eng
NIP: 19820809 201212 1 004	NIK: 201606 910626 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 1 Agustus 2017

Moch. Rizki Cahyadi

NIM: 135150301111041

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memeberikan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul "Implementasi sistem pendeteksi ketinggian air dengan menggunakan wireless sensor network point-to-point".

Banyak kesulitan dan hambatan yang dialami penulis dalam penyusunan skripsi ini, tetapi semua itu dapat diatasi berkat dukungan dan bantuan dari berbagai pihak. Oleh karena itu penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua serta keluarga penulis yang telah memberikan doa dan dukungannya.
2. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku dosen pembimbing pertama yang telah memberikan pengarahan dan bimbingan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibu Edita Rosana W. , S.T., M.T., M.Eng selaku dosen pembimbing kedua yang telah memberikan pengarahan dan bimbingannya kepada penulis sehingga dapat menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
5. Seluruh civitas akademika Fakultas Ilmu Komputer Univeristas Brawijaya yang telah memberikan bantuan dan dukungan selama penulis menempuh studi dan penyelesaian skripsi
6. Seluruh teman-teman angkatan 2013 dan pihak yang telah membantu yang tidak dapat penulis sebutkan satu per satu.
7. Keluarga lulus bareng yang telah menemani proses pengerjaan skripsi.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih.

Malang, 1 Agustus 2017

Penulis

mochrizkic@gmail.com

ABSTRAK

Kondisi daerah aliran sungai saat ini semakin memprihatinkan. Selain adanya pendangkalan yang disebabkan kikisan tanah pada sungai terjadi pula penyempitan yang disebabkan oleh adanya bangunan ilegal yang dibangun di daerah aliran sungai. Kondisi ini menyebabkan daya tampung air sungai menjadi kecil sehingga menyebabkan air sungai meluap dan terjadi banjir. Teknologi yang ada saat ini memungkinkan kita dapat mengetahui akan terjadinya banjir lebih cepat. Penggunaan sensor ultrasonik untuk melakukan pendeteksian ketinggian air dapat dilakukan untuk mengetahui peningkatan ketinggian air di sungai. Setelah data ketinggian diketahui pengiriman data perlu dilakukan agar data tersebut dapat ditampilkan pada komputer. Pengiriman satu data menuju satu komputer ini menjadi tidak efektif karena banyaknya perangkat yang ditempatkan pada komputer. Metode *routing point-to-point* dapat menjadi solusi untuk mengatasi permasalah tersebut. Melalui metode ini pengguna tidak perlu melakukan monitoring satu data dari satu node. Data ketinggian sungai akan dikirimkan oleh node-node menuju gateway sehingga hanya memerlukan satu perangkat yang tersambung dengan komputer. Penelitian ini mengimplementasikan *routing point-to-point* sebagai metode komunikasi sistem pendeteksi ketinggian air sungai menggunakan modul komunikasi nRF24L01. Sistem dirancang agar dapat mengirimkan data yang didapatkan oleh node menuju gateway. Gateway akan menerima data-data dari node tersebut kemudian akan dibuat program untuk menampilkan data tersebut dalam bentuk grafik pada komputer. Dari hasil pengujian yang dilakukan tingkat akurasi dari sensor ultrasonik lebih dari 89%. Pembacaan data pada prototipe memilki ketepatan 80%. Pengiriman data memiliki akurasi 100%. Hasil pengujian fungsional sistem dapat menampilkan grafik yang mengindikasikan berfungsinya fitur mqtt dan *websocket* pada sistem.

Kata kunci: banjir*, routing point-to-point*, monitoring sungai, nRF24L01, sensor ultrasonik

ABSTRACT

The current condition of the watershed is increasingly alarming. In addition to the silting caused by the erosion of soil in the river, there is also a narrowing caused by the illegal buildings built in the watershed. This condition causes the capacity of the river water to be small, causing the river water to overflow and flooding. The current technology allows us to know the floods faster. Ultrasonic sensors can perform water level detection, that can be done to determine the increase in water level in the river. Once the altitude data is known the data needs to be sent so that data can be displayed on the computer. Sending one data to one computer becomes ineffective because of the number of devices placed on the computer. The point-to-point routing method can be a solution to solve the problem. Through this method the user does not need to monitor one data from one node. River height data will be sent by the nodes to the gateway so that only requires one device connected to the computer. This research implements point-to-point routing as communication method of river water level detection system using nRF24L01 communication module. The system is designed to be able to transmit data obtained by the node to the gateway. The gateway will receive data from the node and then the program will be created to display the data in graphical form on the computer. From the results of tests conducted level of accuracy of ultrasonic sensors more than 89%. The data reading on the prototype has an accuracy of 80%. Data delivery has 100% accuracy. System functional testing results can display graphs that indicate the functioning of mqtt and websocket features on the system.

Keywords: flooding, routing point-to-point, river monitoring, nRF24L01, ultrasonic sensor

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang, masalah yang harus diselesaikan, tujuan yang harus dicapai, manfaat yang akan didapatkan, batasan permasalahan yang diangkat serta sistematika pembahasan pada penelitian yang akan dilakukan.

1.1 Latar Belakang

Indonesia merupakan Negara kepulauan terbesar di Dunia yang memiliki 5 pulau terbesar. Pulau Jawa merupakan salah satu pulau besar di Indonesia dengan penduduk yang sangat banyak karena pulau ini menjadi jantung perekonomian Indonesia khususnya di Jakarta yang menjadi Ibu kota Negara. Banyaknya orang yang merantau ke Jakarta menyebabkan Jakarta semakin memiliki banyak penduduk. Banyaknya penduduk tersebut tidak sesuai dengan kapasitas pemukiman yang ada. Kebanyakan perantau tersebut membuat rumah-rumah atau pemukiman yang ilegal dan membuat pemukiman didaerah aliran sungai (BPADJakarta, 2015). Pulau jawa memiliki banyak dataran tinggi yang berupa bukit serta pegunungan. Pegunungan tersebut menjadi tempat penampungan air selama musim hujan berlangsung. Sehingga air akan mengalir melalui sungai untuk kembali ke laut. Namun, karena maraknya peralihan dari hutan menjadi perkebunan atau pemukiman warga menyebabkan air yang harusnya diserap akan dialirkan ke sungai dan menambah volume air di daerah aliran sungai (Kompas.com, 2017).

Aliran air yang cepat dapat menyebabkan dinding sungai terkikis. Kikisan dari dinding sungai ini akan terbawa dan mengendap di berbagai tempat. Akibatnya, sungai yang asalnya dalam akan menjadi dangkal. Sungai yang dangkal mengurangi daya tampung terhadap air yang akan melewati sungai tersebut. Selain pendangkalan pengurangan daya tampung sungai juga terjadi karena adanya bangunan yang dibuat di samping sungai yang menyebabkan penyempitan aliran sungai. Selain penyempitan dengan adanya bangunan ini membuat air yang harusnya tertahan sebagian dan masuk ke dalam tanah tidak dapat dilakukan dan akan mengalir semuanya. Daya tampung sungai yang semakin menyempit ini akan menyebabkan aliran sungai meluap dan terjadi banjir (Kompas.com, 2016).

Kondisi peningkatan air ini dapat dideteksi dengan menggunakan sensor ultrasonik. Kondisi sungai di wilayah hilir seperti Jakarta berarus tidak deras karena biasanya aliran sungai berkelok-kelok (ilmugeografi.com, 2016). Sehingga tidak terjadi gelombang-gelombang air seperti daerah hulu. Sehingga sensor ultrasonik dapat digunakan sebagai pendeteksi ketinggian air sungai dengan cukup efektif. Salah satu penelitian yang menggunakan sensor ultrasonik adalah penelitian yang dilakukan oleh (Martalia, 2016). Menurut penelitian yang dilakukan Martalia (2016) sensor ultrasonik HC-SR04 dapat digunakan sebagai sensor untuk mendeteksi level ketinggian air. Pada penelitian ini level ketinggian air yang dideteksi ada 4. Hanya saja pada penilitian ini data yang didapatkan langsung ditampilkan pada tempat tersebut dan tidak melakukan pengiriman data. Dengan adanya pengiriman data maka akan mempermudah untuk melakukan monitoring

terhadap ketinggian air dan mendeteksi bencana banjir. Sehingga pengawasan dapat dilakukan dimana saja.

Berdasarkan latar belakang diatas penulis mengangkat skripsi dengan judul "IMPLEMENTASI SISTEM PENDETEKSI KETINGGIAN AIR DENGAN MENGGUNAKAN WIRELESS SENSOR NETWORK NODE POINT-TO-POINT" setelah data didapatkan oleh sensor ultrasonik HC-SR04. Data akan dikirimkan dengan menggunakan modul transceiver nRF24L01. Pengiriman data dikirimkan dengan *routing point-to-point*. pengiriman data dilakukan dari node-node menuju gateway. Penggunaan modul komunikasi nRF24L01 memungkinkan sistem yang dibuat memiliki harga yang lebih murah dan modul nRF24L01 juga mudah ditemui.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, terdapat beberapa rumusan masalah yaitu sebagai berikut:

1. Bagaimana sistem pendeteksi ketinggian air dengan *wireless sensor network node point-to-point* dapat di implementasi?
2. Bagaimana performansi sistem dari segi akurasi *sensing* data?
3. Bagaimana data didapatkan setiap node?
4. Bagaimana menerapkan routing point-to-point pada nRF24L01?
5. Bagaimana data dapat dikirimkan node ke gateway?

1.3 Tujuan

Tujuan yang ingin dicapai pada penelitian ini adalah :

1. Membuat sistem pendeteksi ketinggian air yang mengimplementasikan pengiriman data secara point-to-point pada wireless sensor network.
2. Mengetahui kelayakan sensor ultrasonik sebagai sensor ultrasonik sebagai pendeteksi ketinggian air.
3. Dapat membuat node yang dapat mendeteksi ketinggian air.
4. Dapat membuat sistem monitoring yang menggunakan protokol *routing point-to-point* sebagai protokol pengiriman data menggunakan modul komunikasi nRF24L01.
5. Membuat node yang dapat melakukan pengiriman data menuju gateway.

1.4 Manfaat

Manfaat yang ingin dicapai adalah sebagai berikut:

1. Bagi penulis

- Dapat bermanfaat bagi kemajuan teknologi pada bidang komunikasi jaringan nirkabel
- Dengan mengimplementasikan *routing point-to-point* pada pendeteksian ketinggian air menggunakan modul nrf24l01 sebagai modul komunikasi dapat menjadi solusi untuk membantu kehidupan manusia, dimana pendeteksian ketinggian air dapat dilakukan ditempat yang jauh.

2. Bagi mahasiswa lain

   Manfaat untuk mahasiswa lainnya yaitu diharapkan dengan topik ini mahasiswa dapat lainnya dapat mengembangkan penelitian lebih lanjut tentang sistem pedeteksi ketinggian air ini menggunakan *routing* lainnya atau metode lainnya.

1.5 Batasan Masalah

Untuk menghindari pembahasan yang terlalu luas, maka dibuat batasan masalah agar memudahkan dan lebih fokus dalam perancangan dan implementasi. Pembahasan masalah tersebut diantaranya adalah:

1. Sistem yang dibuat adalah sistem pengiriman serta pengolahan data ketinggian air
2. Tahapan *routing* adalah proses pembentukkan node.
3. Grafik dibuat secara statik dengan menggunakan 3 grafik.
4. Pengujian dilakukan menggunakan talang air.

1.6 Sistematika Pembahasan

Sistematika penulisan penulisan yang ditunjukan agar mencapai tujuan diharapkan secara garis besar meliputi beberapa bab, yaitu antara lain:

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan tentang sistem pendeteksi ketinggian air dengan menggunakan *wireless sensor network* node *point-to-point*.

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Membahas tentang kajian pustaka terkait dengan penelitian yang telah ada untuk memperkuat penulisan tugas akhir ini dan dasar teori yang membahas hal-hal teknis yang nantinya diperlukan untuk pembuatan Sistem.

BAB 3 METODOLOGI PENILITIAN

Bab ini menjelaskan langkah-langkah yang dilkaukan pada penelitian mulai dari studi literatur, rekayasa kebutuhan, perancangan perangkat keras dan lunak, implementasi perangkat keras dan lunak serta pengujian dan analisis.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini membahas mengenai semua kebutuhan sistem yang diperlukan dalam merancang sistem pendeteksi ketinggian air dengan *wireless sensor node point-to-point* seperti kebutuhan perangkat keras, perangkat lunak serta kebutuhan fungsional sisem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan mengenai perancangan penelitian seperti perancangan perangkat keras dan perancangan perangkat lunak pada gateway, node dan perancangan pembuatan grafik, serta implementasi perangkat keras dan lunak pada gateway, node dan implementasi pembuatan grafik.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan mengenai langkah pengujian yang dilakukan dalam sistem baik pengujian fungsional mapun pengujian nonfungsional sistem, serta menyajikan analisis dari data hasil pengujian yang dilakukan.

BAB 7 PENUTUP

Bab ini berisi tentang kesimpulan dari hasil pengujian dan analisis yang dilakukan pada sistem, serta saran mengenai system yang telah selesai dibuat serta dianalisis.

BAB 2 KAJIAN PUTAKA DAN DASAR TEORI

Pada bab ini berisi pembahasan kajian pustaka dan dasar teori yang mendukung dan berhubungan dengan sistem yang akan dibangun. Teori yang dibahas antara lain gelombang, arduino, pengujian system dan pengujian akurasi data.

2.1 Kajian Pustaka

Kajian pustaka ini membahas penelitian yang sudah ada dan berkaitan dengan penelitian yang diusulkan. Pada penelitian ini ada kajian pustaka diambil dari beberapa penelitian yang pernah dilakukan.

2.1.1 Kalibrasi Sensor Ultrasonik Hc-Sr04 Sebagai Sensor Pendeteksi Jarak Pada Prototipe Sistem Peringatan Dini Bencana Banjir

Penelitian untuk mengetahui kondisi bencana banjir lebih cepat yang dilakukan oleh Martalia (2016). Sensor yang digunakan pada penelitian ini adalah sensor HC-SR04. Sensor ini merupakan sensor yang biasa digunakan untuk mendeteksi jarak. Pada penelitian ini dilakukan 2 kali proses pengujian dan menggunakan 2 buah sensor. penelitian tersebut dirancang dan dikembangkan agar dapat diterapkan untuk peringatan dini saat akan terjadi bencana banjir.

Prototipe sistem peringatan dini bencana banjir yang diteliti ini didapatkan hasil bahwa sensor HC-SR04 layak untuk digunakan pada sistem peringatan dini bencana banjir. pada penelitian ini menggunakan 4 status yaitu dari siaga 1-4 yang berdasar pada BPBD (Badan Penanggulangan Bencana Daerah). Pembacaan dilakukan setiap 2 detik dan dilakukan 20 kali pembacaan data.

Kekurangan dari penelitian ini adalah tidak adanya pengiriman data. Data yang didapatkan langsung ditampilkan pada serial monitor. Selain itu pada penelitian ini tidak dijelaskan pula bagaimana sensor HC-SR04 mendapatkan jarak dari segi sensor. Serta rumus yang digunakan disesuaikan dengan skema pengukuran yang dilakukan.

2.1.2 Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensornets

Penelitian ini dilakukan untuk mengetahui *scalabilitas* dari *routing point-to-point*. Penelitian ini dilakukan oleh Fronseca dkk (2005). Pada penelitian ini telah dijelaskan pada saat penelitian dilakukan penggunaan *routing* ini masih jarang. Adapun yang telah dilakukan tetapi tidak ditemukan dokumentasi hasilnya sehingga tidak dapat diketahui kemampuan yang sebenarnya. Implementasi penggunaan *point-to-point* lebih kepada fungsionalitas dan memiliki sekala yang kecil. Penelitian ini berfokus kepada penerapan yang lebih luas dan dapat diketahui kemampuan yang sebenarnya.

Penelitian ini mengemukakan tentang penggunaan BVR (*Beacon Vector Routing*) *routing* ini memberikan *scalabilitas* yang lebih tinggi dan memberikan

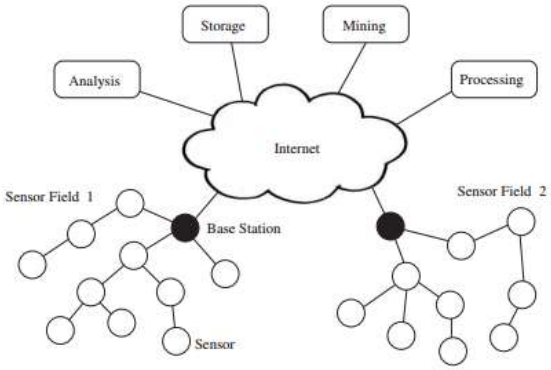
routing state yang lebih kecil dibandingkan dengan algoritma *sortest path*. BVR menggunakan *greedy* untuk *forwarding* antar kordinat node, namun tidak membutuhkan informasi geografis membuat tidak ada asumsi tentang konektivitas radio, dan menggunakan algoritma pembangunan kordinat yang sangat mudah. Akan tetapi pada penelitian ini yang digunakan adalah routing *point-to-point* dasar tidak sampai menggunakan routing BVR.

2.2 Dasar Teori

Dasar teori ini akan membahas mengenai referensi dan dasar teori yang mendasari tentang *routing point-to-point* pada NRF24L01 beserta kebutuhan perangkat yang digunakan.

2.2.1 Wireless Sensor Network

Wireless Sensor Network (WSN) adalah sekumpulan sensor yang terhubung dengan kontroller dan *processing station*. Penambahan jumlah sensor yang berkomunikasi mengkoleksi data secara wireless menuju sebuah *processing station* yang terpusat. Ini sangat penting sejak sangat banyaknya aplikasi jaringan yang membutuhkan ratusan hingga ribuan sensor nodes, sering diguanakan di daerah terpencil dan area yang memiliki akses yang sulit. Bagaimanapun, sebuah *wireless sensor* tidak hanya sebuah komponen *sensing*, akan tetapi juga merupakan *on-board processing*, komunikasi dan kemampuan untuk menyimpan data. Sebuah sensor node tidak hanya bertanggung jawab dalam pengambilan data saja, tetapi juga pada analisa jaringan, mengkorelasi dan menggabungkan data dari nodenya sendiri maupun node lainnya. Ketika sensor bergabung dan melakukan monitoring terhadap lingkungan fisik yang luas, mereka membentuk sebuah *wireless sensor network* (WSN). Selain berkomunikasi dengan node lainnya node dapat berkomunikasi juga dengan *Base station* (BS) menggunakan radio *wireless*, membuat mereka dapat melakukan proses *remote* data sensor, memvisualisasikan, menganalisis dan membuat sistem penyimpanan. Sebagai contoh dapat dilihat pada Gambar 2.1 yang menunjukkan dua lahan dua lokasi geografis yang berbeda dan menyambungkan internet menggunakan *base station*.



