

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab perancangan sistem merupakan bagian yang menjelaskan setiap kegiatan yang berhubungan dengan sistem guna menciptakan sistem tersebut. Pada bagian ini menjelaskan tentang perancangan guna untuk implementasi pada sistem penentuan posisi node pada jaringan sensor berdasarkan *coverage area* sensor menggunakan algoritma genetika. **Gambar 5.1** merupakan diagram dari perancangan sistem.



Gambar 5.1 Perancangan Sistem

Alur perancangan sistem dimulai dari proses perancangan perangkat lunak. Proses perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan perangkat lunak pada matlab, perancangan perangkat lunak pada mikrokontroler Arduino, dan perancangna perangkat lunak web dan xampp. Setelah melakukan perancangan perangkat lunak, selanjutnya proses perancangan perangkat keras / *hardware*. Perancangan perangkat keras tersusun menjadi perancangan node *transmitter* dan perancangan node *receiver*. Selanjutnya tahap implementasi sistem

5.1 Perancangan

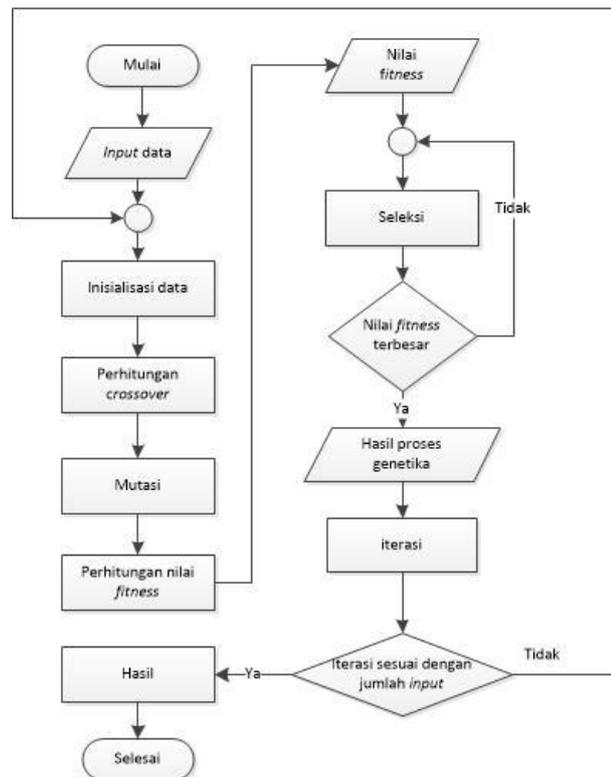
Dalam perancangan sistem ini terdapat beberapa tahap sehingga sistem dapat tersusun sesuai dengan tujuan. Tahapan-tahapan perancangan meliputi perancangan perangkat lunak sistem sebelumnya pada matlab, dan perancangan perangkat keras. Pada perancangan perangkat lunak dimulai dari perhitungan manual genetika yang merupakan salah satu tahap untuk menghitung nilai dari posisi node terbaik dengan menggunakan algoritma genetika. *Input* dari perhitungan ini adalah jumlah kromosom, jumlah node, ukuran jari-jari sensor, selanjutnya *output* berupa kromosom yang memiliki nilai *fitness* terbaik dan kumpulan titik pusat (X dan Y) yang merupakan tempat lokasi node. Pada

perancangan perangkat lunak sistem sebelumnya dilakukan implementasi dari perhitungan genetika yang telah disusun pada matlab. Selanjutnya perancangan perangkat keras yaitu perancangan yang melibatkan alat-alat seperti sensor, arduino, nrf24l01, dan esp8266 sesuai dengan aturan dan tahapan perancangan *hardware*.

5.1.1 Perancangan Perangkat Lunak

5.1.1.1 Perancangan Perangkat Lunak Pada Matlab

Perancangan perangkat lunak pada matlab merupakan sebuah perancangan sistem untuk mendeteksi posisi node terbaik menggunakan algoritma genetika pada matlab. Alur perancangan tersebut sesuai dengan Gambar 5.2.



