PERANCANGAN SISTEM DETEKSI PELANGGARAN RAMBU LALU LINTAS MENGGUNAKAN *WIRELESS SENSOR NETWORK*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Alvin Candra Wijaya
NIM:125150302111001

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

PENGESAHAN

PERANCANGAN SISTEM DETEKSI PELANGGARAN RAMBU LALU LINTAS MENGGUNAKAN *WIRELESS SENSOR NETWORK*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Alvin Candra Wijaya
NIM:125150302111001

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Februari 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Edita Rosana Widasari , S.T, M.T., M.Eng	Gembong Edhi Setyawan, S.T, M.T.
NIK: 201606 910626 2 001	NIK:201208 761201 1 001

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 diter*bit*kan 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 Februari 2017

Alvin Candra Wijaya

NIM: 125150302111001

KATA PENGANTAR

Puja dan puji syukur kami panjatkan kehadirat Allah SWT yang mana atas limpahan rahmat, taufik serta hidayah-Nya, skripsi ini dapat terselesaikan. Skripsi yang berjudul "**Perancangan Sistem Deteksi Pelanggaran Rambu Lalu Lintas Menggunakan *Wireless Sensor Network***" ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer.

Penulis menyadari bahwa penyusunan skripsi ini terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada pihak-pihak yang telah bersedia untuk memberikan bantuan demi kelancaran penyusunan skripsi ini diantaranya:

1. Ibu Edita Rosana Widasari , S.T., M.T., M.Eng dan Bapak Gembong Edhi Setyawan, S.T, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer.
3. Ayahanda dan Ibunda serta seluruh keluarga besar yang saya cintai atas segala dukungan dan semangat serta tiada henti-hentinya memberikan do'a demi terselesaikannya skripsi ini.
4. Seluruh civitas akademik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Rekan-rekan Sistem Komputer 2012 yang selalu memberikan motivasi dalam menyelesaikan skripsi ini.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 1 Februari 2017

Penulis

Alvin Candra Wijaya

ABSTRAK

Intelligent Transportation System (ITS) adalah sistem yang digunakan untuk memberikan kecerdasan buatan pada sistem transportasi. Layanan yang dimiliki oleh ITS sendiri sangat banyak dan menyesuaikan sesuai kebutuhan informasi yang diperlukan pengguna sistem. Salah satu pengguna sistem yang dapat memanfaatkan informasinya adalah Polisi Lalu Lintas (POLANTAS). Informasi yang diperlukan adalah pelanggaran pada rambu, karena angka pelanggaran rambu lalu lintas oleh kendaraan bermotor masih relatif tinggi. Dari masalah tersebut maka dibuatlah sistem yang dapat mendeteksi pelanggaran dan mengirimkannya secara *wireless*. Dalam penelitian ini digunakan empat buah node yaitu dua buah *Road Marking Unit* (RMU) yang diletakkan pada rambu lalu lintas, satu buah *Road Side Unit* (RSU) untuk menjembatani antara RMU dan OPS, serta satu buah *Open Platform Server* (OPS) yang diletakkan pada komputer server. Node-node tersebut akan saling bertukar informasi dengan menggunakan perangkat nRF24L01. Metode transfer data yang digunakan untuk komunikasi antar node dilakukan secara *singlehop* dan *multihop*. Dimana pada hasil pengujian pengiriman data tersebut meliputi *Latency*, *Packet loss*, *Throughput* dan *Jitter.* Pada hasil pengujian menunjukan bahwa pengiriman data *singlehop* menghasilkan nilai *Latency* 1002.22 – 1002.70 ms, *Packet loss* 0 %, *Throughput* 1.00 dps, dan *Jitter* 0.94 – 1.36 ms. Sedangkan pada *multihop* menghasilkan nilai *Latency* 1006.54 – 1007.22 ms, *Packet loss* 0%, *Throughput* 0.99 dps, dan *Jitter* 1.28 – 1.96. Pengiriman data menggunakan nRF24L01 dirasa sangat efektif selama penggunaannya dalam jarak sesuai dengan spesifikasinya dan tidak terdapat banyak halangan. Hal ini ditunjukkan dengan tidak adanya data pelanggaran yang hilang saat proses pengiriman data.

Kata kunci: *Intelligent Transportation System, Wireless Sensor Network,* rambu lalu lintas

ABSTRACT

Intelligent Transportation System (ITS) is a system used to provide artificial intelligence in the transportation system. The service owned by ITS itself is very much and number of service will grow in accordance from the information user needed. One of the users of the system to take advantage of the information is the Traffic Police. The necessary information is a violation of the signs, because the number of violations of traffic signs by vehicles is still relatively high. From that problem can invented a system that can detect violations and send it by wireless. This research used four nodes : two Road Marking Unit (RMU) *are placed on traffic signs, one Road Side Unit* (RSU) *to bridge between the* RMU *and* OPS*, and also one Open Platform Server* (OPS) *that is placed on the server computer. Nodes will exchange information using the nRF24L01. Data transfer method for communication between nodes is used singlehop and multihop. Where on the test results of the data transfer includes Latency, Packet loss, Throughput, and Jitter. In the test results indicate that the data transmission singlehop generate value Latency 1002.22 - 1002.70 ms, Packet loss 0 %, Throughput 1.00 dps, and Jitter 0.94 – 1.36 ms. While in multihop generate value Latency 1006.54 - 1007.22 ms, Packet loss 0 %, Throughpu 0.99 dps, and Jitter 1.28 - 1.96 ms. Send data using nRF24L01 is considered very effective as long as distance between node still in specification range and there are not many obstacle. This is indicated by 0 % data was lost during the data transmission process.*

Keywords: Intelligent Transportation System, Wireless Sensor Network, traffic signs

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Intelligent Transportation System (ITS) adalah sebuah sistem yang membuat sistem transportasi memiliki kecerdasan. ITS merupakan integrasi dari penerapan teknologi di bidang elektronika, komputer dan telekomunikasi guna menyediakan informasi kepada pengguna. Sistem ini melibatkan kendaraan, pengendara, rambu lalu lintas dan perangkat komunikasi yang dihubungkan satu dengan lainnya untuk meningkatkan efisiensi. Pengelompokan layanan yang disediakan oleh ITS meliputi manajemen transportasi, manajemen informasi, manajemen transportasi umum, *e-payment*, manajemen kendaraan pribadi, manajemen darurat dan kontrol kendaraan dan keamanan. Layanan dari setiap kelompok dapat berubah sewaktu-waktu apabila layanan baru ditambahkan. Layanan baru akan ditambahkan sesuai kebutuhan informasi yang diperlukan oleh pengguna (Mathew, T., 2014).

Berbagai pihak dapat menggunakan informasi yang didapatkan oleh ITS seperti contohnya adalah pengguna jalan, kepolisian dan rumah sakit. Beberapa informasi yang dapat dimanfaatkan oleh pihak kepolisian meliputi data rambu lalu lintas, data kendaraan dan data kemacetan. Selanjutnya, data-data ini dapat digunakan untuk mempermudah tugas kepolisian dalam mengurai kemacetan, mendeteksi kecelakaan antar kendaraan, mendeteksi pelanggaran rambu lalu lintas dan lain sebagainya.

Salah satu penelitian yang telah mengimplementasikan ITS pada rambu lalu lintas yaitu "Perancangan Aplikasi Komputer Pusat Untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara *Wireless*". Penelitian tersebut berfokus pada komunikasi data antara kontrol terpusat dan lampu lalu lintas dinamis. Komunikasi data yang digunakan adalah secara *wireless*. Sehingga, pengguna dapat mengontrol lampu lalu lintas baik manual atau otomatis melalui *user interface* yang terdapat di komputer pusat (Faisal, A., 2016).

Dari penelitian di atas maka dapat dikembangkan implementasi ITS lebih lanjut. Penelitian dapat dikembangkan dengan menambahkan rambu lalu lintas lainnya. Pada penelitian ini akan ditambahkan rambu larangan, karena rambu jenis ini masih sering dilanggar. Rambu larangan yang akan ditambahkan meliputi rambu 'dilarang putar arah', 'dilarang berhenti', dan 'dilarang masuk'.

Angka pelanggaran rambu lalu lintas setiap tahunnya semakin meningkat seperti yang ditunjukan pada Tabel 1.1. Banyak faktor yang menyebabkan pelanggaran rambu lalu lintas, seperti kelalaian pengendara, kesengajaan, atau pun ketidak tahuan arti dari rambu. Apabila hal ini terus dilakukan dapat merugikan dan membahayakan pengendara maupun orang lain.

Tabel 1.1 Jumlah Pelanggaran Rambu Lalu Lintas Sepeda Motor Dan Mobil Di Kota Malang

Tahun	Sepeda Motor	Mobil
2009	3.510	286
2010	4.479	354
2011	6.031	927

Sumber: (Satlantas Polres Malang, 2011)

Dalam perancangan ITS diperlukan minimal tiga buah node, yaitu *Road Marking Units* (RMU), *Road Side Units* (RSU) dan *Open Platform Server* (OPS). Penelitian sebelumnya menggunakan tiga node, yaitu node lampu lalu lintas, node forwarder, dan node komputer server. Jumlah minimal node ini sudah cukup karena hanya mengontrol satu lampu lalu lintas. Sedangkan, pada penelitian ini akan menambahkan tiga rambu larangan maka node RMU perlu ditambahkan.

Penelitian ini akan menggunakan dua buah perangkat RMU, yaitu RMU 1 sebagai pendeteksi pelanggaran rambu putar arah dan dilarang berhenti, kemudian RMU 2 sebagai pendeteksi pelanggaran rambu dilarang masuk. Digunakan juga satu buah perangkat RSU yang digunakan sebagai jembatan antara RMU dan OPS apabila jarak terlalu jauh. Selain itu juga digunakan satu perangkat OPS, yaitu OPS yang berperan sebagai penerima data akhir. Semua node ini akan saling berhubungan satu sama lainnya menggunakan teknologi *wireless sensor network*.

Berdasarkan uraian di atas maka dengan adanya sistem deteksi pelanggaran rambu lalu lintas yang menggunakan komunikasi data secara *wireless sensor network* dapat memudahkan dalam mengetahui pelanggaran rambu lalu lintas tanpa harus mendatangi lokasi rambu tersebut. Dengan demikian, lokasi pelanggaran rambu dapat segera diketahui dan diharapkan dapat digunakan sebagai acuan dalam mengurangi tingkat pelanggaran rambu lalu lintas.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan beberapa masalah sebagai berikut.

1. Bagaimana merancang sistem pendeteksian pelanggaran rambu lalu lintas menggunakan *wireless sensor network* dengan perangkat Arduino Nano dan nRF24L01?
2. Bagaimana akurasi pendeteksian pelanggaran rambu lalu lintas menggunakan sensor *Ultrasonic* HC-SR04 dan *Light Dependant Resistor*?
3. Bagaimana performa metode transfer data yang digunakan untuk komunikasi antar node?
4. Bagaimana performa pengiriman data menggunakan *wireless sensor network* pada keseluruhan sistem?

1.3 Tujuan

Tujuan dari penelitian ini antara lain adalah sebagai berikut.

1. Merancang sistem pendeteksian pelanggaran rambu lalu lintas menggunakan *wireless sensor network* dengan perangkat Arduino Nano dan nRF24L01.
2. Menguji tingkat keberhasilan sensor *Ultrasonic* HC-SR04 dan *Light Dependant Resistor* dalam mendeteksi pelanggaran rambu lalu lintas.
3. Mengetahui performasi metode transfer data yang digunakan untuk komunikasi antar node.
4. Mengetahui tingkat keberhasilan pengiriman data pelanggaran rambu lalu lintas menggunakan *wireless* sensor network.

1.4 Manfaat

Manfaat yang ingin diperoleh dari hasil penelitian ini antara lain adalah sebagai berikut.

1. Bagi kepolisian, dapat membantu dalam menentukan ada tidaknya pelanggaran hanya dengan melihat komputer server tanpa harus melihat langsung lokasi kejadian.
2. Bagi pembaca, dapat memahami konsep kerja dari *Intelligent Transportation System* (ITS) terutama pada sistem deteksi pelanggaran lalu lintas.
3. Bagi penulis, dapat mengimplementasikan teori-teori mengenai *Intelligent Transportation System* (ITS) menjadi sebuah sistem deteksi pelanggaran lalu lintas.

1.5 Batasan Masalah

Batasan masalah pada penlitian ini antara lain adalah sebagai berikut.

1. Perancangan sistem deteksi pelanggaran lalu lintas menggunakan *wireless sensor network* hanya terbatas pada *prototype.*
2. Algoritma sensing hanya terbatas pada kendaraan bermotor.
3. Rambu larangan yang digunakan meliputi rambu ‘dilarang putar arah’, ‘dilarang berhenti’ dan ‘dilarang masuk’.
4. Komunikasi data antar node menggunakan *wireless sensor network* pada level *singlehop* dan *multihop*.
5. *User interface* pada sistem menampilkan grafik, tabel *database*, jumlah pelanggaran dan *ping* dari masing-masing RMU.
6. Pengujian pengiriman data meliputi *Latency*, *Packet loss*, *Throughput* dan *Jitter.*

1.6 Sistematika Pembahasan

Penulisan skripsi ini di bagi dalam 6 bab yang masing-masing akan dijelaskan secara singkat di bawah ini :

BAB I PENDAHULUAN

Bab ini terdiri dari latar belakang penellitian, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika pembahasan

BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas tentang penelitian – penelitian *Intelligent Transportation System* (ITS) yang mendukung penelitian mengenai Pembangunan Sistem Deteksi Pelanggaran Rambu Lalu Lintas Menggunakan *Wireless Sensor Network*.

BAB III METODE PENELITIAN

Bab ini berisi pembahasan tentang metode yang dipakai pada penelitian ini.

BAB IV ANALISIS KEBUTUHAN

Bab ini membahas mengenai deskripsi umum dari sistem, kebutuhan antarmuka ekternal, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

BAB V PERANCANGAN DAN IMPLEMENTASI

Di dalam bab ini dibahas perancangan sistem serta implementasi sistem berupa perangkat keras dan perangkat lunak.

BAB VI PENGUJIAN dan ANALISIS

Di dalam bab ini membahas langkah kerja dalam melaksanakan pengujian dan analisis.

BAB VII PENUTUP

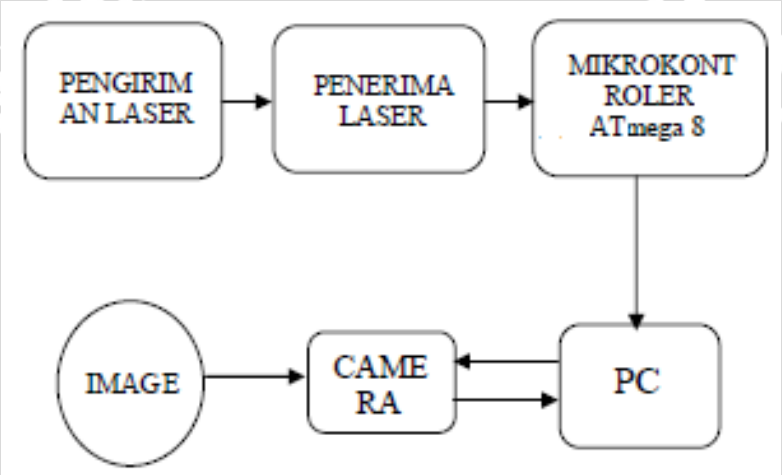
Bab ini membahas mengenai kesimpulan dan saran yang diperoleh dari rumusan masalah penelitian dengan melakukan analisis dan pengujian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka membahas tentang penelitian-penelitian yang sudah ada sebelumnya. Penelitian yang sudah ada kemudian dikaji hasilnya dan dijadikan sebagai dasar dalam pelaksanaan penelitian ini. Pada penelitian ini digunakan tiga hasil penelitian sebelumnya yang dapat mendukung penelitian.

Penelitian pertama ditulis oleh Andri Alfian, Ronny Susetyoko dan Eru Puspita dengan judul "Alat Pendeteksi Pelanggaran Marka Lalu-Lintas Dengan Indikasi Jumlah Pelanggar". Sistem pada penelitian ini dibuat untuk melakukan deteksi pelanggaran marka pada lampu rambu lalu lintas dan kemudian menampilkan jumlah pelanggarn pada *user interface*. Berikut gambar mengenai prinsip kerja sistem :



Gambar 2.1 Prinsip Kerja Alat Pendeteksi Pelanggaran Marka Lalu-Lintas Dengan Indikasi Jumlah Pelanggar

Sumber: (Alfian, et al.)

Pada gambar 2.1 dapat dilihat peneliti memanfaatkan sensor laser untuk melakukan deteksi pelanggaran. Ketika ada pelanggaran terdeteksi kemudian Atmega 8 memberikan perintah pada kamera melalui PC untuk menangkap gambar. Gambar kemudian diproses pada *Visual Basic* untuk menghitung jumlah pelanggar marka. Kekurangan dari penelitian ini adalah tidak adanya komunikasi data secara *wireless*, sehingga pelanggaran tidak dapat terdeteksi dari jarak jauh.

Penelitian kedua ditulis oleh V. Ramya, B. Palaniappan dan M. Aruljothi dengan judul "*Embedded System for Automatic Traffic Violation Monitoring and Alerting*". Sistem pada penelitian ini dibuat untuk melakukan deteksi pelanggaran marka pada lampu rambu lalu lintas dan kemudian menampilkan siapa pengendara yang melanggar pada. Peneliti memanfaatkan sensor *Infra Red* untuk melakukan deteksi pelanggaran marka jalan. Pada kendaraan dipasang rangkaian yang digunakan untuk memberikan identitas mengenai pemilik mobil.

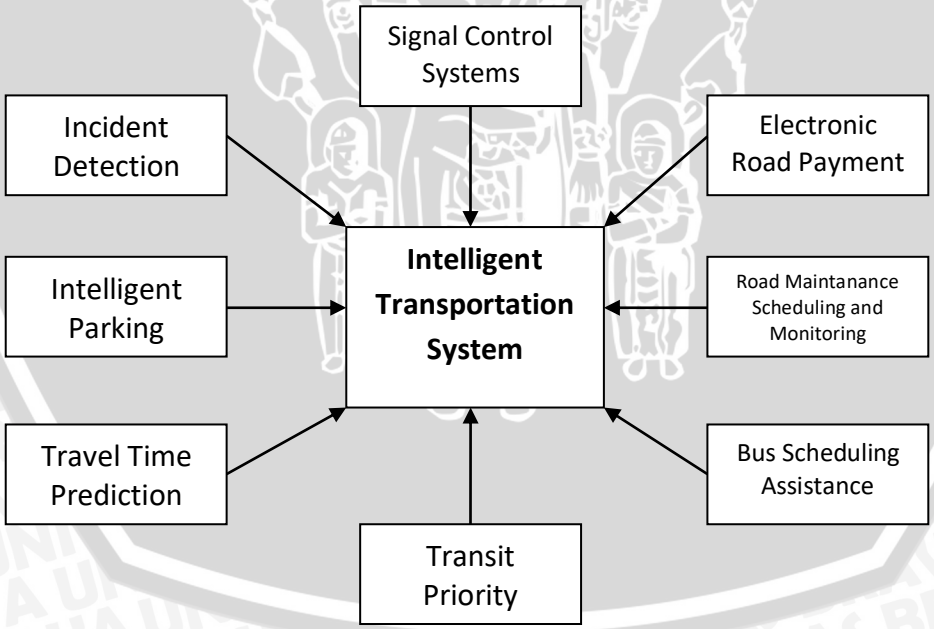
Ketika mobil melakukan pelanggaran, data akan dikirimkan melalui komunikasi RF dan kemudian ditampilkan pada LCD. Kekurangan dari penelitian ini adalah belum ada tempat untuk menyimpan data pelanggaran serta komunikasi hanya *one-to-one*.

Penelitian ketiga ditulis oleh Faisal Amir dengan judul "Perancangan Aplikasi Komputer Pusat untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara *Wireless*". Penelitian ini berfokus pada komunikasi data antara komputer pusat dan lampu lalu lintas. Komunikasi data yang digunakan adalah secara *wireless* dengan menggunakan perangkat nRF24L01. Kontrol lampu lalu lintas dibagi menjadi dua yaitu *auto* dan *manual*. Sehingga, pengguna dapat mengontrol lampu lalu lintas baik otomatis maupun *manual* melalui *user interface* yang terdapat di komputer pusat. Kekurangan pada penelitian ini adalah komunikasi datanya bersifat *one-to-one.*

2.2 Dasar Teori

2.2.1 *Intelligent Transportation System*

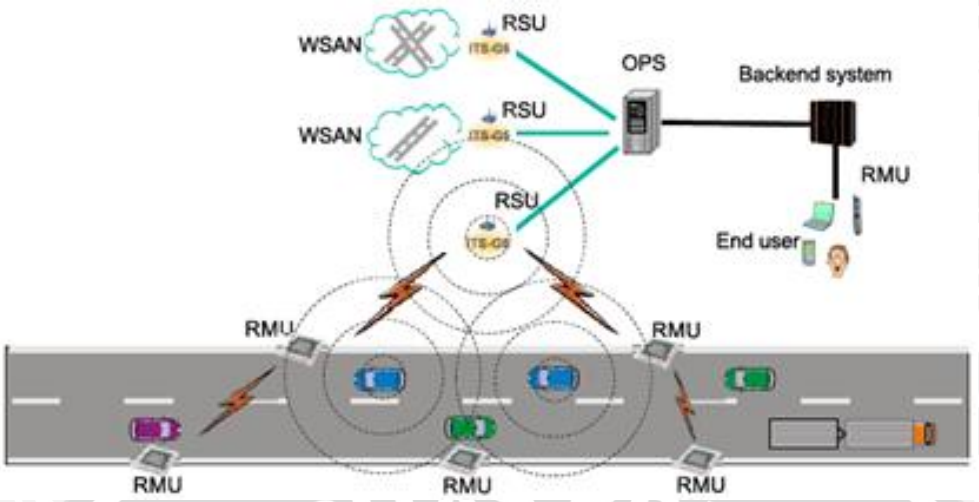
Pada buku *"Transportation Systems Engineering"* dituliskan bahwa *Intelligent Transportation System* (ITS) merupakan sistem yang dapat meningkatkan efisiensi di jalanan. ITS melibatkan kendaraan, pengemudi, rambu lalu lintas dan perangkat komunikasi yang digabungkan menjadi sebuah sistem kompleks yang dapat berinteraksi satu dengan lainnya (Mathew, T., 2014).



Gambar 2.2 Layanan Pada *Intelligent Transportation System*

Sumber: (Mathew, T., 2014)

Gambar 2.1 menunjukkan delapan layanan umum yang diberikan ITS kepada pengguna. Layanan-layanan ini masih bisa terus bertambah menyesuaikan dengan keadaan di jalan.



Gambar 2.3 Komponen Penyusun *Intelligent Transportation System*

Sumber: (Riliskis, et al., 2014)

Disebutkan pada penelitian yang berjudul *"Maestro: An Orchestration Framework for Large-Scale WSN Simulations"* ada beberapa komponen penyusun ITS. *Road Marking Units* (RMU) adalah perangkat otomatis yang diletakkan di jalan untuk melakukan *sensing* dan perhitungan, *Road Side Units* (RSU) adalah node yang menyampaikan data antara RMU dan OPS, *Open Platform Server* (OPS) adalah node yang menyediakan data untuk *backend system* (Riliskis, et al., 2014).

RMU harus memiliki empat komponen wajib yang harus dimiliki seperti yang disebutkan pada penelitian berjudul *"A Survey on Urban Traffic Management System Using Wireless Sensor Networks"*. *Sensing module* digunakan untuk mendapat data, *processing module* digunakan untuk memroses data, *radio module* digunakan untuk berkomunikasi secara *wireless, power module* digunakan sebagai penyedia energi (Nellore, et al., 2016).

2.2.2 *Wireless Sensor Network*

Wireless Sensor Network atau disingkat dengan WSN adalah gabungan dari beberapa sensor yang saling terhubung satu sama lain melalui komunikasi *wireless* atau tanpa kabel (Sohraby, et al,. 2007). Sensor yang digunakan untuk membentuk sebuah jaringan WSN bermacam-macam, mulai dari sensor temperatur, tekanan, cahaya, jarak dan lain-lain. Sensor-sensor ini berguna untuk mengubah data analog menjadi data digital, kemudian data tersebut dikirimkan melalui media komunikasi seperti : Bluetooth, Infrared, Wifi, dan Radio.

Beberapa karakteristik yang dimiliki *wireless sensor network* adalah : dapat digunakan pada daya yang terbatas, dapat diterapkan pada kondisi lingkungan yang keras, dapat digunakan untuk kondisi dan pemrosesan data secara mobile, memiliki topologi jaringan yang dinamis, dan dapat dikembangkan untuk skala

besar Dari karakteristik yang dimiliki WSN pengaplikasian WSN bisa dilakukan dimana saja. Berikut beberapa contoh pengaplokasi WSN :

1. Pada rumah WSN bisa digunakan untuk melakukan kontrol pada perangkat-perangkat di rumah dan juga bisa menjaga keamanan rumah.
2. Bidang kesehatan WSN bisa digunakan untuk memonitoring keadaan pasien secara terus menerus tanpa harus menemui pasien.
3. Bidang militer digunakan WSN bisa digunakan untuk mengetahui pergerakan musuh.