Gambar 2.1 Wireless Sensor Network

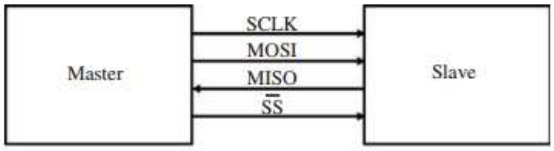
Sumber : (Dargie & Poellabauer, 2010)

Kemampuan sensor node pada sebuah WSN dapat sangat luas. secara mudah sensor node dapat melakukan monitoring terhadap sebuah penomena fisik. Ketika semakin kompleks perangkat akan mengkombinasikan banyak tehnik sensing yang berbeda. mereka juga memiliki banyak cara untuk berkomunikasi, seperti menggunakan *ultrasound*, *infrared* atau teknologi radio yang memiliki data *rate* dan *latencies* yang berbeda. Ketika simple sensor hanya dapat mengambil data dan mengkomunikasikan informasi mengenai hasil observasinya, akan menjadi perangkat yang kuat juga dapat melakukan proses yang luas dan kumpulan fungsi. Termasuk perangkat yang sering diasumsikan memiliki tambahan kemampuan pada sebuah WSN, sebagai contoh mereka dapat membentuk komunikasi *backbones* yang dapat di gunakan oleh sumber perangkat sensor terbatas lainnya untuk dapat mencapai *base station*. akhirnya beberapa perangkat dapat menambahkan tekonologi pendungkung, sebagai contoh, *Global Positioning System* (GPS) *receivers*, yang memberikan mereka lokasi yang sangat akut (Dargie & Poellabauer, 2010)

2.2.2 Serial Peripheral Interface

Serial Periperal Interfacce (SPI) adalah *high-speed, full-duplex synchronous serial bus.* SPI diciptakan di Motorola pada pertengahan 1980. SPI tidak memiliki sebuah standar khusus, tapi pembuatan *device* yang menggunakan SPI harus memiliki kesesuaian saat implementasi spesifikasi dari produk lain untuk menciptakan komukinikasi yang benar (contohnya, *device* harus mennyetujui dalam mengirim akan menggunakan *most significant bit* (MSB) atau *Least Significant bit* (LSB). SPI bus didefinisikan oleh empat pin: (*Master-Out/Slave-In*) MOSI, (*Master-In/Slave-Out*) MISO, (*Serial Clock*) SCLK/SCK, dan (*Chip Select*) CS. CS terkadang mereferensi sebagai (*Slave Select*) SS (Dargie & Poellabauer, 2010). MOSI digunakan untuk melakukan transfer data dari *master* ke *slave* saat *device* diconfigurasikan sebagai sebuah *master*. Pada kasus konfigurasi *device* sebagai *slave* port ini digunakan untuk menerima data dari sebuah *master*. SCK digunakan *master* untuk mengirimkan sinyal clock yang di butuhkan untuk sinkronisasi pengiriman.

Sambungan SPI pada sebuah *Master* dan *Slave* ditunjukkan pada Gambar 2.2**.** Untuk memulai pengiriman data dari *Master* ke *Slave* menggunakan *Clock* yang dikirim master melalui port SCK. Penggunaan *Clock* ini adalah sebagai tanda pengiriman data serta sarat sinkronisasi data pada saat *Master* mengirim ke *Slave* pada port MOSI. Pada saat *Slave* mengirim data ke *Master* pada port MISO menggunakan *Clock* ini juga. Sistem ini akan digunakan untuk pengiriman data pada sambungan antara NRF24L01 sebagai *slave* dan Arduino Nano sebagai *master*.



Gambar 2.2 Single Master Single Slave

Sumber : (Dargie & Poellabauer, 2010)

2.2.3 Point-to-Point Routing

Pada *Wireless Sensor Network* komunikasi antar gateway dan node dapat menggunakan rute yang telah ditentukan. Ketika node *dideploy* pada gaya acak. Misalnya saat penempatan node dilakukan secara acak. Dalam kasus ini topologi yang terbentuk akan menjadi tidak beraturan. Pada hal ini node harus dapat melakukan *self-organize.* Harus bekerja sama dalam penempatan posisi, mengenali tetangga dan mengetahui jalan menuju gateway (Dargie & Poellabauer, 2010). Dalam sistem ini node berkomunikasi dengan cara *point-to-point*.

Point-to-point routing merupakan *routing* yang melakukan komunikasi secara langsung dengan node-node yang ada. *Routing* ini dapat diaplikasikan dengan berbagai cara serta pengorbanan tertentu dalam skalabilitas. Memiliki satu bagian kelas algoritma yang tersekala dengan baik, hal ini karena node hanya perlu mengetahui alamat dari tetangganya (J Ortiz:2007). Setiap node hanya akan berkomunikasi dengan kurang dari dua node. Yang mana kedua node tersebut adalah tetangga dari node utama. *Routing* ini mendefinisikan node baru dengan mencari satu node baru. *Routing* ini sangat cocok digunakan pada system yang memiliki cakupan wilayah yang cukup besar. Pada sistem ini setiap node berkomunikasi dengan satu node tetangganya untuk penerimaan data dan satu node tetangganya untuk pengiriman data.

2.2.4 Gelombang ultrasonik

Gelombang bunyi adalah gelombang *longitudinal* yang dapat merambat melalui gas, zat padat maupun cair dengan kecepatan bergantung pada sifat elastis dan sifat inersia medium rambat. Gelombang bunyi yang dapat didengar manusia memiliki frekuensi antara 20 Hz sampat dengan 20 KHz. Gelombang bunyi di bawah 20 Hz disebut infrasonik, sedangkan gelombang bunyi diatas 20 KHz disebut gelombang Ultrasonik.

Gelombang ultrasonik yang melaui medium mengakibatkan adanya getaran partikel dengan medium amplitude sejajar dengan arah rambat secara longitudinal sehingga menyebabkan bentuk rapatan (*strain*) dan tegangan (*stress*). (MAJDI, 2013). Medium yang umumnya digunakan sebagai perabatan gelombang ultrasonik adalah udara dan air. Kecepatan rambatan gelombang ultrasonic dipengaruhi oleh karakteristik medium yang ia lewati yaitu suhu, kelembaban dan tekanan.

Kecepatan rambat gelobang dapat kita hitung dengan rumus berikut ini.

$$V = \lambda f \quad (2.1)$$

V = Cepat Rambat Gelombang Ultrasonik (m/s)

λ = Panjang Gelombang (m)

f = Frekuensi (Hertz)

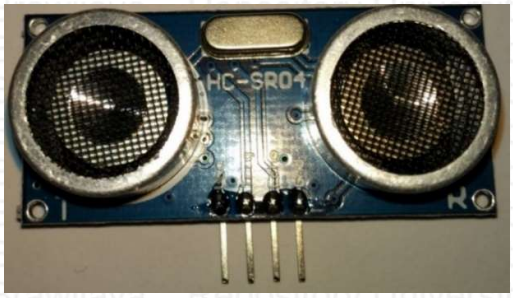
Tabel 2.1 merupakan contoh kecepatan rambat gelombang ultrasonik dalam beberapa medium.

Tabel 2.1 Kecepatan Bunyi Dalam Beberapa Medium

Medium	Cepat Rambat Bunyi (M/s)
Udara ($0^o C$)	331
Udara ($15^o C$)	340
Air ($25^o C$)	1490
Air Laut ($25^o C$)	1530
Aluminium ($20^o C$)	5100
Tembaga ($20^o C$)	3560
Besi ($20^o C$)	5130

2.2.5 Sensor HC-SR04

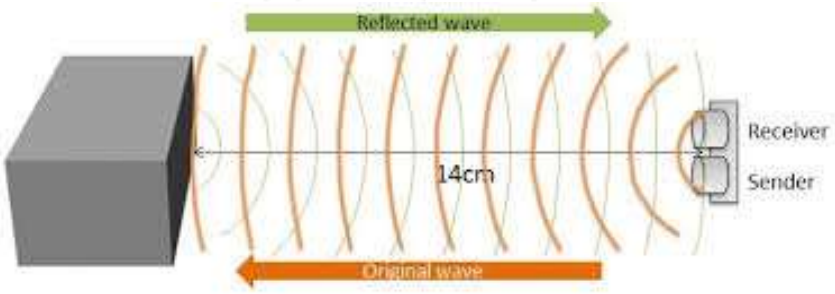
Sensor HC-SRF04 adalah seri dari sensor jarak dengan gelombang ultrasonik. Dalam sensor ini terdapat 2 bagian penting yang berfungsi sebagai transmitter dan receiver. Sensor ini memiliki 4 port yaitu VCC, *Ground, Echo* dan *Trigger*. Berikut Gambar 2.3 dari sensor HC-SR04.



Gambar 2.3 Sensor HC-SR04

Cara kerja sensor ini adalah menembakkan gelombang ultrasonik dan menunggu gelombang tersebut kembali diterima oleh *receiver*. Jarak dapat ditentukan dengan sensor ini dengan cara menghitung waktu ketika sensor melakukan pengiriman dan gelombang tersebut diterima. Gelombang yang ditransmisikan akan mematul kembali ketika menabrak suatu benda dan saat gelombang tersebut diterima maka dapat dihitung jarak dengan menghitung berapa lama gelombang tersebut ditembakkan dan diterima. Serta kecepatan dari gelombang tersebut. Atau dapat diketahui dengan rumus untuk menentukan

jarak, yaitu $S = Vt$. Cara kerja gelombang ultrasonic diperlihatkana pada Gambar 2.4 .



Gambar 2.4 Cara kerja gelombang ultrasonic

Sumber: (Teachengineering, 2013)

2.2.6 Arduino IDE

Arduino IDE adalah IDE atau sebagai perangkat lunak untuk dapat melakukan pengaturan pada Board atau mikrokontroller arduino. Dengan perangkat lunak ini kita dapat melakukan pemrograman ke *board* yang akan kita gunakan. Dalam penilitian ini board yang digunakan adalah mikroontroller Arduino Uno sehingga aplikasi arduino IDE yang kita gunakan sebagai alat untuk melakukan *upload* atau memasukkan baris perintah yang akan dieksekusi oleh mikrokontroller tersebut. Dengan Arduino IDE ini kita dapat memberikan perintah terhadap mikrokontroller agar melakkan *sensing* data atau melakukan pengiriman data dengan perangkat keras lain. Perangkat keras lain yang akan di tambahkan pada mikrokontroller biasanya memiliki *library* yang harus di masukkan agar mikrokontroole dapat mengakses perangkat keras tambahan yang digunakan. Dalam sistem ini perangkat keras tambahan yang menggunakan *library* adalah NRF24L01 dengan menggunakan *library* RF24.

Library RF24 ini memiliki fungsi-fungsi yang digunakan agar mikrokontroller dapat melakukan pengiriman, penerimaan atau perintah lainnya yang dapat dilakukan modul *tranceiver* NRF24L01. Dengan kata lain untuk menjalankan fitur NRF24L01 diperlukan *library* RF24 ini. Selain RF24 masih banyak *library* lainnya yang dapat digunakan. Namun, pada sistem ini hanya menggunakan *library* RF24. *Library* ini hanya dapat mengirimkan data sebanyak 32 byte dalam sekali pengiriman.

2.2.7 Arduino NANO

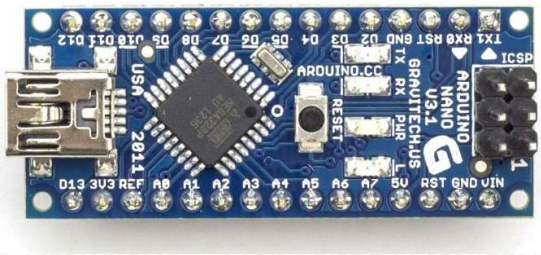
Mikrokontroller yang digunakan untuk penelitian ini adalah Arduino Nano. Arduino Nano adalah sebuah perangkat keras yang dapat melakukan beberapa intruksi yang sebelumnya kita telah rancang. Dengan melakukan *upload* menggunakan arduino IDE ke board ini. Spesifikasi Arduino Nano tertera pada Tabel 2.2.

Tabel 2.2 Spesifikasi Arduino Nano

Microcontroller	**Atmel ATmega168 or ATmega328**
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (*of which 6 provide PWM output*)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Sumber : (Arduino, 2016)

Arduino Nano dihubungkan dengan komputer dengan menggunakan kabel data mini usb. Gambar dari Arduino Nano ditunjukkan pada Gambar 2.5 .



Gambar 2.5 Arduino Nano

Sumber: (Arduino, 2016)

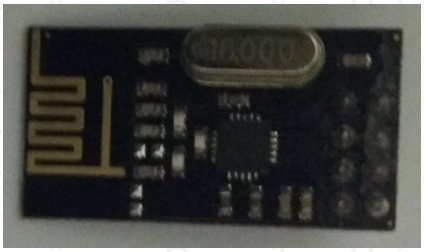
2.2.8 NRF24L01

Modul nRF24L01 adalah modul komunikasi jarak jauh yang memanfaatkan gelombang RF 2.4 GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan tampilan SPI untuk Berkomunikasi. Berikut spesifikasi dari nrf24l01 menurut *datasheet*.

- Memiliki 3 jenis datarate yaitu 250 Kbps, 1 Mbps dan 2 Mbps.
- Validasi dan deteksi paket otomatis
- Panjang playload dinamik 1 – 32Bytes.
- Mengirim ulang otomatis

- 6 Multireceiver data pipe
- Menggunakan interfase SPI untuk komunikasina
- Bekerja pada tegangan 1.9 sampai 3.6V
- Bekerja pada suhu -40 sampai +80 ^{o}C

nRF24L01 berhubungan dengan arduino secara *Serial Peripheral Interface* (SPI). Gambar nRF24L01 ditunjukkan pada Gambar 2.6 .



Gambar 2.6 nRF24L01

2.2.9 Library Python

Sejak dikenalkan oleh Guido van Rossum pada 1991, python telah berkembang menjadi salah satu yang paling luas digunakan untuk keperluan umum, bahasa pemrograman level tinggi dan didukung oleh salah satu komunitas *developer* terbesar. Python adalah bahasa pemrograman *open source* yang mendukung sangat banyak *library*. *library* ini adlah fitur terbaik yang dimiliki python. Dengan *library* ini kita dapat memperbanyak fitur serta menambah fungsi yang digunakan.

Semakin banyak *library* yang digunakan pada pemrograman python ini semakin luas pula jangkauan pemrograman yang dimiliki. Tentu saja kita tidak dapat memasukkan sembarang *library* pada program yang kita buat. Kita hanya akan menggunakan *library* yang memang dibutuhkan. Selain menambah fitur, *library* ini dapat pula digunakan untuk mempermudah proses *programming. Library* ini tersedia pada berbagai sumber atau bentuk biner gratis untuk semua *platform* utama, dan dapat didistribusikan dengan bebas (Desai, 2015)

Library Pyserial adalah *library* tambahan pada bahasa pemrograman python. *Library* ini harus diinstall terlebih dahulu untuk dapat digunakan. *Library* ini digunakan agar program yang kita buat dalam bahasa python dapat membaca data serial. Data serial ini didapat dari *board* arduino yang digunakan pada sistem ini.

Library TXWS (*Twisted Websocket*) adalah *library* python yang digunakan untuk menjalankan *websocket* server pada python. *Websocket* server ini akan diakses dari web untuk dapat diambil datanya. Txws ini juga mendukung pada versi *Websocket* terbaru.

Library paho-mqtt adalah *library* yang digunakan pada python agar dapat menggunakan fitur mqtt. Fitur tersebut adalah *publish* dan *subscribe*. Data yang didapatkan akan di-*publish* menuju *broker* dengan fungsi *publish* yang tersedia

pada *library* ini. Kemudian datanya dapat di-*subscribe* dengan fungsi *subscribe* yang ada pada *library* ini. Sedangkan untuk *broker* harus di-*install manual* dan berjalan sendiri tidak termasuk *library* python. *Broker* yang digunakan di sistem operasi Windows adalah *mosquitto*. Setelah menjalankan program *mosquitto* ini barulah kita dapat melakukan *publish* dan *subscribe*.

2.2.10 MQTT

MQTT (*Message Queuing Telemetry Transport*) adalah sebuah *messaging* protokol ringan yang menyediakan pembatasan sumberdaya jaringan *client* dengan mudah mendistibusikan informasi *telemetri*. Protokol ini menggunakan pola komunikasi *publish/subscribe*. protokol ini digunakan untuk komunikasi *machine-to-machine* (M2M) dan memainkan bagian terpenting dalam *Internet of Things* (IoT).

MQTT membiarkan perangkat untuk mengirim (*publish*) informasi yang diberikan topik ke sebuah server yang berfungsi sebagai MQTT *message broker*. *Broker* kemudian melakukan push informasi menuju *client* yang sebelumnya telah melakukan *subscribe* pada topik *client*. Topik ini terlihat seperti sebuah hirarki lokasi file. *Client* dapat melakukan *subscribe* ke level spesifik sebuah topik atau menggunakan karakter *wildcard* untuk *subscribe* ke banyak level (TechTarget, 2005).

2.2.11 Websocket

Websocket adalah protokol jaringan yang mendefinisikan bagaimana server dan client dapat berkomunikasi melalui web. protokol setuju terhadap aturan untuk berkomunikasi. *websocket* membuat komunikasi *real-time* lebih effisien. Juga dapat digunakan untuk *polling* melalui HTTP untuk menerima notifikasi pada HTTP, bagaimanapun *websocket* dapat menghemat *bandwith*, *cpu power* dan waktu respon. *Websocket* memungkinkan untuk mempermudah secara dramatis koneksi yang berorientasi pada komunikasi pada aplikasi *real-time* (Vannesa, et al., 2013).

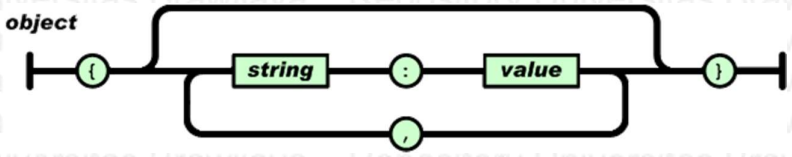
2.2.12 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON tidak bergantung pada bahasa pemrograman apapun karena penggunaan bahasa yang umum digunakan oleh programmer keluarga C, *java, JavaScript*, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, JSON menjadi ideal sebagai bahasa untuk pertukaran data. Dalam kasus ini pengiriman dilakukan dari bahasa pemrograman python menuju *JavaScript* (Json.org, 2013).

JSON dibangun dalam dua struktur yaitu sebuah kumpulan dari nama/nilai. Dalam berbagai bahasa, hal ini dinyatakan sebagai sebuah objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*) atau *associtive array.* Sebuah nilai terurutkan. Pada

kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vaktor (*vector*), daftar (*list*), atau urutan (*squence*). Struktur-struktur data ini disebut sebagai setruktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur dalam bentuk yang sama ataupun berbeda. JSON menggunakan bentuk sebagai berikut.

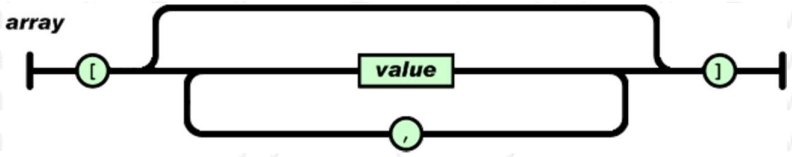
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan '{' (kurung kurawal buka) kemudian dengan nama yang diikuti dengan ':' (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh ',' (koma) kemudian diakhiri dengan '}' (kurung kurawal tutup).