Gambar 5.2 Diagram Alir Perancangan Perangkat Lunak Matlab

Pada gambar 5.2 diatas menunjukkan tahapan-tahapan proses dalam merancang perangkat lunak pada matlab. Tahapan-tahapan tersebut sesuai dengan proses perhitungan genetika dimulai dari *user* memasukkan data, selanjutnya inisialisasi data. Ketika inisialisasi data telah berhasil maka selanjutnya proses crossover. Didalam proses crossover, awal mula terjadi proses pemilihan induk, selanjutnya baru memulai perhitungan crossover dengan menggunakan crossover aritmatika. Ketika proses crossover selesai maka proses selanjutnya adalah proses mutase. Proses mutasi memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Selanjutnya adalah proses perhitungan nilai *fitness*. Awal dari nilai *fitness* yaitu menggabungkan seluruh parent dan offspring setelah itu mulai proses pencarian nilai *fitness*. Ketika telah mendapat

nilai fitness dari setiap kromosom, maka selanjutnya adalah proses seleksi. Proses seleksi merupakan proses mencari nilai terbaik, dalam system ini nilai terbaik adalah nilai terbesar. Apabila telah menemukan nilai fitness merupakan nilai terbesar maka lanjut ke proses selanjutnya. Sedangkan apabila belum merupakan nilai fitness terbesar maka kembali pada proses seleksi. Ketika telah mendapatkan nilai fitness terbesar maka kromosom yang memiliki nilai fitness terbesar tersebut merupakan hasil dari proses genetika. Selanjutnya system akan menjalankan tahap iterasi sebanyak jumlah yang dimasukkan oleh *user*. Berikut merupakan perhitungan manual dari proses genetika:

1. Perhitungan Manual Genetika

Implementasi dilakukan pada sebuah ruangan yang memiliki luas $6,5 \text{ m}^2$ dengan dimensi $3\text{m} \times 2,5\text{m}$. Pada ruangan ini dibutuhkan alat pendeteksi kebakaran salah satunya dengan mendeteksi api. Untuk mendeteksi api secara tepat maka semua titik ruangan harus terdeteksi oleh sensor. Pada implementasi ini menggunakan 2 jenis node, yaitu node A dan node B. Pada dasarnya sistem penentuan posisi node yang sudah ada sebelumnya memberikan 2 ukuran jari-jari sensor yang berbeda pada tiap node yang sudah ditentukan oleh sistem, akan tetapi pada implementasi sistem ini dibuat sensor api dengan jenis jari-jari yang sama. Kalaupun *user* ingin mengimplementasikan dengan sensor yang berbeda ukuran jari-jarinya, *user* dapat langsung memasukkan ukuran jari-jari yang diinginkan pada sistem, karena sistem ini telah *dimodifikasi* ukuran jari-jari yang dibuat dinamis.

Tahapan-tahapan perhitungan genetika:

a. Inisialisasi Parameter

Dimisalkan parameter-parameter untuk membentuk populasi awal pada algoritma genetika yaitu:

- Jumlah kromosom : 5
- Probabilitas *crossover* : 0.5
- Probabilitas mutasi : 0.2
- Ukuran Jari-jari sensor A : 1
- Ukuran jari-jari sensor B : 1
- Jumlah node A : 1
- Jumlah node B : 3

b. Membangkitkan Populasi Awal

Membangkitkan populasi awal dilakukan dengan proses pengolahan gen sehingga dapat terbentuk suatu kromosom. Kromosom merupakan sekumpulan dari beberapa gen. Pada perhitungan genetika ini yang menjadi titik solusi dari permasalahan adalah kumpulan dari titik-titik pusat dengan ketentuan titik X adalah $0 \leq X \leq 3$ dan titik Y adalah $0 \leq Y \leq 2,5$. Ketentuan

dari titik pusat tersebut diperoleh dari dimensi ruangan yang akan menjadi tempat untuk implementasi yang memiliki dimensi 3m x 2,5m. Kumpulan titik pusat ini merupakan kumpulan titik posisi dimana node akan diletakkan. Titik node ini merupakan solusi dari permasalahan. Satu populasi berisikan kumpulan kemungkinan titik posisi node yang menjadi solusi dari permasalahan.

Populasi awal dibangkitkan dengan menentukan beberapa pasangan titik pusat secara acak, yang setiap pasangannya memiliki nilai titik pusat yang berbeda. Titik pusat inilah yang merupakan kromosom dari permasalahan ini. Dalam membangkitkan populasi awal ini, jumlah kromosom sebanyak yang telah ditentukan. Misal kromosom berjumlah 5, node A berjumlah 1, dan node B berjumlah 3. **Tabel 5.1** merupakan tabel membangkitkan populasi awal.