2.2.3 *Quality of Service*

Quality of Service atau disingkat QoS adalah kemampun atau kualitas suatu jaringan untuk menyediakan layanan yang baik. QoS bisa diketahui dengan menghitung beberapa parameter seperti : *Latency, Packet loss, Throughput,* dan *Jitter. Latency* adalah waktu dari *request* data hingga menerima data. *Packet loss* adalah keadaan dimana terjadi data yang tidak sampai ke peminta data. *Throughput* adalah berapa besar data yang dapat dikirimkan dalam sekali pengiriman. *Jitter* adalah variansi *delay* yang terjadi selama pengiriman. Berikut persamaan – persamaan yang digunakan untuk menghitung QoS :

$$Latency = \frac{\sum delay}{\sum data} \quad (2.1)$$

Latency dapat dihitung dengan membagi total waktu pengiriman dengan jumlah data yang dikirimkan. *Latency* disajikan dengan satuan *microseconds* (ms) atau *nanoseconds* (ns).

$$Packet\ Loss = \frac{\sum paket\ terkirim}{\sum paket\ diterima} \quad (2.2)$$

Packet loss dapat dihitung dengan membagi total data yang terkirim dibagi dengan total data diterima. *Packet loss* disajikan dalam persen (%).

$$Throughput = \frac{\sum request\ data - \sum\ data\ terkirim}{\sum request\ data} \times 100\ \% \quad (2.3)$$

Throughput dapat dihitung dengan menjumlah data yang dikirimkan dan dibagi dengan total waktu pengiriman. *Throughput* disajikan dengan satuan *bit per second* (bps), *Byte per second* (Bps), atau *data per second* (dps).

$$Jitter = \frac{\sum((delay_n - delay_{n-1}) + (delay_{n-1} - delay_{n-2}) + \dots)}{\sum data} \quad (2.4)$$

Jitter dapat dihitung dengan menjumlah selisih tiap-tiap delay dari tiap pengiriman kemudian dibagi dengan total data yang dikirimkan. *Jitter* disajikan dengan satuan *microseconds* (ms) atau *nanoseconds* (ns).

2.2.4 Rambu Lalu Lintas

Rambu lalu lintas adalah bagian dari jalan yang mengatur tata tertib di jalan raya. Rambu lalu lintas memiliki bentuk dan simbol yang berbeda satu dengan lainnya agar pengguna jalan dapat mengenalinya dengan mudah. Yang terdapat pada rambu antara lain adalah : lambang, huruf, angka, kalimat yang digunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi pengguna jalan.

Rambu yang terdapat di jalan raya dibedakan menjadi tiga, yaitu : rambu peringatan, rambu dilarang, dan rambu perintah. Rambu perintah adalah peringatan bahaya atau tempat berbahaya pada jalan di depan pemakai jalan. Warna dasar rambu peringatan berwarna kuning dengan lambang atau tulisan berwarna hitam. Rambu larangan adalah perintah yang wajib dilakukan oleh pemakai jalan. Warna dasar rambu larangan berwarna putih dan lambang atau tulisan bewarna hitam atau merah. Rambu petunjuk adalah penunjuk mengenai jurusan, jalan, situasi, kota tempat, pengaturan, fasilitas dan lain-lain bagi pemakai jalan. Rambu perintah berbentuk bundar berwarna biru dan lambang atau tulisan berwarna putih serta merah untuk garis serong sebagai batas akhir perintah (Affandi, F. 2016). Berikut gambar 2.4 menunjukkan beberapa contoh rambu lalu lintas berdasarkan kategorinya :

Gambar 2.4 Contoh Rambu Lalu Lintas Berdasarkan Kategorinya

Sumber: (Affandi, F)

Dari berbagai macam rambu lalu lintas yang ada untuk penelitian ini digunakan tiga diantaranya yaitu rambu dilarang berhenti, rambu dilarang putar arah dan rambu dilarang masuk. Berikut penjelasan dari ketiga rambu-rambu yang digunakan :

1. Rambu Dilarang Berhenti

Rambu dilarang berhenti adalah rambu larangan yang melarang pengguna kendaraan bermotor untuk berhenti sampai jarak 15 meter dari tempat pemasangan rambu. Jarak langgar rambu bisa lebih dari 15 meter apabila ada papan tambahan yang menyatakan “sampai rambu berikut”. Pasal 1 angka 16 Undang Undang Lalu Lintas dan Angkutan Jalan berbunyi berhenti adalah keadaan kendaraan berhenti atau tidak bergerak untuk beberapa saat dan tidak ditinggalkan pengemudinya.

Gambar 2.5 Rambu Dilarang Berhenti

Sumber: (Affandi, F)

2. Rambu Dilarang Putar Arah

Rambu dilarang putar arah adalah rambu larangan yang melarang pengguna kendaraan bermotor maupun tidak bermotor untuk tidak berbalik arah. Rambu ini biasanya dipasang pada jalan bebas hambatan yang biasanya penuh dengan kendaraan dengan kecepatan tinggi. Berikut gambar 2.6 menunjukkan bentuk rambu dilarang putar arah :

Gambar 2.6 Rambu Dilarang Putar Arah

Sumber: (Affandi, F)

3. Rambu Dilarang Masuk

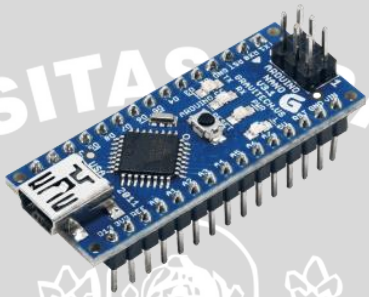
Rambu dilarang masuk adalah rambu larangan bagi semua kendaraan bermotor maupun tidak bermotor untuk melewati jalan tertentu. Rambu ini biasanya dipasang pada jalan satu arah sehingga kendaraan yang dari arah lain tidak boleh lewat jalan tersebut. Berikut gambar 2.7 menunjukkan bentuk dari rambu dilarang masuk :

Gambar 2.7 Rambu Dilarang Masuk

Sumber: (Affandi, F)

2.2.5 Arduino Nano

Arduino Nano adalah sebuah *board* yang berukuran kecil yang dibuat berdasarkan Atmega328 (Arduino Nano 3.x) atau Atmega168 (Arduino Nano 2.x). Board ini salah satu mikrokontroller yang populer dikarenakan memiliki dimensi ukuran yang kecil sehingga membuat mikrokontroller praktis untuk digunakan. Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, hanya berbeda pada ukurannya saja. Kekurangan dari board ini adalah tidak memiliki port untuk DC power, dan hanya bekerja dengan kabel Mini-B USB. Nano didesain dan diproduksi oleh Gravitech (Arduino, 2016). Berikut gambar 2.8 menunjukkan bentuk fisik Arduino Nano :



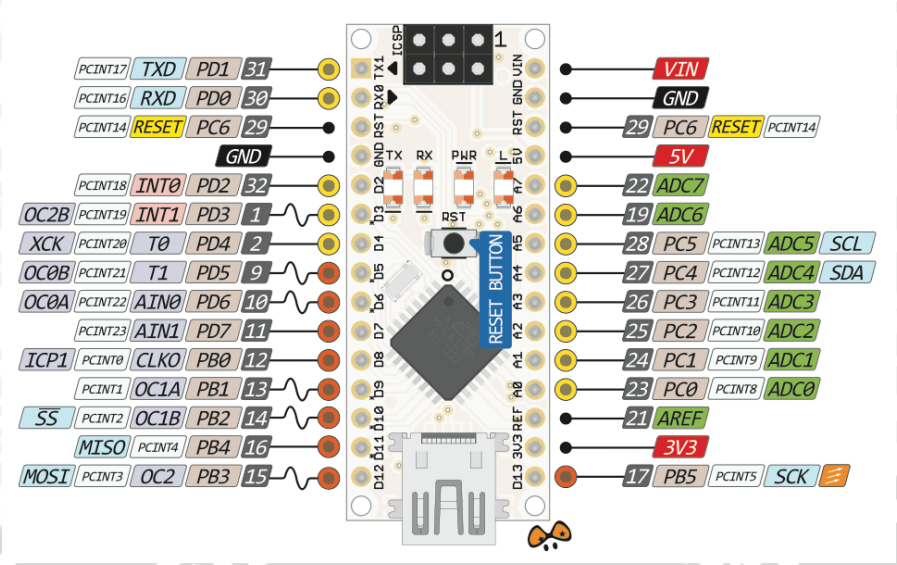
Gambar 2.8 Arduino Nano

Sumber: (bombayelectronics.in)

Berikut spesifikasi dari Arduino Nano :

1. Microcontroller : Atmel ATmega168 or ATmega328;
2. Operating Voltage (logic level) : 5 V;
3. Input Voltage (recommended) : 7-12 V;
4. Input Voltage (limits) : 6-20 V;
5. Digital I/O Pins : 14;
6. Analog Input Pins : 8;
7. DC Current per I/O Pin : 40 mA;
8. Flash Memory : 16 KB (ATmega168) or 32 KB (ATmega328);
9. SRAM : 1 KB (ATmega168) dan 2 KB (ATmega328);
10. EEPROM : 512 *byte*s (ATmega168) or 1 KB (ATmega328);
11. Clock Speed : 16 MHz;
12. Dimensions : 0.73" x 1.70";
13. Length : 45 mm;
14. Width : 18 mm;
15. Weigth : 5 g;

Arduino Nano memiliki beberapa pin yang memiliki fungsinya masing-masing. Berikut gambar 2.9 menunjukkan tata letak pin dan fungsi pada Arduino Nano :



Gambar 2.9 Pin Pada Arduino Nano

Sumber: (forum.arduino.cc)

2.2.6 Modul *Wireless* nRF24L01

Modul *Wireless* nRF24L01 adalah sebuah modul komunikasi jarak jauh yang memanfaatkan gelombang RF 2.4GHz. Modul ini menggunakan antarmuka SPI untuk berkomunikasi. nRF24L01 memiliki true ULP solution, yang memungkinkan daya tahan baterai berbulan-bulan hingga bertahun-tahun. Modul ini dapat digunakan untuk pembuatan pheriperal PC, piranti permainan, piranti fitnes dan olahraga, mainan anak-anak dan alat lainnya (nordicsemi.com, 2007). Berikut gambar 2.10 menunjukkan bentuk fisik nRF24L01 :

Gambar 2.10 Modul nRF24L01

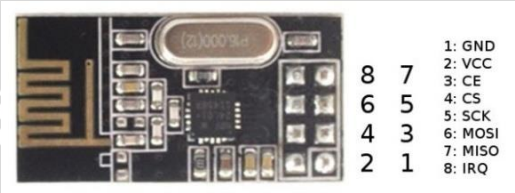
Sumber: (hallard.me, 2013)

Berikut spesifikasi dari Modul *Wireless* nRF24L01 :

1. Worldwide 2.4GHz ISM band operation;
2. Jangkauan data di udara yaitu 250kbps, 1Mbps and 2Mbps;
3. Ultra low power operation;
4. 11.3Ma TX at 0dBm output power;
5. 13.5Ma RX at 2Mbps air data rate;
6. Power turun pada 900Na ;

7. 26mA in standby-I;
8. On chip voltage regulator;
9. Sumber tegangan yang digunakan 1.9 sampai 3.6V;
10. Enhanced ShockBurst™;
11. Automatic packet handling;
12. Auto packet transaction handling;
13. 6 data pipe MultiCeiver™;
14. Drop-in compatibility with nRF24L01;

Modul nRF24L01 memiliki delapan buah pin yang masing-masing pin memiliki fungsinya masing-masing. Berikut gambar 2.11 menunjukkan tata letak pin Modul nRF24L01 :

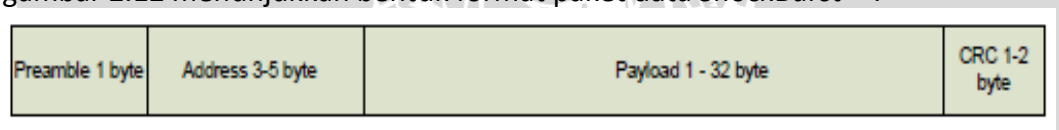


Gambar 2.11 Pin Pada Modul nRF24L01

Sumber: (nordicsemi.com, 2015)

Untuk mengoneksikan nRF24L01 agar dapat berhubungan dengan Arduino Nano maka pin 1 dan 2 dihubungkan dengan GND dan VCC yang terdapat pada Arduino, sedangkan untuk pin 3 sampai 8 dihubungkan dengan port digital input/output pada Arduino.

Pada *datasheet* nRF24L01 telah disediakan sebuah format data *ShockBurst™* yang dapat dipakai untuk melakukan pengiriman antar nRF24L01. Berikut gambar 2.12 menunjukkan bentuk format paket data *ShockBurst™* :



Gambar 2.12 Format Paket Data *ShockBurst™*

Sumber: (nordicsemiconductor, 2007)

Premble adalah *bit* tambahan yang dipergunakan untuk sinkronisisi antara penerima dan pengirim. Nilai *premble* hanya ada dua yakni 01010101 dan 10101010. Nilai ini didapatkan berdasarkan *bit* pertama pada *Address,* apabila bernilai 0 maka menjadi 01010101 dan apabila bernilai 1 maka menjadi 10101010. *Address* adalah *bit* yang menentukan alamat penerima. Panjang *address* bisa diganti antara 3, 4 atau 5 *byte* pada `AW` register. *Payload* adalah data yang dikirimkan. Panjang *payload* bisa diganti dari 1 – 32 *byte* pada *code* yang ditulis. CRC atau *Cyclic Redundancy Check* digunakan untuk mengetahui apakah telah terjadi perubahan data pada proses pengiriman (nordicsemiconductor, 2007). *P*reamble dan CRC telah ditangani oleh modul *transceiver*, sehingga pengguna hanya mengirimkan alamat tujuan dan data saja (delta-electronic).

2.2.7 Sensor *Ultrasonic* HC-SR04

Sensor *ultrasonic* HC-SR04 bekerja seperti halnya kelelawar yaitu menggunakan sonar untuk mengukur jarak dengan sebuah objek. HC-SR04 memiliki akurasi tinggi dan stabilitas dalam pembacaa datanya. Sensor ini bekerja pada jarak 2 cm sampai 400 cm, selain itu juga pembacaan sensor tidak akan terganggu oleh sinar matahari atau pun benda berwarna hitam. Sensor ini sudah dilengkapi dengan modul *transmitter* dan *receiver* yang digunakan untuk mengirim dan menerima sinyal (randomnerdtutorials.com, 2013). Pada gambar 2.13 ditunjukkan bentuk fisik dari sensor *ultrasonic* HC-SR04.

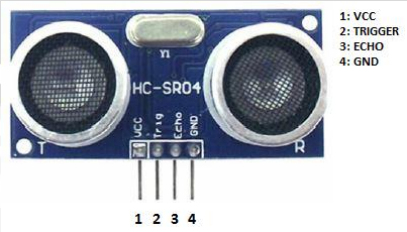
Gambar 2.13 Sensor *Ultrasonic* HC-SR04

Sumber: (randomnerdtutorials.com, 2013)

Berikut spesifikasi dari Sensor *Ultrasonic* HC-SR04 :

1. Power Supply : +5V DC;
2. Quiescent Current : <2mA;
3. Working Current : 15mA;
4. Effectual Angle : <15°;
5. Ranging Distance : 2cm – 400 cm/1" – 13ft;
6. Resolution : 0.3 cm;
7. Measuring Angle : 30 degree;
8. Trigger Input Pulse width : 10uS;
9. Dimension : 45mm x 20mm x 15mm;

Sensor *ultrasonic* HC-SR04 memiliki embat buah pin yang digunakan. Agar *ultrasonic* dapat terhubung dengan Arduino Nano maka pin VCC dan GND pada sensor harus dihubungkan dengan VCC dan GND milik Arduino, sedangkan untuk pin TRIG dan ECHO dihubungkan dengan pin digital pada Arduino. Berikut pada gambar 2.14 menunjukkan tata letak dari pin Sensor *Ultrasonic* HC-SR04.



Gambar 2.14 Pin Pada Sensor *Ultrasonic* HC-SR04

Sumber: (learn.linksprite.com, 2014)

2.2.8 Sensor *Light Dependant Resistor*

Sensor *Light Dependant Resistor* (LDR) adalah perangkat yang sensitif terhadap cahaya yang sering digunakan untuk menunjukkan ada tidaknya cahaya, atau untuk mengukur intensitas cahaya. Dalam keadaan gelap resistansi yang dihasilkan sensor bisa sangat tinggi, sedangakan saat ada cahaya resistansi akan turun bahkan sampai beberapa ohm saja. Pada gambar 2.15 menunjukkan bentuk fisik dari sensor LDR :

Gambar 2.15 Sensor *Light Dependant Resistor*

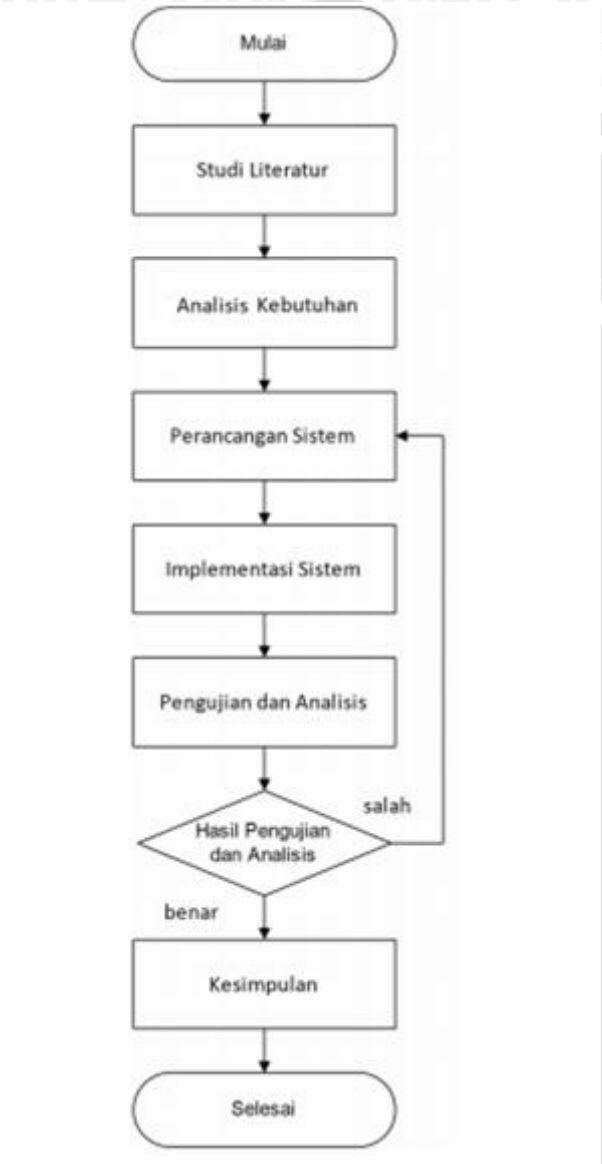
Sumber: (learn.linksprite.com, 2014)

Pada sisi bagian atas LDR terdapat suatu garis / jalur melengkung yang menyerupai bentuk kurva. Jalur tersebut terbuat dari bahan *cadmium sulphida* yang sangat sensitif terhadap pengaruh dari cahaya. Jalur *cadmium sulphida* dibuat melengkung menyerupai kurva agar jalur tersebut dapat dibuat panjang dalam area yang sempit. *Cadmium sulphida* (CdS) merupakan bahan semi-konduktor yang memiliki gap energi antara elektron konduksi dan elektron valensi. Ketika cahaya mengenai *cadmium sulphida*, maka energi proton dari cahaya akan diserap sehingga terjadi perpindahan dari band valensi ke band konduksi. Akibat perpindahan elektron tersebut mengakibatkan hambatan dari *cadmium sulphida* berkurang dengan hubungan kebalikan dari intensitas cahaya yang mengenai LDR.

BAB 3 METODOLOGI

3.1 Alur Metode Penelitian

Alur metode penelitian menjelaskan langkah-langkah penulis dalam mengerjakan penelitian. Berikut gambar diagram alir penelitian yang digunakan :



Gambar 3.1 Diagram Alir Penelitian

3.2 Studi Literatur

Studi literatur dilakukan untuk mencari penelitian-penelitian sebelumnya sebagai pendukung penelitian yang akan dilakukan. Selain itu, juga untuk menyusun teori-teori yang digunakan dalam penelitian dan perancangan sistem. Teori-teori yang diperlukan untuk merancang sistem antara lain :

1. Teori *Wireless Sensor Network* menjelaskan mengenai pengerian WSN, perangkat apa saja yang dibutuhkan untuk merancang WSN, karakteristik WSN, dan contoh-contoh penerapan WSN.
2. Teori Rambu Lalu Lintas menjelaskan pengertian rambu lalu lintas, jenis-jenis rambu, serta menjelaskan maksud dari rambu dilarang berhenti, dilarang putar arah, serta rambu dilarang masuk.
3. Teori Arduino Nano menjelaskan apa itu Arduino Nano, spesifikasi yang dimiliki, dan menunjukkan pin-pin yang tersedia.
4. Teori Modul *Wireless* nRF24L01 menjelaskan apa itu nRF24L01, spesifikasi yang dimiliki, dan menampilkan pin-pin yang tersedia.
5. Teori Sensor *Ultrasonic* HC-SR04 menjelaskan bagaimana sensor bekerja, spesifikasi sensor, dan menampilkan pin-pin yang dimiliki.
6. Teori Sensor *Light Dependant Resistor* menjelaskan bagaimana sensor bekerja.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan merupakan tahapan yang bertujuan untuk menganalisis semua perangkat yang diperlukan untuk merancang sistem.

3.3.1 Gambaran Umum Sistem

Gambaran umum sistem menjelaskan maksud dan tujuan dari sistem ini. Selain itu juga dijelaskan mengenai bagaimana sistem bekerja mulai dari awal sampai akhir.

3.3.2 Kebutuhan Antarmuka Eksternal

Kebutuhan antarmuka eksternal menjelaskan perangkat apa saja yang akan digunakan untuk membuat sistem. Perangkat yang dibutuhkan meliputi perangkat keras maupun perangkat lunak.

3.3.3 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan mengenai fungsi-fungsi yang harus dimiliki oleh sistem. Baik fungsi yang harus dimiliki perangkat keras maupun perangkat lunak.

3.3.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan batasan-batasan yang dimiliki oleh sistem. Batasan ini didapatkan berdasarkan kemampuan dari masing-masing perangkat, baik perangkat keras maupun lunak.

3.4 Perancangan

Perancangan adalah tahapan melakukan perancangan sistem sesuai dengan analisis kebutuhan sistem yang sudah dibuat. Pada perancangan sistem ini akan dilakukan beberapa tahapan, yaitu :

1. Perancangan RMU, RSU Dan OPS

Perancangan meliputi pembuatan rangkaian perangkat keras pada tiap-tiap node serta membuat algoritma untuk sensor agar sensor dapat bekerja. Masing-masing node memiliki peran masing-masing dimana RMU berperan sebagai penyedia data, RSU berperan sebagai penghubung antara RMU dan OPS apabila jarak terlalu jauh, dan OPS berperan sebagai penerima data akhir.

2. Perancangan Komunikasi Data
   Perancangan meliputi pembuatan algoritma pengiriman data agar dalam proses pengirimannya tidak terjadi tabrakkan data serta membuat format data yang dapat dikenali oleh seluruh node.
3. Perancangan *User interface* Pada Server
   Perancangan dilakukan dengan membuat daftar fungsi yang harus dimiliki oleh *User interface.*

3.5 Implementasi

Pada implementasi sistem dilakukan pembuatan sistem berdasarkan perancangan yang sudah dibuat. Pada implementasi sistem ini akan dilakukan beberapa tahapan, yaitu :

1. Implementasi RMU, RSU Dan OPS
   Implementasi meliputi pembuatan perangkat keras berupa dua buah RMU, satu buah RSU, dan satu buah OPS berdasarkan rangkaian yang sudah dibuat. Selain itu juga dibuat *source code* dari algoritma sensor yang sudah dibuat pada perancangan.
2. Implementasi Komunikasi Data
   Implementasi meliputi pembuatan *source code* dari algoritma pengiriman data dan perancangan format data.
3. Implementasi *User interface* Pada Server
   Implementasi meliputi pembuatan *source code* sesuai dengan daftar fungsi yang harus dimiliki *User interface*.

3.6 Pengujian

Pengujian dari penelitian ini difokuskan pada kinerja sensor dan pengiriman data, berikut skenario pengujian yang dilakukan :

3.6.1 Pengujian Fungsional *Hardware*

Pengujian sensor dilakukan dua kali, yakni pada RMU 1 dan RMU 2. Pada RMU 1 diuji apakah sensor *ultrasonic* HC-SR04 dapat mendeteksi pelanggaran rambu putar arah dan sensor LDR dapat mendeteksi pelanggaran rambu dilarang berhenti. Sedangakan pada RMU 2 diuji apakah sensor *ultrasonic* HC-SR04 dapat mendeteksi pelanggaran rambu dilarang masuk dan apakah lampu LED sudah menyala sesuai dengan waktu yang sudah ditentukan. Objek yang digunakan dalam pengujian adalah kendaraan bermotor.