Gambar 2.7 Struktur Object pada JSON

Sumber : (Json.org, 2013)

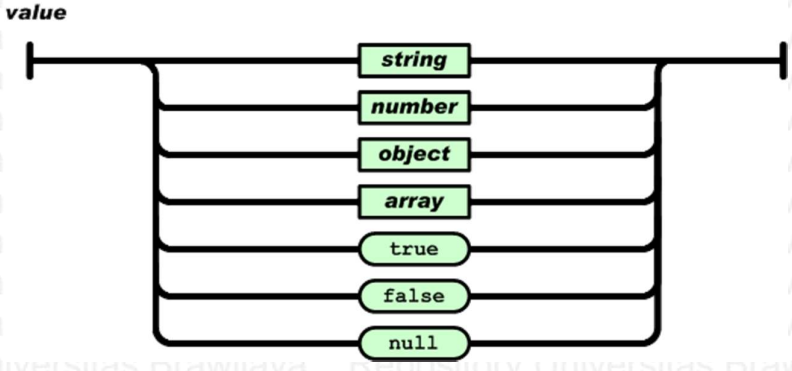
Array atau larik adalah nilai terurutkan. Larik dimulai dengan '[' (kurung kotak terbuka) kemudian diisi dengan nilai, setiap nilai dipisahkan menggunakan ',' (koma) kemudian diakhiri dengan ']' (kurung kotak tertutup.



Gambar 2.8 Struktur Array

Sumber : (Json.org, 2013)

Setiap data yang dikirimkan dalam bentuk objek dapat menggunakan semua nilai (*value*). Pada bagian value dapat menggunakan format data larik (*array*) atau nilai-nilai lainnya. Nilai dapat berupa string dalam tanda kutip ganda, angka, atau true atau false atau null, atau sebuah larik atau objek. Struktur tersebut dapat disusun bertingkat.



Gambar 2.9 Struktur Tingkatan nilai (*value*)

Sumber : (Json.org, 2013)

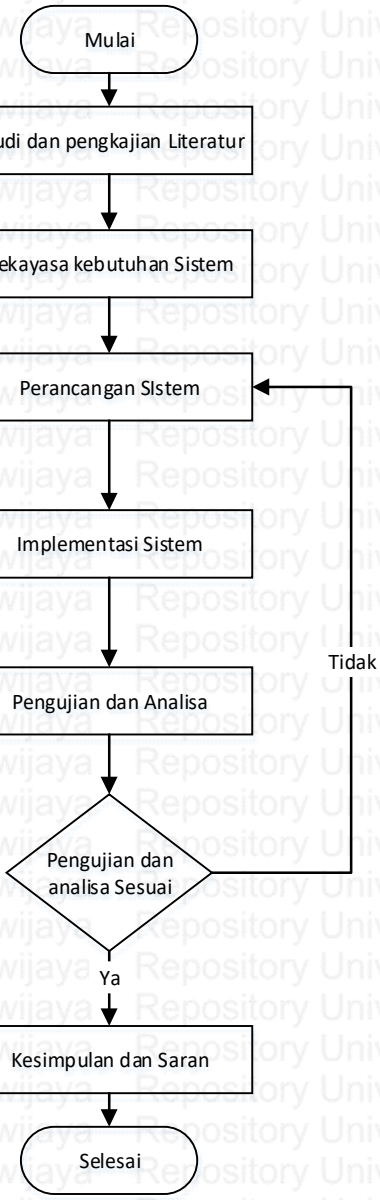
2.2.13 Google Chart

Google *chart* menyediakan cara yang sempurna untuk memvisualisasikan data pada *website*. Dari *chart* mudah sampai hirarki tree yang kompleks, *chart galery* menyediakan banyak tipe *chart* yang siap digunakan. Umumnya google *chart* digunakan dengan *simple JavaScript* yang di tanamkan pada *website*. Melakukan *load* pada sebagian google *chart library*, daftar data yang akan di buat grafik, pilih opsi untuk menyesuaikan grafik, dan buatlah sebuah chart objek dengan sebuah id. kemudian pada *web page* buat sebuah dengan id untuk menampilkan google *chart* (Google, 2017).

Untuk dapat menampilkan *chart* kita harus terkoneksi dengan internet. Hal ini dikarenakan kita memanggil *loader* yang berada pada server google. File ini bertipe *JavaScript*. Meskipun kita dapat membuka dan mengunduh file ini grafik tidak akan terbentuk dan memang harus *online*.

BAB 3 METODOLOGI PENELITIAN

Bab metodologi penelitian berisi penjelasan mengenai metodologi yang digunakan dalam penelitian. Gambar 3.1 merupakan diagram alir yang berisi tahapan-tahapan yang dilakukan dalam metodologi penelitian sistem monitoring kedalaman sungai sebagai pendeteksi pendangkalan sungai. Tahapan tersebut dimulai dari studi literatur, perancangan, implementasi, pengujian dan analisis serta pengambilan kesimpulan dan saran.



Gambar 3.1 Tahapan metodologi penelitian

3.1 Studi Literatur

Sudi literatur merupakan tahapan pencarian literatur yang akan menunjang penyusunan dasar teori dalam system. Literatur ini diperoleh dari buku, jurnal, dan *website* terkait yang meliputi.

1. Gelombang Ultrasonik adalah gelombang suara yang berada pada frekuensi diatas 20KHz. Gelombang ini tidak dapat didengar oleh manusia. Gelombang ini akan memantul ketika mengenai suatu objek yang lebih padat dari medium yang digunakan.
2. Sensor HC-SR04 adalah sensor pengukur jarak yang menggunakan gelombang ultrasonik. Cara kerja sensor ini adalah menembakkan gelombang ultrasonik kemudian diterima kembali oleh *receiver* ultrasonik. Waktu ketika sensor ditembakkan dan diterima dihitung dan dikonversi menjadi jarak.
3. Arduino IDE adalah perangkat lunak yang digunakan untuk melakukan pemrograman dan *upload* program yang kita buat menuju mikrokontroller atau mikrokomputer. Bahasa pemrograman yang digunakan pada IDE ini adalah bahasa C.
4. Arduino NANO adalah sebuah mikrokontoller yang dapat digunakan sebagai pemroses atau pengontrol dari sensor yang digunakan. Hasil dari sensor dapat diketahui dengan memasukkan program yang sesuai dengan sensor dan menghubungkan sensor dengan mikrokontroller arduino Nano ini.
5. Modul *transceiver* NRF24L01 adalah modul pengiriman data yang menggunakan gelombang radio untuk melakukan pengiriman datanya. Modul ini berhubungan dengan mikrokontroller secara *Serial Peripheral Interface* (SPI).
6. *Serial Peripheral Interface* adalah sebuah *interface bus* yang biasanya digunakan untuk melakukan pengiriman data antara mikrokontroller dan *peripheral* kecil termasuk *shift register*, sensor dan SD *cards*.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan adalah proses pengetahuan mengenai kebutuhan yang akan digunakan dalam penilitian. Kebutuhan ini harus dipenuhi agar sistem dapat berjalan sesuai dengan yang diinginkan. Bab ini juga menjelaskan kebutuhan-kebutuhan yang harus dipenuhi dalam kebutuhan fungsional dan non-fungsional. Dalam garis besar kebutuhan yang harus dipenuhi terbagi menjadi dua bagian yaitu pada sisi perangkat keras dan Perangkat lunak.

1. Kebutuhan perangkat keras
   a. Arduino Nano
   b. nRF24L01
   c. Sensor HC-SR04
   d. Komputer/Laptop

2. Kebutuhan perangkat lunak
   a. Operating Sytem windows

b. Arduino IDE

c. *Library* RF24

d. Python

e. *Library websocket* dan mqtt

f. GoogleChart

3.3 Perancangan

Perancangan merupakan tahapan yang dilakukan setelah seluruh kebutuhan dari sistem diperoleh dalam proses rekayasa kebutuhan. Perancangan implementsi sistem pendeteksi ketinggian air dengan *wireless sensor node point-to-point* dibagi menjadi perancangan perangkat keras dan lunak. Untuk masing-masing perancangan dilakukan perancangan pada perangkat gateway dan node.

3.3.1 Perancangan Perangkat Keras

Perancangan yang dilakukan meliputi perancangan perangkat keras pada gateway dan node.

1. Perancangan Gateway

Perancangan perangkat keras pada gateway dimulai dengan menyambungkan perangkat-perangkat yang akan digunakan. Perangkat itu adalah mikrokontroller arduino nano dan modul komunikasi nRF24L01. Gateway akan bertugas sebagai penerima keseluruhan data yang dikirimkan oleh node. Rancangan dari gateway pada system ini diperlihatkan pada Gambar 3.2 .

Gambar 3.2 Perancangan Gateway

2. Perancangan Node

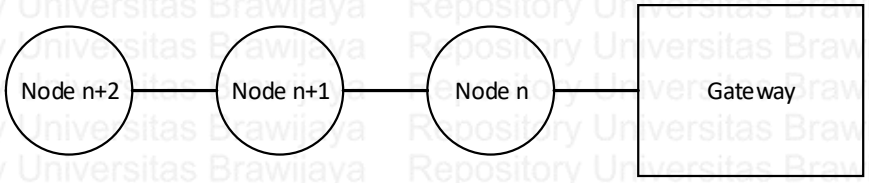
Perancangan perangkat keras pada node dimulai dengan menghubungkan perangkat keras yaitu mikrokontroller arduino nano, modul komunikasi nRF24L01 dan sensor HC-SR04. Dalam penilitian ini node yang digunakan memiliki spesifikasi dan perangkat keras yang sama. Node-node ini akan melakukan pengiriman data dari hasil pendeteksian sensor dan data yang dikirimkan oleh node lainnya menuju gateway. Diagram dari node-node yang akan dibuat pada sistem ini diperlihatkan pada Gambar 3.3 .

Gambar 3.3 Perancangan Node

3.3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak ini melputi perancangan yang dilakukan pada gateway dan node. Perancangan perangkat lunak dilakukan dengan pembuatan diagram alir. Dari diagram alir ini nantinya akan implementasikan menjadi bahasa pemrograman yang digunakan oleh mikrokontroller dan dijalankan. Perancangan perangkat lunak pada gateway dimulai dengan membuat diagram alir yang berfungsi untuk membentuk hubungan dengan node dan menerima data dari node. Perancangan perangkat lunak pada node dimulai dengan diagram alir yang berfungsi untuk membentuk hubungan dengan gate, membuat hubungan dengan node lainnya, melakukan *sensing* data, mengirimkan data dan menerima data.

Dari perancangan perangkat lunak yang dilakukan gateway dan node akan membentuk sebuah topologi. Topologi yang terbentuk dapat dilihat pada Gambar 3.4 . Topologi terbentuk dimulai dari gateway yang membentuk hubungan dengan node n. Kemudian node n membentuk hubungan dengan node selanjutnya sampai dengan node terakhir yang sesuai dengan perancangan perangkat lunak yang dilakukan .



Gambar 3.4 Topologi Sistem

3.4 Implementasi

Implementasi mengacu pada hasil perancangan yang telah dilakukan. Tahapan implementasi yang dilakukan meliputi implementasi perangkat lunak dan perangkat keras.

3.4.1 Implementasi Perangkat Keras

Implementasi perangkat keras yang dilakukan meliputi implementasi perangkat keras pada gateway dan node. Implementasi sistem dimulai dengan spesifikasi sistem. Setelah itu sistem dirangkai agar dapat terhubung satu sama lain

1. Implementasi perangkat keras gateway

Implementasi perangkat keras gateway dimulai dengan spesifikasi sistem. Perangkat yang digunakan pada gateway adalah modul komunikasi nRF24L01 dan mikrokontroller arduino nano. modul nRF24L01 dihubungkan dengan arduino nano sebagai perangkat untuk mengirimkan data.

2. Implementasi perangkat keras node

Implementasi perangkat keras node dimulai dengan spesifikasi sitem. Perangkat yang digunakan pada node adalah sensor ultrasonik HC-SR04, modul komunikasi nRF24L01 dan mikrokontroller arduino nano. Sensor HC-SR04

dihubungkan dengan arduino nano sebagai sensor pendeteksi ketinggian. Sedangkan untuk modul nRF24L01 dihubungkan dengan arduino nano sebagai perangkat untuk mengirimkan data.

3.4.2 Implementasi Perangkat Lunak

Implementasi yang dilakukan pada perangkat lunak meliputi implementasi perangkat lunak pada gateway dan node. Implementasi sistem dilakukan dengan mengimplementasikan kode program yang dibuat untuk gateway dan node. Kemudian mengimplementasikan routing yang telah dirancang sebelumnya. Dijelaskan juga mengenai batasan yang ada dalam implementasi sistem.

3.5 Pengujian dan Analisis

Pengujian dan analisis ini dilakukan pada sistem agar dapat diketahui apakah sistem telah berjalan sesuai dengan spesifikasi dari rancangan yang telah dibuat. Pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian Nilai sensor yang digunakan
2. Pengujian sensor pada lingkungan *prototype*
3. Pengujian pengiriman data oleh node
4. Pengujian monitoring pada *prototype*

Dari pengujian yang telah dilakukan, data yang diterima dari node akan diperiksa apakah telah sesuai serta data yang diterima pada gateway merupakan data dari tiap node. Serta akan diketahui pula bagaimana *routing* tersebut berjalan sesuai dengan apa yang diinginkan serta tidak adanya kesalahan atau pertukaran dari data yang diterima. Analisis hasil pengujian ini dilakukan agar dapat mengetahui apakah sistem berjalan sesuai dengan perancangan yang dilakukan.

3.6 Pengambilan Kesimpulan dan Saran

Kesimpulan terdiri dari ringkasan hasil pengujian dan analisa dari sistem yang dibuat dalam penelitian ini. Sehingga dapat pula membentuk saran untuk pengembangan sistem di masa yang akan datang.

BAB 4 REKAYASA KEBUTUHAN

Dalam bab ini menjelaskan persyaratan yang harus dipenuhi untuk perancangan hingga implementasi. Dengan harapan perancangan dan implementasi sistem pendeteksi ketinggian air dengan *wireless sensor node point-to-point* dapat berjalan dengan baik.

4.1 Deskripsi

Sistem yang dibuat penulis adalah sistem pengiriman data menggunakan modul komunikasi nRF24l01 dengan menggunakan *routing Point-to-Point*. Untuk penerapannya sendiri penulis menggunakan sistem ini untuk mengirimkan data ketinggian sungai yang didapat oleh node-node yang akan melakukan pengecekkan ketinggian sungai dengan sensor ultrasonik dan mengirimkan data yang didapatkan secara *point-to-point* dengan nRF24l01.

4.1.1 Tujuan

Tujuan dari sistem ini adalah nRF24L01 dapat melakukan pengiriman data sensor pada setiap node dengan menggunakan *routing Point-to-Point* untuk dapat mengirim data menuju gateway. Data sensor yang didapat adalah ketinggian air yang mana ketinggian air ini akan dibagi menjadi 4 level ketinggian.

4.1.2 Ruang Lingkup

Judul dari Skripsi ini adalah sistem pendeteksi ketinggian air dengan *wireless sensor node point-to-point*.

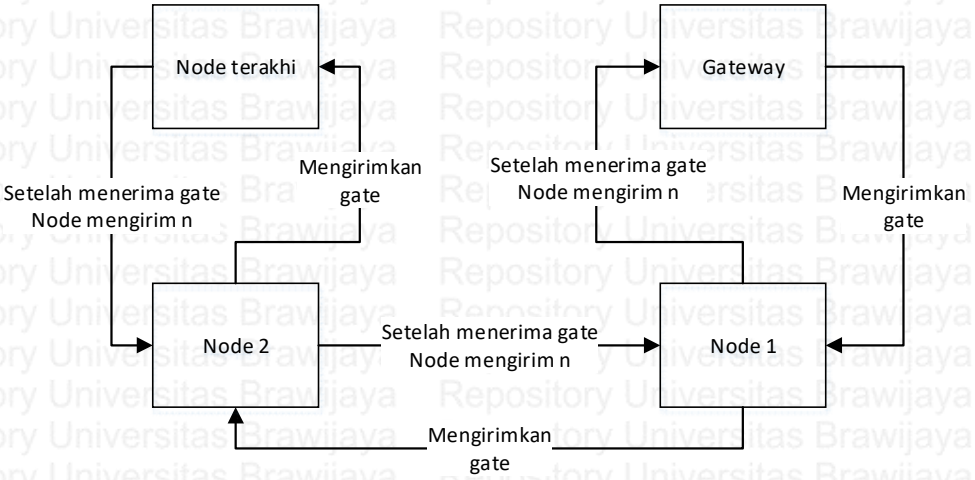
Skripsi ini memberikan informasi mengenai cara membuat modul komunikasi nRF24l01 untuk melakukan komunikasi/pengiriman data secara *point-to-point*.

4.2 Deskripsi Umum

Sisem pengiriman data secara *point-to-point* ialah mendapatkan data dari node-node terjauh dari gateway. Setiap node akan mendapatkan dan mengirim data dari node sebelumnya hingga seluruh data pada node dapat diterima dan diolah oleh gateway.

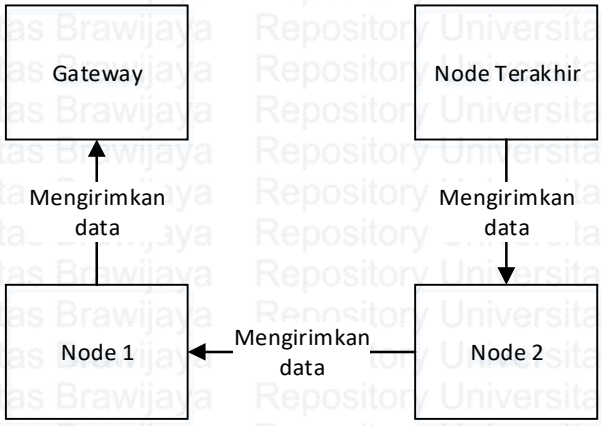
4.2.1 Persfektif Produk / Sistem

Sistem ini dibentuk dengan menentukan jumlah node yang akan digunakan. Jumlah node ini akan memengaruhi lamanya hubungan antar node terbentuk. Node akan melakukan *sensing* data apabila seluruh node telah tersambung dalam hal ini secara *point-to-point* yang mana setiap node akan berhubungan hanya dengan node pengirim dan node penerima. Gambaran pembentukan node dapat dilihat pada Gambar 4.1 .



Gambar 4.1 Diagram Pembentukan Node

Pada Gambar 4.1 diatas gateway bekerja pertama kali dengan melakukan pengiriman data sebagai pembetukan koneksi pada node pertama. Setelah node pertama menerima data maka node pertama akan membalas data tersebut. Setelah balasan diterima gateway, gateway akan bersiap menerima data pada *channel* yang sama dengan saat node mengirimkan balasan. Setelah node pertama berhubungan dengan gateway, node pertama membentuk koneksi dengan node ke-dua dengan prosedur sama seperti yang dilakukan oleh gateway pada node 1. Dan dilakukan secara terus menerus hingga node terakhir ditemukan. Sedangkan untuk pengiriman data menuju gateway ditunjukkan pada Gambar 4.2.



Gambar 4.2 Diagram Pengiriman data dari Node

Pada Gambar 4.2 diatas seluruh node telah terhubung satu sama lain. Pengiriman data dilakukan pertama kali oleh node terakhir atau node ke-n atau node terakhir. setelah itu menuju node ke-(n-1) dan seterusnya sampai ke gateway. Node pertama akan mengirimkan data node dia sendiri dan seluruh node yang terbentuk menuju ke gateway.

4.2.2 Kegunaan

Sistem ini memiliki keguanaan yang sangat efektif untuk proses penyaluran data satu arah seperti data ketinggian air pada jalur sungai yang panjang. Setiap node dapat ditempatkan berjauhan sesuai dengan jarak jangkauan modul komunikasi yang digunakan. Node akan mengirim data yang ia dapatkan menuju ke gateway untuk diolah.