Tabel 5.1 Populasi Awal

No. Kromosom	Node A		Node B		Node B		Node B	
	X	Y	X	Y	X	Y	X	Y
K1	3	2	2	1	1	2	3	1
K2	1	2	1	1	2	1	3	2
K3	3	1	3	2	2	1	1	2
K4	2	2	1	1	3	1	3	2
K5	1	2	3	2	2	2	2	1

Tabel 5.1 menunjukkan kromosom berjumlah 5. Satu kromosom memiliki 3 node A dan 1 node B. Pada permasalahan ini gen kumpulan titik pusat node yang berupa nilai X dan Y dan merupakan bilangan integer. Jumlah gen dapat ditentukan dengan mengalikan jumlah kromosom dengan jumlah node.

c. Perhitungan Crossover

Perhitungan *crossover* adalah proses memindah-silangkan dua kromosom. Pindah silang hanya bisa dilakukan jika bilangan acak (*random*) yang dibangkitkan kurang dari probabilitas *crossover* yang ditentukan, umumnya diset mendekati 1. Jumlah kromosom yang mengalami *crossover* dipengaruhi oleh parameter *crossover_rate* (pc). Probabilitas *crossover* (pc) ini yang nantinya akan digunakan untuk menentukan banyaknya gen yang di *crossover*. Pada permasalahan ini telah ditentukan nilai dari probabilitas *crossover* (pc) sebesar 0.5. Probabilitas *crossover* sebesar 0.5 berarti terdapat 50% gen dalam satu kromosom yang akan dipilih sebagai induk untuk di *crossover* dan nantinya akan menghasilkan *offspring*. *Offspring* merupakan kromosom baru yang dihasilkan setelah melalui proses-proses sebelumnya.

- Langkah awal untuk perhitungan *crossover* adalah memilih induk secara random yang nantinya akan di *crossover*. Pemilihan induk dilakukan dengan membangkitkan bilangan random sebanyak jumlah kromosom, selanjutnya bilangan random yang dipilih adalah $\text{random} < pc$. Missal yang terpilih sebagai induk kromosom 1 dan kromosom 3.

Kromosom yang terpilih sebagai induk [1 3]

K1	3	2	2	1	1	2	3	1
K3	3	1	3	2	2	1	1	2

- Selanjutnya pilih posisi kolom gen yang akan di *crossover* dengan bilangan random. Mis: [3 1 5 2]

Telah terpilih kolom yang akan di *crossover*:

K1	3	2	2	1	1	2	3	1
K3	3	1	3	2	2	1	1	2

Kolom: 1 2 3 4 5 6 7 8

- Langkah selanjutnya hitung nilai pada kolom yang sudah terpilih menggunakan *crossover* aritmatika:

K1	3	2	2	1	1	2	3	1
K3	3	1	3	2	2	1	1	2

Kolom: 1 2 3 4 5 6 7 8

- Misal pada K1 dan K3 yang terpilih adalah kolom 1:

Kolom 1 bernilai $X_1 = 3$, dan $X_2 = 3$

Misal $\lambda_1 = 0,3$ dan $\lambda_2 = 0,7$, maka:

$$O_1 = 0,3 \cdot 3 + 0,7 \cdot 3 \quad \Rightarrow \quad O_1 = 3,0 = 3$$

$$O_2 = 0,3 \cdot 3 + 0,7 \cdot 3 \quad \Rightarrow \quad O_2 = 3,0 = 3$$

- Misal pada K1 dan K3 yang terpilih adalah kolom 2:

Kolom 2 bernilai $X_1 = 2$, dan $X_2 = 1$

$$O_1 = 0,3 \cdot 2 + 0,7 \cdot 1 \quad \Rightarrow \quad O_1 = 1,3 = 1$$

$$O_2 = 0,3 \cdot 1 + 0,7 \cdot 2 \quad \Rightarrow \quad O_2 = 1,7 = 2$$

- Misal pada K1 dan K3 yang terpilih adalah kolom 3:

Kolom 3 bernilai $X_1 = 2$, dan $X_2 = 3$

$$O_1 = 0,3 \cdot 2 + 0,7 \cdot 3 \quad \Rightarrow \quad O_1 = 2,7 = 3$$

$$O_2 = 0,3 \cdot 3 + 0,7 \cdot 2 \quad \Rightarrow \quad O_2 = 2,3 = 2$$

- Misal pada K1 dan K3 yang terpilih adalah kolom 5:

$$O_1 = 0,3 \cdot 1 + 0,7 \cdot 2 \quad \Rightarrow \quad O_1 = 1,7 = 2$$

$$O2 = 0,3 \cdot 2 + 0,7 \cdot 1 \implies O2 = 1,3 = 1$$

- Selanjutnya dari perhitungan *crossover* aritmatika diatas, diperoleh nilai O1 dan O2. Ganti nilai kolom yang sudah terpilih nilai X1 dan X2 dengan nilai O1 dan O2:

K1	3	2	2	1	1	2	3	1
K3	3	1	3	2	2	1	1	2

↓ ↓ ↓ ↓

K1	3	1	3	1	2	2	3	1
K3	3	2	2	2	1	1	1	2

Dari perhitungan diatas maka diperoleh tabel hasil *crossover* yaitu:

Tabel 5.2 Hasil Crossover

Offspring	Node A		Node A		Node A		Node B	
	X	Y	X	Y	X	Y	X	Y
O1	3	1	3	1	2	2	3	1
O2	3	2	2	2	1	1	1	2

d. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi merupakan suatu operator pada algoritma genetika yang berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Sebelumnya telah ditentukan nilai probabilitas mutasi (pm) sebesar 0,2. Dengan nilai pm 0,2 diharapkan gen yang akan mutasi tidak sampai setengah dari total gen, sehingga untuk mutasi ditentukan nilai pm 0,2. Jika semakin besar pm maka gen yang akan dimutasi juga semakin banyak dan membuat nilai *offspring* memiliki sifat jauh berbeda dari induknya.

- Langkah awal mutasi yaitu membangkitkan bilangan random lebih dari 0 dan kurang dari 1 sebanyak total gen. Bilangan random yang dibangkitkan tidak boleh sama. Misal total gen 5x8, angka 5 didapat dari jumlah kromosom, sedangkan 8 didapat dari jumlah node dikali dengan 2.

0,93	0,25	0,63	0,24	0,04	0,15	0,28	0,74
0,76	0,81	0,19	0,42	0,95	0,06	0,47	0,33
0,31	0,71	0,03	0,77	0,75	0,66	0,09	0,4
0,02	0,89	0,36	0,43	0,39	0,87	0,73	0,49

0,32	0,98	0,68	0,37	0,39	0,58	0,01	0.9
------	------	------	------	------	------	------	-----

- Mutasi nilai gen yang mempunyai nilai < pm. Berikut cell tabel yang memiliki nilai gen < pm adalah: [5 6 11 14 19 23
25 39]

Mengubah nilai gen dengan X bilangan random 1 sampai 6, dan Y bilangan random 1 sampai 4. Nilai yang diubah adalah nilai yang menempati cell yang sudah terpilih.

Tabel 5.3 Hasil Mutasi

No. Kromosom	Node A		Node A		Node A		Node B	
	X	Y	X	Y	X	Y	X	Y
O3	3	2	2	1	1	1	3	1
O4	1	2	3	1	2	2	3	2
O5	3	1	2	2	2	1	3	2
O6	1	2	1	1	3	1	3	2
O7	1	2	3	2	2	2	3	1

e. Gabungan Parent dan Offspring

Pada tahap gabungan *parent* dan *offspring* merupakan penggabungan dari generasi awal dan juga nilai hasil perhitungan sebelumnya yaitu perhitungan *crossover* dan mutasi.

Tabel 5.4 Gabungan Parent dan Offspring

Nama	Nama Awal/Asal	Node A		Node A		Node A		Node B	
		X	Y	X	Y	X	Y	X	Y
P1	K1	3	2	2	1	1	2	3	1
P2	K2	1	2	1	1	2	1	3	2
P3	K3	3	1	3	2	2	1	1	2
P4	K4	2	2	1	1	3	1	3	2
P5	K5	1	2	3	2	2	2	2	1
O1	-	3	1	3	1	2	2	3	1
O2	-	3	2	2	2	1	1	1	2
O3	-	3	2	2	1	1	1	3	1