3.6.2 Pengujian Performa Pengiriman Data Menggunakan WSN

Pengujian pengiriman data dilakukan dua kali, yakni *singlehop* dan *multihop*. Pada pengujian *singlehop* RMU 1 dan RMU 2 mengirimkan data ke RSU, sedangkan pengujian *multihop* menambahkan pengiriman dari RSU ke OPS.

3.6.3 Pengujian Algoritma Sistem Secara Keseluruhan

Pengujian dimulai dari saat RMU membaca data kemudian mengirimkannya ke RSU, dari RSU kemudian data dikirimkan ke OPS. Saat data sampai OPS kemudian diolah dan ditampilkan pada *User interface* dan disimpan pada *Database.*

3.7 Kesimpulan

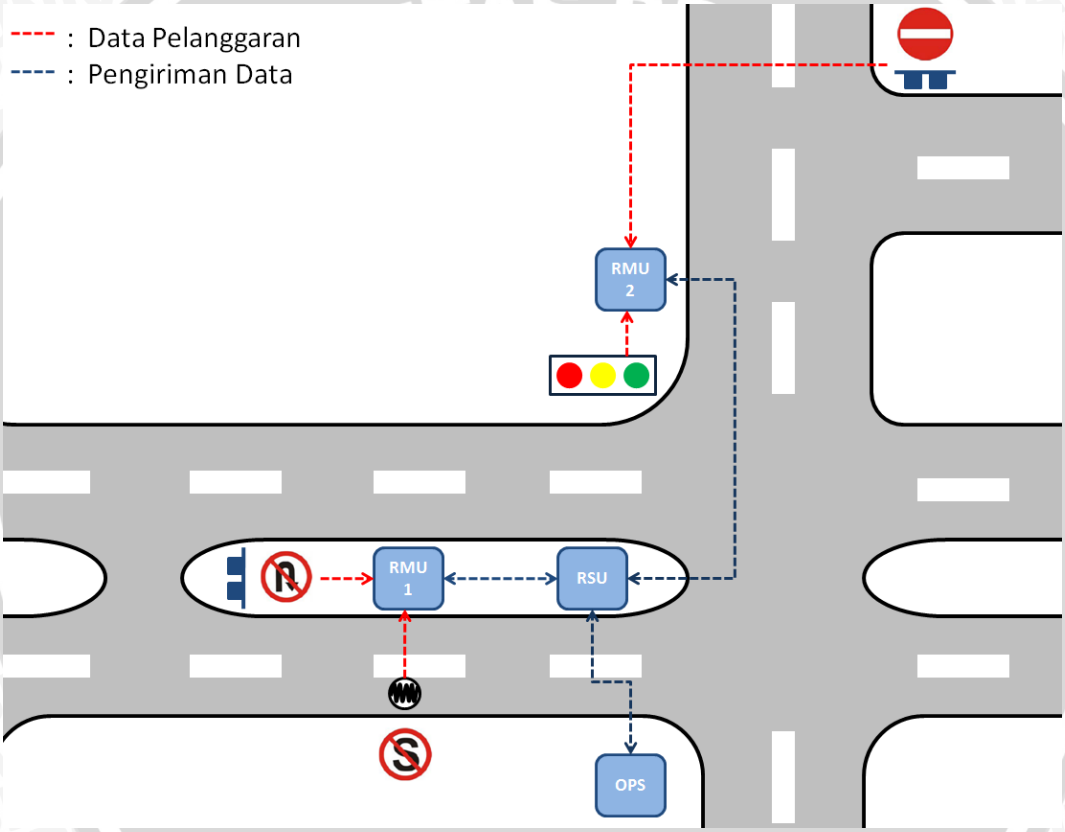
Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, pengujian dan analisis sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis yang dibuat. Penarikan kesimpulan diharapkan dapat menjadi acuan bagi penelitian lain untuk mengembangkan sistem deteksi pelanggaran rambu lalu lintas.

Tahap ini juga berisi saran untuk mengembangkan dan memperbaiki kesalahan terhadap penelitian ini. Selain itu juga memberikan ide pada peneliti selanjutnya untuk menambahkan perangkat keras ataupun perangkat lunak agar alat ini dapat menjadi lebih baik.

BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem digunakan untuk mengetahui pelanggaran rambu lalu lintas tanpa harus melihat langsung lokasi terjadinya pelanggaran. Sistem terdiri dari empat buah node, yaitu : RMU 1 yang digunakan untuk mengambil data pelanggaran rambu dilarang putar arah dan dilarang berhenti, RMU 2 yang digunakan untuk mengambil data pelanggaran dilarang masuk dan monitor lampu lalu lintas, RSU yang digunakan untuk menjembatani pengiriman data antara RMU dan OPS, dan OPS sebagai pengirim *request* dan penerima data akhir. Berikut gambar bagaimana cara kerja sistem secara umum :



Gambar 4.1 Cara Kerja Sistem Secara Umum

Pada gambar 4.1 ditunjukkan sistem akan mendeteksi pelanggaran rambu menggunakan sensor. Sensor jarak digunakan untuk mendeteksi jenis pelanggaran di rambu dilarang putar arah dan dilarang masuk. Sensor jarak diletakkan seperti posisi pada gambar sehingga apabila ada kendaraan yang melewati sensor dengan jarak tertentu akan terdeteksi melakukan pelanggaran. Sensor cahaya digunakan untuk mendeteksi pelanggaran di rambu dilarang berhenti. Sensor cahaya diletakkan pada permukaan jalan, sehingga ketika ada kendaraan yang berhenti di atas sensor selama kurun waktu tertentu maka akan dianggap melakukan pelanggaran.

Agar komputer server mengetahui adanya pelanggaran maka OPS akan mengirimkan *request* ke RSU dan kemudian diteruskan ke RMU. Kemudian data yang didapatkan oleh RMU dikembalikan ke OPS melalui RSU. Data pelanggaran yang diterima oleh OPS akan diolah dan ditampilkan pada *user interface.*

4.2 Kebutuhan Antarmuka Eksternal

4.2.1 Antarmuka Pengguna

Pada sub bab ini akan dianalisis kebutuhan antramuka pengguna yang digunakan *user* untuk berinteraksi dengan sistem. Dalam sistem dibuat sebuah *user interface* dengan menggunakan Embarcadero RAD Studio 2010 untuk memudahkan *user* membaca data pelanggaran yang ada. Data pelanggaran akan ditampilkan dalam bentuk grafik agar *user* dapat membaca data pelanggaran lebih mudah.

4.2.2 Antarmuka Perangkat Keras

Pada sub bab ini akan dianalisis kebutuhan perangkat keras apa saja yang akan digunakan dalam penelitian ini agar sistem nantinya dapat bekerja sesuai dengan harapan. Berikut daftar perangkat keras yang digunakan dalam sistem :

a. Arduino Nano

   Arduino berperan sebagai *Control Processing Unit (CPU)* dari sistem. Arduino memiliki beberapa tugas antara lain : memerintah nRF24L01 untuk saling bertukar data antar node, memerintah sensor untuk membaca data pelanggaran, dan memroses data yang sudah didapatkan oleh sensor.

b. nRF24L01

   nRF24L01 berperan sebagai alat komunikasi yang digunakan node untuk berhubungan satu dengan lainnya.

c. Sensor *Ultrasonic* HC-SR04

   Sensor ini berperan sebagai pembaca pelanggaran pada rambu dilarang putar arah dan rambu dilarang masuk.

d. Sensor *Light Dependant Resistor*

   Sensor ini berperan sebagai pembaca pelanggaran pada rambu dilarang berhenti.

e. Lampu LED

   Lampu LED berperan sebagai pengganti lampu merah, maka dari itu diperlukan tiga buah Lampu LED yang masing-masing berwarna merah, kuning dan hijau.

f. *Power Supply*

   Power Supply digunakan untuk memberikan tenaga kepada Arduino Nano agar dapat bekerja dengan baik.

g. Kabel Mini USB

   Kabel Mini USB digunakan untuk mengupload *source code* ke Arduino Nano.

h. Laptop / Komputer

Laptop / Komputer digunakan untuk membuat *code* pada Arduino ataupun *user interface*, memberi perintah memasukkan *code* pada Arduino, dan menampilkan data pada monitor.

4.2.3 Antarmuka Perangkat Lunak

Pada sub bab ini akan dianalisis kebutuhan perangkat lunak apa saja yang akan digunakan dalam penelitian ini agar sistem nantinya dapat bekerja sesuai dengan harapan. Berikut daftar perangkat lunak yang digunakan dalam sistem :

a. Arduino IDE

   Arduino IDE digunkan untuk membuat *code* serta mengupload *code* pada Arduino Nano.

b. Embarcadero RAD Studio 2010

   Embarcadero RAD Studio 2010 digunakan untuk membuat *user interface* pada sistem.

c. Fritzing

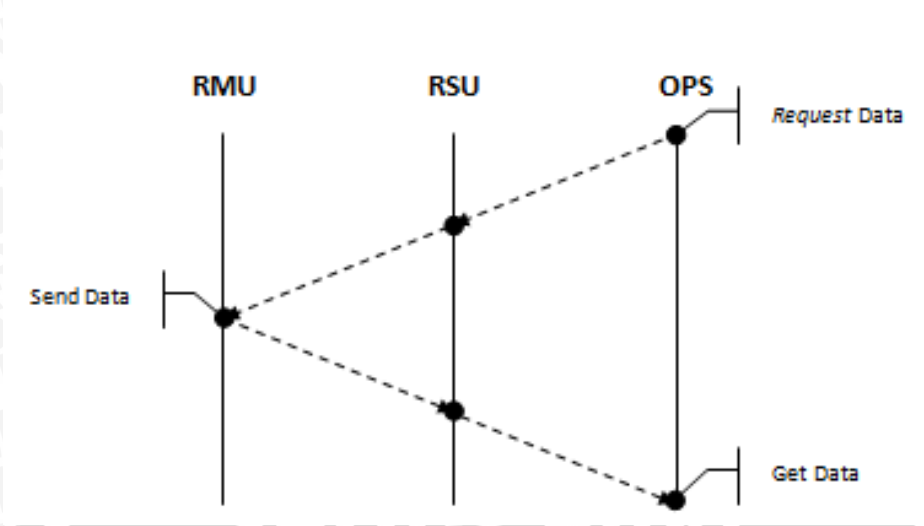
   Fritzing digunakan untuk membuat rancangan perangkat keras pada node.

4.2.4 Antarmuka Komunikasi

Pada sub bab ini akan dianalisis kebutuhan antarmuka komunikasi yang diperlukan agar sistem dapat bekerja dengan baik. Sistem ini menggunakan perangkat keras nRF24L01 sebagai pengirim datanya. nRF24L01 memerlukan *library* tertentu agar dapat berkerja. *Library* yang digunakan pada sistem adalah *Mirf.* Dalam proses pengiriman menggunakan *Mirf* memerlukan adanya pengaturan *channel* dan nama untuk tiap-tiap node. *Channel* yang digunakan pada sistem adalah 123. Sedangkan untuk nama dari tiap-tiap node adalah sebagai berikut : RMU 1 dan RMU 2 menggunakan *address* ‘clnt’, RSU menggunakan *address* ‘mstr’, sedangkan OPS menggunakan *address* ‘srvr’.

4.3 Kebutuhan Fungsional

Sistem yang dibuat memerlukan empat buah node yakni RMU 1, RMU 2, RSU serta OPS. Tiap node memiliki fungsinya masing-masing namun saling berhubungan satu dengan lainnya menggunakan nRF24L01 agar dapat memonitor pelanggaran rambu lalu lintas secara jauh tanpa harus mendatangi lokasi rambu berada.



Gambar 4.2 Sirkulasi Data

Pada gambar 4.1 awal sistem berjalan OPS akan mengirimkan *request* kepada RMU. Pengiriman *request* dilakukan oleh OPS ke RMU baik secara langsung apabila jarak dirasa mencukupi atau melalui RSU apabila jarak dari OPS ke RMU terlalu jauh.

Setelah *request* sampai pada RMU, kemudian RMU mulai melakukan pembacaan data dengan sensor. Pada RMU 1 sensor akan membaca ada atau tidaknya pelanggaran putar arah serta pelanggaran dilrang berhenti. Sedangkan pada RMU 2 sensor akan membaca ada atau tidaknya pelanggaran dilarang masuk serta memonitor keadaan lampu merah. Setelah RMU selesai membaca data kemudian yang dilakukan adalah menyusun data yang didapatkan menjadi sebuah format yang nantinya dapat memudahkan dalam membedakan data antara RMU 1 atau RMU 2. Kemudian data yang sudah disusun sesuai format data dikirimkan ke OPS.

Agar dapat menampilkan data yang sudah diterima oleh OPS diperlukan *user interface* yang dapat membaca data dari *COM Port*. Salah satu IDE yang dapat berhubungan dengan Arduino melalui *COM Port* adalah Embarcadero RAD Studio 2010.

Data yang dikirimkan dalam bentuk format akan diproses oleh *user interface* agar memudahkan dalam pembacaan data pelanggaran. Proses yang dilakukan pertama adalah menentukan data yang diterima dari RMU 1 atau RMU 2. Kemudian data pelanggaran akan diambil dan disimpan pada *database*. Untuk mempermudah pembacaan ada tidaknya pelanggaran dibuat empat buah grafik. Grafik pertama untuk membaca pelanggaran rambu dilarang putar arah, grafik kedua untuk membaca pelanggaran rambu dilarang masuk, grafik ketiga untuk membaca pelanggaran rambu dilarang berhenti, dan grafik yang keempat digunakan untuk menampilkan keadaan lampu merah. Berikut tabel analisis kebutuhan kinerja sistem yang diperlukan :

Tabel 4.1 Analisis Kebutuhan Kinerja RMU 1

No.	Anailisa Kebutuhan
1	Menerima *request* dari RSU/OPS
2	Membaca data dari sensor *Ultrasonic*
3	Membaca data dari sensor LDR
4	Menyusun data menjadi sebuah format
5	Mengirimkan data ke RSU/OPS

Tabel 4.2 Analisis Kebutuhan Kinerja RMU 2

No.	Anailisa Kebutuhan
1	Menerima *request* dari RSU/OPS
2	Membaca data dari sensor *Ultrasonic*
3	Membaca data dari lampu LED
4	Menyusun data menjadi sebuah format
5	Mengirimkan data ke RSU/OPS

Tabel 4.3 Analisis Kebutuhan Kinerja RSU

No.	Anailisa Kebutuhan
1	Menerima *request* dari OPS
2	Menerima data dari RMU
3	Mengirimkan data ke OPS atau RMU

Tabel 4.4 Analisis Kebutuhan Kinerja OPS

No.	Anailisa Kebutuhan
1	Mengirimkan *request* ke RMU atau RSU
2	Menerima data dari RMU atau RSU

Tabel 4.5 Analisis Kebutuhan Kinerja *User interface*

No.	Anailisa Kebutuhan
1	Dapat berhubungan dengan OPS melalui *COM Port*
2	Dapat membedakan data dari RMU 1 atau 2
3	Memecah data yang sudah terformat
4	Menyimpan data pada *database*
5	Menampilkan data pada *database*
6	Memproses data pada *database* menjadi sebuah grafik
7	Menampilkan jumlah pelanggaran
8	Menampilkan waktu pengiriman antara OPS dan RMU

4.4 Kebutuhan Non-Fungsional

Sistem ini akan bekerja dengan baik apabila jarak pengiriman antar node maksimal 100 meter, dikarenakan jarak maksimal pengiriman nRF24L01 adalah 100 meter. Jarak masksimal tersebut dapat berkurang apabila terdapat halangan seperti gedung dan lain sebagainya. Sensor jarak dapat mendeteksi benda pada jarak 2 – 400 cm.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi perancangan dan implementasi untuk penelitian. Perancangan membahas tentang rancangan yang ada pada penelitian. Dan implementasi membahas tentang pengimplementasian penelitian pada sistem yang ada di lingkungan.

5.1 Perancangan

Tahap perancangan berisi tentang analisis perangkat keras dan perangkat lunak yang digunakan untuk membuat sistem. Ada beberapa tahap di antarannya sebagai berikut :

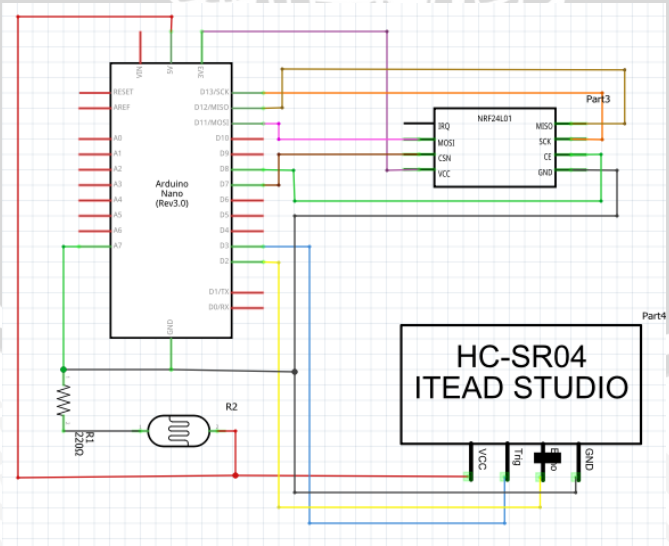
5.1.1 Perancangan RMU, RSU, Dan OPS

Perancangan node digunakan untuk mempermudah dalam proses merangkai perangkat keras. Perancangan dibagi menjadi dua yaitu perancangan rangkaian dan perancangan algoritma sensor.

5.1.1.1 Perancangan Rangkaian Node

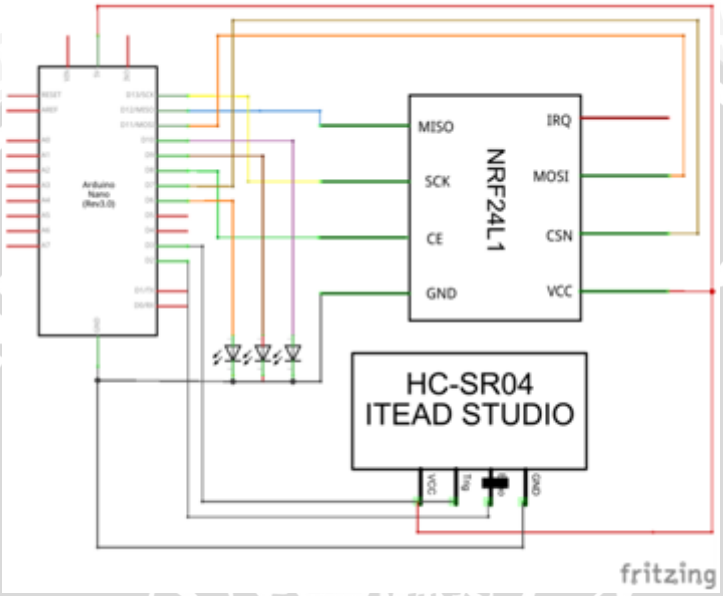
Perancangan perangkat keras pada ke empat node semuanya menggunakan Arduino Nano dan nRF24L01. Untuk RMU 1 ditambahkan sensor *Ultrasonic* HC-SR04 dan sensor LDR. Sedangkan RMU 2 ditambahkan sensor *Ultrasonic* HC-SR04 dan tiga lampu LED berwarna merah, kuning dan hijau.

Sensor *ultrasonic* pada RMU 1 menggunakan 4 pin pada Arduino Nano yaitu VCC, GND, DIGITAL 2 dan DIGITAL 3. Sensor LDR sendiri menggunakan pin ANALOG 7 dan VCC. Sedangakan untuk nRF24L01 menggunakan 7 pin yaitu SCK, MISO, MOSI, 3V3, DIGITAL 7, DIGITAL 8 dan GND. Berikut gambar skematik perancangan perangkat keras pada RMU 1 :



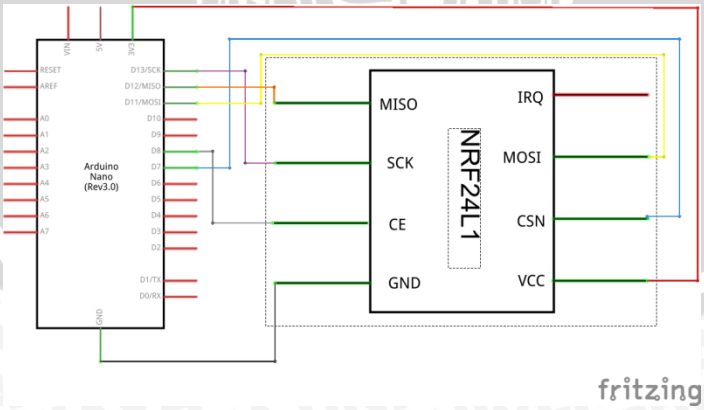
Gambar 5.1 Skematik RMU 1

Pada RMU 2 juga menggunakan sensor *Ultrasonic* yang menggunakan 4 pin yaitu VCC, GND, DIGITAL 2 dan DIGITAL 3. Lampu LED membutuhkan masing-masing 2 pin, jadi apabila menggunakan tiga buah lampu LED maka membutuhkan 6 pin. Lampu LED warna merah terhubung dengan pin DIGITAL 6 dan GND, LED warna kuning terhubung dengan pin DIGITAL 9 dan GND, dan LED warna hijau terhubung dengan pin DIGITAL 10 dan GND. Sedangakan untuk nRF24L01 menggunakan 7 pin yaitu SCK, MISO, MOSI, 3V3, DIGITAL 7, DIGITAL 8 dan GND. Berikut gambar skematik perancangan perangkat keras pada RMU 2 :



Gambar 5.2 Skematik RMU 2

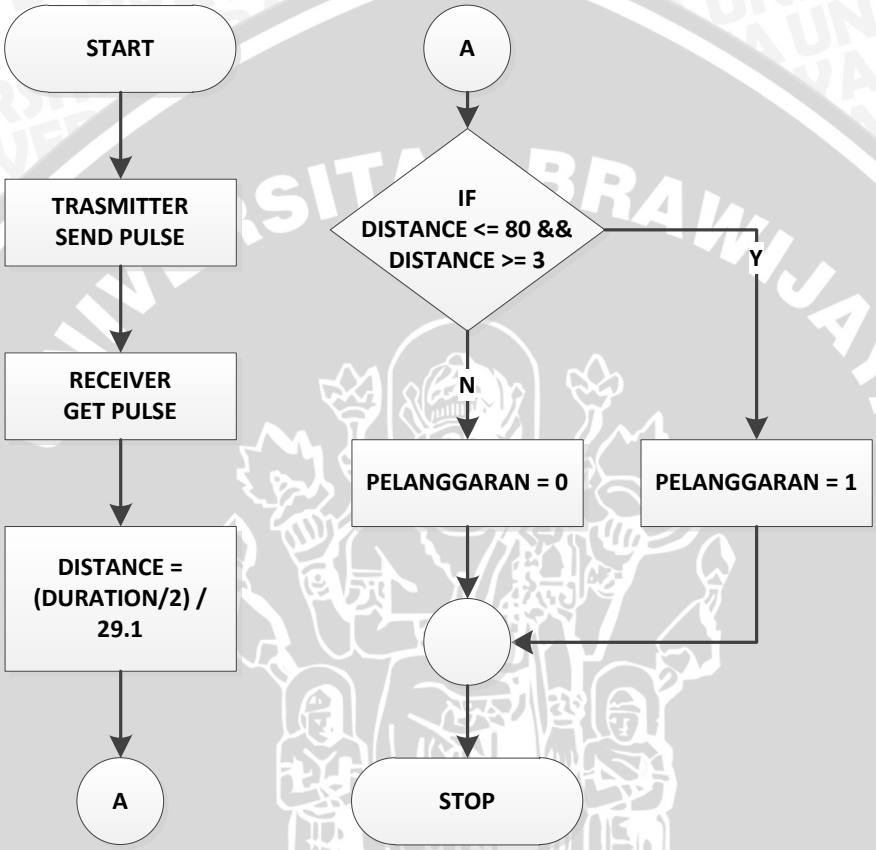
Perancangan perangkat keras RSU dan OPS sama yaitu hanya membutuhkan Arduino Nano dan nRF24L01. Pin yang digunakan oleh nRF24L01 adalah SCK, MISO, MOSI, 3V3, DIGITAL 7, DIGITAL 8 dan GND. Berikut gambar skematik perancangan perangkat keras pada RSU dan OPS :



Gambar 5.3 Skematik RSU Dan OPS

5.1.1.2 Perancangan Algoritma Sensor

Pada penelitian ini digunakan beberapa sensor untuk dapat mendeteksi adanya pelanggaran di jalan raya. Ada dua jenis sensor yang digunakan yaitu sensor *Ultrasonic* HC-SR04 dan sensor LDR. Sensor-sensor ini tidak dapat bekerja secara otomatis tanpa adanya sebuah algoritma perhitungan. Berikut algortima perhitungan pada sensor *Ultrasonic* HC-SR04 :

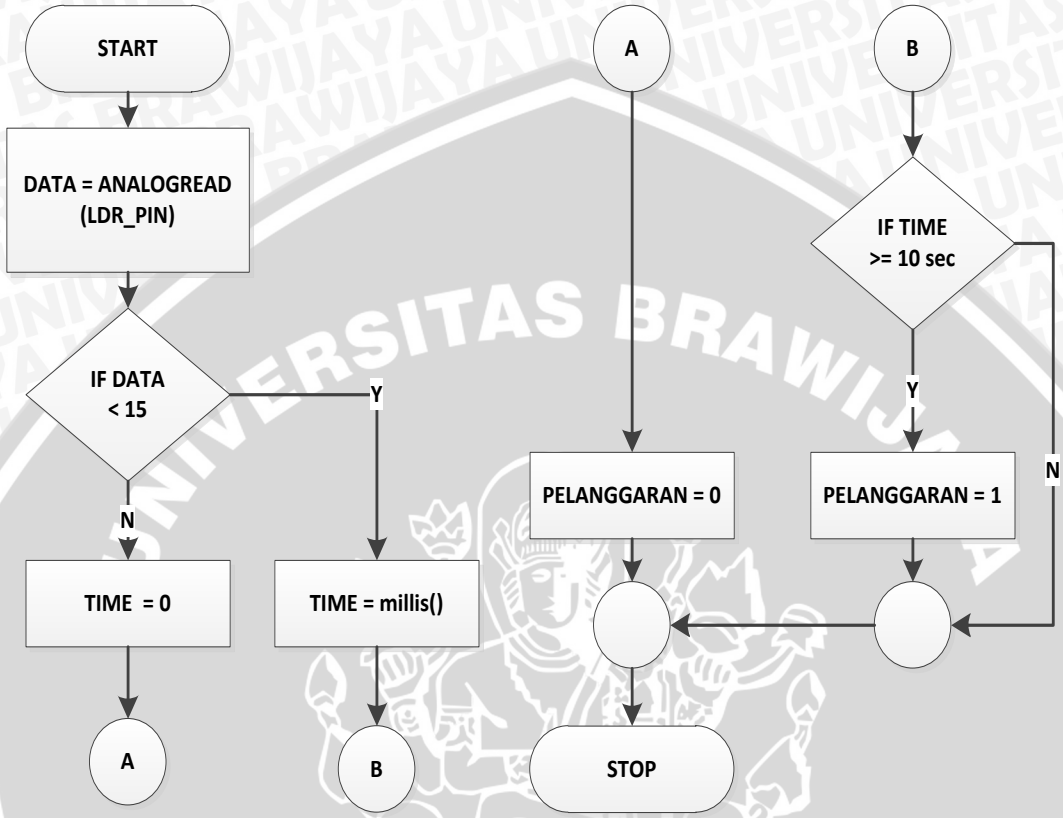


Gambar 5.4 Diagram alir Algoritma Sensor *Ultrasonic* HC-SR04

Gambar 5.4 merupakan sebuah proses sensor mendeteksi adanya pelanggaran pada rambu dilarang putar arah dan rambu dilarang masuk. Awal algoritma sensor akan mengukur jarak antara rambu dengan kendaraan. Untuk mengetahui jarak sensor akan mengirimkan sinyal kepada objek kendaraan, kemudian sinyal akan dipantulkan kembali ke sensor. Dengan ini sensor akan mendapatkan lama waktu saat sinyal dikirimkan hingga dipantulkan kembali. Dari waktu yang didapatkan dapat dihitung berapa jaraknya.