4.2.3 Karakteristik Pengguna

Pengguna sistem ini adalah orang-orang yang dapat menggunakan Arduino IDE dan sedikit mengerti mengenai pemrograman. Karena tidak disediakannya tampilan (GUI/CLI) dan tidak adanya input. Sehingga pengaturan-pengaturan dilakukan dalam kode program. Hal ini menyebabkan orang awam tidak dapat menggunakannya. Sehingga user setidaknya mengerti mengoperasikan komputer dan mengetahui sedikit tentang program. Sementara untuk melihat hasilnya dapat dilakukan oleh siapapun karena hasil monitoring ditampilkan dalam bentuk grafik.

4.2.4 Lingkungan Operasi

Lingkungan pada sistem ini adalah di sungai daerah hilir yang memiliki kondisi aliran air tidak deras yang cenderung stabil. Pada sistem ini lingkungan *prototype* adalah talang air. Talang air ini memiliki panjang 100 cm.

4.2.5 Batasan Perancangan dan Implementasi

Batasan perancangan dan implementasi pada sistem ini adalah jumlah node yang digunakan berjumlah 3 node dengan 1 gateway. *Routing point-to-point* yang digunakan dalam sistem ini adalah untuk pembentukan node yang nantinya digunakan untuk melakukan pengiriman data menuju gateway. Pada sistem ini sensor yang digunakan adalah sensor ultrasonik. Sensor ini digunakan untuk mengetahui ketinggian air sungai. Ketika node berdekatan saat pembentukkan node, node harus di nyalakan atau dijalankan satu persatu.

4.3 Kebutuhan Antarmuka Eksternal

Bagian ini akan menjelaskan mengenai antarmuka yang ada dalam sistem yang meliputi antarmuka pengguna, antarmuka perangkat keras, antar muka perangkat lunak dan antarmuka komunikasi.

4.3.1 Antarmuka Pengguna

Pengguna akan diberikan tampilan GUI grafik. Grafik ini dapat ditampilkan melalui web browser. Grafik yang ditampilkan untuk satu node. Sehingga memiliki 3 grafik karena memiliki 3 node.

4.3.2 Antarmuka Perangkat Keras

Antar muka perangkat keras sistem adalah:

1. Arduino Nano

2. nRF24L01
3. Sensor HC-SR04
4. Komputer/Laptop

4.3.3 Antarmuka perangkat Lunak

Antar muka perangkat lunak sistem adalah:

1. *Operating System* Windows
2. Arduino IDE
3. *Library* RF24
4. Python
5. *Library websocket* dan mqtt
6. GoogleChart

4.3.4 Antarmuka Komunikasi

Antarmuka komunikasi sistem adalah:

1. *Websocket*
2. MQTT
3. *Routing Point-to-Point*

4.4 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi agar sistem dapat berjalan dengan semestinya. Kebutuhan fungsional dibagi menjadi dua bagian yaitu kebutuhan fungsional aplikasi dan kebutuhan fungsional sistem.

4.4.1 Kebutuhan Fungsional Aplikasi

Kebutuhan fungsional aplikasi antaralain:

1. Program python harus dapat membaca data serial pada port yang digunakan oleh gateway pada komputer.
2. Data serial yang didapat harus dapat di-*subscribe* setelah di-*publish* menuju mqtt *broker*.
3. Data harus dapat dibaca oleh *javascript* dan ditampilkan menjadi grafik.

4.4.2 Kebutuhan Fungsional Sistem

Pada sistem pendeteksi ketinggian air ini memiliki 2 subsistem yaitu subsistem node dan subsistem gateway.

Kebutuhan fungsional subsistem node antara lain:

1. Sensor dapat membaca data jarak antara sensor dan permukaan air.
2. Data diterima mikrokontroller dari sensor secara digital
3. Mikrokontroller mengolah data sensor untuk dijadikan level ketinggian.
4. nRF24L01 berkomunikasi dengan mikrokontroller secara SPI

5. nRF24L01 mengirim data sensor yang diolah mikrokontroller menuju gateway

Kebutuhan fungsional subsistem gateway

1. nRF24L01 menerima data level ketinggian yang dikirimkan oleh node
2. nRF24L01 berkomunikasi dengan mikrokontroller secara SPI
3. Mikrokontroller mecetak data yang diterima dari node secara serial

4.5 Kebutuhan Non-Fungsional

Kebutuhan Non-fungsional sistem antara lain:

1. Pembentukkan node ditentukkan jumlah node yang akan digunakan.

   Node pada saat melakukan pembentukkan akan melakukan listening pada channel yang ditentukkan batasannya. Node terakhir yang terhubung dapat diketahui pada batas nilai pencarian channel yang ditentukkan oleh jumlah node yang akan dibentuk.

2. Semakin banyak jumlah node maka pembentukan akan berlangsung lebih lama pada setiap nodenya.

   Node yang melakukan pembentukkan akan melakukan listening pada *channel* yang disediakan. *Channel* yang tersedia ditentukkan oleh banyaknya jumlah node. Sehingga semakin banyak node maka *channel* yang tersedia akan semakin besar dan membuat node semakin lama terhubung karena harus melakukan *listening* pada seluruh *channel* yang ada.

3. Pengiriman data dilakukan setelah node terakhir terhubung

   Node yang telah terbentuk semua sampai node terakhir akan melakukan pengiriman data. Node selain node terakhir tidak akan melakukan pengiriman data sebelum menerima data dari node terakhir. Sehingga ketika node terakhir diberikan waktu untuk melakukan pengiriman data dalam satu menit sekali maka gateway akan menerima data setiap satu menit. Node lainnya akan menunggu selama satu menit atau menunggu data dari node sebelumnya diterima sebelum dapat mengirim data.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas tentang perancangan dan implementasi sistem pendeteksi ketinggian air dengan menggunakan *wireless sensor network node point to point* yang meliputi perancangan perangkat lunak dan keras, implementasi perangkat lunak dan keras serta pendukung sistem lainnya.

5.1 Perancangan Sistem

perancangan sistem ini meliputi perancangan perangkat lunak, perancangan perangkat keras serta alur kerja sistem. Perancangan perangkat lunak membahas program yang ditanamkan pada mikrokontroller agar masing-masing komponen dapat bekerja sesuai dengan spesifikasi sistem. Perancangan perangkat keras membahas tentang komponen dan perangkat yang digunakan serta hubungan antar perangkat tersebut. Sedangkan alur kerja sistem akan membahas bagaimana alur dari sistem saat sistem dijalankan.

5.1.1 Perancangan Perangkat keras

Perancangan perangkat keras meliputi perancangan gateway dan node. Perancangan perangkat keras menjelaskan tentang perangkat keras yang digunakan dalam sistem.

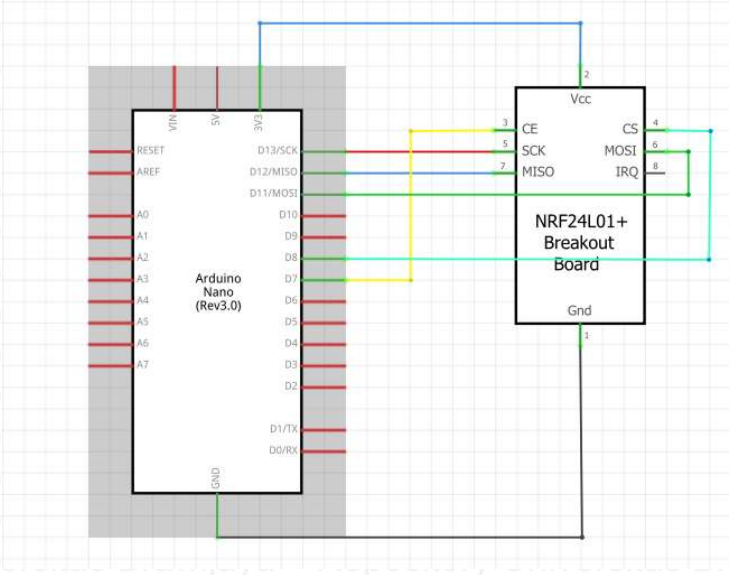
5.1.1.1 Perancangan Perangkat Keras Gateway

Perangkat keras yang digunakan pada gateway adalah Arduino Nano, nRF24L01 dan laptop. Arduino Nano berfungsi sebagai mikrokontroller yang mana mengontrol perangkat keras lain yang digunakan yaitu nRF24L01. nRF24L01 ini adalah modul komunikasi yang digunakan gateway untuk membentuk hubungan dan menerima data dari node. Arduino nano dihubungkan dengan nRF24L01 menggunakan sambungan pin yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Pin Penghubung Arduino Nano dan NRF24L01

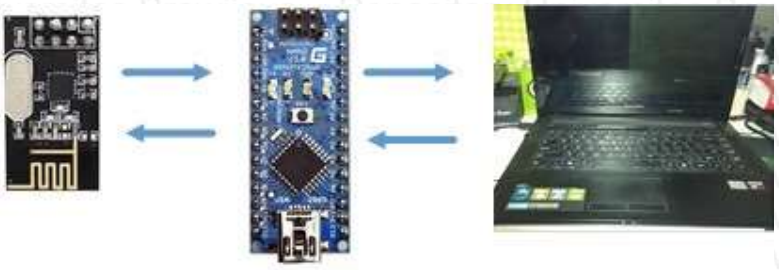
No	Pin Arduino Nano	Pin nRF24L01
1	3.3V	VCC
2	GROUND	GROUND
3	D7	CE
4	D8	CSN
5	D11	MOSI
6	D12	MISO
7	D13	SCK

Skematik perancangan perangkat keras gateway dapat dilihat pada Gambar 5.1, yang memperlihatkan sambungan antara nrf24l01 dan mikrokontroller yang dirangkai menjadi satu.



Gambar 5.1 Skematik Rangkaian Gateway

Arduino Nano juga akan dihubungkan dengan laptop menggunkan kabel data USB untuk menampilkan data yang didapatkan oleh node menjadi sebuah grafik yang akan diambil secara serial. Keseluruhan hubungan antar perangkat keras pada gateway ditunjukkan pada Gambar 5.2.



Gambar 5.2 Hubungan Antar Perangkat Keras Gateway

Data yang diperoleh node dari sensor HC-SR04 akan di kirimkan oleh nRF24L01 menuju node lainnya sampai menuju gateway data yang diterima gateway akan ditampilkan di laptop dalam bentuk grafik.

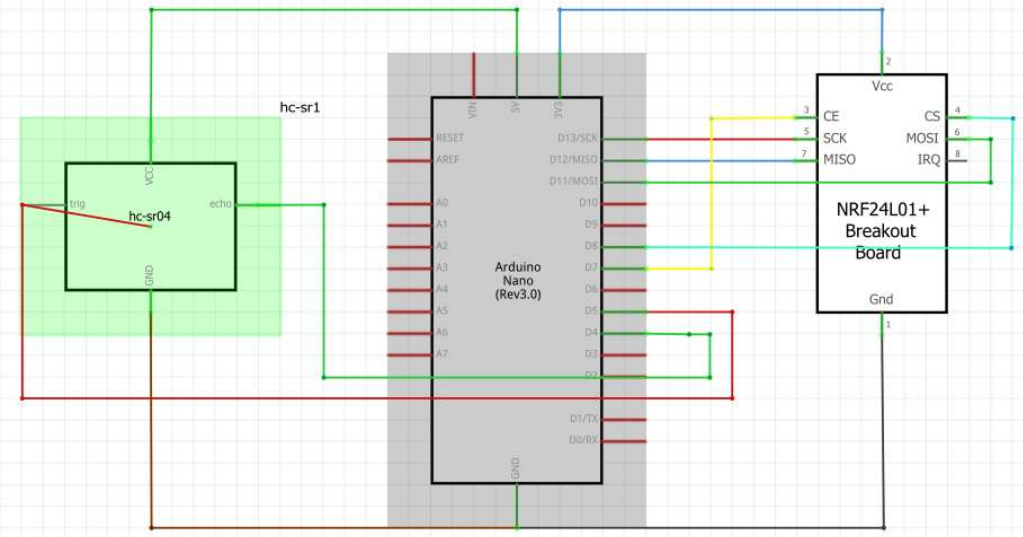
5.1.1.2 Perancangan Perangkat Keras Node

Perangkat keras yang digunakan pada node sedikit berbeda dengan gateway. Pada node diberikan tambahan sensor ultrasonik. Sensor ultrasonik yang digunakan adalah sensor HC-SR94. Sensor ultrasonik ini akan dihubungkan dengan arduino nano untuk mendeteksi jarak antara permukaan air dengan sensor yang kemudian akan dirubah menjadi level ketinggian air. Pin yang menghubungkan seluruh perangkat keras node dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pin Penghubung Arduino Nano dan Sensor HC-Sr04

No	Pin Arduino Nano	Pin nRF24L01	Pin HC-SR04
1	3.3V	VCC	
2	GND	GND	GND
3	D7	CE	
4	D8	CSN	
5	D11	MOSI	
6	D12	MISO	
7	D13	SCK	
8	5V		VCC
9	D4		TRIG
10	D5		ECHO

Pada subsistem node diberikan sensor yang berfungsi untuk melakukan *sensing* data. Data tersebut adalah data jarak yang dideteksi oleh sensor ultrasonik HC-SR04. Setelah data didapatkan data akan dikirimkan menggunakan modul komunikasi NRF24L01. Gambar skematik hubungan perangkat keras node dapat dilihat pada Gambar 5.3 .



Gambar 5.3 Skematik Rangkaian Node

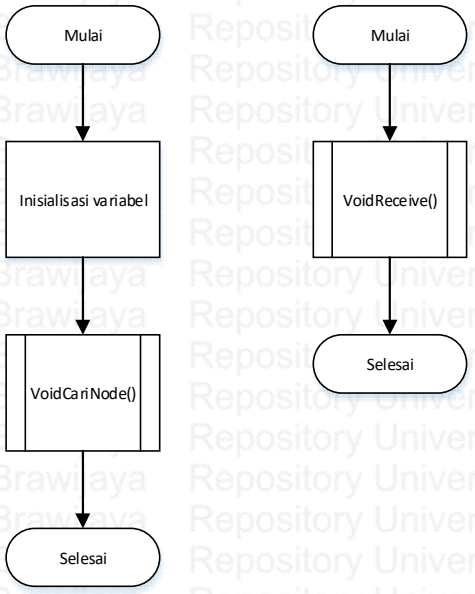
Sensor ultrasonik akan menentukkan ketinggian air. Setelah ketinggian air ditentukkan maka akan dibandingkan dengan ketentun ketinggian untuk menentukan status siaga. Pada system ini digunakan sebuah penampung air dengan panjang 100 cm, lebar 11 cm dan tinggi 11 cm. ketinggian sensor saat ditempatkan adalah 16 cm. Ketinggian air pada siaga IV adalah 3-5 cm, Siaga III 5-7 cm, Siaga II 7-9 cm sedangkan untuk siaga >9 cm. Data siaga inilah yang akan dikirimkan oleh node menuju gateway.

5.1.2 Perancangan Perangkat Lunak

Berdasarkan pada metodologi penelitian maka perancangan perangkat lunak dilakukan setelah rekayasa kebutuhan. Perancangan perangkat lunak meliputi program Arduino untuk pembentukan node, pembacaan nilai dari sensor SR04 dan pengiriman nilai sensor dari node menuju gateway oleh mikrokontroller Arduino Nano. Untuk melakukan pembentukan node menggunakan nRF24l01 program harus ditambah dengan *library* nRF24l01 dan RF24 yang akan mengatur komunikasi data seperti penerimaan data, pengiriman data, pengalamatan dan frekuensi data yang digunakan. Program ini terdiri dari dua bagian, program gateway dan program node.

5.1.2.1 Perancangan Perangkat Lunak Gateway

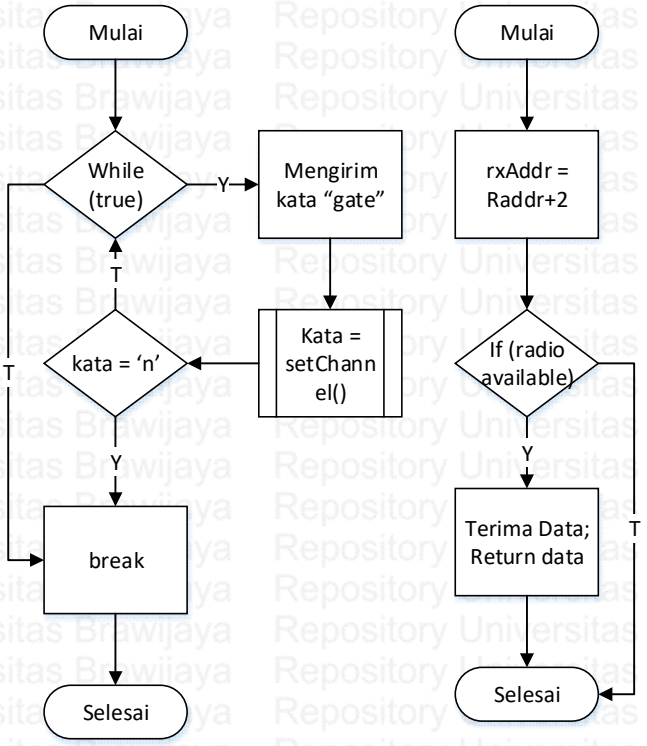
program gateway akan membuat node yang menjalankannya melakukan pencarian node pertama dan pengolahan data level keringgian air yang diterima dari node. Data ketinggian air tersebut akan dikirimkan secara serial menuju laptop. Program utama gateway ditunjukkan oleh *flowchart* pada Gambar 5.4 Program gateway pertama kali akan mencari node dengan cara mengirimkan data pada suatu *channel* tertentu. *Channel* ini akan menjadi channel yang menjadi penghubung gateway untuk mendapatkan data dari node. Data yang diterima akan diolah oleh gateway.



Gambar 5.4 *Flowchart* Fungsi utama Program gateway

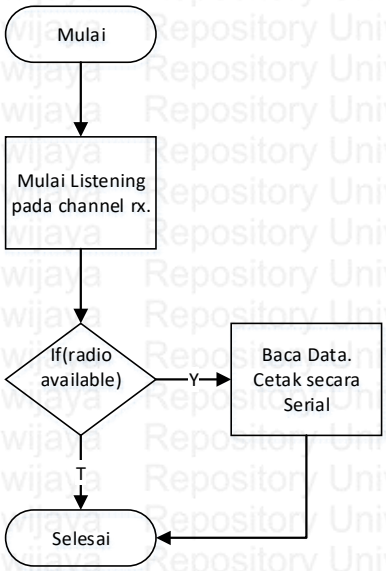
Fungsi utama ada dua yaitu fungsi *setup* dan *loop*. Pada fungsi *setup* program yang dimasukkan disana akan dijalankan satu kali selama program menyala. Fungsi *setup* digunakan untuk inisialisasi. Sedangkan fungsi *loop* adalah bagian fungsi yang akan dijalankan secara terus menerus.