O4	-	1	2	3	1	2	2	3	2
O5	-	3	1	2	2	2	1	3	2
O6	-	1	2	1	1	3	1	3	2
O7	-	1	2	3	2	2	2	3	1

Tabel P1 sampai P5 merupakan nilai generasi awal, O1 dan O2 merupakan hasil dari operasi *crossover*, O3 sampai O7 merupakan hasil operasi mutasi.

f. Perhitungan Nilai Fitness

Perhitungan nilai fitness dilakukan untuk mencari generasi mana yang merupakan individu terbaik dibanding dengan individu lainnya. Dalam perhitungan genetika pencarian nilai *fitness* ditentukan dengan cara yang berbeda-beda sesuai dengan masalah yang sedang diselesaikan. Pada penelitian ini nilai *fitness* digunakan untuk mencari posisi node terbaik dengan parameter luas area ruangan yang diukur dan luas daerah yang *tercover* sensor. Luas area ruangan yang diukur dan luas daerah yang *tercover* sensor diukur sistem dengan menggunakan nilai pixel. Hasil dari perhitungan genetika disimpan dalam bentuk .png.

Tabel 5.5 Nilai *Fitness*

Nama	Nama Awal/Asal	Node A		Node A		Node A		Node B		Nilai <i>Fitness</i>
		X	Y	X	Y	X	Y	X	Y	
P1	K1	3	2	2	1	1	2	3	1	8264
P2	K2	1	2	1	1	2	1	3	2	10834
P3	K3	3	1	3	2	2	1	1	2	7781
P4	K4	2	2	1	1	3	1	3	2	9875
P5	K5	1	2	3	2	2	2	2	1	7659
O1	-	3	1	3	1	2	2	3	1	10745
O2	-	3	2	2	2	1	1	1	2	9604
O3	-	3	2	2	1	1	1	3	1	11423
O4	-	1	2	3	1	2	2	3	2	8451
O5	-	3	1	2	2	2	1	3	2	11925
O6	-	1	2	1	1	3	1	3	2	10728
O7	-	1	2	3	2	2	2	3	1	8304

g. Proses Seleksi

Seleksi dilakukan untuk menentukan individu terbaik dari setiap populasi. Individu terbaik dipilih dengan melihat nilai *fitness* dari setiap kromosom.

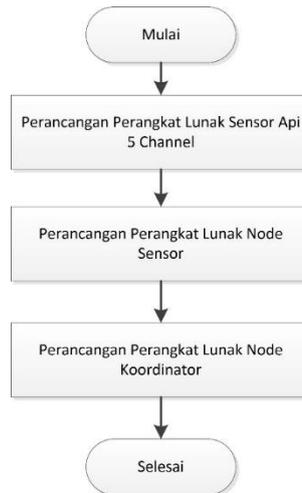
Tabel 5.6 Seleksi

Nama	Nama Awal/Asal	Node A		Node A		Node A		Node B		Nilai <i>Fitness</i>
		X	Y	X	Y	X	Y	X	Y	
P1	K1	3	2	2	1	1	2	3	1	8264
P2	K2	1	2	1	1	2	1	3	2	10834
P3	K3	3	1	3	2	2	1	1	2	7781
P4	K4	2	2	1	1	3	1	3	2	9875
P5	K5	1	2	3	2	2	2	2	1	7659
O1	-	3	1	3	1	2	2	3	1	10745
O2	-	3	2	2	2	1	1	1	2	9604
O3	-	3	2	2	1	1	1	3	1	11423
O4	-	1	2	3	1	2	2	3	2	8451
O5	-	3	1	2	2	2	1	3	2	11925
O6	-	1	2	1	1	3	1	3	2	10728
O7	-	1	2	3	2	2	2	3	1	8304

Kromosom terdiri dari beberapa titik pusat. Kumpulan titik pusat yang memiliki nilai *fitness* tertinggi atau terbesar maka kumpulan titik pusat tersebut yang terpilih dan diambil menjadi sebuah individu baru. Pada table 5.6 terlihat bahwa yang memiliki nilai *fitness* terbesar adalah kromosom O5 dan merupakan hasil dari proses genetika tersebut. O5 akan muncul sebagai individu baru.

5.1.1.2 Perancangan Perangkat Lunak Mikrokontroler Arduino

Perancangan perangkat lunak pada mikrokontroler Arduino akan menjelaskan system kerja dari awal hingga akhir system pada arduino. Perancangan keseluruhan system pada mikrokontroler arduino dapat dilihat pada Gambar 5.3 yang merupakan diagram alir dari system.