Setelah jarak sudah didapatkan, Arduino dapat menentukan apakah terjadi pelanggaran atau tidak dengan menggunakan *if-else.* Apabila jarak yang didapatkan dalam rentan 3 – 80 cm maka bisa dikatakan terjadi pelanggaran, sedangkan apabila diluar rentan tersebut maka tidak terjadi pelanggaran.

Jenis sensor yang kedua adalah sensor LDR, pada sistem ini LDR digunakan untuk membaca pelanggaran dilrang berhenti. Berikut algoritma perhitungan dalam sensor LDR :

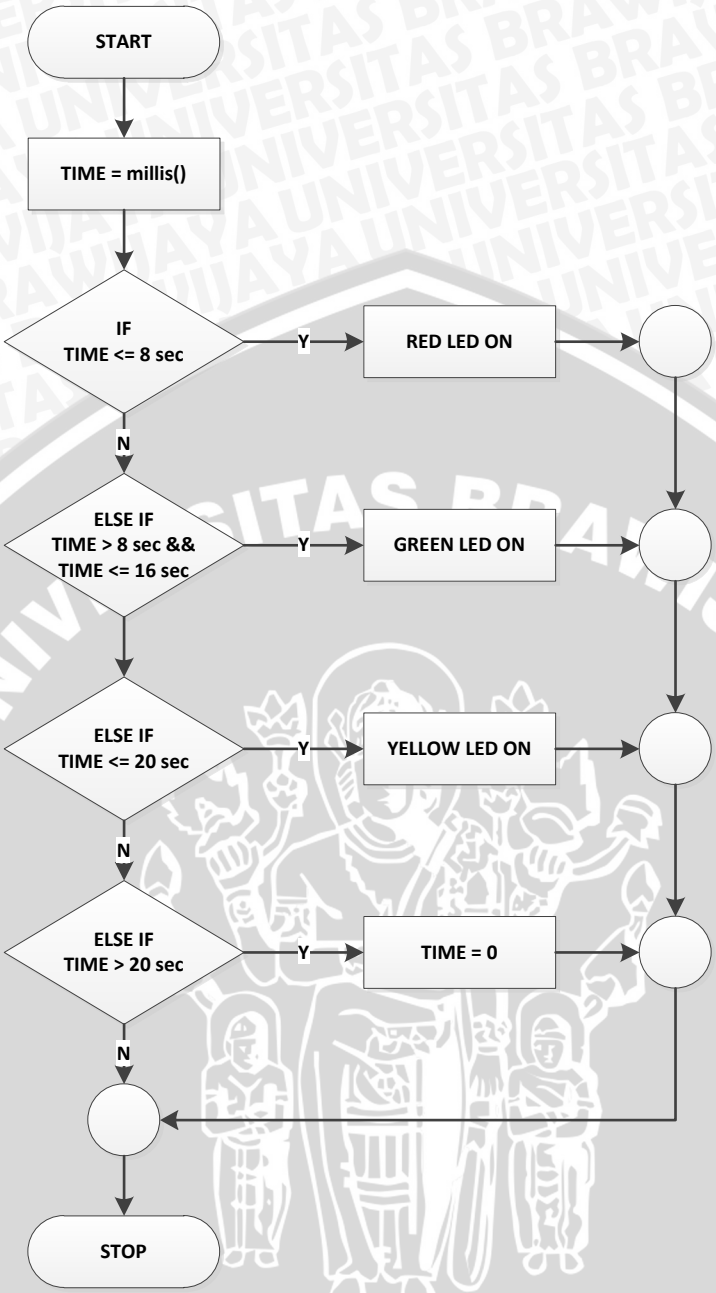


Gambar 5.5 Diagram alir Algoritma Sensor LDR

Gambar 5.5 adalah sebuah proses sebuah sensor LDR dapat menentukan ada tidaknya pelanggaran yang terjadi pada rambu dilarang berhenti. Pada awal algoritma Arduino mencoba membaca data dari pin analog yang digunakan oleh sensor. Data yang sudah didapatkan kemudian diproses dalam *if-else*, apabila nilai analog dari sensor adalah kurang dari 15 berarti ada kendaraan yang berhenti. Sedangkan apabil nilai analog lebih dari 15 maka tidak ada kendaraan yang berhenti.

Variabel waktu dalam algoritma digunakan untuk menentukan berapa detik sebuah kendaraan berhenti pada rambu dilarang berhenti. Apabila waktu lebih dari 10 detik atau lebih berarti sedang terjadi pelanggaran, sedangkan apabila waktu kurang dari 10 detik maka tidak terjadi pelanggaran.

Selain sensor dalam sistem ini juga dibuat sebuah miniatur lampu merah yang terbuat dari tiga buah lampu LED. Berikut algoritma lampu LED yang digunakan :



Gambar 5.6 Diagram alir Algoritma Lampu Merah LED

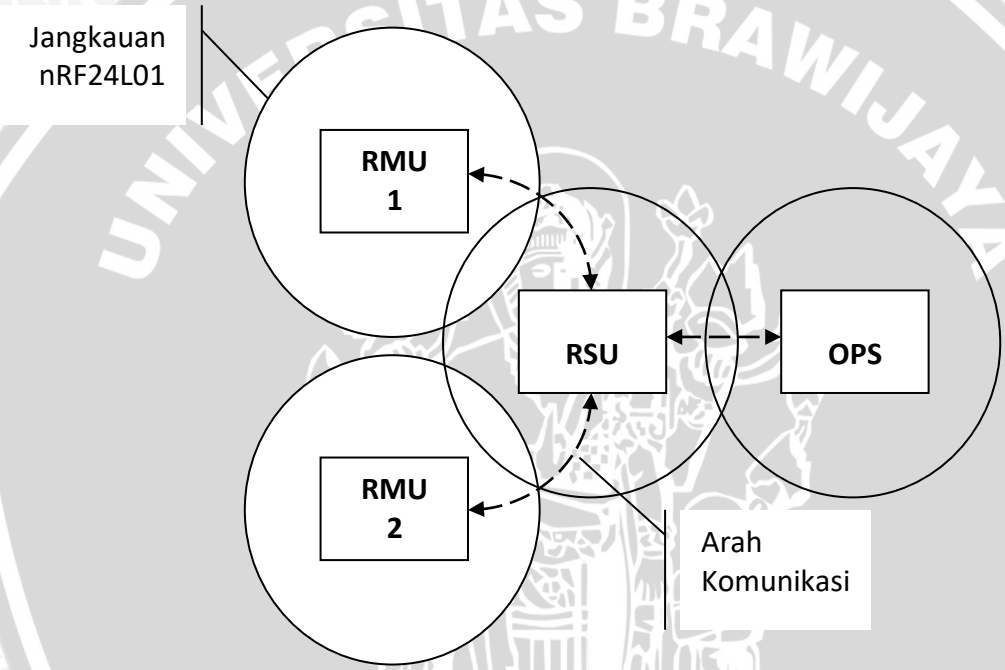
Gambar 5.6 adalah algoritma bagaimana lampu lalu lintas bekerja pada sistem ini. Digunakan sebuah variabel waktu untuk menentukan kapan tiga lampu LED dapat menyala secara bergantian. Apabila waktu bernilai 8 atau kurang maka lampu merah menyala, jika waktu bernilai antara rentan 9 – 16 maka lampu hijau yang menyala, jika waktu bernilai antara rentan 17 - 20 maka lampu kuning yang menyala, sedangkan apabila waktu lebih dari 20 maka waktu akan di reset menjadi 0.

5.1.2 Perancangan Komunikasi Data

Perancangan komunikasi data dibuat agar tidak terjadi tabrakan data dalam pengiriman antar node nantinya. Perancangan komunikasi data dibagi menjadi dua bagian yaitu perancangan algoritma pengiriman data dan perancangan format data.

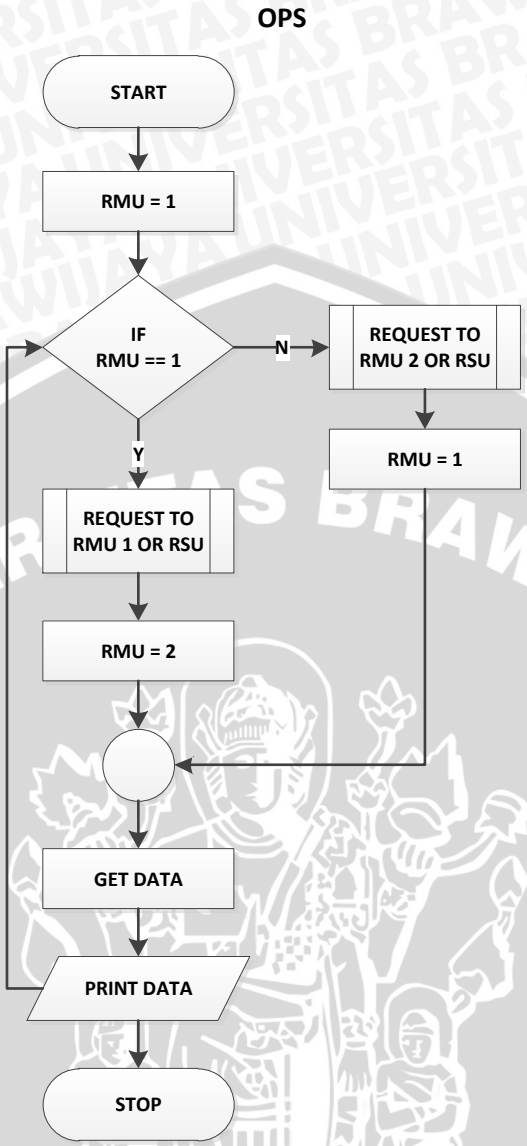
5.1.2.1 Perancangan Algoritma Pengiriman Data

Komunikasi yang dilakukan antar node bersifat *wireless* sehingga diperlukan sebuah algoritma agar tidak terjadi tabrakan data dalam proses pengiriman data. Berikut gambar arah komunikasi data pada sistem :



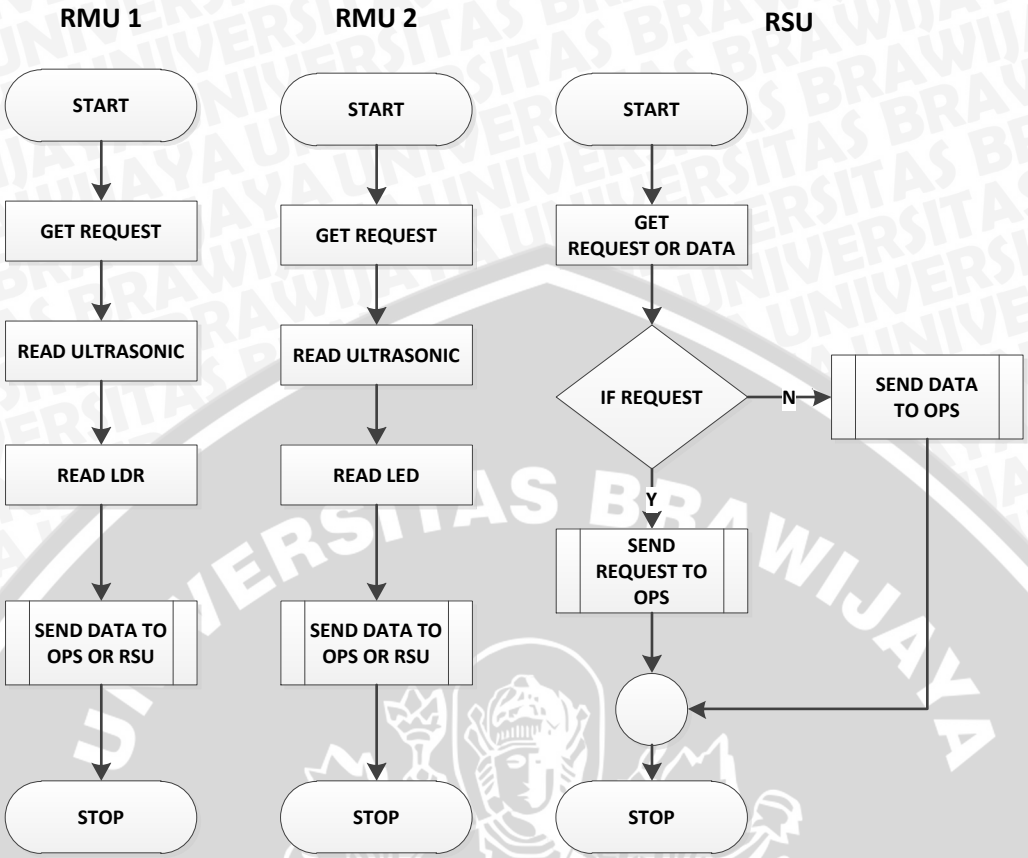
Gambar 5.7 Arah Komunikasi Data

Gambar 5.7 menunjukkan arah komunikasi data dari keempat node yang digunakan. *Request* dilakukan oleh OPS ke RMU secara langsung jika jarak pengiriman data masih aman. *Request* dikirimkan ke RSU apabila jarak pengiriman antara OPS dan RMU dirasa kurang aman sehingga diperlukan tambahan node ditengah-tengahnya. RSU digunakan sebagai penerus data antara OPS dan RMU. Berikut diagram alir cara kerja pada OPS :



Gambar 5.8 Diagram alir Algoritma Komunikasi OPS

Gambar 5.8 merupakan diagram alir dari komunikasi OPS yang diterapkan pada sistem. OPS akan menentukan kepada siapa *request* dikirimkan terlebih dahulu. Jika nilai variabel RMU adalah 1 maka *request* akan dikirimkan ke RMU 1 dan apabila nilai variabel RMU adalah 2 maka *request* akan dikirimkan ke RMU 2. OPS akan mengirimkan *request* secara langsung pada RMU jika jarak pengiriman dirasa aman atau dikirimkan ke RSU jika jarak pengiriman antara OPS dan RMU dirasa kurang aman. Setelah mengirimkan *request* OPS akan menunggu data sampai. Data yang sudah sampai pada OPS kemudian akan ditampilkan pada *serial monitor.*



Gambar 5.9 Diagram alir Algoritma Komunikasi Data RMU 1, 2 Dan RSU

Gambar 5.9 menunjukkan diagram alir komunikasi data pada RMU 1, RMU 2, dan RSU. Setelah RMU menerima *request* kemudian sensor akan membaca ada tidaknya pelanggaran dan kemudian mengirimkan datanya ke OPS apabila jarak pengiriman aman atau dikirimkan ke terlebih dahulu RSU apabila jarak pengiriman antara OPS dan RMU tidak aman. RSU bekerja saat menerima *request* dari OPS atau data dari RMU. Ketika data yang diterima dari OPS maka RSU akan mengirimkan *request* ke RMU. Ketika data yang diterima dari RMU maka RSU akan mengirimkan data ke OPS.

5.1.2.2 Perancangan Format Data

Perancangan format paket data diperlukan agar tiap-tiap node yang terlibat dalam pengiriman data dapat saling berkirim pesan. Sistem ini menggunakan format data *ShockBurst*™ yang sudah disediakan oleh nRF24L01. Berikut tabel format data *ShockBurst*™ :

Tabel 5.1 Format Data ShockBurst™

Preamble	*Address*	*Payload Data*	*CRC*
1 *byte*	3 – 5 *byte*	1 – 32 *byte*	1 – 2 *byte*

Pada sistem masing-masing *address* yang dibuat memiliki panjang 4 *byte* sehingga format data hanya memerlukan 4 *byte*. Data pelanggaran yang dihasilkan oleh sensor pada RMU bernilai 0 dan 1. Jika nilai 0 maka tidak ada pelanggaran dan 1 ada pelanggaran, sehingga hanya dibutuhkan 1 *bit*. Nilai yang dihasilkan oleh lampu lalu lintas adalah 0, 1 dan 2. Dimana nilai 0 adalah hijau, 1 adalah kuning, dan 2 adalah merah. Nilai lampu membutuhkan panjang data 2 *bit*. Dari kedua jenis data tersebut maka dibutuhkan *payload data* 3 *bit*. Agar OPS dapat mengenali asal data maka diperlukan 2 *bit* tambahan pada *sender*. Dikarenakan minimal *payload data* yang dikirimkan adalah 1 *byte* maka ditambahkan 3 *bit* tambahan sebagai penutup data. Berikut tabel format data yang digunakan :

Tabel 5.2 Format Data Yang Digunakan

HEADER		**DATA**	**FOOTER**
Address	*Sender*	*Payload Data*	*Foot*
4 *byte*	2 *bit*	3 *bit*	3 *bit*

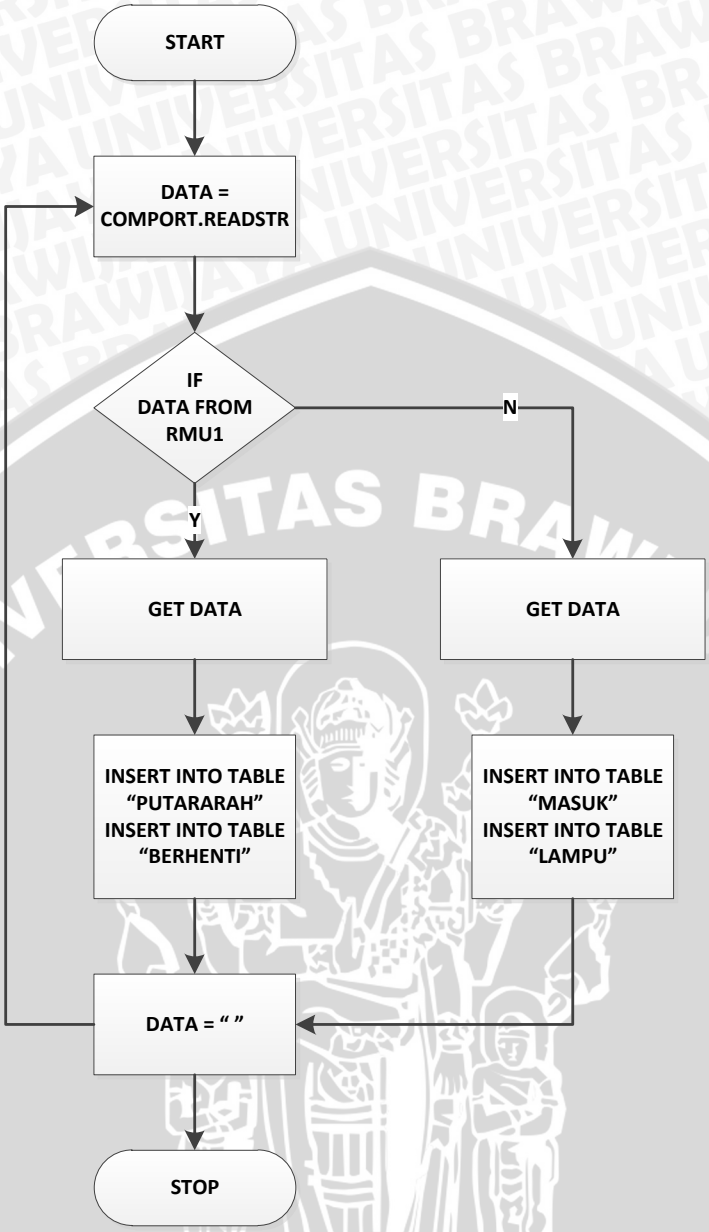
Address diisi oleh alamat RMU 1, RMU 2, RSU dan OPS seperti yang sudah dibuat, yakni : 'clnt' sebagai RMU 1 dan RMU 2, 'mstr' sebagai RSU dan 'srvr' sebagai OPS. *Sender* diisi dengan nilai 01 dan 10, dimana nilai 01 mewakili RMU 1 dan nilai 10 mewakili RMU 2. *Payload Data* diisi sesuai dengan data pelanggaran yang didapatkan dari sensor. *Foot* diisi 111 sebagai penutup data.

5.1.3 Perancangan *User interface*

User interface dibuat agar memudahkan saat melakukan monitoring pelanggaran rambu lalu lintas. Tampilan sistem memerlukan beberapa fungsi penting yaitu dapat mengambil data dari Arduino melalui *COM Port*, memecah data yang sudah terformat, menyimpan data pada *database*, dan menampilkan data pada grafik.

5.1.3.1 Perancangan Algoritma Pemecah Format Data

Dalam proses memecah data diperlukan algoritma agar data yang didapatkan dapat disimpan pada *database* sesuai dengan tabelnya masing-masing. Berikut algoritma pemecah data yang digunakan :



Gambar 5.10 Diagram alir Algoritma Pemecahan Data

Gambar 5.10 merupakan algoritma pemecahan data dari data yang sudah terformat menjadi data yang siap dimasukkan dalam *database*. Awal algoritma variabel data akan diisi oleh data dari Arduino. Kemudian dilakukan pengecekan apakah data yang diterima benar-benar ada isinya atau tidak.

Setelah itu dilakukanlah pengecekan apakah data berasal dari RMU 1 atau RMU 2. Data yang sudah diketahui darimana asalnya kemudian diambil datanya. Data yang diambil berada pada 1 *byte* terakhir format paket, yakni *bit* ke-3 sampai ke-5. *Bit* ke- 4 dari RMU 1 merupakan data pelanggaran dilarang putar arah dan *bit* ke-5 merupakan data pelanggaran dilarang berhenti. Jika data dari RMU 2, *bit* ke-3 merupakan data pelanggaran dilarang masuk dan *bit* ke-4 dan

ke-5 data lampu rambu lalu lintas. Setelah data dipecah kemudian dimasukkan ke *database* sesuai dengan tabelnya masing-masing

5.1.3.2 Perancangan Tabel Pada *Database*

Diperlukan sebuah *database* yang digunakan untuk menyimpan data-data yang telah didapatkan. Dibuat empat buah tabel karena menyesuaikan dengan jumlah rambu-rambu yang ada. Berikut perancangan tabel yang dibutuhkan :

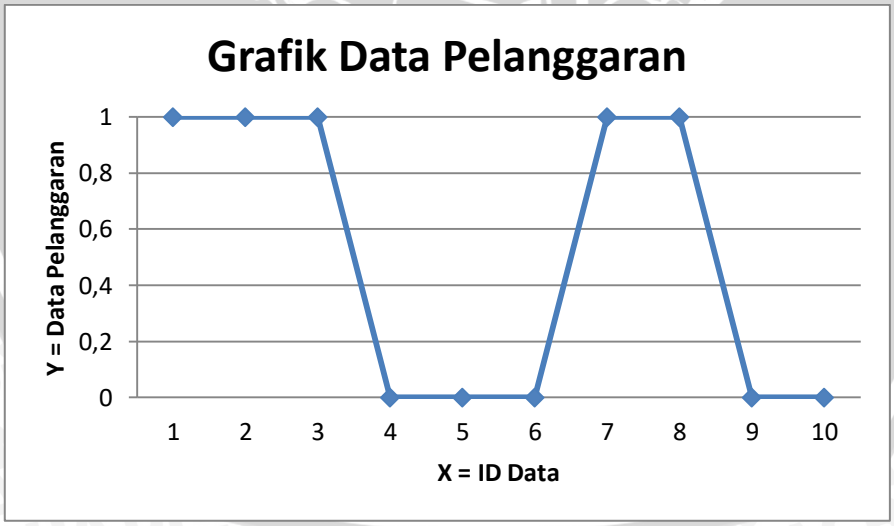
Tabel 5.3 Perancangan Tabel *Database*

No.	Nama Tabel	ID	Waktu	Data	Ping
1	Putar Arah	INT(10)	TIME(6)	INT(1)	INT(4)
2	Berhenti	INT(10)	TIME(6)	INT(1)	INT(4)
3	Masuk	INT(10)	TIME(6)	INT(1)	INT(4)
4	Lampu	INT(10)	TIME(6)	INT(1)	INT(4)

Tiap-tiap tabel memerlukan empat buah kolom, yaitu : ID, WAKTU, DATA dan PING. Tipe data yang digunakan adalah *integer* dan *time.* Kolom ID menggunakan tipe data *integer* dengan panjang 10 karakter, kolom WAKTU menggunakan tipe data *time* dengan panjang 6 karakter, kolom DATA menggunakan tipe data *integer* dengan panjang 1 karakter, dan kolom PING menggunakan tipe data *integer* dengan pnjng 10 krakter.