Pembentukan node baru pada gateway dilakukan pada fungsi *setup.* Pembentukkan node ini dilakukan sekali setelah dijalankan. Program dibuat *loop* secara terus menerus kemudian setelah node ditemukan program akan berhenti berjalan dan akan masuk menuju fungsi *loop*. *Flowchart* pembentukkan node pada gateway dapat dilihat pada Gambar 5.5. *Flowchart* pencarian node ini memiliki 2 fungsi yaitu fungsi cariNode dan fungsi *setChannel*. Fungsi cariNode melakukan *looping* sampai node ditemukan. Pada saat *looping* fungsi cariNode mengirimkan data gateway pada *channel* 2 yang akan di-*listening* oleh node. Sedangkan untuk fungsi *setChannel* melakukan penerimaan data pada saat node membalas atau mengirimkan data. Kemudian data yang diterima akan digunakan oleh fungsi cariNode untuk memverifikasi node dan menghentikkan pencarian node. *Flowchart* fungsi cariNode dapat dilihat pada Gambar 5.5a dan *flowchart* fungsi *setChannel* dapat dilihat pada Gambar 5.5b.



Gambar 5.5 *Flowchart* Pencarian Node Pada Gateway

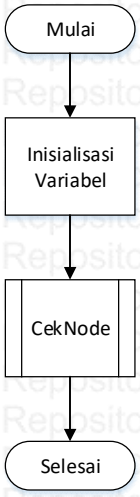
Fungsi yang ada pada gateway selain fungsi pembentuk node adalah fungsi *receive*. Fungsi ini digunakan untuk menerima data yang dikirimkan oleh node. Setelah data diterima data akan diambil secara serial oleh pc/laptop. Setelah data didapatkan oleh laptop maka akan dibuat tampilan grafiknya menggunakan laptop. *Flowchart* fungsi receive pada gateway dapat dilihat pada Gambar 5.6.



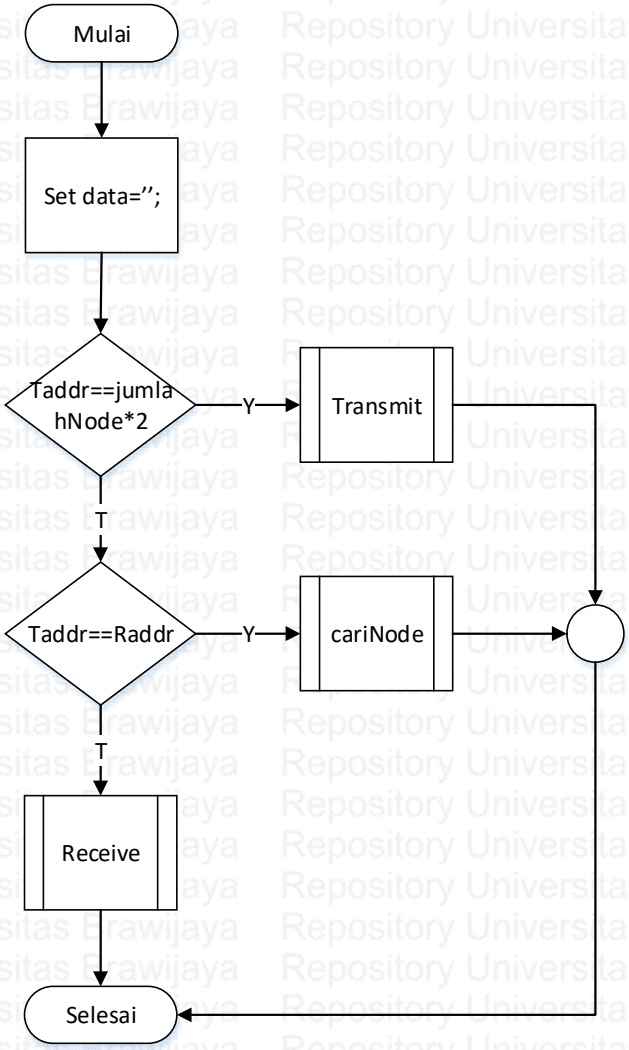
Gambar 5.6 *Flowchart* Fungsi Receive Pada Gateway

5.1.2.2 Perancangan Perangkat Lunak Node

Program utama node memliki beberapa fungsi yang sama dengan program gateway, namun memiliki kompleksitas yang lebih banyak. Hal ini disebabkan program yang sama akan dijalankan oleh beberapa node. Sehingga program tersebut harus lebih fleksibel. *Flowchart* program utama node ditunjukkan oleh Gambar 5.7 **.** Pada fungsi void *setup* node akan memanggil dan menjalankan fungsi CekNode. Yang akan berjalan sekali fungsi cekNode ini berfungsi untuk membalas pesan dan membentuk hubungan dengan gateway. Sedangkan pada fungsi *Loop* program node memiliki tiga fungsi yang akan dijalankan yaitu *Transmit*, cariNode dan *receive*. Fungsi yang akan dijalankan bergantung pada kondisi yang diterima node pada saat menjalankan fungsi *setup*. Gambar *flowchart* fungsi setup ditunjukkan pada Gambar 5.7 dan *flowchart* fungsi *loop* pada Gambar 5.7



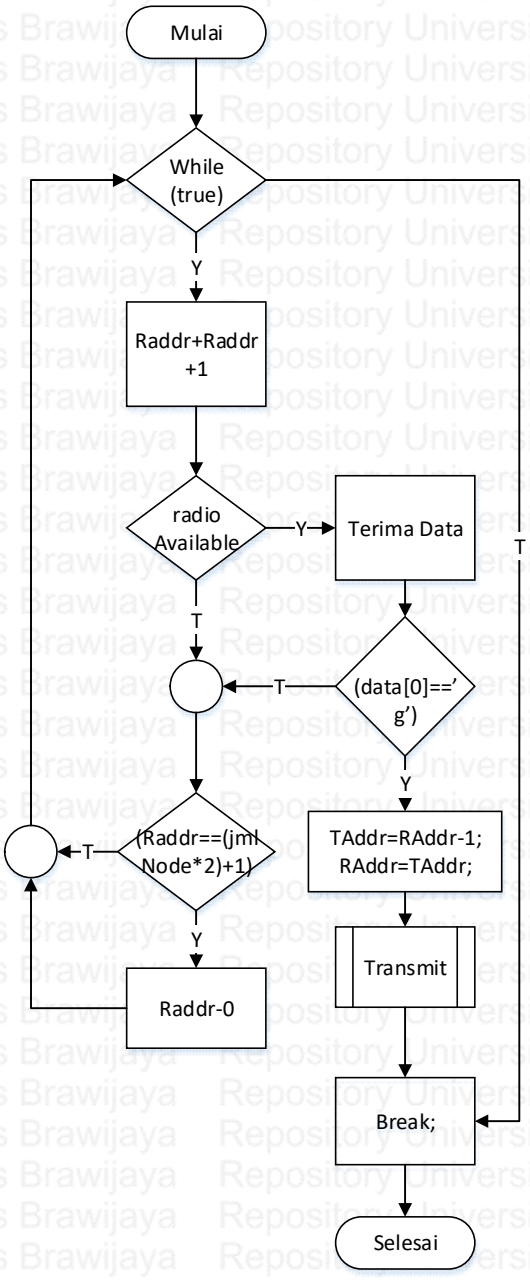
(a)



(b)

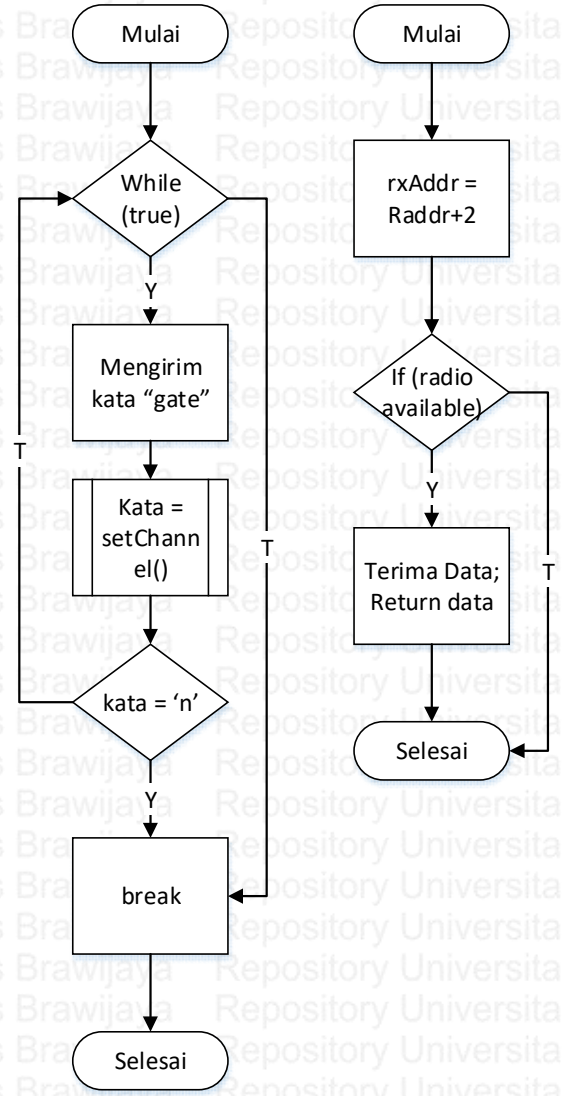
Gambar 5.7 *Flowchart* Fungsi utama node

Fungsi cekNode yang djalankan pada fungsi *setup* akan melakukan *listening* pada *channel* yang disediakan. *Channel* yang disediakan ini dihitung dari jumlah node yang akan dibentuk di kali 2. Setelah melakukan *listening* terhadap seluruh *channel* dan tidak mendapatkan data "gate" maka *listening channel* akan direset kembali ke 0. Dan melakukan *listening* kembali sampai node menerima data "gate". Setelah node mendapatkan "gate" maka node akan menjalankan fungsi *transmit* yang mengirimkan balasan 'n' menuju *channel* yang sama pada saat node menerima data "gate". Gambar *flowchart* fungsi ceknode dapat dilihat pada Gambar 5.8.



Gambar 5.8 Flowchart Fungsi CekNode

Program node juga melakukan pencarian node. Pencarian node ini dilakukan ketika node telah terhubung dengan gateway. Setiap node akan berhubungan dengan dua node kecuali node terakhir hanya berhubungan dengan satu node. Kedua node tersebut adalah node untuk mengirim data atau node untuk menerima data. Node hanya akan menerima data dari satu node dengan *channel* tertentu dan mengirim data ke node lainnya dengan channel yang berbeda. Proses pencarian node baru pada node sama seperti pencarian node pada gateway yaitu menggunakan fungsi cariNode dan setChannel. *Flowchart* dari fungsi cariNode dan setchannel dapat dilihat pada Gambar 5.9a dan Gambar 5.9b.

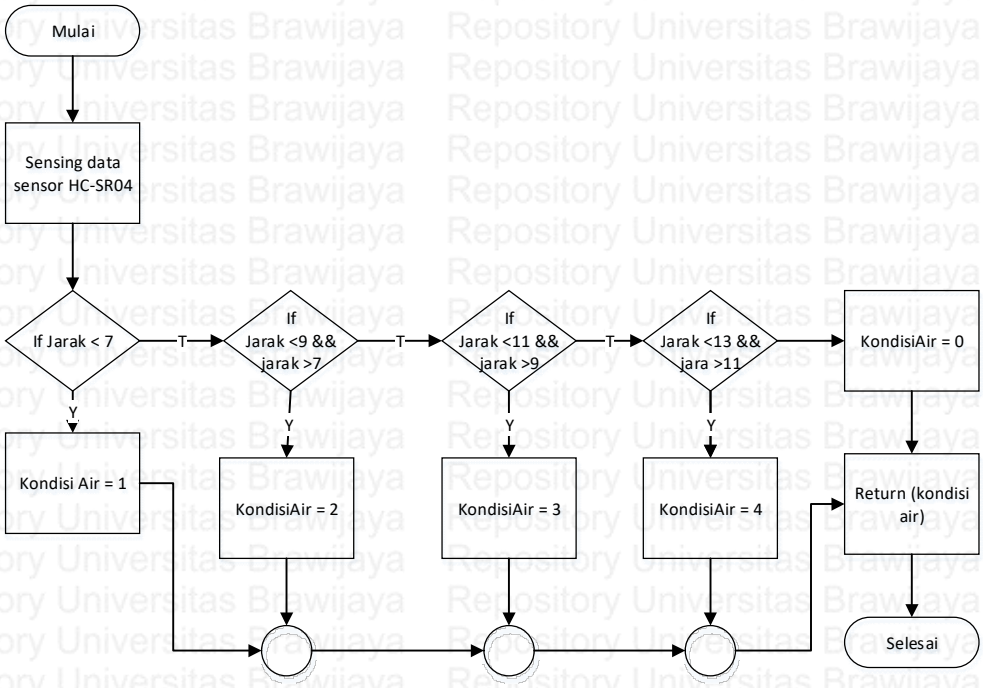


(a) **(b)**

Gambar 5.9 Flowchart Pencarian Node pada Node

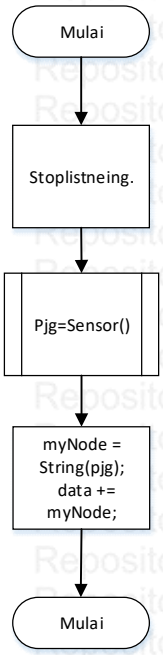
Setelah semua node terhubung maka node terakhir akan mengirimkan data. Data yang dikirimkan adalah data dari ultrasonik yang telah dimodifikasi. *Flowchart* dari fungsi sensor dapat dilihat pada Gambar 5.10 . Fungsi ini akan dipanggil oleh fungsi *transmit*. Sebelum mengirimkan data node akan melakukan pendeteksian data jarak yang dilakukan oleh sensor HC-SR04. Setelah jarak didapatkan maka jarak akan dibandingkan dengan kondisi level ketinggian. Semakin kecil jarak yang terdeteksi maka semakin tinggi level ketinggian air. Level ketinggian dibagi menjadi empat dari siaga 1 sampai siaga 4. Level tertinggi ada pada kondisi siaga 1. Kondisi siaga ini berdasarkan pada BPBD (Badan Penanggulangan Bencana Daerah) yang memberikan arti pada setiap kondisi siaga dari siaga 4 sampai siaga 1.

Siaga IV : Belum ada peningkatan debit air secara mencolok. Komando di lapangan, termasuk membuka atau menutup pintu air serta akan dikemanakan arah air cukup dilakukan oleh komandan pelaksana dinas atau wakil komandan operasional wilayah. Siaga III : Bila hujan yang terjadi menyebabkan terjadinya debit air meningkat di pintu - pintu air tetapi kondisinya masih belum kritis dan membahayakan. Meski demikian, bila status siaga III sudah ditetapkan, masyarakat sebaiknya mulai berhati-hati dan mempersiapkan segala sesuatunya dari berbagai kemungkinan bencana banjir. Siaga II : Bila hujan yang terjadi menyebabkan debit air mulai meluas, maka akan ditetapkan Siaga II, penanggung jawab untuk siaga II ini adalah Kepala Badan Penanggulangan Bencana Daerah Prov. DKI Jakarta yaitu Sekretaris Daerah. Siaga I : Bila dalam enam jam debit air tersebut tidak surut dan kritis maka ditetapkan Siaga I. Penanggung jawab penanganan status siaga I langsung di tangan Gubernur (Kumparan.com, 2017).



Gambar 5.10 Flowchart Fungsi Sensor

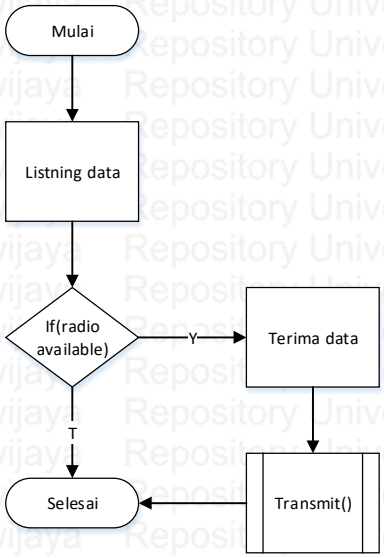
Setelah data pada node didapatkan maka node akan mengirimkan data dari sensor ini menuju node selanjutnya sampai menuju gateway. Untuk node terakhir akan langsung menjalankan fungsi ini terlebih dahulu. Sedangkan untuk node lainnya akan melakukan receive terlebih dahulu sebelum menjalankan fungsi *transmit* ini. *Flowchart* fungsi transmit dapat dilihat pada Gambar 5.11 . Fungsi transmit akan memanggil fungsi sensor untuk mendapatkan datanya dan menyambungkan dengan data dari node sebelumnya sebelum datanya dikirimkan.



Gambar 5.11 Flowchart Fungsi Transmit

Setelah node terakhir mengirimkan datanya node lainnya akan menerima data. Penerimaan data dilakukan menggunakan fungsi *receive* pada node. *Flowchart* fungsi *receive* pada node dapat dilihat pada

Gambar 5.12. Fungsi *transmit* akan dipanggil pada fungsi *Receive* ini. Fungsi *transmit* dipanggil atau dijalankan untuk mengirimkan data yang baru saja diterima oleh node. Data yang diterima oleh node ini akan digabungkan dengan data yang diperoleh dari sensor dan mengirimkannya kembali menuju node selanjutnya.



Gambar 5.12 Flowchart Fungsi Receive

5.1.2.3 Pembuatan Grafik

Data yang dikirimkan oleh node-node tersbut kemudian akan ditampilkan pada grafik. Masing-masing node menggunakan satu grafik. Pembuatan grafik ini melalui tahapan sebagai berikut.

1. Pengambilan data menggunakan python

   Data yang diterima pada node akan diambil oleh program python secara serial. Untuk itu dibutuhkan *library* PySerial untuk melakukannya. *Library* ini harus di-*install* terlebih dahulu pada python yang kita gunakan. Setelah data didapatkan data tersebut akan dipublish menuju *broker*. Sebelum dapat melakukan *publish* program harus meng-*install* terlebih dahulu *library* paho-mqtt. *Library* ini digunakan untuk dapat melakukan *publish* dan *subscribe* pada *broker*.

2. Data pada *broker* di *subscribe*

   Broker harus di-*install* dan dijalankan terlebih dahulu agar proses *publish* dan *subscribe* dapat di lakukan. *Broker* ini ada pada port 1883. Ketika telah berjalan kita dapat dilihat dengan cara mengetikkan netstat –an kemudian cari port 1883 jika ada dan *listening* maka broker telah berjalan dan dapat digunakan. Data yang sebelumnya di-*upload* menuju *broker* akan dapat di-*subscribe*. Data yang di-*subscribe* disesuaikan dengan topik yang digunakan pada saat *publish*. Kemudian data yang di-*subscribe* akan dipilah dan di *encript* menjadi data json. Setelah itu dikirimkan menuju *websocket* server.