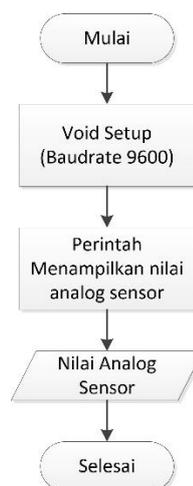


Gambar 5.3 Diagram Alir Perancangan Perangkat Lunak pada Arduino

Pada digram alir diatas menunjukkan tahapan-tahapn perancangan perangkat lunak secara keseluruhan pada mikrokontroler Arduino. Terdapat tiga node sensor. Dalam setiap node sensor dapat mendeteksi api, yang berarti setiap node sensor mendapatkan beberapa informasi. Selanjutnya node sensor terhubung dengan node koordinator. Informasi dari node sensor dikirim pada node koordinator. Dalam node koordinator terjadi pengolahan beberapa informasi yaitu keberadaan api, delay, throughput, jitter. Informasi tersebut akan ditampilkan pada komputer berupa tampilan web. Dalam perancangan perangkat lunak Arduino tersusun atas beberapa perancangan yaitu:

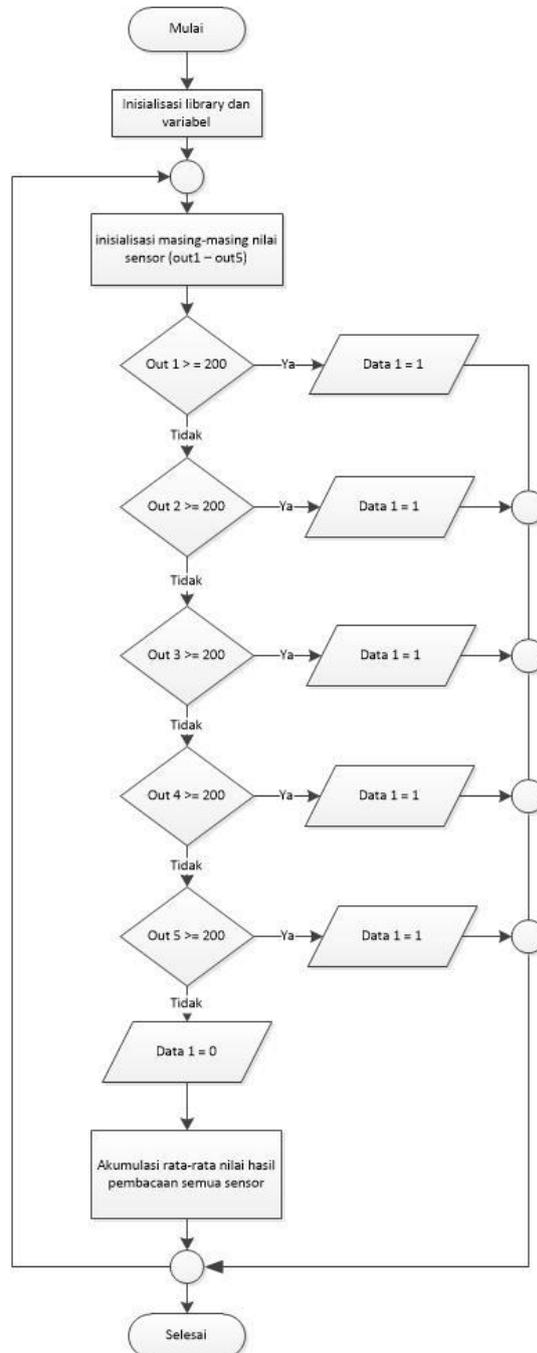
1. Perancangan Perangkat Lunak Sensor Api 5 Channel

Pada bagian ini akan menjelaskan tentang perancangan untuk mengoperasikan sensor api 5 channel yang terhubung pada Arduino. Gambar 5.4 merupakan diagram alir proses perancangan perangkat lunak sensor api 5 channel untuk pengujian *coverage area* sensor.