5.1.3.3 Perancangan Grafik Data

Data-data yang sudah tersimpan pada *database* kemudian akan ditampilkan dalam bentuk grafik. Berikut gambar grafik yang akan digunakan :



Gambar 5.11 Perancangan Grafik Data

Gambar 5.11 menunjukkan gambaran grafik yang akan ditampilkan pada *user interface.* Dibutuhkan empat grafik yang masing-masing mewakili tabel dari *database*. ID dari *database* akan mempresentasikan sumbu x, sedangkan DATA akan mempresentasikan sumbu y.

5.2 Implementasi

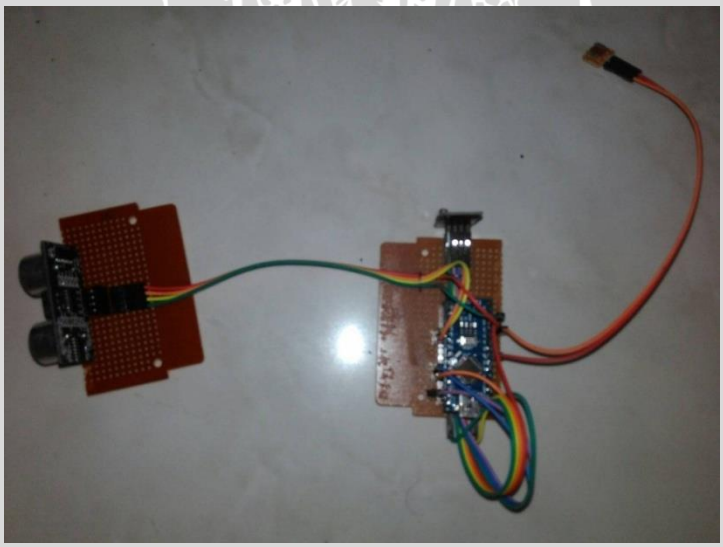
Sub bab ini membahas mengenai tahapan-tahapan membuat sebuah sistem yang sudah dibuat perencangannya. Implementasi meliputi pembuatan perangkat keras dan perangkat lunak dan kemudian digabungkan menjadi sebuah sistem. Implementasi meliputi penjelasan lingkungan implementasi, batasan implementasi dan implementasi sistem.

5.2.1 Implementasi RMU, RSU Dan OPS

Implementasi perangkat keras merupakan tahapan merangkai beberapa komponen perangkat keras menjadi satu. Implementasi dilakukan dengan mengikuti rangkaian perangkat keras pada perancangan. Beberapa komponen yang digunakan untuk membuat node, yaitu : Arduino Nano, nRF24L01, Sensor *Ultrasonic* HC-SR04 dan Sensor LDR.

5.2.1.1 Implementasi Rangkaian Node

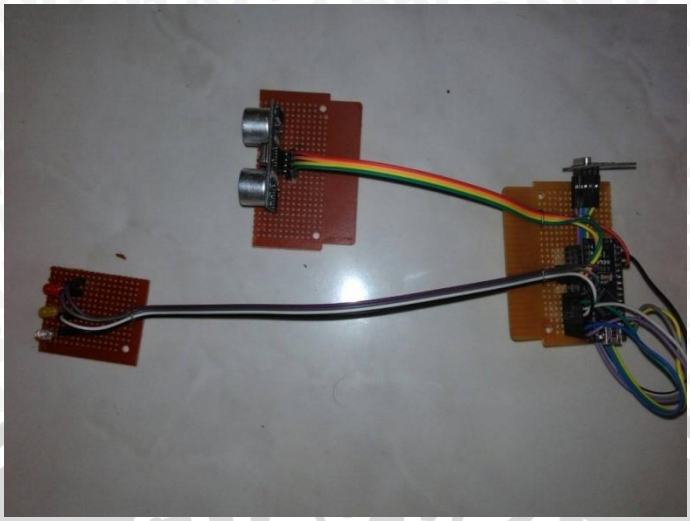
Implementasi perangkat keras pada RMU 1 yaitu pemasangan Arduino Nano, sensor *Ultrasonic* HC-SR04, sensor LDR dan nRF24L01 menjadi satu. Dengan pemasangan perangkat-perangkat ini node dapat melakukan pembacaan data dan dapat berkomunikasi dengan node lainnya.



Gambar 5.12 Implementasi RMU 1

Gambar 5.12 merupakan hasil jadi dari pengimplementasian RMU 1. Implementasi pada RMU 1 didasarkan pada gambar 5.1 yaitu skematik perancangan node 1.

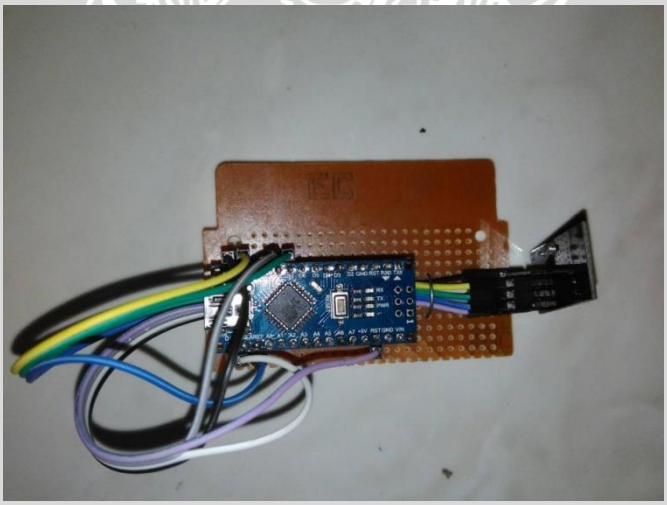
Implementasi perangkat keras pada RMU 2 yaitu pemasangan Arduino Nano, sensor *Ultrasonic* HC-SR04, Lampu LED dan nRF24L01 menjadi satu. Dengan pemasangan ini node dapat melakukan pembacaan data dan dapat berkomunikasi dengan node lainnya.



Gambar 5.13 Implementasi RMU 2

Gambar 5.13 merupakan hasil jadi dari pengimplementasian RMU 2. Implementasi pada RMU 2 didasarkan pada gambar 5.2 yaitu skematik perancangan node 2.

Implementasi perangkat keras pada RSU dan OPS yaitu pemasangan Arduino Nano dan nRF24L01 menjadi satu. Dengan pemasangan perangkat-perangkat ini node dapat melakukan komunikasi dengan node lainnya.



Gambar 5.14 Implementasi RSU Dan OPS

Gambar 5.14 merupakan hasil jadi dari pengimplementasian RSU dan OPS. Implementasi pada RSU dan OPS didasarkan pada gambar 5.3 yaitu skematik perancangan RSU dan OPS.

5.2.1.2 Implementasi Algoritma Sensor

Implementasi algoritma sensor merupakan tahapan pemberian kecerdasan pada perangkat keras agar sensor dapat bekerja sesuai dengan yang diinginkan. Implementasi dilakukan dengan mengkonversi algoritma-algoritma pada

perancangan menjadi sebuah *source code.* Pada tahapan ini digunakan Arduino IDE untuk proses compiling *source code.*

Implementasi algoritmna sensor pada RMU 1 ini dibuat berdasarkan diagram alir pada gambar 5.4 dan 5.5 yang merupakan algoritma sensor *ultrasonic* dan LDR. Berikut potongan *code* algoritma sensor *ultrasonic* dan LDR :

```
1.  …
2.  void setup(){
3.    …
4.  }
5.
6.  void sense(){
7.    …
8.    digitalWrite(trigPin, LOW);
9.    delayMicroseconds(2);
10.   digitalWrite(trigPin, HIGH);
11.   delayMicroseconds(10);
12.   digitalWrite(trigPin, LOW);
13.   long duration = pulseIn(echoPin, HIGH);
14.   long distance = (duration/2) / 29.1;
15.
16.   if (distance <= 80 && distance >= 3){
17.     x = 1;
18.   } else {
19.     x = 0;
20.   }
21.   …
22.
23.   int LDRReading = analogRead(LDR_Pin);
24.   delay(10);
25.   if (LDRReading < 15){
26.     if(violance == 1){
27.       stoptime = millis();
28.       violance = 0;
29.     }
30.   } else if(LDRReading >= 15){
31.     stoptime = millis();
32.     violance = 1;
33.   }
34.
35.   if(millis() - stoptime > 10000){
36.     y = 1;
37.   } else {
38.     y = 0;
39.   }
40. …
41. }
42.
43. void loop(){
44.   …
45.       sense();
46.   …
47. }
```

Potongan *code* di atas menunjukkan *code* pembacaan data pelanggaran pada rambu dilarang putar arah dan rambu dilarang berhenti pada RMU 1. Baris 8 – 20

pengimplementasian diagram alir pembacaan sensor *ultrasonic* pada gambar 5.4. Sedangkan pada baris 23 – 39 pengimplementasian diagram alir pembacaan sensor LDR pada gambar 5.5.

Implementasi algoritmna sensor pada RMU 2 ini dibuat berdasarkan diagram alir pada gambar 5.4 dan 5.6 yang merupakan algoritma sensor *ultrasonic* dan Lampu LED. Berikut potongan *code* algoritma sensor *ultrasonic* dan LED :

```
1.  …
2.  void setup(){
3.    …
4.  }
5.
6.  void sense(){
7.    readtime = (millis());
8.    digitalWrite(trigPin, LOW);
9.    delayMicroseconds(2);
10.   digitalWrite(trigPin, HIGH);
11.   delayMicroseconds(10);
12.   digitalWrite(trigPin, LOW);
13.   long duration = pulseIn(echoPin, HIGH);
14.   long distance = (duration/2) / 29.1;
15.
16.   if (distance <= 80 && distance >= 3){
17.     x = 1;
18.   } else {
19.     x = 0;
20.   }
21.   …
22. }
23.
24. void led(){
25.   if (millis() - ledtime <= 8000){
26.     ledcolour = 'M';
27.     y = 0;
28.     z = 1;
29.     digitalWrite(red, HIGH);
30.     digitalWrite(green, LOW);
31.     digitalWrite(yellow, LOW);
32.   } else if ((millis() - ledtime > 8000) && (millis() -
33. ledtime <= 16000)){
34.     ledcolour = 'H';
35.     y = 1;
36.     z = 0;
37.     digitalWrite(red, LOW);
38.     digitalWrite(green, HIGH);
39.     digitalWrite(yellow, LOW);
40.   } else if (millis() - ledtime <= 20000){
41.     ledcolour = 'K';
42.     y = 1;
43.     z = 1;
44.     digitalWrite(red, LOW);
45.     digitalWrite(green, LOW);
46.     digitalWrite(yellow, HIGH);
47.   }
48.   else{
49.     ledtime = millis();
50.   }
```

```
51.  }
52.
53.  void loop(){
54.    led();
55.    …
56.        sense();
57.    …
58.  }
```

Potongan *code* di atas menunjukkan *code* pembacaan data pelanggaran pada rambu dilarang masuk dan lampu merah. Baris 7 – 20 pengimplementasian diagram alir pembacaan sensor *ultrasonic* pada gambar 5.4. Sedangkan pada baris 24 – 51 pengimplementasian diagram alir pembacaan lampu LED pada gambar 5.6.

5.2.2 Implementasi Komunikasi Data

Implementasi komunikasi data merupakan tahapan pemberian kecerdasan pada perangkat keras agar tiap-tiap node dapat mengirimkan data sesuai dengan yang diinginkan. Implementasi dilakukan dengan mengkonversi algoritma-algoritma pada perancangan menjadi sebuah *source code.* Pada tahapan ini digunakan Arduino IDE untuk proses compiling *source code.*

5.2.2.1 Implementasi Algoritma Pengiriman Data

Implementasi algoritma pengiriman data pada RMU 1 ini dibuat berdasarkan diagram alir pada gambar 5.9 yang merupakan algoritma pengiriman data RMU 1. Berikut potongan *code* algoritma pengiriman data RMU 1 :

```
1.  …
2.  void setup() {
3.    …
4.  }
5.
6.  void sense() {
7.    …
8.
9.    Mirf.setTADDR((byte *)"srvr");
10.   Mirf.send((byte *)datasend);
11.   while (Mirf.isSending()) {
12.   }
13.   delay(10);
14. }
15.
16. void loop() {
17.   if (!Mirf.isSending() && Mirf.dataReady()) {
18.     Mirf.getData(datareceive);
19.     if (datareceive[0] == 127) {
20.       …
21.       sense();
22.     }
23.   }
24. }
```

Potongan *code* di atas menunjukkan *code* pengiriman data pada RMU 1 yang dibuat berdasarkan diagram alir pada gambar 5.9. Baris 6 – 14 RMU 1 mengirimkan data ke OPS. Pada baris 16 – 24 merupakan proses dimana RMU 1 menerima *request* dari OPS.

Implementasi algoritma pengiriman data pada RMU 2 ini dibuat berdasarkan diagram alir pada gambar 5.9 yang merupakan algoritma pengiriman data RMU 2. Berikut potongan *code* algoritma pengiriman data RMU 2 :

```
1.  …
2.
3.  void setup() {
4.    …
5.  }
6.
7.  void sense() {
8.    …
9.
10.   Mirf.setTADDR((byte *)"srvr");
11.   Mirf.send((byte *)datasend);
12.   while (Mirf.isSending()) {
13.   }
14.   delay(10);
15. }
16.
17. void led() {
18.   …
19. }
20.
21. void loop() {
22.   …
23.   if (!Mirf.isSending() && Mirf.dataReady()) {
24.     Mirf.getData(datareceive);
25.     if (datareceive[0] == 191) {
26.       …
27.       sense();
28.     }
29.   }
30. }
```

Potongan *code* di atas menunjukkan *code* pengiriman data pada RMU 2 yang dibuat berdasarkan diagram alir pada gambar 5.9. Baris 7 – 15 RMU 2 mengirimkan data pada ke OPS. Pada baris 21 – 30 merupakan proses dimana RMU 2 menerima *request* dari OPS.

Implementasi ini dibuat berdasarkan diagram alir pada gambar 5.9 yang merupakan algoritma pengiriman data pada RSU. Berikut potongan *code* algoritma pengiriman data pada RSU :

```
1.  …
2.  void setup() {
3.    …
4.  }
5.
6.  void rmu(int id_rmu) {
7.    …
8.    Mirf.setTADDR((byte *)"clnt");
```

```
9.      Mirf.send((byte *)datasend);
10.     while (Mirf.isSending()) {}
11.     delay(5);
12.   }
13.
14.   void ops() {
15.     …
16.     Mirf.setTADDR((byte *)"srvr");
17.     Mirf.send((byte *)datasend);
18.     while (Mirf.isSending()) {}
19.     delay(5);
20.   }
21.
22.   void loop() {
23.     …
24.         rmu(getdata);
25.     …
26.         ops();
27.       }
28.     }
29.   }
```

Potongan *code* di atas menunjukkan *code* pengiriman data pada RSU yang dibuat berdasarkan diagram alir pada gambar 5.9. Baris 6 – 12 proses *request* dari RSU ke RMU. Baris 14 – 20 proses pengiriman data dari RSU ke OPS. Baris 22 – 29 digunakan untuk menentukan kapan RSU meminta data ke RMU dan kapan harus mengirim data ke OPS.

Implementasi ini dibuat berdasarkan diagram alir pada gambar 5.8 yang merupakan algoritma pengiriman data pada OPS. Berikut *code* algoritma pengiriman data pada OPS :

```
1.    …
2.
3.    void setup() {
4.      …
5.    }
6.
7.    void recv() {
8.      if (!Mirf.isSending() && Mirf.dataReady()) {
9.        Mirf.getData(datareceive);
10.       …
11.     }
12.   }
13.
14.   void request() {
15.     if (rmu == 1) {
16.       rmu = 2;
17.       datasend[0] = 127;
18.     } else if (rmu == 2) {
19.       rmu = 1;
20.       datasend[0] = 191;
21.     }
22.     Mirf.setTADDR((byte *)"clnt");
23.     Mirf.send((byte *)datasend);
24.     while (Mirf.isSending()) {}
25.     delay(5);
```

```
26.   …
27. }
28.
29. void loop() {
30.   …
31.   request();
32.   recv();
33. }
```

Potongan *code* di atas menunjukkan *code* penerimaan data pada OPS yang dibuat berdasarkan diagram alir 5.8. Baris 7 – 12 merupakan proses pembacaan data yang diterima oleh OPS. Baris 14 – 27 merupakan proses *request* ke RMU.

5.2.2.2 Implementasi Format Data

Implementasi format data pada RMU 1 dibuat berdasarkan tabel 5.2 yang merupakan format data RMU. Berikut *code* untuk membuat format data pada RMU 1 :

```
1.  …
2.  void sense(){
3.    …
4.    if (x == 0 && y == 0) {
5.      datasend[0] = 71;
6.    } else if (x == 0 && y == 1) {
7.      datasend[0] = 79;
8.    } else if (x == 1 && y == 0) {
9.      datasend[0] = 87;
10.   } else if (x == 1 && y == 1) {
11.     datasend[0] = 95;
12.   }
13.   …
14. }
15. …
16.
```

Potongan *code* di atas menunjukkan *code* pembuatan format data pada RMU 1. Baris 4 – 12 merupakan isi dari *byte* terakhir yaitu data sensor dan *tail*.

Implementasi format data pada RMU 2 dibuat berdasarkan tabel 5.2 yang merupakan format data RMU. Berikut *code* untuk membuat format data pada RMU 2 :

```
1.  …
2.  void sense(){
3.    …
4.    if (x == 0 && y == 0 && z == 1) {
5.      datasend[0] = 143;
6.    } else if (x == 0 && y == 1 && z == 0) {
7.      datasend[0] = 151;
8.    } else if (x == 0 && y == 1 && z == 1) {
9.      datasend[0] = 159;
10.   } else if (x == 1 && y == 0 && z == 1) {
11.     datasend[0] = 175;
12.   } else if (x == 1 && y == 1 && z == 0) {
13.     datasend[0] = 183;
14.   } else if (x == 1 && y == 1 && z == 1) {
```

```
15.       datasend[0] = 191;
16.     }
17.     …
18.   }
```

Potongan *code* di atas menunjukkan *code* pembuatan format data pada RMU 2. Baris 4 – 16 merupakan isi dari *byte* terakhir yaitu data sensor dan *tail*.

Implementasi format data pada RSU dibuat berdasarkan tabel 5.5 yang merupakan format data RSU. Berikut *code* untuk membuat format data pada RSU :

```
1.  …
2.  void request(int id_req){
3.    if (rmu == 1) {
4.      rmu = 2;
5.      datasend[0] = 127;
6.    } else if (rmu == 2) {
7.      rmu = 1;
8.      datasend[0] = 191;
9.    }
10. }
11. …
```

Potongan *code* di atas menunjukkan *code* pembuatan format data pada OPS. Baris 3 – 5 merupakan format *request* data yang akan dikirimkan ke RMU 1. Baris 6 – 8 merupakan format *request* data yang akan dikirimkan ke RMU 2.

5.2.3 Implementasi *User interface* Pada Server

Implementasi *User interface* Pada Server merupakan tahapan pembuatan tampilan sesuai dengan fungsi-fungsi yang ada pada percangan. Implementasi dilakukan dengan mengkonversi algoritma dan perancangan lainnya menjadi sebuah *source code.* Pada tahapan ini digunakan Embarcadero RAD Studio 2010 untuk proses compiling *source code* dan XAMPP untuk proses pembuatan *database.*

5.2.3.1 Implementasi Algoritma Pemecah Format Data

Implementasi algoritma pemecah format data adalah proses konversi dari algoritma pada gambar 5.10 menjadi *source code*. Berikut potongan *code* yang digunakan untuk memcah data yang sudah terformat :

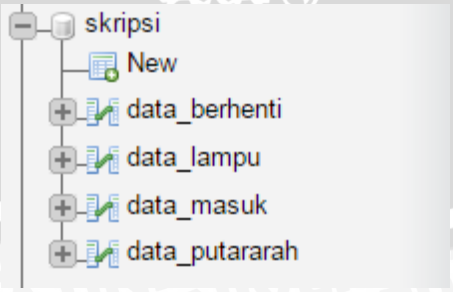
```
1.  procedure   TForm1.ComPortRxChar(Sender:   TObject;   Count:
2.  Integer);
3.  ...
4.  begin;
5.  ...
6.  ComPort.ReadStr(Str, Count);
7.  Data := Data + Str;
8.  ...
9.  if(pos('Z', Str) > 0) then
10.   begin
11.     if((pos('1', Str) = 1) and (pos('Z', Str) = 6)) then
12.     begin
```

```
13.       nilai := StrToInt(Copy(Data, pos('A', Data) + 1,
14.                (pos('B', Data) 1) - (pos('A', Data))));
15.       Data_PutarArah := (IntToStr(nilai));
16.       nilai := StrToInt(Copy(Data, pos('B', Data) + 1,
17.                (pos('Z', Data) - 1) - (pos('B', Data))));
18.       Data_Berhenti := (IntToStr(nilai));
19.       nilai := StrToInt(Copy(Data, pos('Z', Data) + 1,
20.                (pos('P', Data) - 1) - (pos('Z', Data))));
21.       Ping_Node1 := (IntToStr(nilai));      …
22.       …
23.     if((pos('2', Str) = 1) and (pos('Z', Str) = 7))then
24.     begin
25.       nilai := StrToInt(Copy(Data, pos('A', Data) + 1,
26.                (pos('B', Data)- 1) - (pos('A', Data))));
27.       Data_Masuk := (IntToStr(nilai));
28.       nilai := StrToInt(Copy(Data, pos('B', Data) + 1,
29.                (pos('Z', Data) - 1) - (pos('B', Data))));
30.       Data_Lampu := (IntToStr(nilai));
31.       nilai := StrToInt(Copy(Data, pos('Z', Data) + 1,
32.                (pos('P', Data) - 1) - (pos('Z', Data))));
33.       Ping_Node2 := (IntToStr(nilai));
34.       …
35.     Data := '';
36.   end;
37. end;
```

Pada baris 6 merupakan proses pengambilan data dari Arduino, sedangakan baris 7 merupakan proses memasukkan data pada sebuah variabel *Data.* Baris 9 digunakan untuk mengecek apakah varibel *Data* benar-benar ada isinya atau tidak. Baris 11 – 21 merupakan algoritma untuk memecah data dari RMU 1. Baris 23 – 33 merupakan algoritma untuk memecah data dari RMU 2. Baris 35 digunakan untuk mengosongkan variabel *Data.*

5.2.3.2 Implementasi Tabel Pada *Database*

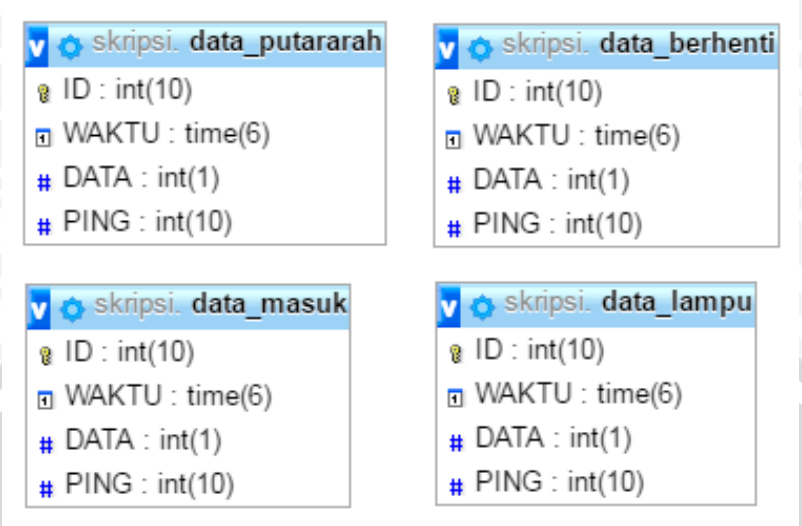
Implementasi tabel pada *database* adalah proses pembuatan tabel-tabel pada *database* sesuai dengan perancangan yang sudah ada. Proses pembuatan dilakukan sesuai dengan tabel perancangan 5.7 dengan menggunakan XAMPP. Berikut *database* serta tabel-tabel yang sudah dibuat :



Gambar 5.15 *Database* Sistem

Gambar 5.15 menunjukkan *database* dengan nama skirpsi, kemudian terdapat empat buah tabel dengan nama data_berhenti, data_lampu,

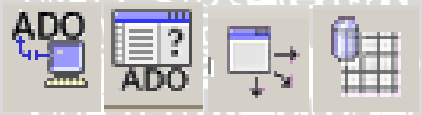
data_masuk dan data_putararah. Berikut struktur dari masing-masing tabel yang telah dibuat :



Gambar 5.16 Struktur Tabel *Database*

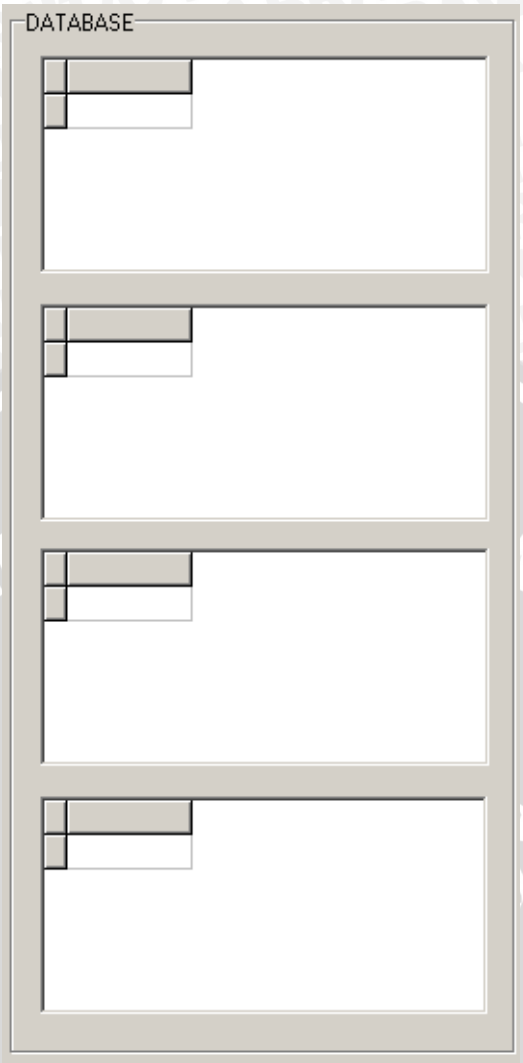
Gambar 5.16 merupakan struktur tabel dari masing-masing tabel pada *database*. Seperti yang sudah dibahas pada tabel perancangan 5.7 tiap-tiap tabel memiliki empat kolom yang bernama ID, WAKTU, DATA dan PING. Kolom ID pada tabel merupakan *primary key* dari masing-masing tabel.