3. Data ditampilkan pada browser dengan JavaScript

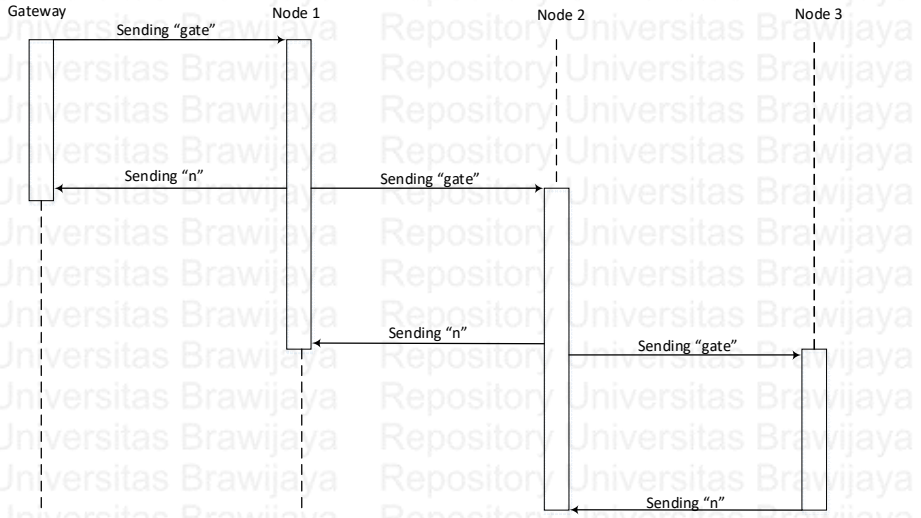
   Kemudian browser akan dibuat program *JavaScriptnya* yang akan melakukan *listen* terhadap *websocket* server yang dibuat pada proses sebelumnya. Setelah terhubung maka Browser akan menerima data json yang dikirimkan dan diolah menggunakan Bahasa Pemrograman *JavaScript*. Yang kemudian ditampilkan datanya menjadi bentuk grafik. Data grafik ini dibuat menggunakan fitur google *chart*.

5.1.3 Alur Kerja Sistem

Sistem ini bekerja melalui 2 alur utama, yaitu pembentukkan node dan pengiriman data. Sebelum data dikirim node harus terbentuk terlebih dahulu. Pembentukkan node dimulai oleh gateway sehingga apabila gateway belum siap maka node tidak akan pernah terhubung. Setelah semua terhubung node akan mengirimkan data menuju gateway. Pada pengiriman data menggunakan *channel* yang telah dibentuk sebelumnya yaitu pada tahap pembentukkan node.

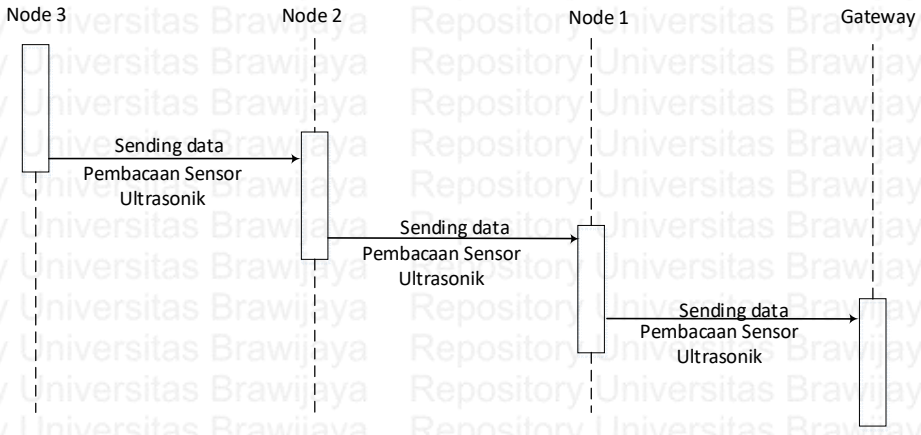
Tahapan pembentukkan node yang berjalan pada sistem ini setelah program dijalankan dapat dilihat pada Gambar 5.13. Pada siklus dibawah menggunakan 3 buah node dan satu gateway. Pada siklus tersebut ada 4 tahapan utama yaitu gateway akan melakukan request berupa pengiriman “gate” yang mengindikasikan bahwa pengirim adalah gateway, node akan melakukan listening menunggu *request* pada *channel* channel 1-7, setelah request berupa data “gate”

diterima node akan mengirimkan balasan berupa data “n” yang mengindikasikan bahwa penerima data adalah node, pada channel request diterima dan melakukan *set channel* untuk menerima dan mengirim data. Berikut gambar siklus pembentukan node untuk 3 node dan 1 gateway.



Gambar 5.13 Diagram Siklus Pembentukkan Node

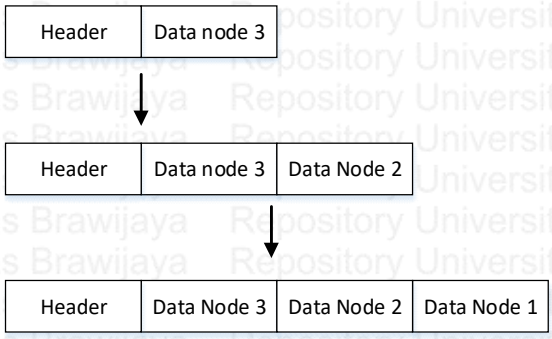
Setelah node terbentuk maka node akan melakukan pengiriman data. Pengiriman data dilakukan setelah seluruh node terhubung. Untuk siklus pengiriman data dapat dilihat pada Gambar 5.14**.** *Channel* yang digunakan untuk mengirim adalah *channel* yang digunakan node untuk mengirimkan “N,” pada saat pembentukkan Node. Pengiriman data dimulai dari node terakhir mengirimkan data dari hasil pembacaan sensor ultrasonik menuju node 2 setelah node 2 menerima data node 2 mengirimkan data node 3 dan node 2 menuju node selanjutnya hingga data sampai di gateway.



Gambar 5.14 Diagram Siklus Pengiriman Data

5.1.4 Struktur data

Pengiriman data akan selalu dilakukan setelah node terkoneksi seluruhnya. Setelah seluruh node terhubung pengiriman dilakukan dari node terakhir menuju ke gateway. Pengiriman dilakukan dalam segala kondisi. Struktur data yang dikirim untuk 3 node dapat dilihat pada Gambar 5.15.



Gambar 5.15 Struktur Data Pada Pengiriman data

Data yang dikirimkan adalah angka 1-4. Angka tersebut mewakili dari status yang nantinya digunakn oleh system. Angka ini mewakili kondisi dari sungai yang diberikan node. Kondisi ini terbagi menjadi 4, yaitu siaga 1, 2, 3 dan siaga 4. Siaga ini berdasar pada ketinggian air yang diterima oleh sistem. Semakin tinggi air maka tingkat siaga semakin kecil dan darurat. Untuk perbedaan ketinggian sendiri akan berbeda pada setiap kondisi air yang dimonitor. Perbedaan kondisi ini berdasarkan pada daya tampung tempat dilakukannya monitor. *Header* data memiliki panjang data 1bit data. Nilai pembaca data yang akan dikirimkan adalah 1 bit data. Pada pengiriman node 3 mengirimkan data sebesar 4 bit. 1bit untuk header, 1bit untuk data pembacaan sensor ultrasonik dan 2 bit data untuk pembatas antar data.

5.2 Implementasi Sistem

Implementasi pengiriman data dengan *routing point to point* ini dilakukan setelah tahap rekayasa kebutuhan yang meliputi kebutuhan perangkat lunak dan keras. Selanjutnya adalah perancangan perangkat lunak dan keras, serta implementasi perangkat lunak dan keras.

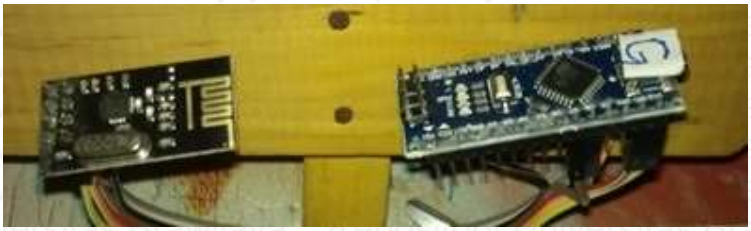
5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras sistem pendeteksi ketinggian air dengan menggunakan *wireless sensor network node point-to-point* dibagi menjadi dua bagian yaitu bagian gateway dan node. Node digunakan untuk mendapatkan data dan mengirimkannya sampai ke gateway dan gateway akan menampikan data yang didapatkan oleh node. Pengiriman data pada node sampai menuju gateway menggunakan modul transceiver NRF24L01.

5.2.1.1 Implementasi Perangkat Keras Sebagai Gateway

Pada gateway implementasi perangkat keras tidak menggunakan sensor karena hanya digunakan sebagai *receiver*. Data dari node dikirimkan

menggunakan modul komunikai nRF24L01. Gambar implementasi perangkat keras pada node dapat dilihat pada Gambar 5.16 berikut.



Gambar 5.16 Implementasi Alat Pada Gateway

Data yang didapatkan akan ditampilkan secara serial pada Arduino IDE di laptop. Mikrokontroller pada gateway akan mengolah data kemudian dikirimkan menuju laptop dengan tampilan yang dapat mudah dilihat.

5.2.1.2 Implementasi Perangkat Keras Sebagai Node

Pada node implementasi perangkat keras menggunakan sensor untuk mendapatkan data. Data ini nantinya akan dikirimkan menuju gateway. Selain data sensor yang node tersebut dapatkan, node akan mengirimkan data-data dari node tetangganya menuju ke gateway. Sensor yang digunakan pada sistem ini adalah sensor ultrasonic. Gambar 5.17 berikut ini adalah gambar node setelah sistem diimplementasikan.

Gambar 5.17 Implementasi Alat Pada Node

5.2.1.3 Implementasi Node Pada Prototype

Prototype yang digunakan adalah talang air. Pada talang air ini akan diisi air yang di deteksi oleh sensor pada node. Talang air ini memiliki panjang 100 cm. Sensor ultrasonik diletakkan dibagian ujung talang air. Agar sensor berada diatas talang air digunakan *sterofoam*. Gambar 5.18 adalah gambar dari implementasi node pada *prototype.*

Gambar 5.18 Impelementasi node pada *prototype*

5.2.2 Implementasi Perangkat Lunak Aplikasi

Pada implementasi perangkat lunak terdapat dua bagian yaitu program pada gateway dan program pada node. Program-program ini memiliki bagian-bagian yang sama. Yaitu pada bagian pencarian node memiliki bagian yang sama antara gateway dan node.

5.2.2.1 Program Gateway

Program pada gateway memiliki 2 proses. Yaitu proses pembentukan node dan penerimaan data. Setelah proses satu berjalan dan menemukan node, proses dua akan berjalan menunggu data yang dikirimkan oleh node.

1. Proses pembentukkan node

Proses pembentukan node dilakukan pada saat inisialisasi void setup. Sehingga fungsi ini akan berjalan hanya sekali dan akan berhenti berjalan setelah terhubung dengan node. Proses pembentukkan node terdiri dari dua fungsi, yaitu fungsi cariNode dan setChannel. Fungsi cari node ini akan dilakukan berulang-ulang sampai data diterima node dan dibalas oleh node. Dibawah ini adalah *source code* dari fungsi cari node.

```
1   ……
2   while(true)
3     {
4         Serial.println("cari Node/0/0/0");
5         radio.setRetries(15, 15);
6         radio.openWritingPipe(rxAddr);
7         radio.stopListening();
8         const char text[] = "gate";
9         radio.write(&text, sizeof(text));
10        delay(100);
11        char kata=SetChannel();
12        if(kata=='n')
```

```
13        {
14          break;
15        }
16    }
17  ……
```

Untuk fungsi *setchannel* proses yang pertamakali dilakukan adalah menetapkan alamat untuk menerima data. Karena gateway hanya akan menerima data maka penentuan *channel* pengirim data tidak dilakukan pada gateway. Setelah alamat penerima data di tentukan maka gateway akan menunggu node mengirimkan suatu data yang nantinya akan dikembalikan ke fungsi carinode untuk diolah. Dibawah ini adalah source code dari fungsi *setchannel*.

```
1   ……
2   rxAddr=RAddr+2;
3       radio.openReadingPipe(0, rxAddr);
4       radio.startListening();
5       delay(100);
6       if (radio.available())
7       {
8         char text[3]={0};
9         radio.read(&text, sizeof(text));
10        data = text;
11        return(text[0]);
12    }
13  ……
```

2. Proses penerimaan data

Setelah terhubung dengan node maka gateway akan melakukan penerimaan data. Data yang diterima oleh node bertipe string dengan format header/node3/node2/node1. *Source code* saat gateway melakukan penerimaan data dapat dilihat di bawah ini.

Data yang diterima pada gateway akan dikirimkan menuju laptop secara serial. Kemudian data tersebut akan dijadikan sebuah grafik. Karena pembuatan grafik menggunakan data serial sehingga ketika melakukan perintah Serial.print pada gateway harus memperhatikan format karena akan terjadi *error* pada saat menjalankan program yang menggunakan data serial.

```
1   ……
2   radio.openReadingPipe(0, rxAddr);
3       radio.startListening();
4       if (radio.available())
5       {
```

```
6       char text[50];
7       radio.read(&text, sizeof(text));
8       Serial.println(text);
9     }
10  ……
```

5.2.2.2 Program Node

Program pada node memiliki 5 proses yaitu, proses pencarian gateway atau node yang mengarah menuju gateway, pencarian node baru, menerima data, mendapatkan data sensor dan mengirimkan data. Ketika node dijalankan proses-proses tersebut tidak akan dilewati kecuali pada node terakhir ada beberapa proses yang dilewati. Yaitu proses pencarian node baru dan menerima data.

1. Proses pencarian gateway

Proses pencarian gateway ini berjalan pada saat inisialisasi pada void setup sehingga hanya akan berjalan sekali. Hanya saja program ini tidak akan berhenti atau melanjutkan ke proses berikutnya apabila belum terhubung dengan gateway. Proses ini berjalan secara terus-menerus sampai gate ditemukan. Jumlah *channel* yang akan dilisten bergantung pada jumlah node yang kita inisialisasi terlebih dahulu. Proses ini memiliki 2 fungsi yaitu fungsi ceknode dan fungsi transmit. Dibawah ini *source code* fungsi ceknode.

```
1   ……
2     while(true)
3       {
4         RAddr=RAddr+1;
5         radio.openReadingPipe(0, RAddr);
6         radio.startListening();
7         if (radio.available())
8         {
9           char text[32] = {0};
10          radio.read(&text, sizeof(text));
11          if(text[0]=='g')
12          {
13            TAddr=RAddr-1;
14            RAddr=TAddr;
15            transmit();
16            break;
17          }
18          data = text;
19        }
```

```
20          if(RAddr==(jumlahNode*2)+1)
21          {
22            RAddr=0;
23          }
24      }
25  ……
```

Source code transmit akan ditampilkan pada saat proses pengiriman data. Karena menggunakan fungsi yang sama dengan fungsi yang digunakan pada proses ini.

2. Proses pencarian node baru

Setelah terhubung dengan gateway selain node terakhir akan menjalankan pencarian node baru. Pencarian node baru ini memiliki *source code* yang mirip dengan gateway saat melakukan pencarian node. Hanya saja pada node fungs-fungsi tersebut dijalankan pada fungsi *loop* arduino karena untuk melakukan pengecekkan apakah node tersebut node terakhir atau bukan dan untuk menjalankan fungsinya terus-menerus sampai node baru ditemukan. Proses ini dibagi menjadi 2 fungsi yang saling berhubungan juga seperti pada gateway. *Source code* fungsi carinode dapat dilihat dibawah ini.

```
1   ……
2   radio.setRetries(15, 15);
3   radio.openWritingPipe(rxAddr);
4   radio.stopListening();
5   const char text[] = "gate";
6   radio.write(&text, sizeof(text));
7   char kata=SetChannel();
8      if(kata=='n')
9         {
10          RAddr=RAddr+2;
11       }
12  ……
```

Fungsi *setchannel* ini sama dengan fungsi setchannel pada gateway. Berikut *source code* fungsi setchannel dapat dilihat dibawah ini.

```
1   ……
2      rxAddr=RAddr+2;
3      radio.openReadingPipe(0, rxAddr);
4      radio.startListening();
5      if (radio.available())
6      {
7        Serial.println("data masuk");
```

```
8       char text[32] = {0};
9       radio.read(&text, sizeof(text));
10      return(text[0]);
11    }
12  ……
```

3. Proses pengiriman data

Setelah node baru ditemukan node tersebut akan menunggu data pada *channel* penerima, yaitu *channel* yang digunakan pada saat membentuk node baru. Proses pengiriman data ini dilakukan pertamakali oleh node terakhir. Sebelum node terarkhir ditemukan data pada setiap node tidak akan dikirimkan, karena node lainnya akan menunggu data diterima terlebih dahulu baru mengirimkan datanya. Berikut *source code* proses pengiriman data yang dijadikan fungsi *transmit*.

```
1   ……
2   radio.setRetries(15, 15);
3   radio.openWritingPipe(TAddr);
4   radio.stopListening();
5   String myNode;
6   String pjg;
7   pjg= sensor();
8   myNode = String(pjg);
9   data += myNode;
10  Serial.print("data yang dikirim : ");
11  Serial.println(data);
12  char text[50];
13  data.toCharArray(text, sizeof(text));
14  radio.write(&text, sizeof(text));
15  ……
```

4. Proses pembacaan data sensor

Pada saat sensor menjalankan fungsi transmit dipanggil pula fungsi sensor. Fungsi sensor adalah fungsi yang digunakan pada saat proses pembacaan data sensor. Selain dibaca data sensor dirubah menjadi suatu konidis. Setelah itu datanya dibaca pada fungsi transmit untuk dikirimkan menuju node selanjutnya. *Source code* fungsi sensor dapat dilihat dibawah ini.

```
1   ……
2   digitalWrite(trigPin, LOW);
3   delayMicroseconds(10);
4   digitalWrite(trigPin, HIGH);
5   delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
jarak= duration*0.034/2;
  if (jarak < 7){
    Serial.print("Status siaga I : ");
    kondisiAir = "1/";
  }else if(jarak<9 && jarak >7){
    Serial.print("Status siaga II : ");
    kondisiAir = "2/";
  }else if(jarak<11 && jarak >9){
    Serial.print("Status siaga III : ");
    kondisiAir = "3/";
  }else if(jarak < 13 && jarak >11){
    Serial.print("Status siaga IV : ");
    Serial.println(jarak);
    kondisiAir = "4/";
  }else{
    kondisiAir = "0/";
  }
return (kondisiAir);
……
```

5. Proses penerimaan data

Setelah node terakhir mengirim data node selanjutnya akan menerima data dari node terakhir. Setelah menerima data node selanjutnya akan mengirimkan data juga. Proses pengiriman dan pembacaan di node ini sama dengan proses pada node terakhir. Bedanya node terakhir tidak menunggu adanya data yang diterima terlebih dahulu. Node yang menerima data akan menerima data pada *channel* yang digunakan untuk menghubungkan dengan node selanjutnya. Berikut *source code* penerima data pada node.

```
……
  radio.openReadingPipe(0, RAddr);
  radio.startListening();
  if (radio.available())
  {
     char text[32] = {0};
     radio.read(&text, sizeof(text));
     data = text;
     delay(100);
     transmit();
```

```
11    }
12  ……
```

5.2.2.3 Pembuatan Grafik

Data yang dikirimkan oleh node-node tersbut kemudian akan ditampilkan pada grafik. Masing-masing node menggunakan satu grafik. Pembuatan grafik ini melalui tahapan sebagai berikut.