User interface dapat mengakses *database* dengan menggunakan *OLEDB Connection* yang sudah disediakan oleh *Microsoft.* Diperlukan beberapa komponen agar dapat menyimpan dan menampilkan tabel *database*. Berikut gambar komponen-komponen yang diperlukan :



Gambar 5.17 Komponen Untuk Mengakses *Database*

Gambar 5.17 menunjukkan empat buah komponen yang dinamkan *ADOConnection*, *ADOQuery*, *DataSource* dan *DBGrid. ADOConnection* digunakan untuk mengakses *database* yang sudah dibuat. *ADOQuery* digunakan untuk memberikan perintah *Query*pada *database*. *DataSource* digunakan untuk mengakses tabel yang berada pada *database* sesuai dengan perintah yang sudah dimasukkan dalam *ADOQuery.DBGrid* digunakan untuk menampilkan tabel yang sudah diakses oleh *DataSource.* Berikut gambar tampilan tabel pada *User interface* yang akan ditampilkan :



Gambar 5.18 Tampilan Tabel Pada Form

Gambar 5.18 merupakan tampilan tabel yang akan ditampilkan pada form. Masing-masing *DBGrid* mewakilik tabel-tabel yang terdapat pada *database*. Selain menggunakan beberapa komponen agar dapat menampilkan dan menyimpan pada *database* juga diperlukan beberapa baris *code*. Berikut *code* yang digunakan untuk menyimpan data pada *database* :

```
1.  procedure TForm1.ComPortRxChar(Sender: TObject; Count:
2.  Integer);
3.  ...
4.  begin
5.  ...
6.  ADOQuery1.FieldByName('WAKTU').AsString:=formatdatetime('hh
7.  :nn:ss', Time);
8.  ADOQuery1.FieldByName('DATA').AsString:=Data_PutarArah;
9.  ADOQuery1.FieldByName('PING').AsString:=Ping_Node1;
10. ADOQuery1.Post;
11. ...
12. ADOQuery2.FieldByName('WAKTU').AsString:=formatdatetime('hh
13. :nn:ss', Time);
14. ADOQuery2.FieldByName('DATA').AsString:=Data_Berhenti;
```

```
15.  ADOQuery1.FieldByName('PING').AsString:=Ping_Node1;
16.  ADOQuery2.Post;
17.  ...
18.  ADOQuery3.FieldByName('WAKTU').AsString:=formatdatetime('hh
19.  :nn:ss', Time);
20.  ADOQuery3.FieldByName('DATA').AsString:=Data_Masuk;
21.  ADOQuery3.FieldByName('PING').AsString:=Ping_Node2;
22.  ADOQuery3.Post;
23.  ...
24.  ADOQuery4.FieldByName('WAKTU').AsString:=formatdatetime('hh
25.  :nn:ss', Time);
26.  ADOQuery4.FieldByName('DATA').AsString:=Data_Lampu;
27.  ADOQuery4.FieldByName('PING').AsString:=Ping_Node2;
28.  ADOQuery4.Post;
29.  ...
30.  end;
```

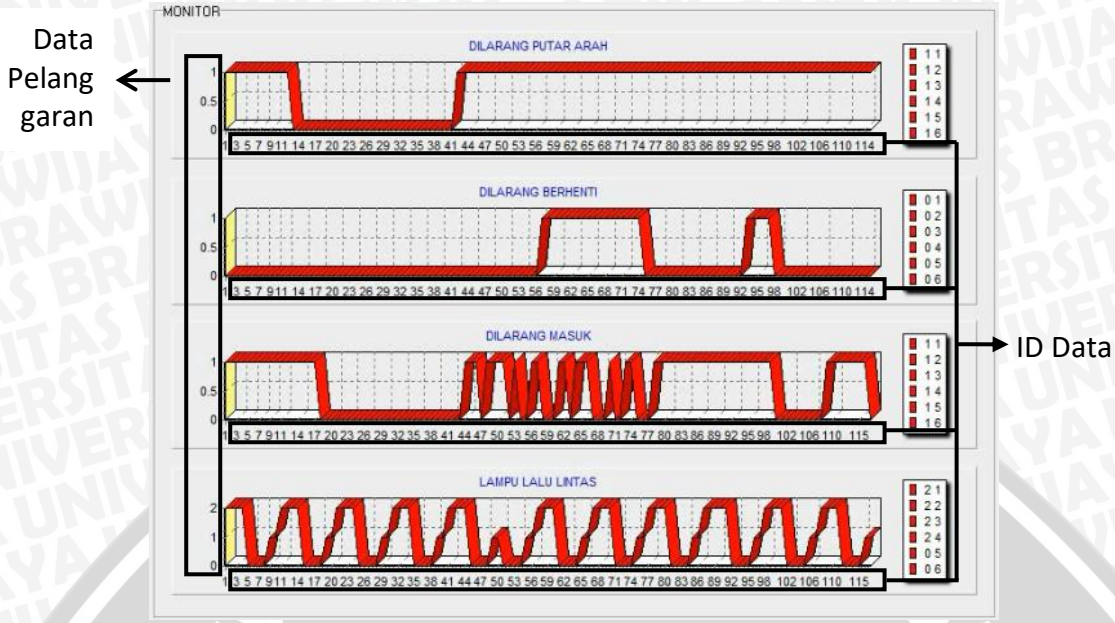
Tiap *DBGrid* diberikan masing-masing satu *ADOQuery* agar dapat menampilkan dan menyimpan data pada tabel yang sesuai. Baris 6 – 10 digunakan untuk menyimpan data pelanggaran rambu dilarang putar arah. Baris 12 – 16 digunakan untuk menyimpan data pelanggaran rambu dilarang berhenti. Baris 18 – 22 digunakan untuk menyimpan data pelanggaran rambu dilarang masuk. Baris 24 – 28 digunakan untuk menyimpan data lampu merah.

5.2.3.3 Implementasi Grafik Data

Implementasi grafik data adalah proses konversi data dari *database* menjadi sebuah grafik. Karena grafik dibuat dengan mengambil data dari *database* sehingga diperlukan sebuah komponen tambahan. Berikut gambar komponen yang digunakan :

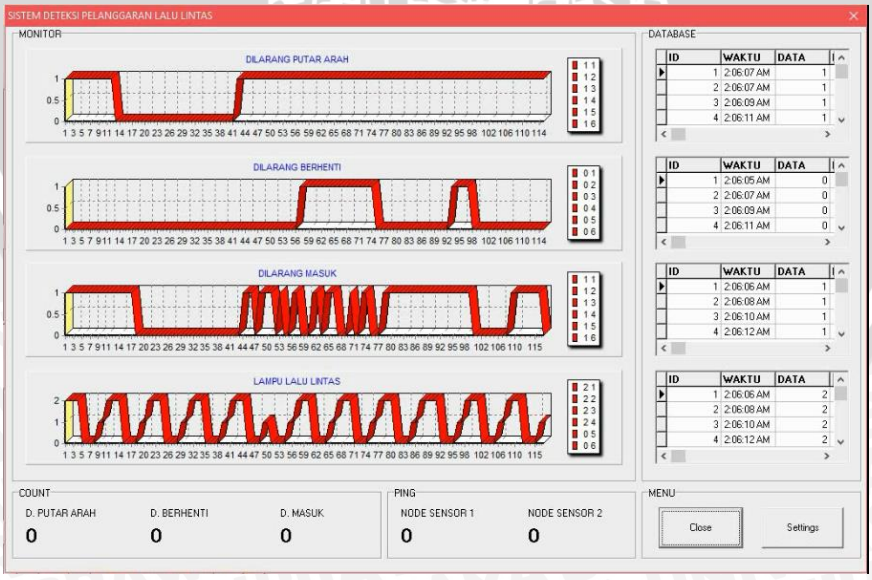
Gambar 5.19 Komponen Untuk Menampilkan Grafik

Gambar 5.19 menunjukkan gambar ikon dari komponen *DBChart*, komponen inilah yang merubah data dari masing-masing tabel menjadi grafik. Berikut gambar grafik yang dihasilkan oleh *DBChart* :



Gambar 5.20 Tampilan Grafik Pada Form

Gambar 5.20 menunjukkan grafik yang akan ditampilkan pada *user interface* sistem. Terdapat empat buah grafik yang masing-masing mewakili tiap-tiap rambu. *DBChart* memerlukan bantuan dari *ADOQuery* untuk dapat menghasilkan garis pada grafik. Sumbu x pada grafik merupakan ID dari masing-masing data pelanggaran, sedangkan sumbu y bernilai 0, 1 dan 2. Pada grafik dilarang putar arah, dilarang berhenti dan dilarang masuk jika bernilai 1 maka terdeteksi ada pelanggaran dan jika bernilai 0 maka tidak terdeteki adanya pelanggaran. Sedangkan pada grafik lampu lalu lintas nilai 0 berarti lampu sedang menyala merah, nilai 1 ketika lampu sedang menyala kuning dan nilai 2 ketika lampu menyala hijau.



Gambar 5.21 *User interface* Sistem

Gambar 5.21 menunjukkan tampilan sistem yang sudah diberikan fungsi-fungsi seperti pada tabel perancangan 4.5. Pada form terdapat *GroupBox* MONITORING untuk meletakkan grafik, *GroupBox DATABASE* untuk meletakkan tabel *database*, *GroupBox* COUNT digunakan untuk menghitung pelanggaran yang sudah terjadi, *GroupBox* PING digunakan untuk menampilkan berapa waktu data sampai ke OPS dan *GroupBox* MENU digunakan untuk meletakkan tombol-tombol.

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk mengetahui bahwa semua kebutuhan fungsional maupun non-fungsional yang dirancang sebelumnya telah terpenuhi. Hal-hal yang diuji pada sistem ini yaitu mengenai fungsional dari perangkat keras dan perangkat lunak yang digunakan pada sistem tersebut.

Analisis sistem dilakukan dengan membandingkan data pada pengujian dibandingkan dengan hipotesa dan ditarik kesimpulan dari setiap pengujian yang ada pada setiap sub bab.

6.1 Pengujian Fungsional Hardware

Pengujian fungsional hardware dilakukan untuk menguji perangkat keras yang digunakan pada sistem. Perangkat keras yang akan diuji antara lain : sensor *ultrasonic* HC-SR04, sensor LDR dan lampu LED.

6.1.1 Pengujian Sensor *Ultrasonic* HC-SR04

Sensor *ultrasonic* HC-SR04 adalah sensor yang digunakan untuk membaca data pelanggaran pada rambu dilarang putar arah dan dilarang masuk. Sensor ini akan mendeteksi adanya pelanggaran dan memberikan keluaran 1. Sedangakan sensor akan memberikan keluaran 0 saat tidak ada pelanggaran.

6.1.1.1 Tujuan

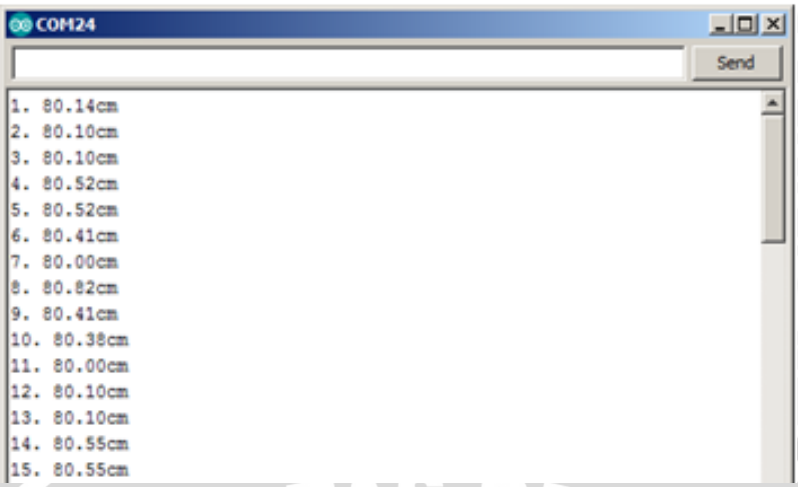
Pengujian sensor *ultrasonic* HC-SR04 dilakukan untuk mengetahui berapa rentan jarak yang dapat dibaca sensor secara akurat dan berapa rentan jarak yang dibaca sensor secara tidak akurat.

6.1.1.2 Skenario

Pengujian dilakukan dengan menggunakan sensor *ultrasonic* jenis HC-SR04 yang kemudian dihubungkan dengan Arduino Nano. Pada pengujian akan ditentukan jarak yang akan diukur oleh sensor menggunakan alat ukur. Agar sensor dapat melakukan perhitungan jarak diletakkan sebuah objek di depan sensor. Pengujian dilakukan dengan menambahkan jarak antara sensor dan objek sampai sensor tidak dapat membaca secara akurat lagi. Hasil pembacaan sensor akan ditampilkan pada *Serial Monitor* pada komputer. Hasil yang akan ditampilkan adalah jarak dalam satuan *centimeter*. Data dari sensor dianggap akurat apabila jarak pembacaan sensor sama dengan pengukuran secara manual menggunakan penggaris.

6.1.1.3 Hasil

Pada hasil pengujian sensor *ultrasonic* HC-SR04 yang akan ditunjukkan adalah jarak yang diuji, hasil pembacaan sensor, dan rata-rata keakuratan pembacaan sensor. Hasil didapatkan dengan menguji sensor sebanyak 50 kali pada jarak yang sudah ditentukkan.



Gambar 6.1 Contoh Pengujian Sensor *Ultrasonic* HC-SR04

Gambar 6.1 merupakan salah satu hasil pembacaan sensor *ultrasonic* HC-SR04 pada *Serial Monitor*. Pada gambar dapat dilihat jarak yang didapatkan oleh sensor antara 80 cm yang dimana jarak tersebut sensor masih bisa melakukan pembacaan jarak dengan akurasi 100%. Nilai keakuratan data didapat dengan rumus sebagai berikut :

$$Keakuratan\ Data = \frac{\sum data\ yang\ akurat}{\sum data}$$

Berikut contoh perhitungan pada pengujian jarak 80 cm :

$$Keakuratan\ Data = \frac{90}{100} = 90\ \%$$

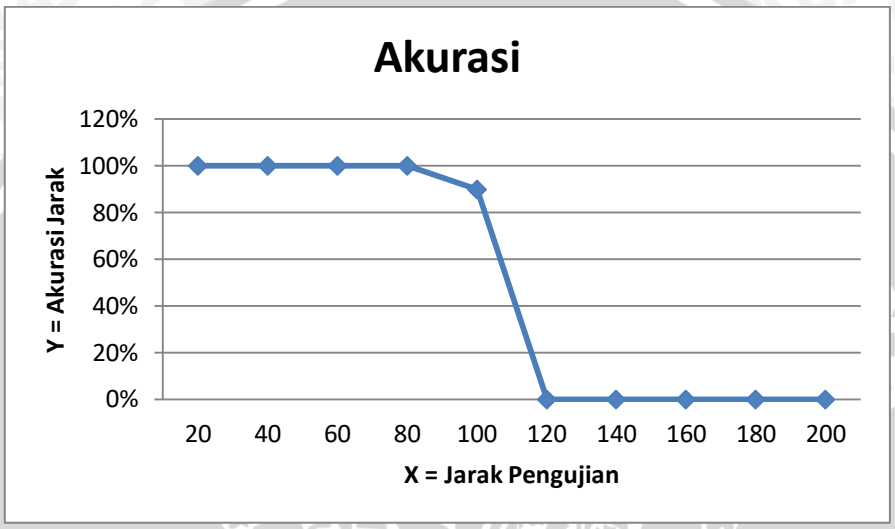
Tabel 6.1 Hasil Pengujian Sensor *Ultrasonic* HC-SR04

No	Jarak (cm)	Keakuratan Data (%)	Keterangan
1	20	100	Akurat
2	40	100	Akurat
3	60	100	Akurat
4	80	100	Akurat
5	100	90	Kurang Akurat
6	120	0	Tidak Akurat
7	140	0	Tidak Akurat
8	160	0	Tidak Akurat
9	180	0	Tidak Akurat
10	200	0	Tidak Akurat

Tabel 6.1 menunjukkan hasil dari pengujian sensor *ultrasonic* HC-SR04 pada RMU. Tiap jarak pengujian didapatkan 50 data yang kemudian dirata-rata berapa data yang sesuai dengan jarak sebenarnya.

6.1.1.4 Analisis

Dari pengujian sensor *ultrasonic* HC-SR04 didapatkan rata-rata keakuratan data yang dapat dibaca oleh sensor. Dari hasil ini dapat ditarik kesimpulan berapa jarak aman yang dapat digunakan sensor *ultrasonic* HC-SR04 untuk membaca jarak.



Gambar 6.2 Hasail Pengujian Sensor *Ultrasonic* HC-SR04

Pada pengujian sensor *ultrasonic* dapat dilihat pada gambar 6.2 sensor dapat membaca adanya objek secara akurat pada jarak maksimal 80 cm. Sedangkan pada jarak 100 cm jarak yang didapatkan oleh sensor kurang akurat. Pada jarak 120 cm – 200 cm jarak yang didapatkan oleh sensor tidak akurat. Dari hasil tersebut bisa disimpulkan bahwa sensor *ultrasonic* HC-SR04 dapat membaca objek secara akurat pada rentan jarak 20 cm – 80 cm.

6.1.2 Pengujian Sensor LDR

Sensor LDR digunakan untuk melakukan deteksi pelanggaran pada rambu dilarang berhenti. Sensor akan memberikan keluaran 1 apabila di atas sensor terdapat objek yang menghasilkan bayang-bayang. Sedangakn sensor akan memberikan keluaran 0 apabila tidak ada objek di atas sensor atau keadaan terang.

6.1.2.1 Tujuan

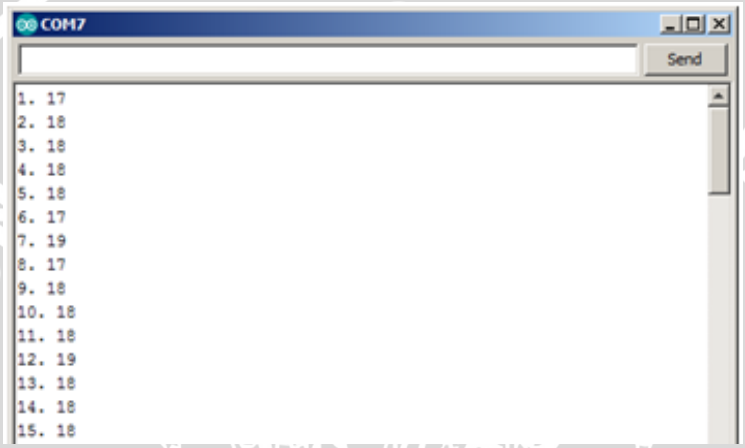
Pengujian sensor LDR dilakukan untuk mengetahui apakah sensor dapat mendeteksi adanya sebuah objek di atasnya atau tidak. Selain itu juga untuk mengetahui berapa nilai analog yang dihasilkan sensor apabila dalam keadaan ada penghalang dan tidak ada penghalang.

6.1.2.2 Skenario

Pengujian dilakukan dengan menggunakan sensor LDR yang dihubungkan pada Arduino Nano. Kemudian di atas sensor diletakkan sebuah objek yang menghasilkan bayangan. Dari bayangan tersebut sensor akan membacanya dan merubahnya menjadi nilai analog. Nilai analog yang didapatkan akan ditampilkan pada *Serial Monitor.*

6.1.2.3 Hasil

Pada hasil pengujian sensor LDR akan ditampilkan rentan nilai analog pembacaan sensor pada tiap-tiap kondisi. Hasil didapatkan dengan menguji sensor sebanyak 50 kali pada tiap-tiap kondisi.



Gambar 6.3 Contoh Pengujian Sensor LDR

Gambar 6.3 merupakan salah satu contoh hasil pembacaan sensor LDR pada *Serial Monitor*. Pada gambar dapat dilihat nilai analog yang didapatkan oleh sensor adalah 17 –19 yang menunjukkan bahwa tidak ada benda yang menghalangi sensor. Nilai rata-rata nilai analog didapatkan dengan rumus sebagai berikut :

$$Rata - Rata\ Nilai\ Analog = \frac{\sum nilai\ analog}{\sum data}$$

Berikut contoh perhitungan saat tidak ada penghalang dengan jumlah nilai analog yang didapatkan 905 :

$$Rata - Rata\ Nilai\ Analog = \frac{905}{50} = 18.1$$

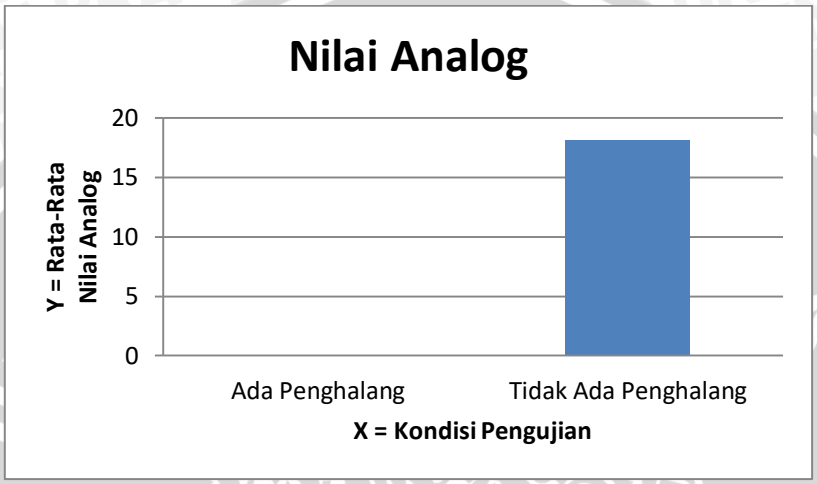
Tabel 6.2 Hasil Pengujian Sensor LDR

No	Keadaan	Rata-Rata Nilai Analog	Objek
1	Ada Penghalang	0	Ada
2	Tidak Ada Penghalang	18.1	Tidak Ada

Tabel 6.2 menunjukkan hasil dari pengujian sensor LDR pada RMU. Hasil pengujian didapatkan dengan melihat hasil pembacaan sensor dari tiap-tiap keadaan kemudian diambil nilai rata-ratanya.

6.1.2.4 Analisis

Dari pengujian sensor LDR didapatkan nilai analog yang dihasilkan oleh sensor ketika ada penghalang dan tidak ada penghalang. Dari hasil ini dapat ditarik kesimpulan berapa nilai analog saat ada objek dan tidak ada objek.



Gambar 6.4 Hasil Pengujian Sensor LDR

Pada pengujian sensor LDR dapat dilihat pada gambar 6.4 sensor akan mendeteksi adanya sebuah objek pada saat nilai analog 0. Sedangkan pada nilai analog 18.1 sensor tidak mendeteksi adanya objek. Dari hasil tersebut bisa disimpulkan bahwa sensor LDR dapat mendeteksi adanya kendaraan dijalan selama nilai analog yang dihasilkan oleh sensor adalah 0.

6.1.3 Pengujian Lampu LED

Lampu LED digunakan untuk menampilkan warna merah, kuning, dan hijau pada RMU 2. Lampu LED menyala merah pada waktu 0 – 8 detik, LED menyala hijau pada waktu 9 – 16 detik, dan LED menyala kuning pada waktu 17 – 20 detik.

6.1.3.1 Tujuan

Pengujian lampu LED dilakukan untuk mengetahui apakah nyala lampu LED sudah sesuai dengan rentan waktu yang sudah ditentukan pada *source code.*

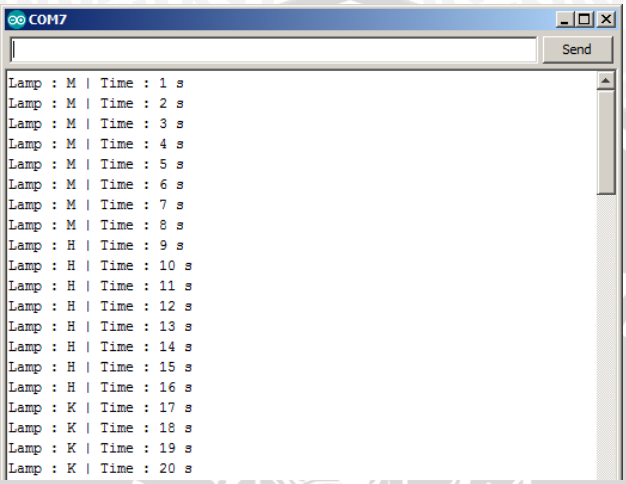
6.1.3.2 Skenario

Pengujian dilakukan dengan menghubungkan tiga buah lampu LED berwarna merah, kuning, dan hijau pada Arduino Nano. Kemudian dilihat nyala lampu LED apakah sudah sesuai dengan waktu yang sudah ditentukan. Selain melihat langsung nyala lampu LED pada *Serial Monitor* akan ditampilkan huruf M, K, dan H yang masing-masing memiliki arti Merah, Kuning dan Hijau. Keterangan waktu

juga akan ditampilkan pada *Serial Monitor* dengan memanfaatkan fungsi *millis()* dan *delay()*.

6.1.3.3 Hasil

Pada hasil pengujian lampu LED akan ditampilkan waktu, keterangan nyala lampu, dan gambar lampu LED.



Gambar 6.5 Pengujian Lampu LED

Gambar 6.5 merupakan salah hasil keluaran pada *Serial Monitor* berupa keterangan nyala lampu dan waktu. Pada gambar dapat dilihat waktu berada pada detik ke 1 - 8 keterangan nyala lampu adalah M, pada detik 9 – 16 keterangan nyala lampu adalah H, dan pada detik 17 – 20 keterangan nyala lampu adalah K.

Tabel 6.3 Hasil Pengujian Lampu LED

No.	Waktu (detik)	Keterangan	Gambar
1	0 – 8	M	
2	9 – 16	H	
3	17 – 20	K	

Tabel 6.3 menunjukkan hasil dari pengujian lampu LED pada RMU. Hasil pengujian didapatkan dengan melihat gambar apakah nyala lampu LED sudah sesuai dengan keterangan yang ada pada *Serial Monitor.*

6.1.3.4 Analisis

Pada pengujian lampu LED dapat dilihat pada gambar 6.3 dan tabel 6.5 lampu LED dapat menyala sesuai dengan rentan waktunya masing-masing. Pada rentan waktu 0 – 8 detik lampu LED menyala merah, rentan waktu 9 – 16 lampu LED menyala hijau, dan rentan waktu 17 – 20 lampu LED menyala kuning.

6.2 Pengujian Performa Pengiriman Data Menggunakan WSN

Pengujian performa pengiriman data menggunakan WSN dilakukan untuk menguji pengiriman data antar node pada sistem. Node yang akan diuji antara lain : RMU 1, RMU 2, RSU dan OPS.

6.2.1 Pengujian Pengiriman *Singlehop*

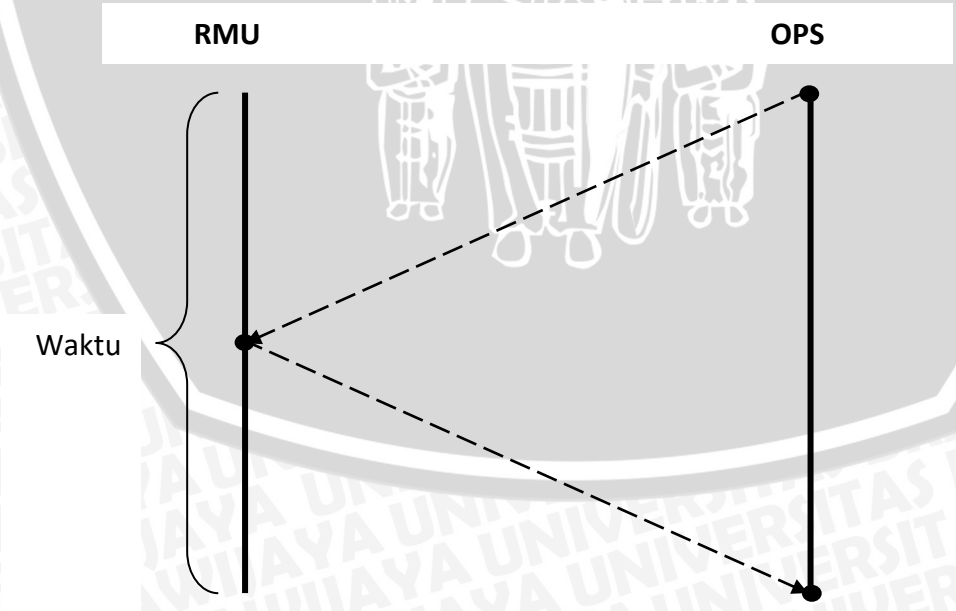
Pengujian pengiriman data yang pertama dilakukan secara *singlehop*. Pada pengujian ini melibatkan RMU 1, RMU 2 dan OPS. OPS bertugas sebagai peminta data dan RMU bertugas sebagai penyedia data.

6.2.1.1 Tujuan

Pengujian secara *singlehop* dilakukan untuk mengetahui beberapa parameter, yaitu : *Latency*, *Packet loss*, *Throughput* dan *Jitter*. Hal ini dilakukan untuk mengetahui performa komunikasi data pada nRF24L01.

6.2.1.2 Skenario

Pengujian *singlehop* dilakukan dengan melihat *Serial Monitor* pada tiap-tiap node. Pengujian dilakukan dengan jarak antar node 10 meter. Pada *Serial Monitor* ditampilkan waktu pengiriman datanya.



Gambar 6.6 Skenario Pengujian *Singlehop*

Gambar 6.6 merupakan arah pengiriman data pada skenario pengujian *singlehop.* OPS akan mengirimkan *request* pada RMU , kemudian RMU akan

mengirimkan data ke OPS. Waktu yang diukur adalah total waktu saat OPS meminta data sampai OPS menerima data dari RMU.

6.2.1.3 Hasil

Pada hasil pengujian pengiriman data *singlehop* akan dihitung *Latency*, *Packet loss*, *Throughput* dan *Jitter*. Pengujian dilakukan dengan melakukan pengiriman data sebanyak 50 kali. Pengujian menggunakan persamaan 2.1 untuk menghitung *Latency*, persamaan 2.2 untuk menghitung *Packet loss*, persamaan 2.3 untuk menghitung *Throughput*, dan persamaan 2.4 untuk menghitung *Jitter*. Berikut contoh perhitungan yang dilakukan pada pengujian ke-1 dimana total waktu pengirimannya adalah 50122 ms dan total waktu perbedaan pengiriman tiap kali pengiriman adalah 66 ms :

$$Latency = \frac{50122\ ms}{50} = 1002.44\ ms$$

$$Packet\ Loss = \frac{50 - 50}{50} \times 100\ \% = 0\ \%$$

$$Throughput = \frac{50\ data}{50122\ ms} = 1\ dps$$

$$Jitter = \frac{66\ ms}{50} = 1.32\ \text{ms}$$

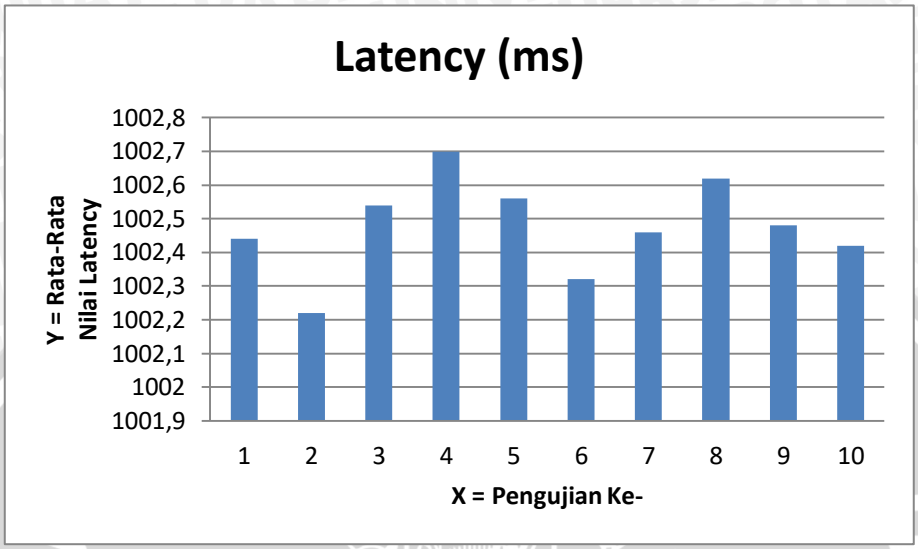
Berikut tabel 6.4 mengenai hasil pengujian QoS pada pengiriman data secara *singlehop* :

Tabel 6.4 Hasil Pengujian *Singlehop*

Pengujian Ke-	*Latency* (ms)	*Packet loss* (%)	*Throughput* (dps)	*Jitter* (ms)
1	1002.44	0	1.00	1.32
2	1002.22	0	1.00	1.14
3	1002.54	0	1.00	1.34
4	1002.70	0	1.00	0.94
5	1002.56	0	1.00	1.34
6	1002.32	0	1.00	0.96
7	1002.46	0	1.00	1.32
8	1002.62	0	1.00	1.30
9	1002.48	0	1.00	1.36
10	1002.42	0	1.00	1.02

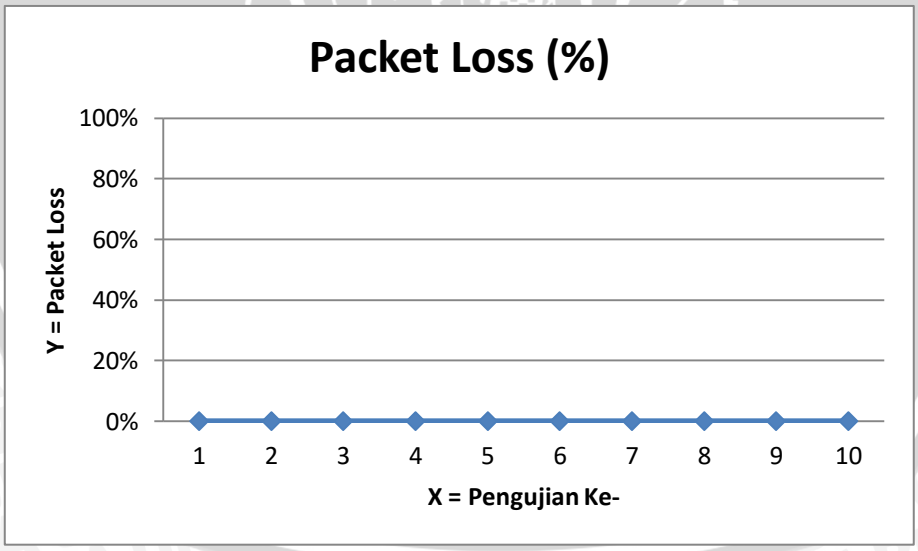
6.2.1.4 Analisis

Dari pengujian pengiriman data secara *singlehop* didapatkan *Latency*, *Packet loss*, *Throughput* dan *Jitter* saat OPS meminta data hingga OPS menerima data dari kedua RMU.



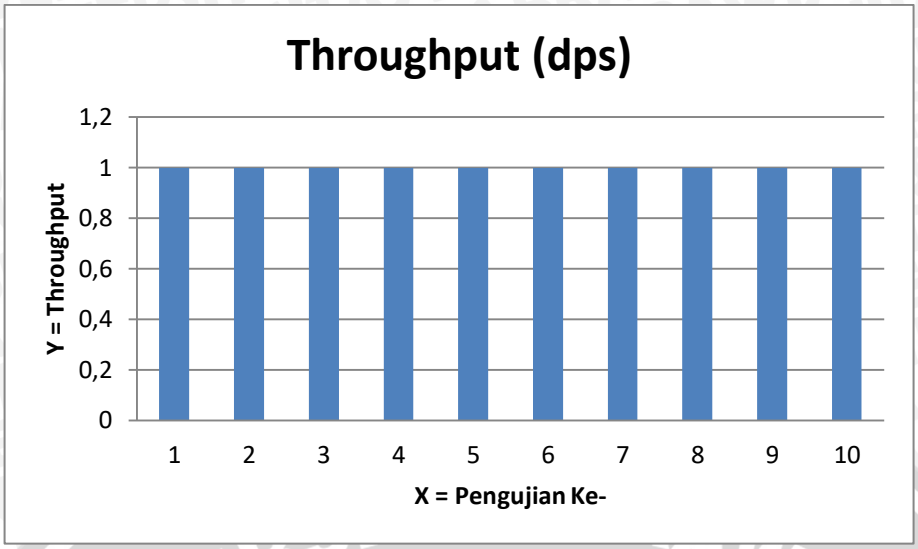
Gambar 6.7 *Latency Singlehop*

Gambar 6.7 menunjukkan nilai *Latency* dari pengiriman data *singlehop*. Nilai *Latency* dipengaruhi oleh waktu pengiriman data. Nilai yang didapatkan dari hasil pengujian adalah 1002.22 – 1002.70 ms.



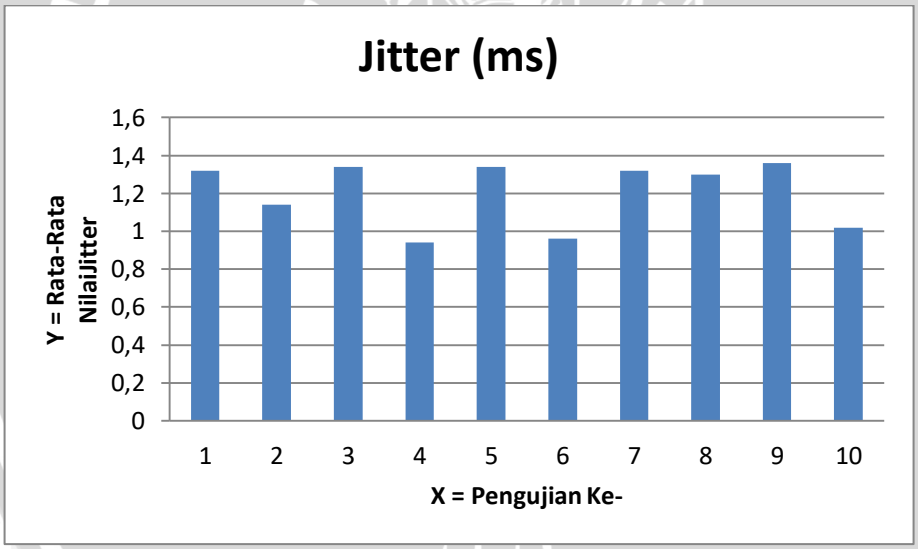
Gambar 6.8 *Latency Singlehop*

Gambar 6.8 menunjukkan nilai *Packet loss* dari pengiriman data sebanyak 50 data. Dari 10 kali pengujian tidak didapati adanya *Packet loss* dikarenakan jarak pengujian pengiriman data masih dalam rentan jarak aman pengiriman nRF24L01.



Gambar 6.9 *Throughput Singlehop*

Gambar 6.9 menunjukkan nilai *Throughput* dari pengiriman data secara *singlehop.* Dari pengujian ini didapati nilai *Throughput* adalah 1. Sehingga setiap detiknya sistem dapat mengirimkan seluruh data yang dikirimkan.



Gambar 6.10 *Jitter Singlehop*

Gambar 6.10 menunjukkan nilai *Jitter* dari pengiriman data *singlehop*. Nilai *Jitter* mempengatuhi lama waktu pengiriman data pada sistem. Nilai yang didapatkan dari hasil pengujian adalah 0.94 – 1.36 ms.

6.2.2 Pengujian Pengiriman *Multihop*

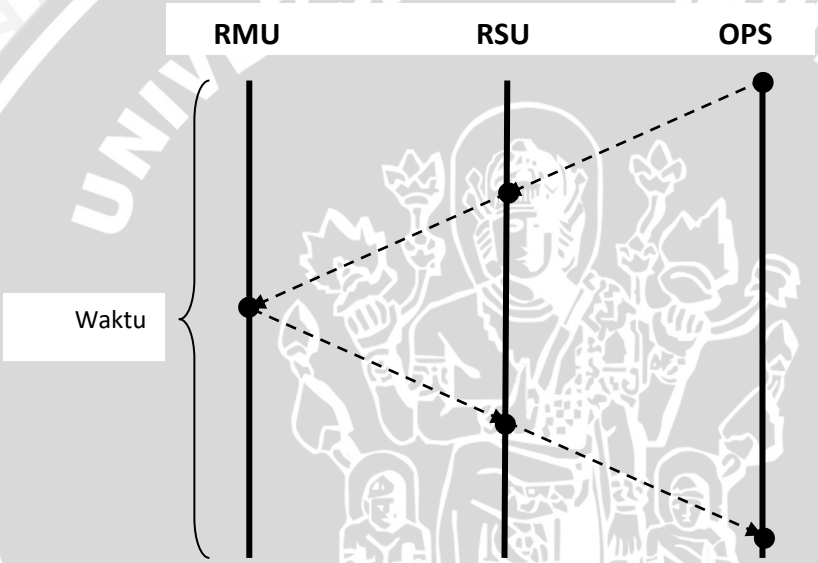
Pengujian pengiriman data yang kedua dilakukan secara *multihop*. Pada pengujian ini melibatkan RMU 1, RMU 2, RSU dan OPS. OPS bertugas sebagai peminta data, RSU sebagai jembatan antara RMU dan OPS, dan RMU sebagai pendeteksi pelanggaran.

6.2.2.1 Tujuan

Pengujian secara *multihop* dilakukan untuk mengetahui beberapa parameter seperti : *Latency*, *Packet loss*, *Throughput* dan *Jitter*. Hal ini dilakukan untuk mengetahui performa komunikasi data pada nRF24L01.

6.2.2.2 Skenario

Pengujian *multihop* dilakukan dengan melihat *Serial Monitor* pada tiap-tiap node. Pengujian dilakukan dengan jarak antar node 10 meter. Pengujian dilakukan denPada *Serial Monitor* ditampilkan data dari sensor serta waktu pengiriman datanya. Pengujian dilakukan dengan mengirimkan *request* dari OPS ke RSU dan diteruskan ke RMU. Kemudian data pelanggaran dari RMU akan dikirimkan ke RSU. Data pelanggaran yang diterima oleh RSU kemudian akan diteruskan OPS.



Gambar 6.11 Skenario Pengujian *Multihop*

Gambar 6.11 merupakan arah pengiriman data pada skenario pengujian *multihop.* Waktu yang diukur adalah waktu dari OPS mengirim *request* sampai OPS menerima data.

6.2.2.3 Hasil

Pada hasil pengujian pengiriman data *multihop* akan dihitung *Latency*, *Packet loss*, *Throughput* dan *Jitter*. Pengujian dilakukan dengan melakukan pengiriman data sebanyak 50 kali. Berikut contoh perhitungan yang dilakukan pada pengujian ke-1 dimana total waktu pengirimannya adalah 50360 ms dan total waktu perbedaan pengiriman tiap kali pengiriman adalah 70 ms :

$$Latency = \frac{50360\ ms}{50} = 1007.20\ ms$$

$$Packet\ Loss = \frac{50 - 50}{50} \times 100\ \% = 0\ \%$$

$$Throughput = \frac{50\ data}{50360\ ms} = 0.99\ dps$$

$$Jitter = \frac{70\ ms}{50} = 1.40\ \text{ms}$$

Berikut tabel 6.5 mengenai hasil pengujian QoS pada pengiriman data secara *singlehop* :

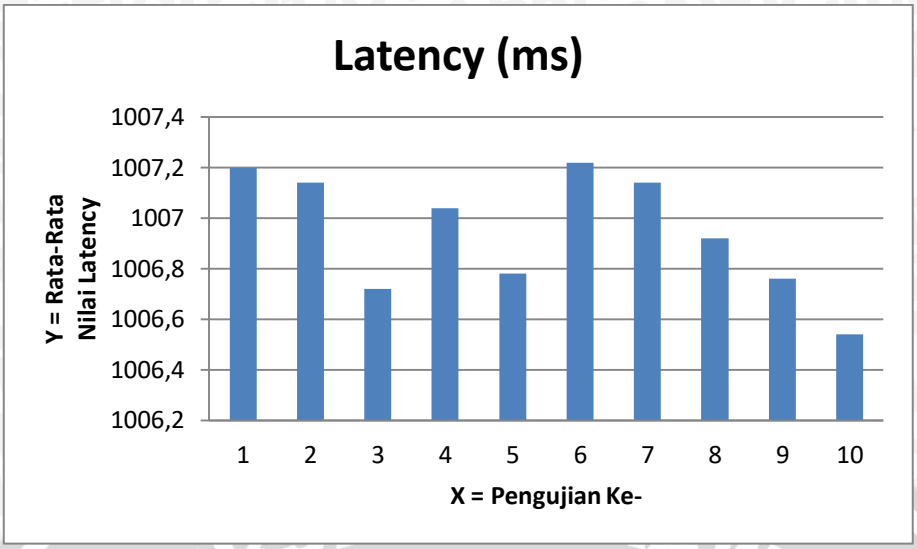
Tabel 6.5 Hasil Pengujian Multihop

Pengujian Ke-	*Latency* (ms)	*Packet loss* (%)	*Throughput* (dps)	*Jitter* (ms)
1	1007.20	0	0.99	1.40
2	1007.14	0	0.99	1.66
3	1006.72	0	0.99	1.28
4	1007.04	0	0.99	1.96
5	1006.78	0	0.99	1.62
6	1007.22	0	0.99	1.62
7	1007.14	0	0.99	1.78
8	1006.92	0	0.99	1.78
9	1006.76	0	0.99	1.70
10	1006.54	0	0.99	1.34

Tabel 6.5 merupakan hasil pengujian pengiriman data secara *multihop*. Hasil didapatkan dengan menghitung *Latency*, *Packet loss*, *Throughput* dan *Jitter*.

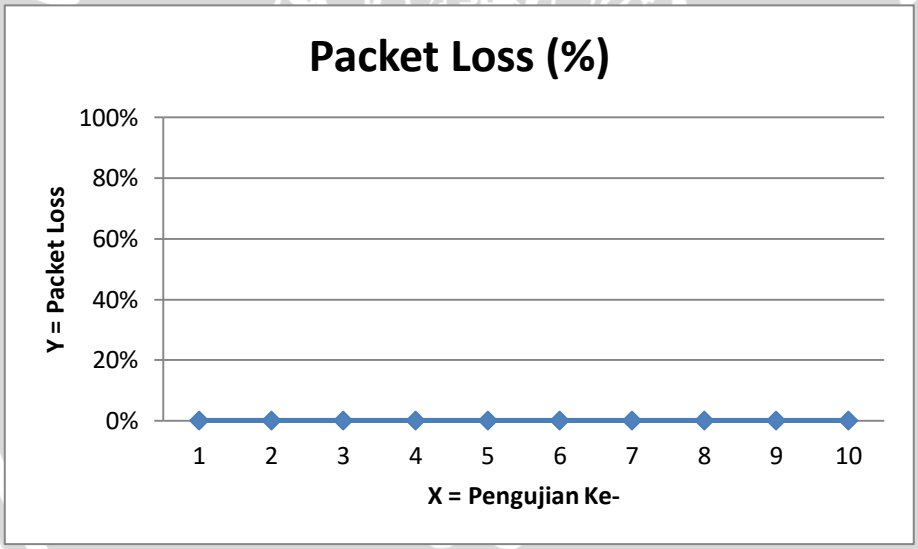
6.2.2.4 Analisis

Dari pengujian pengiriman data secara *multihop* didapatkan *Latency*, *Packet loss*, *Throughput* dan *Jitter* saat RSU meminta data hingga OPS menerima data dari kedua RMU.