1. Pengambilan data menggunakan python

Data yang diterima oleh gateway akan diambil oleh program python secara serial. Untuk dapat melakukan menggunakan *library* PySerial. Setelah data didapatkan data tersebut akan di-*publish* menuju *broker*. Alamat *broker* yang digunakan adalah pada alamat ip "127.0.0.1" dan port 1883. Alamat Ip tersebut merupakan alamat *Ip local* pada komputer. Data yang dipublish didapat dari serial dengan port "com24" dan *baudrate* 9600. Berikut ini *source code* program *publisher* untuk menampilkan grafik.

```
1   ……
2       radio.openReadingPipe(0, RAddr);
3       radio.startListening();
4       if (radio.available())
5       {
6           char text[32] = {0};
7           radio.read(&text, sizeof(text));
8           data = text;
9           delay(100);
10          transmit();
11      }
12  ……
```

2. Data pada *broker* di-*subscribe*

Data yang sebelumnya di-*upload* menuju *broker* akan dapat di-*subscribe*. Data di-*subscribe* disesuaikan dengan topik yang digunakan pada saat *publish*. Pada proses *subscribe* ini data akan di baca dan di *encript* dalam format json. Data yang di-*subscribe* ada pada topic dengan folder skripsi/sensor dan terhubung dengan alamat I*p local* dan port 1883. Berikut *source code* pada saat proses *subscribe* data.

```
1   ……
2   class MQTTtoweb(protocol.Protocol):
3       def __init__(self):
4           global counter
5           self.no = counter
6           self.ruang = []
```

```
7       counter += 1
8   def connectionMade(self):
9       self.mqttc = self.startMQTT()
10  def connectionLost(self, reason):
11      self.mqttc.loop_stop()
12  def save(self,data):
13      topic = data["topic"].split("/")
14      ada = False
15      for ruang in self.ruang:
16          if ruang["nama"] == topic[0]:
17              ada = True
18              if topic[2] == "node1":
19                  ruang["node1"] = shiftLeft(ruang["node1"], int(data["msg"]))
20              elif topic[2] == "node2":
21                  ruang["node2"] = shiftLeft(ruang["node2"], int(data["msg"]))
22              elif topic[2] == "node3":
23                  ruang["node3"] = shiftLeft(ruang["node3"], int(data["msg"]))
24      if not ada:
25          if topic[2] == "node1":
26              node1 = int(data["msg"])
27              node2 = 0
28              node3= 0
29          elif topic[2] == "node2":
30              node1 = 0
31              node2 = int(data["msg"])
32              node3= 0
33          elif topic[2] == "node3":
34              node1 = 0
35              node2 = 0
36              node3= int(data["msg"])
37          dict = {
38              "nama": topic[0],
39              "node1": [0, 0, 0, 0, 0, 0, 0, 0, 0, node1],
40              "node2": [0, 0, 0, 0, 0, 0, 0, 0, 0, node2],
41              "node3" : [0, 0, 0, 0, 0, 0, 0, 0, 0, node3]
42          }
```

```
43                  self.ruang.append(dict)
44              print self.no,". "+json.dumps(self.ruang)
45              self.transport.write(json.dumps(self.ruang))
46          def terima(self,mqttc, obj, msg):
47              data = {"topic": msg.topic,
48                              "msg": msg.payload
49                              }
50              self.save(data)
51          def startMQTT(self):
52              mqttc = mqtt.Client("sub1", clean_session=False)
53              mqttc.on_message = self.terima
54              mqttc.connect("127.0.0.1", 1883)
55              mqttc.subscribe("+/sensors/+", qos=0)
56              mqttc.loop_start()
57              return mqttc
58      ……
```

Data yang di-*subscribe* pada folder sensor diambil datanya dan di masukkan ke dalam format json. Setelah berubah menjadi format json data akan dikirimkan menggunakan *websocket* dan dibaca oleh *javascript*. Berikut *source code* untuk dapat menggunakan fitur *websocket*. *Websocket* ini akan di jalankan pada port 9000. Pengiriman data dilakukan dengan *transport.write* yang ditunjukkan pada *source code*.

```
1       ……
2               self.transport.write(json.dumps(self.ruang))
3       ……
4           factory = protocol.ServerFactory()
5           factory.protocol = MQTTtoweb
6           reactor.listenTCP(9000, WebSocketFactory(factory))
7           reactor.run()
8       ……
```

3. Data ditampilkan pada *browser* dengan *JavaScript*

Kemudian *browser* akan dibuat program *JavaScriptnya* yang akan melakukan *listen* terhadap *websocket server* yang dibuat pada proses sebelumnya. Setelah terhubung maka *Browser* akan menerima data json yang dikirimkan dan diolah menggunakan Bahasa Pemrograman *JavaScript*. Data tersebut berformat json. Sehingga dilakukan *encoding* terlebih dahulu. Setelah itu data akan dimuat dalam grafik disesuaikan dengan variabel yang diset dalam format json. Grafik yang ditampilkan menggunakan fitur google *chart*. Berikut *source code* untuk menghubungkan program web dan mengambil data dari *websocket*.

```
1  ……
2        if (! ("WebSocket" in window)) WebSocket =
   MozWebSocket; // firefox
3      var socket = new WebSocket("ws://localhost:9000");
4  ……
5     socket.onopen = function(event) {
6          socket.send('connected\n');
7          socket.onmessage = function(e) {
8               $("#output").text(e.data);
9               skripsi = JSON.parse(e.data);
10              google.charts.setOnLoadCallback(drawChart);
11              google.charts.setOnLoadCallback(drawChart2);
12              google.charts.setOnLoadCallback(drawChart3);
13         };
14     }
15 ……
```

Data yang didapatkan dimasukkan kedalam variabel skripsi untuk diproses kembali menjadi data-data yang dapat dibuat grafiknya. Agar grafik melakukan *update* saat program *websocket* mendapatkan data baru maka fungsi pembuatan *chart* akan dipanggil dan dimasukkan kedalam *source code* diatas. Berikut *source code* pembuatan grafik menggunakan fitur google *chart*.

```
1  ……
2  google.charts.load('current', {'packages':['corechart']});
3  function drawChart(){
4     var headers = ["Detik"];
5     var contents = [[]];
6     for (var i=1; i <= skripsi.length; i++){
7          headers[i] = skripsi[i-1].nama;
8     }
9     for (var i=0; i < 10; i++){
10         contents[i] = [];
11         contents[i][0] = i;
12         for (var j=1; j <= skripsi.length; j++){
13              contents[i][j] = skripsi[j-1].node1[i];
14          }
15         }
16    graphs.node1.data = new google.visualization.DataTable();
17    graphs.node1.data.addColumn('number','Detik');
18    for (var i=0; i < skripsi.length; i++){
```

```
19        graphs.node1.data.addColumn('number', skripsi[i].nama);
20        }
21    graphs.node1.data.addRows(contents);
22    graphs.node1.options = {
23          hAxis: {
24              title: 'Waktu'
25          },
26          vAxis: {
27              title: 'Kondisi Siaga'
28          },
29          title: 'Node1 ',
30    };
31    var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));
32    chart.draw(graphs.node1.data, graphs.node1.options);
33 ……
```

Untuk dapat ditampilkan pada *browser* dibuat sebuah file dengan format html. File ini akan dijalankan oleh *browser* dengan membukanya menggunakan *browser*. File ini merupakan code program yang akan menjalankan *javascript* serta merupakan penampil data pada *browser*. Berikut *source code* yang terdapat pada file html. Agar grafik dapat ditampilkan laptop yang digunakan untuk menampilkan data harus terhubung dengan internet. Jika tidak terhubung grafik tidak akan ditampilkan.

```
1  <html>
2     <head>
3        <!--Load the AJAX API-->
4        <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
5        <script type="text/javascript" src="./js/jquery.min.js"></script>
6        <script type="text/javascript" src="./js/graphLoader.js"></script>
7     </head>
8
9     <body>
10       <!--Div that will hold the pie chart-->
11       <div id="output"></div>
12       <div id="curve_chart"></div>
13       <div id="curve_chart2"></div>
```

```
14          <div id="curve_chart3"></div>
15      </body>
16  </html>
```

5.2.3 Implementasi Topologi

Pada saat node saling terhubung maka akan terbentuk topologi. Proses pembentukkan topologi ini berdasarkan pada proses pencarian node. Pembentukkan topologi pada sistem ini adalah sebagai berikut.

1. Gateway akan melakukan pencarian node dengan cara mengirimkan teks gate pada *channel* tertentu. Gateway juga akan melakukan *listening* setelah melakukan pengiriman data tersebut pada *channel* yang sama.
2. Node akan melakukan *listening* pada setiap *channel* yang disediakan. Setelah menerima data gate maka node akan mengirimkan data n pada *channel* yang sama saat node menerima data. Setelah node terhubung dengan gateway node mencari node lainnya dengan metode yang sama saat gateway melakukan pencarian node.
3. Node selanjutnya melakukan hal yang sama seperti pada poin no 2.
4. Node terakhir tidak akan melakukan pencarian node baru. Dan akan mengirimkan datanya menuju node yang memiliki channel yang sama dengan *channel transmit* node terakhir tersebut.
5. Topologi yang terbentuk dimulai dari gateway yang mendapatkan node ke 1. Setelah itu node ke 1 mencari node ke 2 dan seterusnya sampai bertemu node terakhir.

5.2.4 Batasan Implementasi

Dalam merancang sistem diperlukan pembatasan pada implementasi agar sistem yang diimplementasikan dengan semestinya. Batasan implementasi sistem ini adalah:

1. Node yang digunakan sebanyak 4 buah, dengan 1 sebagai gateway dan 3 node.
2. Gateway harus tersedia sebelum node dapat tersambung. Untuk node harus ada node yang telah mengarah ke gateway.
3. Data dikirim setelah semua node terhubung atau terkoneksi.
4. Monitoring data yang diterima gateway dilakukan dengan menggunakan 3 grafik sesuai dengan jumlah node yaitu 3. Ketiga grafik ditampilkan pada *browser*.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai proses pengujian system**.** Pengujian yang dilakukan adalah pengujian sensor dan pengujian menggunakan *prototype.*

6.1 Pengujian Sensor HC-SR04

Dalam sistem sensor yang digunakan adalah sensor HC-SR04. Sensor HC-SR04 adalah sensor ultrasonik. Sensor ini merupakan sensor digital yang mendapatkan data dengan cara menembakkan gelombang kemudian gelombang tersebut diterima kembali. Sensor akan menghitung waktu dari gelombang ditembakkan dan kembali. Untuk mendapatkan jarak kita menggunakan rumus jarak yaitu $S = Vt$. Kemudian nilainya dapatnya dianalisis terkait tingkat keakuratan dari sensor untuk mendeteksi jarak.

6.1.1 Tujuan Pengujian

Pengujian sensor HC-SR04 ini bertujuan untuk mengetahui seberapa besar akurasi dari sensor HC-SR04 saat melakukan *sensing* data jarak. Jarak yang dideteksi disesuaikan dengan kondisi pada *prototype* yang dibuat. Agar dapat memastikan keakuratan dari prototype yang dibuat.

6.1.2 Skenario Pengujian

Skenario pengujian dilakukan dengan cara mengarahkan sensor HC-SR04 menuju suatu benda dengan jarak yang ditentukkan. Jarak ini disesuaikan dengan jarak yang nantinya digunakan pada batas *prototype*. Jarak yang dideteksi adalah 16, 13, 11, 9 dan 7 cm. Mikrokontroller yang dihubungkan dengan sensor HC-SR04 tersebut dihubungkan dengan laptop untuk mendapatkan hasil pendeteksian jarak dari benda yang disimpan pada jarak deteksi. Data deteksi yang ditampilkan adalah setiap satu detik sekali. Pengujian dilakukan menggunakan 3 sensor ultrasonik sebanyak 15 kali pembacaan data pada setiap jarak yang dideteksi.

6.1.3 Hasil Pengujian Akurasi Sensor HC-SR-4

Hasil pengujian akurasi sensor ultrasonic dengan variasi ketinggian disesuaikan dengan kondisi status yang nantinya digunakan dalam sistem dapat dilihat pada Tabel 6.1.

Tabel 6.1 Pengujian Akurasi Sensor HC-SR04

No	Data Sensor Jarak 7 cm			Data Sensor Jarak 9 cm			Data Sensor Jarak 11 cm			Data Sensor Jarak 13 cm			Data Sensor Jarak 16 cm		
	u1	u2	u3	u1	u2	u3	u1	u2	u3	u1	u2	u3	u1	u2	u3
1	6.43	6.63	6.41	8.13	8.28	8.14	10.76	10.47	10.3	12.3	12.5	12.7	16	16.18	16.4
2	6.43	6.63	6.39	8.11	8.38	8.14	10.51	10.52	10.3	12.3	12.5	12.8	16	16.2	16.4
3	6.32	6.63	6.29	8.01	8.38	8.04	10.51	10.52	10.3	12.3	12.5	12.8	16	16.2	16.4
4	6.43	6.51	6.29	8.01	8.28	8.04	10.49	10.52	10.3	12.3	12.5	12.7	16.1	16.2	16.4
5	6.43	6.63	6.39	8.13	8.28	8.08	10.51	10.52	10.3	12.3	12.5	12.7	16	16.2	16.4
6	6.43	6.63	6.39	8.11	8.28	8.14	10.49	10.52	10.3	12.3	12.5	12.7	16	16.2	16.4
7	6.32	6.53	6.29	8.13	8.38	8.14	10.51	10.52	10.3	12.3	12.5	12.7	16	16.2	16.4
8	6.43	6.63	6.41	8.02	8.38	8.04	10.49	10.63	10.3	12.3	12.5	12.7	16	16.2	16.4
9	6.44	6.61	6.39	8.01	8.28	8.02	10.49	10.63	10.3	12.3	12.6	12.7	16	16.2	16.4
10	6.32	6.53	6.29	8.13	8.28	8.13	10.51	10.63	10.3	12.4	12.6	12.7	16	16.2	16.4
11	6.44	6.53	6.39	8.11	8.28	8.14	10.49	10.63	10.3	12.4	12.6	12.8	16	16.2	16.4
12	6.44	6.63	6.39	8.11	8.38	8.14	10.51	10.63	10.3	12.4	12.5	12.8	16	16.2	16.4
13	6.32	6.63	6.29	8.01	8.4	8.02	10.49	10.29	10.3	12.3	12.5	12.7	16	16.2	16.4
14	6.32	6.53	6.29	8.02	8.28	8.04	10.39	10.27	10.3	12.3	12.5	12.7	16.1	16.2	16.4
15	6.43	6.63	6.39	8.11	8.28	8.14	10.39	10.27	10.3	12.3	12.5	12.7	16	16.2	16.4

6.1.4 Analisis Hasil Pengujian Akurasi Sensor HC-SR04

Berdasarkan pengujian tersebut, dapat dianalisis bahwa sensor HC-SR04 yang mendeteksi jarak dapat mengukur dengan baik. Hal tersebut ditunjukkan pada analisis pengujian di Tabel 6.2.

Tabel 6.2 Analisis hasil pengujian tingkat akurasi Sensor HC-SR04

No	Jarak Sebenarnya (cm)	Rata-rata Data Sensor			ketepatan sensor %		
		u1	u2	u3	u1	u2	u3
1	7	6.40	6.59	6.35	91.36	94.20	90.75
2	9	8.08	8.32	8.09	89.74	92.46	89.92
3	11	10.50	10.50	10.31	95.48	95.50	93.77
4	13	12.29	12.53	12.68	94.53	96.38	97.51
5	16	16.02	16.19	16.40	99.88	98.82	97.57

Sensor yang digunakan yaitu HC-SR04 memiliki kemampuan yang baik dalam membaca data jarak. Akan tetapi performanya semakin menurun dengan semakin dekatnya jarak yang dideteksi. Meskipun tingkat akurasinya menurun sensor ini masih memiliki ketepatan yang tinggi yaitu masih dalam besaran 90%. Untuk kondisi terburuk dalam pengujian ini adalah pada saat jarak yang dideteksi berada pada jarak 9 cm. Pada jarak ini sensor memiliki ketepatan terendah yaitu 89.92%. untuk jarak terbaik berada pada jarak 16 cm dengan ketepatan 97.57%. Akurasi yang tidak mencapai 100% ini salah satunya disebabkan oleh kurang sesuaiannya kecepatan dari ultrasonik. Kecepatan yang digunakan adalah 340 m/s. Kecepatan ini adalah pada saat suhu udara 15^{o} C.

6.2 Pengujian Level Ketinggian Air

Pengujian ini dilakukan menggunakan *prototype* sebuah talang air dengan panjang 100 cm, tinggi 10 cm dan lebar 13,5 cm. Sensor ditempatkan pada ketinggian 15.5 cm dari dasar talang air. Batasan untuk setiap state adalah 2cm dari 3 sampai 9 cm. Nilai 3 sampai 9 cm didapat dari ketinggian prototipe talang air agar tidak terjadi luapan. Sehingga untuk kondisi siaga 4 sensor harus mendeteksi ketinggian 13-11cm, siaga 3 ketinggian 11-9cm siaga 2 ketinggian 9-7cm siaga 1 diatas 7cm.

6.2.1 Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui apakah level ketinggian air dapat dideteksi dengan baik.

6.2.2 Skenario Pengujian

Pengujian dilakukan dengan mengisi air pada prototype dengan ketinggian yang ditentukkan. Ketinggian dari air ini adalah 2.5, 4.5, 7.5, 8.5 dan 9.5 cm. Ketinggian air ini diukur dengan meteran atau penggaris yang memiliki skala

ukuran. Ketinggian air akan dideteksi dan akan dijadikan status pada serial monitor. Sensor akan diletakkan diatas talang air yang akan diisi air. Pada saat ketinggian air naik sensor akan mendapatkan jarak yang semakin kecil dengan semakin mendekatnya jarak antara sesor dan permukaan air. Level siaga didapatkan dari jarak yang terdeteksi antara sensor dan permukaan air.

6.2.3 Hasil Pengujian Level Ketinggian Air

Hasil pengujian level ketinggian air pada *prototype* dapat dilihat padaTabel 6.3 berikut ini.

Tabel 6.3 Hasil Pengujian Level Ketinggian Air

No	Sensor	Ketinggian air (cm)	Pembacaan Sensor	Kondisi Seharusnya	Hasil Pengujian
1	1	2.5	Siaga 4	Siaga 4	Tepat
2	1	4.5	Siaga 3	Siaga 3	Tepat
3	1	7.5	Siaga 2	Siaga 2	Tepat
4	1	8.5	Siaga 2	Siaga 1	Salah
5	1	9.5	Siaga 1	Siaga 1	Tepat
6	2	2.5	Siaga 4	Siaga 4	Tepat
7	2	4.5	Siaga 3	Siaga 3	Tepat
8	2	7.5	Siaga 2	Siaga 2	Tepat
9	2	8.5	Siaga 2	Siaga 1	Salah
10	2	9.5	Siaga 1	Siaga 1	Tepat
11	3	2.5	Siaga 4	Siaga 4	Tepat
12	3	4.5	Siaga 3	Siaga 3	Tepat
13	3	7.5	Siaga 2	Siaga 2	Tepat
14	3	8.5	Siaga 2	Siaga 1	Salah
15	3	9.5	Siaga 1	Siaga 1	Tepat

6.2.4 Analisis Hasil Pengujian Level Ketinggian Air

Berdasarkan pengujian tersebut didapatkan hasil analisis sebagai berikut. Air dapat memantulkan gelombang ultrasonik, sehingga didapatkan jarak yang dapat diperiksa oleh sensor. Setiap jarak yang didapatkan sesuai dengan kondisi yang diinginkan. Hal tersebut ditunjukkan pada Tabel 6.4 berikut.