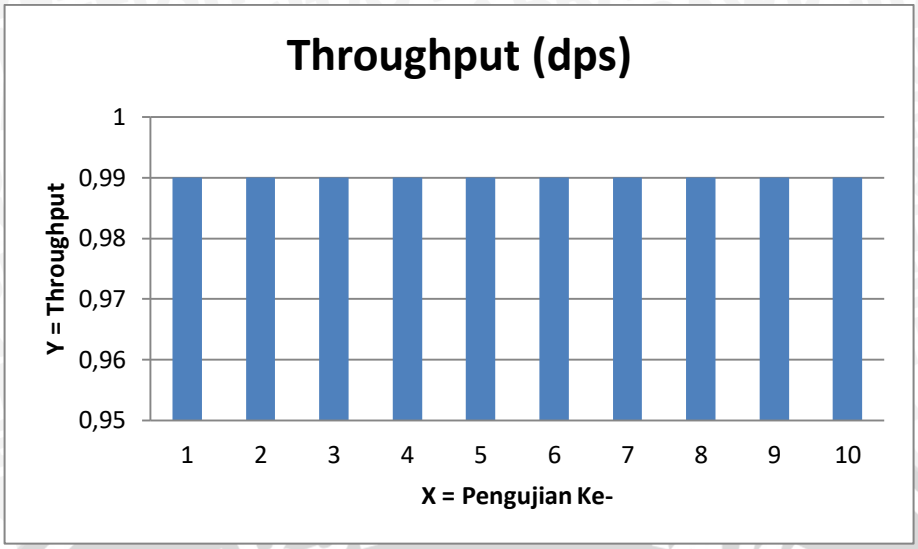
Gambar 6.12 *Latency Multihop*

Gambar 6.12 menunjukkan nilai *Latency* dari pengiriman data *multihop*. Nilai *Latency* mempengatuhi lama waktu pengiriman data pada sistem. Nilai yang didapatkan dari hasil pengujian adalah 1006.54 – 1007.22 ms.



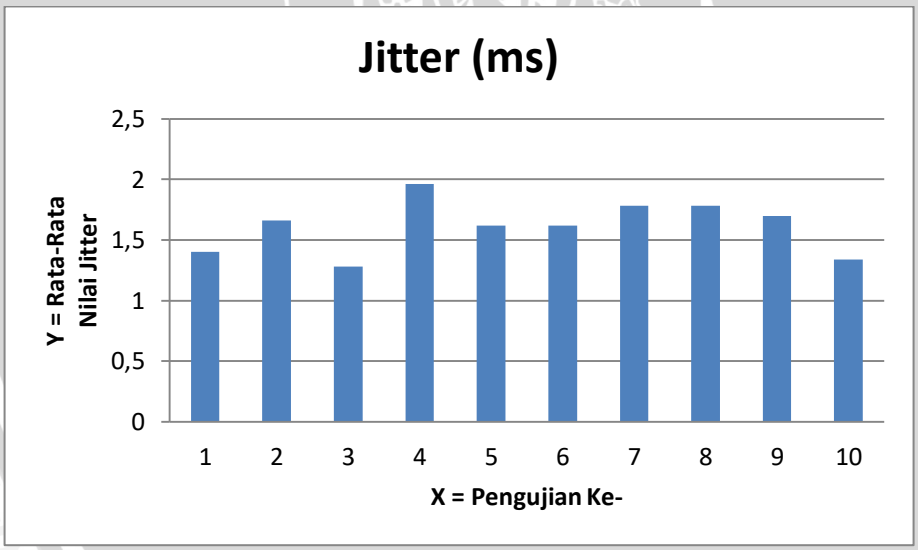
Gambar 6.13 *Packet loss Multihop*

Gambar 6.13 menunjukkan nilai *Packet loss* dari pengiriman data sebanyak 50 data. Dari 10 kali pengujian tidak didapati adanya *Packet loss* dikarenakan jarak pengujian pengiriman data masih dalam rentan jarak aman pengiriman nRF24L01.



Gambar 6.14 *Throughput Multihop*

Gambar 6.14 menunjukkan nilai *Throughput*dari pengiriman data secara *multihop.* Dari pengujian ini didapati nilai *Throughput* adalah 0,99 yang apabila dibulatkan adalah 1. Sehingga setiap detiknya sistem dapat mengirimkan seluruh data yang dikirimkan.



Gambar 6.15 *Jitter Multihop*

Gambar 6.15 menunjukkan nilai *Jitter* dari pengiriman data *multihop*. Nilai *Jitter* mempengatuhi lama waktu pengiriman data pada sistem. Nilai yang didapatkan dari hasil pengujian adalah 1.28 – 1.96 ms.

6.3 Pengujian Algoritma Sistem Secara Keseluruhan

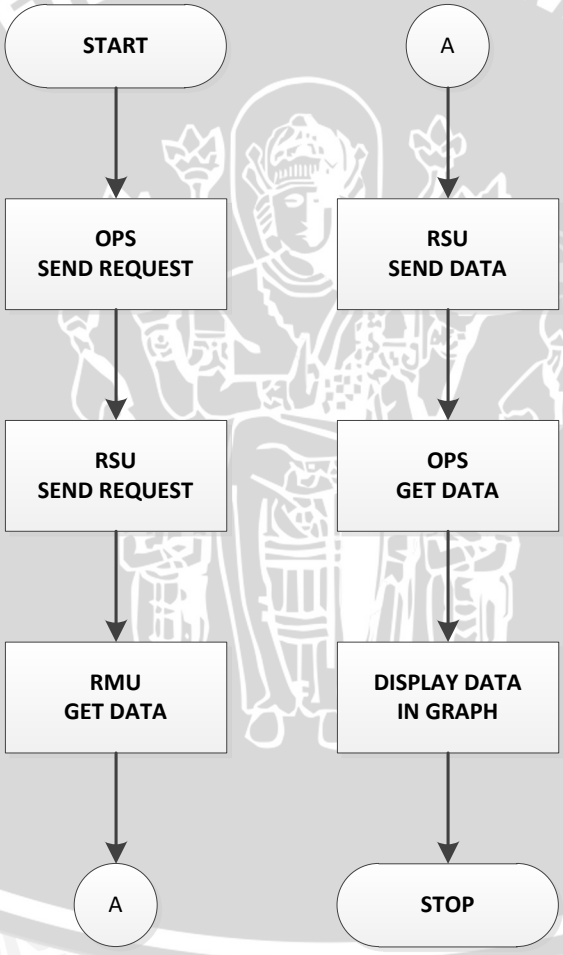
Pengujian sistem secara keseluruhan adalah pengujian yang dilakukan untuk menguji kemampuan sistem saat dijalankan. Langkah-langkah pengujian dilakukan sebagaimana pada kenyataan sistem diuji.

6.3.1 Tujuan

Pengujian dilakukan untuk mengetahui kemampuan sistem secara keseluruhan, mulai dari pembacaan sensor, alur komunikasi data antar node, dan menampilkan data pada *user interface.* Dalam proses pengujian yang diuji adalah akurasi data serta waktu pengiriman datanya.

6.3.2 Skenario

Pengujian dilakukan di Jalan Simpang Raya Candi 6 Malang dengan jarak antar node sepanjang 10 m. Sensor *ultrasonic* HC-SRF04 diletakkan setinggi 50 cm di atas tanah, sedangkan sensor LDR diletakkan di permukaan tanah. Pengujian sistem secara keseluruhan dilakukan sesuai dengan langkah-langkah penggunaan sistem sebagai berikut :



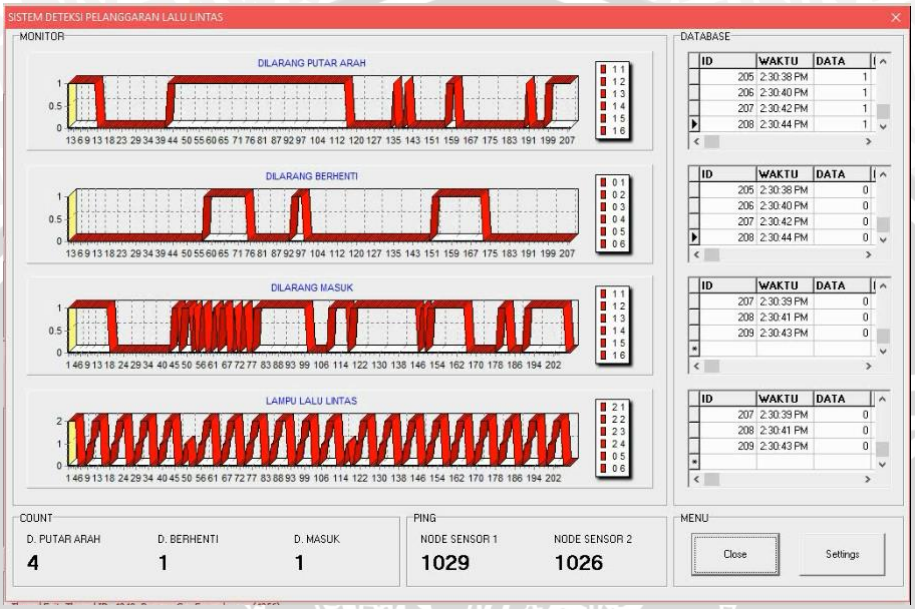
Gambar 6.16 Skenario Pengujian Keseluruhan Sistem

Gambar 6.16 menunjukkan skenario pengujian keseluruhan sistem. Dimulai dari RSU meminta data pada RMU, RMU membaca data, RMU mengirimkan data

ke RSU, RSU meneruskan data ke OPS, *user interface* menampilkan data dalam bentuk grafik.

6.3.3 Hasil

Pada hasil pengujian keseluruhan sistem ditampilkan *Latency*, *Packet loss*, *Throughput* dan *Jitter*. Pengujian dilakukan dengan melakukan pengiriman data sebanyak 50 kali. Berikut contoh *screenshot* pada grafik :



Gambar 6.17 Contoh Saat *User interface* Berjalan

Gambar 6.17 merupakan tampilan saat *user interface* berjalan. Data yang didapatkan oleh OPS akan ditampilkan pada *user interface.* Sumbu x menandakan ID data dan sumbu y menandakan ada tidaknya pelanggaran. Berikut contoh perhitungan yang dilakukan pada pengujian ke-1 dimana total waktu pengirimannya adalah 50362 ms dan total waktu perbedaan pengiriman tiap kali pengiriman adalah 80 ms :

$$Latency = \frac{50362\ ms}{50} = 1007.24\ ms$$

$$Packet\ Loss = \frac{50 - 50}{50} \times 100\ \% = 0\ \%$$

$$Throughput = \frac{50\ data}{50362\ ms} = 0.99\ dps$$

$$Jitter = \frac{80\ ms}{50} = 1.60\ \text{ms}$$

Berikut tabel 6.6 mengenai hasil pengujian QoS pada algoritma keseluruhan sistem :

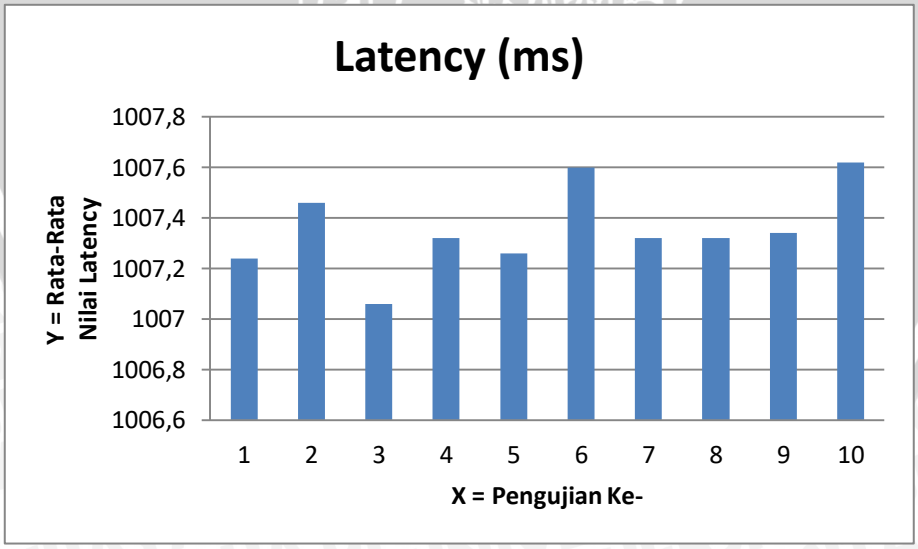
Tabel 6.6 Hasil Pengujian Keseluruhan Sistem

Pengujian Ke-	*Latency* (ms)	*Packet loss* (%)	*Throughput* (dps)	*Jitter* (ms)
1	1007.24	0	0.99	1.60
2	1007.46	0	0.99	1.48
3	1007.06	0	0.99	1.64
4	1007.32	0	0.99	1.56
5	1007.26	0	0.99	1.08
6	1007.60	0	0.99	1.16
7	1007.32	0	0.99	1.42
8	1007.32	0	0.99	1.44
9	1007.34	0	0.99	1.14
10	1007.62	0	0.99	1.16

Tabel 6.6 merupakan hasil pengujian sistem secara keseluruhan. Hasil didapatkan dengan menghitung *Latency*, *Packet loss*, *Throughput* dan *Jitter*.

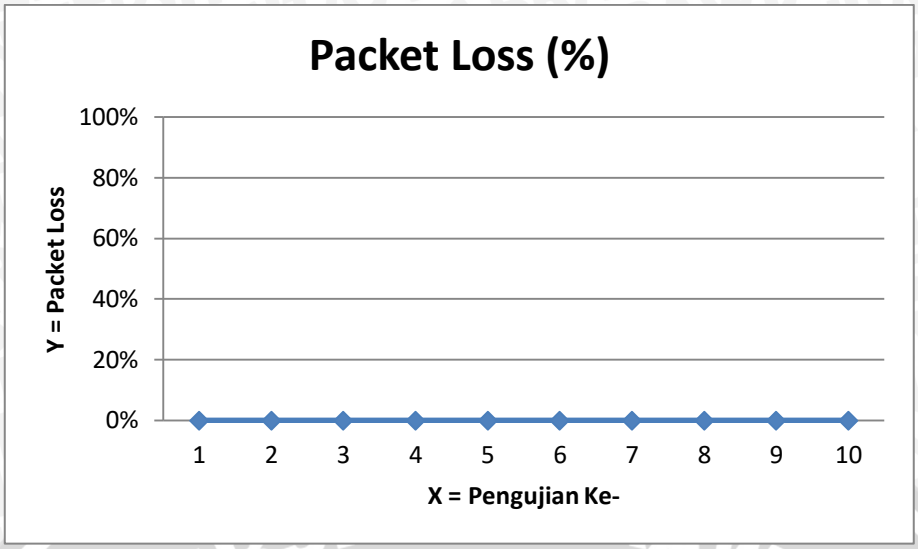
6.3.4 Analisis

Dari pengujian sistem secara keseluruhan didapatkan *Latency*, *Packet loss*, *Throughput* dan *Jitter* saat OPS meminta data hingga komputer server menampilkan data pada *user interface.*



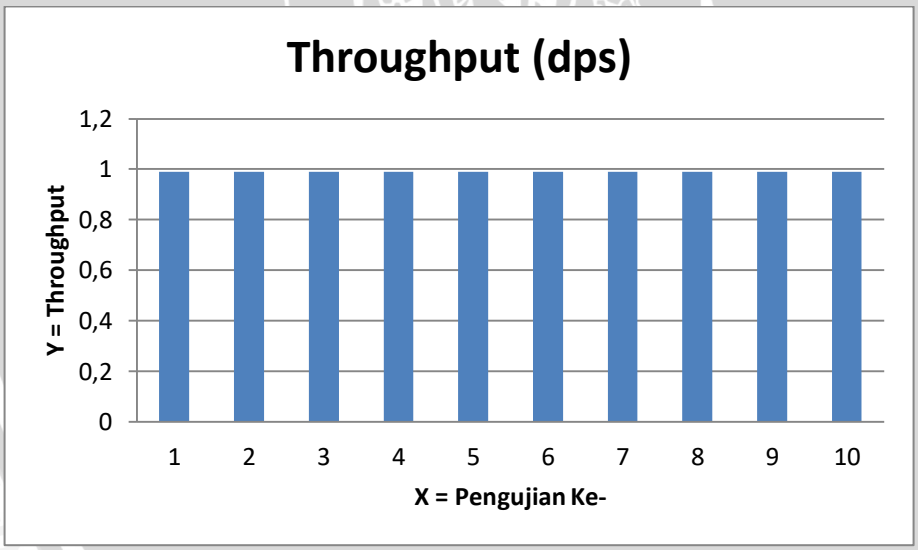
Gambar 6.18 *Latency* Keseluruhan Sistem

Gambar 6.18 menunjukkan nilai *Latency* dari pengiriman data *multihop*. Nilai *Latency* mempengatuhi lama waktu pengiriman data pada sistem. Nilai yang didapatkan dari hasil pengujian adalah 1007.06 – 1007.62 ms.



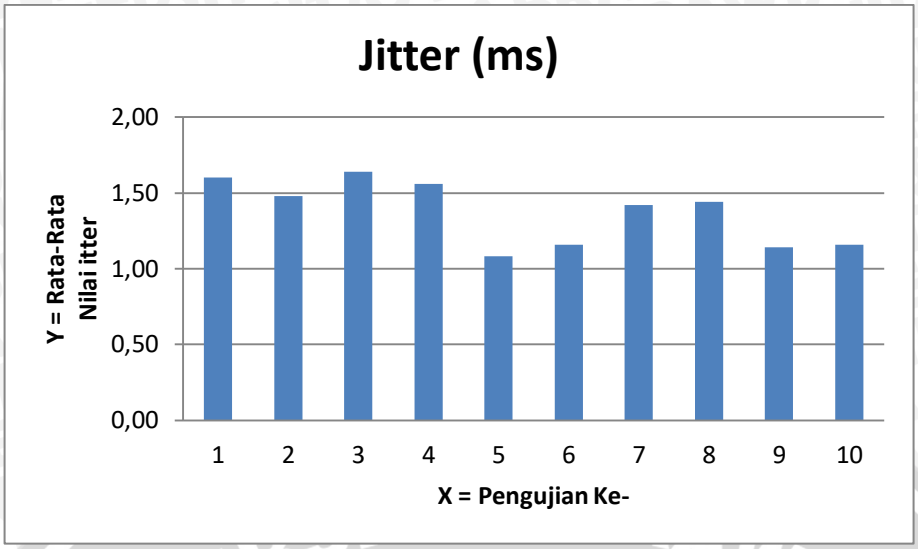
Gambar 6.19 *Packet loss* Keseluruhan Sistem

Gambar 6.19 menunjukkan nilai *Packet loss* dari pengiriman data sebanyak 50 data. Dari 10 kali pengujian tidak didapati adanya *Packet loss* dikarenakan jarak pengujian pengiriman data masih dalam rentan jarak aman pengiriman nRF24L01.



Gambar 6.20 *Throughput* Keseluruhan Sistem

Gambar 6.20 menunjukkan nilai *Throughput* dari pengiriman data secara *multihop.* Dari pengujian ini didapati nilai *Throughput* adalah 0,99 yang apabila dibulatkan adalah 1. Sehingga setiap detiknya sistem dapat mengirimkan seluruh data yang dikirimkan.



Gambar 6.21 *Jitter* Keseluruhan Sistem

Gambar 6.21 menunjukkan nilai *Jitter* dari pengiriman data *multihop*. Nilai *Jitter* mempengaruhi lama waktu pengiriman data pada sistem. Nilai yang didapatkan dari hasil pengujian adalah 1.08 – 1.64 ms.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan rumusan masalah yang sudah dibuat maka dapat dituliskan kesimpulan sebagai berikut :

1. Perancangan sistem deteksi pelanggaran rambu lalu lintas menggunakan Arduino Nano sebagai pemroses data, nRF24L01 sebagai pengirim data dan penerima data secara *wireless*, sensor *ultrasonic* HC-SR04 dan LDR untuk mendapatkan data, dan lampu LED sebagai pengganti lampu lalu lintas.
2. Pembacaan data oleh sensor *ultrasonic* HC-SR04 dengan akurasi 100% hanya pada rentan jarak 3 – 80 cm, sedangkan pada jarak 100 cm akurasi yang didapatkan menurun menjadi 90%. Ketika jarak melebihi 100 cm data yang didapatkan menghasilkan akurasi 0%. Pembacaan data oleh sensor *Light Dependant Resistor* menghasilkan rata-rata nilai analog 0 apabila terdapat penghalang. Ketika keadaan normal tanpa penghalang sensor akan menghasilkan rata-rata nilai analog sebesar 18,1.
3. Metode transfer data dibagi menjadi dua yakni *singlehop* dan *multihop.* Pengiriman data *singlehop* menghasilkan nilai *Latency* 1002.22 – 1002.70 ms, *Packet loss* 0 %, *Throughput* 1.00 dps, dan *Jitter* 0.94 – 1.36 ms. Sedangkan pada *multihop* menghasilkan nilai *Latency* 1006.54 – 1007.22 ms, *Packet loss* 0%, *Throughput* 0.99 dps, dan *Jitter* 1.28 – 1.96. Proses pengujian dipengaruhi oleh waktu pengiriman data antara OPS dengan RMU.
4. Pengiriman data menggunakan nRF24L01 dirasa sangat efektif selama penggunaannya dalam jarak sesuai dengan spesifikasinya dan tidak terdapat banyak halangan. Hal ini ditunjukkan dengan tidak adanya data pelanggaran yang hilang saat proses pengiriman data. Dari pengujian keseluruhan sistem didapatkan nilai *Latency* 1007.06 – 1007.62 ms, *Packet loss* 0 %, *Throughput* 0.99 dps, dan *Jitter* 1.08 – 1.64 ms.

7.2 Saran

Beberapa saran bagi peneliti yang ingin mengembangkan penelitian ini, antara lain :

1. Proses pembacaan data pelanggaran yang hanya menggunakan sensor bisa dikembangkan lebih lanjut agar proses deteksi pelanggaran bisa lebih akurat.
2. Perangkat komunikasi nRF24L01 bisa ditambahkan antena agar sinyal pengiriman menjadi lebih kuat. Karena proses pengiriman sangat dipengaruhi oleh jarak dan banyak tidaknya penghalang.

DAFTAR PUSTAKA

Affandi, F, Rambu Lalu Lintas Jalan Di Indonesia. [Online] Tersedia di : < http://xa.yimg.com/kq/groups/17366665/1193281615/name/Gambar+Rambu+Lalu+Lintas.pdf> [Diakses 11 Februari 2016]

Alfian, et al., Alat Pendeteksi Pelanggaran Marka Lalu-Lintas dengan Indikasi Jumlah Pelanggar. Indonesia: PENS-ITS.

Amir, F., 2016. Perancangan Aplikasi Komputer Pusat untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara Wireless. Indonesia: Universitas Brawijaya.

Anonimous, Nordic Semiconductor, nRF24L01 Product Specification V2.0. [Online] Tersedia di : <http://www.nordicsemi.com/eng/nordic/download_resource/8041/1/62435711> [Diakses 1 Februari 2016].

Anonimous, Photo Resistor. [Online] Tersedia di : < http://www.resistorguide.com/photoresistor/> [Diaskses 11 Februari 2016]

Anonimous. Arduino And Shields. [Online] Tersedia di : <http://www.bombayelectronics.in/arduino-and-shields> [Diakses 9 Februari 2016]

Anonimous. Arduino Nano. [Online] Tersedia di : <https://www.arduino.cc/en/Main/ArduinoBoardNano> [Diakses 9 Februari 2016].

Delta-Electronic. Pengiriman Data Serial Tanpa Kabel Menggunakan Transceiver 2.4 Ghz Bagian 2.

Charles, 2013. nRF24L01 real life range test. [Online] Tersedia di : < https://hallard.me/nrf24l01-real-life-range-test/> [Diakses 1 Februari 2016]

Jin, Alvin, 2014. Use HC-SR04 *Ultrasonic* Sensor to Measure Distance on Arduino. [Online] Tersedia di : <http://learn.linksprite.com/arduino/advanced-learning-kit-for-arduino/use-hc-sr04-*ultrasonic*-sensor-to-measure-distance-on-arduino/> [Diakses 1 Februari 2016]

Mathew, Tom V., 2014. Transportation Systems Engineering. India: IIT Bombay.

Nellore, Kapileswar. et al., 2016. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. South Africa: University of Pretoria.

Nordic Semiconductor. 2007. nRF24L01 Single Chip 2.4GHz Transceiver.

Pighixx. 2013. Arduino NANO Pinout Diagram. [Online] Tersedia di : <http://forum.arduino.cc/index.php?topic=147582.0> [Diakses 9 Februari 2016]

Ramya, et al., 2012. Embedded System for Automatic Traffic Violation Monitoring and Alertin. USA: Foundation of Computer Science FCS

Riliskis, Laurynas. et al., 2014. Maestro: An Orchestration Framework for Large-Scale WSN Simulations. Sweden: Lulea University of Technology.

Santos, Rui, 2013. Complete Guide for *Ultrasonic* Sensor HC-SR04. [Online] Tersedia di : <http://randomnerdtutorials.com/complete-guide-for-*ultrasonic*-sensor-hc-sr04/> [Diakses 1 Februari 2016]

Sohraby, Kazem et al., 2007. Wireless Sensor Network Technology, Protocols, and Applications. The Wiley Bicentennial.