Tabel 6.4 Analisis hasil pengujian Level Ketinggian Air

No	Sensor	Nilai sesuai	Nilai tidak sesuai	Ketepatan data
1	1	4	1	80%
2	2	4	1	80%
3	3	4	1	80%

Ketidak sesuaian *output* dengan kondisi yang seharusnya terjadi pada level ketinggian yang sama yaitu saat ketinggian air 8.5 cm. Sensor seharusnya memberikan data 6.8, akan tetapi sensor memberikan data 7.3. Sehingga status tetap berada pada siaga 2 tidak naik ke siaga 1. Seperti pada pengujian sebelumnya kondisi sensor pada jarak 7 cm memiliki *error* yang tertinggi sehingga ketika dilakukan pengujian dekat dengan perubahan level ketinggian akan terjadi *error*.

6.3 Pengujian Pengiriman Data

Pengujian yang dilakukan menggunakan modul komunikasi nRF24L01. Pengujian menggunakan *routing point-to*-point menggunakan 3 buah node menuju sebuah gateway.

6.3.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui apakah data yang diterima gateway sesuai dengan data yang dikirim oleh node. Selain itu pengujian ini juga bertujuan untuk mengetahui pembentukkan node serta apakah node telah terhubung sesuai dengan perancangan dan implementasi yang dilakukan.

6.3.2 Skenario Pengujian

Pengujian ini dilakukan pertama kali dengan menyambungkan gateway dengan setiap node. Pertama gateway akan dinyalakan dan dihubungkan dengan node yang akan menjadi node satu. Setelah terhubung maka node selanjutnya dinyalakan hingga node terakhir. Node terakhir akan melakukan pengiriman pertamakali. Kemudian diterima oleh node selanjutnya. Node selanjutnya akan mengirimkan datanya dan data yang diterima dari node sebelumnya hingga sampai di gateway. Ketika data sampai digateway maka akan ditampilkan dan dilihat apakah data sesuai dengan apa yang dikirimkan oleh node.

6.3.3 Hasil Pengujian Pengiriman Data

Hasil pengujian pengiriman data dengan menggunakan metode *point-to-point* dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian Pengiriman Data

No	Data Node 3	Data Node 2	Data Node 1	Gateway	Hasil Pengujian
1	0	0	0	000	Tepat
2	1	2	3	123	Tepat
3	2	1	1	211	Tepat
4	2	0	0	200	Tepat
5	0	1	0	010	Tepat
6	0	0	3	003	Tepat
7	1	0	1	101	Tepat
8	0	1	1	011	Tepat
9	4	1	1	411	Tepat
10	1	4	3	143	Tepat

6.3.4 Analisis Hasil Pengujian Pengiriman Data

Dari hasil pengujian pengiriman diatas, dapat dianalisis bahwa setiap data yang dikirimkan oleh masing-masing node dapat diterima dengan sesuai oleh gateway. Hasil analisis tersebut dapat dilihat pada Tabel 6.6.

Tabel 6.6 Ananlisis Hasil Pengujian Pengiriman Data

Jumlah Data yang diterima salah	0
Jumlah Data yang diterima benar	10
Tingkat akurasi data	100%

Tidak adanya kesalahan pada penerimaan data ini disesbabkan oleh pembentukkan hubungan dengan node dapat berjalan sesuai dengan hasil perancangan dan implementasi. Serta setiap node dapat menerima dan mengirim pada *channel* yang berbeda.

6.4 Pengujian Sistem Pada Prototype

Pengujian dilakukan pada prototipe sebuah talang air dengan panjang 100 cm, lebar 11 cm dan tinggi 10,5 cm. Node yang digunakan adalah 3 node. 2 node berada pada tempat yang sama sedangkan 1 node berada pada jarak 100 cm dari 2 node tadi. Dengan *routing point-to-point* kita akan membuat node yang berdekatan menjadi node 1 dan 3 sedangkan node 2 nya adalah 1 node yang terpisah 100 cm dengan begitu jarak tiap node dapat dikatakan 100 cm. Pada sistem ini node 3 akan mengirim datanya menuju node 2. Node 2 akan mengirim datanya beserta data dari node 3. Sedangkan node 1 akan mengirimkan datanya dan data dari node 2, yaitu data node 2 dan node 3 menuju gateway.

6.4.1 Tujuan Pengujian

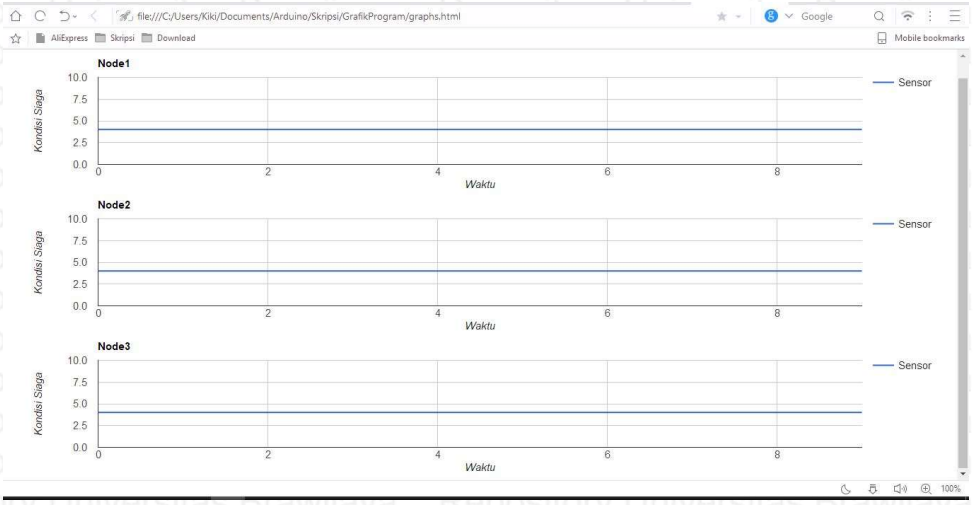
Pengujian ini bertujuan untuk mengetahui apakah sistem dapat berjalan sesuai dengan fungsinya. Serta berapa banyak node yang nantinya dibutuhkan pada kondisi yang sebenarnya berdasarkan hasil pengujian yang akan didapatkan. Pada pengujian ini *prototype* yaitu talang air akan diisi dengan air. Pada saat ketinggian air mencapat kondisi yang ditentukan maka akan dilihat apakah data yang dikirim sesuai atau tidak.

6.4.2 Skenario Pengujian

Pengujian ini dilakukan dengan mengisi air pada talang air secara berkala. Gateway akan dinyalakan dan dihubungkan dengan laptop dan diambil datanya secara serial. Node akan dinyalakan satu-persatu setalah gateway menyala. Gateway yang terhubung dengan laptop akan dibaca datanya secara serial kemudian ditampilkan hasil nya dengan menggnakan grafik. Grafik yang ditampilkan dibuat dengan program python dan menggunakan fitur mqtt untuk dapat membuat grafik menjadi realtime atau melakukan update setiap kali gateway menerima data dari node.

6.4.3 Hasil Pengujian Sistem Pada Prototipe

Pada pengujian ini hasil pengujiannya merupakan terbentuknya grafik yang sesuai dengan kondisi yang diberikan oleh sensor. Pada sistem ini memiliki 4 kondisi status ketinggian air sehingga akan ditampilkan 4 grafik. Setiap grafik akan menampilkan status dari data yang dibaca oleh sensor node. Setiap node di sesuaikan dengan kondisi status ketinggian air. Gambar 6.1 menunjukkan grafik pada saat status ketinggian air berada pada siaga 4.

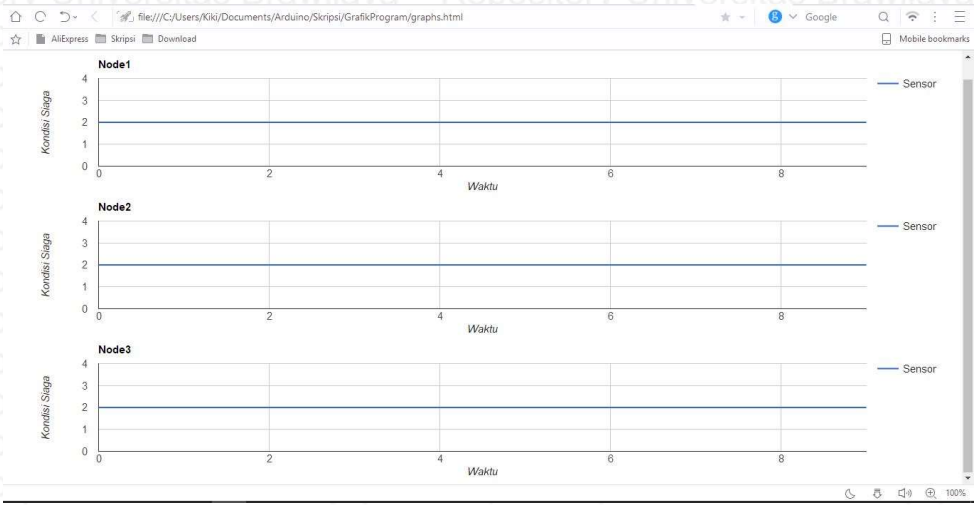


Gambar 6.1 Grafik Siaga 4

Gambar 6.2 di bawah ini adalah gambar pada saat status ketinggian air berada pada kondisi siaga 3

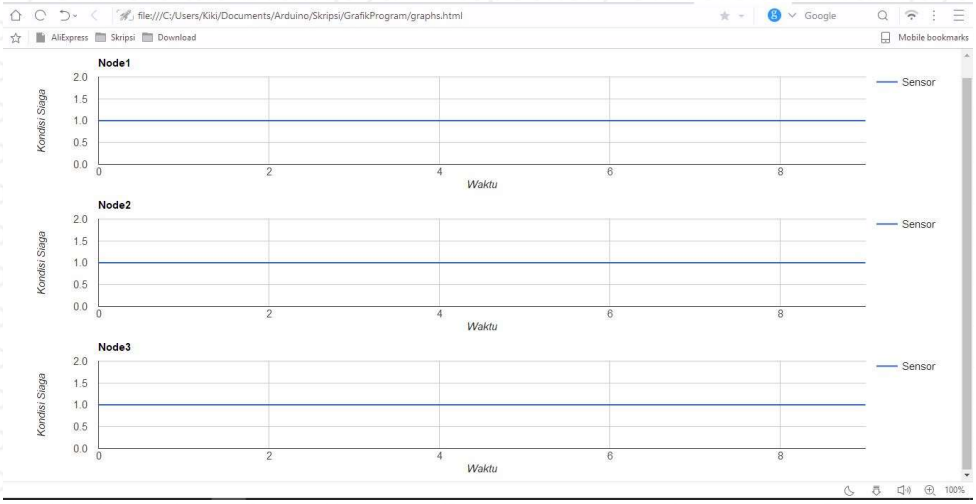
Gambar 6.2 Grafik Siaga 3

Gambar 6.3 di bawah ini adalah gambar pada saat status ketinggian air berada pada kondisi siaga 2



Gambar 6.3 Grafik Siaga 2

Gambar 6.4 di bawah ini adalah gambar pada saat status ketinggian air berada pada kondisi siaga 1



Gambar 6.4 Grafik Siaga 1

6.4.4 Analisis hasil pengujian sistem pada prototipe

Dari hasil pengujian yang dilakukan disubbab sebelumnya didapatkan hasil analisa sebagai berikut. Setiap node akan memberikan kondisinya masing-masing ketika status node tersebut telah sampai pada status siaga. Hal ini dapat dilihat pada Gambar 6.1 sampai 6.4 yang memiliki perbedaan nilai pada bagian akhir atau pada bagian awal grafik. Ketika nilai berada pada 0 atau blum memasuki kondisi siaga maka grafik akan menunjukkan pada nilai 0. Gambar 6.1 sampai 6.4 berhasil menampilkan grafik dikarenakan sistem telah berhasil menerapkan fitur mqtt yang mengirmkan datanya menggunakan fitur *websocket* dan ditampilkan dalam bentuk grafik dengan menggunakan google *chart*.

BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

7.1 Kesimpulan

Berdasarkan perancangan, implementasi, hasil pengujian dan analisis yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Pengimplementasian sistem pendeteksian ketinggian air dilakukan dengan merancang dan mengimplementasi perangkat keras meliputi arduino nano, modul nrf24l01, dan sensor HC-SR04. Sedangkan perancangan perangkat lunak dimulai dengan pembentukkan node, pengiriman data, pengambilan data secara serial dan menampilkan data dalam bentuk grafik.
2. Sensor HC-SR04 dapat digunakan sebagai pendeteksian ketinggian karena memiliki tingkat akurasi lebih dari 89%.
3. Data node diperoleh dari sensor HC-SR04 yaitu sensor ultrasonik dengan mengukur jarak antara sensor dan permukaan air.
4. *Routing point-to-point* diterapkan dengan cara node melakukan pengiriman data pada suatu *channel*. Node yang akan menjadi tetangganya akan melakukan *listening* pada beberapa channel termasuk *channel* node melakukan pengiriman data. Setalah data diterima maka node yang akan dihubungkan akan mengirimkan balasan pada *channel* tersebut dan menjadikan channel tersebut sebagai *channel transmit* node baru tersebut. Sedangkan untuk node pengirim data akan menjadikan *channel* tersebut sebagai *channel receive*.
5. Pengiriman dilakukan dari node terakhir yang terhubung dan mengirimkannya menuju *channel transmit* menuju node selanjutnya. Node tersebut akan mengirimkan datanya bersama dengan data dari node sebelumnya yang ia terima. Pengiriman ini terus dilakukan hingga data sampai pada gateway. Berdasarkan hasil pengujian yang dilakukan data diterima 100%. Seluruh data yang dikirimkan node dapat diterima oleh gateway.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan selanjutnya terhadap penelitian yang dilakukan ini adalah :

1. Gateway dapat terhubung dengan internet agar data yang didapatkan dapat langsung dikirim menuju internet. Sehingga monitoring dapat dilakukan dimana saja.
2. Penelitian selanjutnya dapat menggunakan metode lain dan menambahkan metode *low power* dalam penellitiannya.

3. Penelitian selanjutnya dapat menggunakan *library* lain yang dapat mengatasi pengiriman data yang lebih besar, sehingga dapat menambah jumlah maksimum node yang dapat dibentuk.

DAFTAR PUSTAKA

Archtz, 2015. *Mengenal Arduino Nano.* [Online] Tersedia di: http://archtz.wordpress.com/2015/05/12/mengenal-arduino-nano [Diakses 4 Oktober 2016].

Arduino, 2016. *Arduino Nano.* [Online] Tersedia di: https://www.arduino.cc/en/Main/ArduinoBoardNano [Diakses 4 Oktober 2016].

BPADJakarta, 2015. *Daerah Aliran Sungai (DAS) DKI Jakarta.* [Online] Tersedia di: http://jakartapedia.bpadjakarta.net/index.php/Daerah_Aliran_Sungai_(DAS)_DKI_Jakarta [Diakses 21 Juni 2017].

Dargie, W. & Poellabauer, C., 2010. *FUNDAMENTALS OF WIRELESS SENSOR NETWORKS : THEORY AND PRACTICE.* 1st penyunt. Chichester: John Wiley & Sons Ltd..

Desai, P., 2015. *Python Programming for Arduino.* Birmingham: Packt Publishing Ltd.

Fonseca, R. et al., 2005. *Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless.* Boston, Proceedings of the 2nd Symposium on Networked Systems Design and Implementation.

Freaks, E., 2015. *2.4G Wireless nRF24L01p with PA and LNA.* [Online] Tersedia di: http://www.elecfreaks.com/wiki/index.php?title=2.4G_Wireless_nRF24L01p_with_PA_and_LNA [Diakses 5 Oktober 2016].

Google, 2017. *Google Chart.* [Online] Tersedia di: https://developers.google.com/chart/interactive/docs/ [Diakses 5 Juni 2017].

ilmugeografi.com, 2016. *24 Ciri-ciri Sungai Bagian Hulu dan Hilir – Daerah Aliran Sungai.* [Online] Tersedia di: http://ilmugeografi.com/ilmu-bumi/sungai/ciri-ciri-sungai-bagian-hulu-dan-hilir [Accessed 7 Juli 2017].

Json.org, 2013. *JSON.* [Online] Tersedia di: http://www.json.org/ [Diakses 9 Juni 2017].

Kaazing Corporation, 2017. *About HTML5 WebSocket.* [Online] Tersedia di: https://www.websocket.org/aboutwebsocket.html [Diakses 6 Juni 2017].

Kompas.com, 2016. *Penataan Sungai-sungai di Jakarta Kian Mendesak.* [Online]
Tersedia di: http://megapolitan.kompas.com/read/2016/08/29/17000001/penataan.sungai-sungai.di.jakarta.kian.mendesak
[Diakses 22 Juni 2017].

Kompas.com, 2017. *Ini Penyebab Banjir Jakarta Menurut UPC.* [Online]
Tersedia di: http://properti.kompas.com/read/2017/02/21/193000321/ini.penyebab.banjir.jakarta.menurut.upc
[Diakses 22 Juni 2017].

Kompas, 2010. *Banjir Akibat Pendangkalan Sungai.* [Online]
Tersedia di: http://regional.kompas.com/read/2010/09/02/16414331/Banjir.Akibat.Pendangkalan.Sungai
[Diakses 1 Oktober 2016].

Kumparan.com, 2017. *Bagaimana Tinggi Banjir Diukur? Apa Bedanya Siaga I, II, III dan IV?.* [Online]
Tersedia di: https://kumparan.com/maria-duhita/bagaimana-tinggi-banjir-diukur-apa-bedanya-siaga-i-ii-iii-dan-iv
[Accessed 16 Maret 2017].

MAJDI, K., 2013. *GELOMBANG ULTRASONIK DALAM BERBAGAI BIDANG.* [Online]
Tersedia di: http://sains-resources.blogspot.co.id/2013/07/gelombang-ultrasonik-dalam-berbagai.html
[Diakses 22 Juni 2017].

Martalia, A., 2016. KALIBRASI SENSOR ULTRASONIK HC-SR04 SEBAGAI SENSOR PENDETEKSI JARAK PADA PROTOTIPE SISTEM PERINGATANDINI BENCANA BANJIR. *Prosiding Seminar Nasional Fisika (E-Journal) SNF2016,* Volume V, pp. SNF2016-43 - SNF2016-46.

Nordic, S., 2007. *nRF24L01 Single Chip 2.4GHz Transceiver Product Spesification.[pdf].* [Online]
Tersedia di: https://www.nordicsemi.com/chi/nordic/content_download/2730/34105/file/nRF24L01_Product_Specification_v2_0.pdf
[Diakses 4 Oktober 2016].

Ortiz, J., Moon, C. B. D. & Fonseca, R., 2007. *Beacon location service: a location service for point-to-point routing in wireless.[pdf].* [Online]
Tersedia di: http://cs.brown.edu/~rfonseca/pubs/bls-spots07.pdf
[Diakses 5 Oktober 2016].

python, 2001. *python.org.* [Online] Tersedia di: https://www.python.org/doc/essays/blurb/ [Diakses 6 Juni 2017].

Teachengineering, 2013. *Measuring Distance With Sound Waves.* [Online] Tersedia di: https://www.teachengineering.org/activities/view/nyu_soundwaves_activity1 [Diakses 5 Oktober 2016].

TechTarget, 2005. *IoT Agenda.* [Online] Tersedia di: http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport [Diakses 6 Juni 2017].

Vannesa, W., Salim, F. & Moskovits, P., 2013. *The Definitive Guide to HTML5 WebSocket.* First penyunt. New York: Paul Manning.