Gambar 5.4 Alur Program Sensor Api 5 Channel untuk Pengujian Coverage Sensor

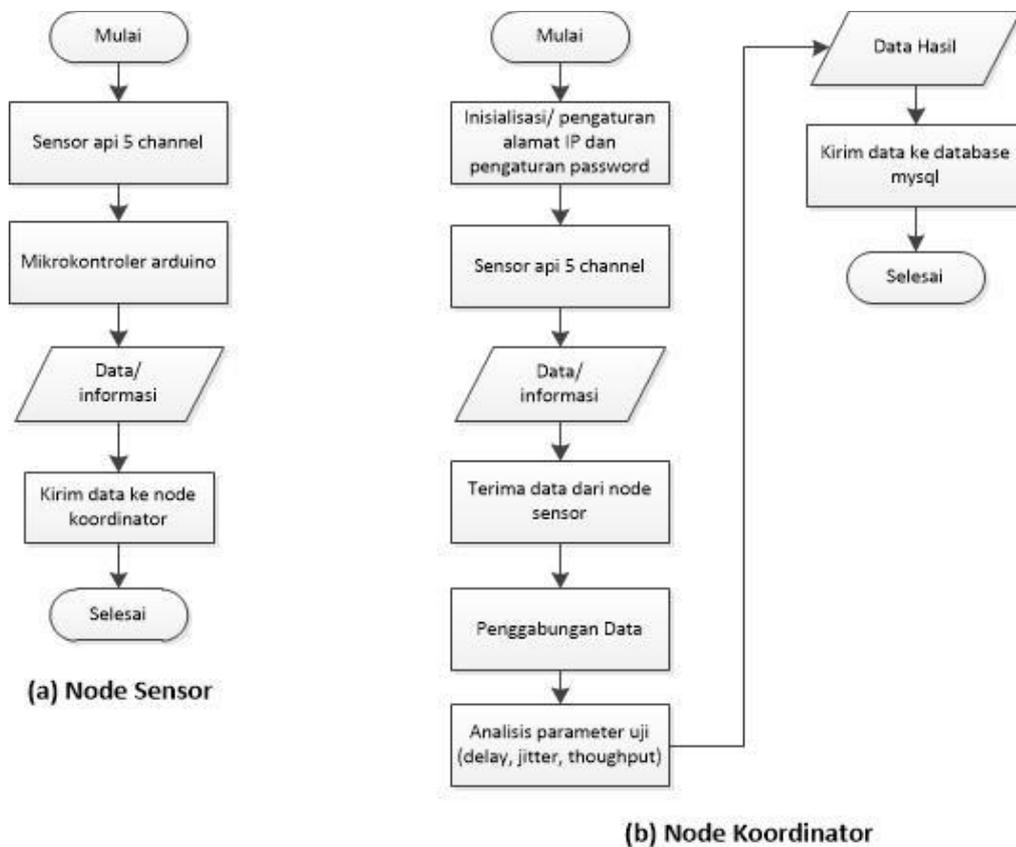
Dimulai dari program untuk mengatur baudrate pada void setup, selanjutnya membuat perintah untuk menampilkan nilai analog sensor pada void loop. Dan menampilkan nilai analog sensor pada serial monitor. Selain Program sensor api 5 channel untuk pengujian, selanjutnya adalah program sensor api 5 channel pada node sensor dan node koordinator. **Gambar 5.5** merupakan alur dari program sensor api 5 channel pada node sensor dan node koordinator.



Gambar 5.5 Diagram Alir Perancangan Sensor Api 5 Channel Node Sensor dan Node Koordinator

Diagram alir diatas menunjukkan proses pengolahan sensor api 5 channel pada node sensor dan node koordinator. Sensor api 5 channel terdapat 5 channel untuk mendeteksi api. Kelima channel tersebut dijadikan nilai dari sensor yaitu out 1 sampai dengan out 5. Ketika out 1 sampai dengan out 5 memiliki nilai lebih dari sama dengan 200, maka selanjutnya akan disimpan pada data 1 = 1 berarti terdeteksi, sedangkan apabila nilai dari sensor kurang dari 200, maka masuk pada data 1 = 0 yang berarti tidak terdeteksi.

2. Pada penjelasan keseluruhan sistem pada Arduino sebelumnya telah disebutkan bahwa sistem terdiri dari node sensor dan node koordinator. Gambar 5.6 (a) menunjukkan diagram alir node sensor dan gambar 5.6 (b) menunjukkan diagram alir node koordinator. Secara umum rancangan node sensor dan node koordinator dapat dilihat sebagai berikut.



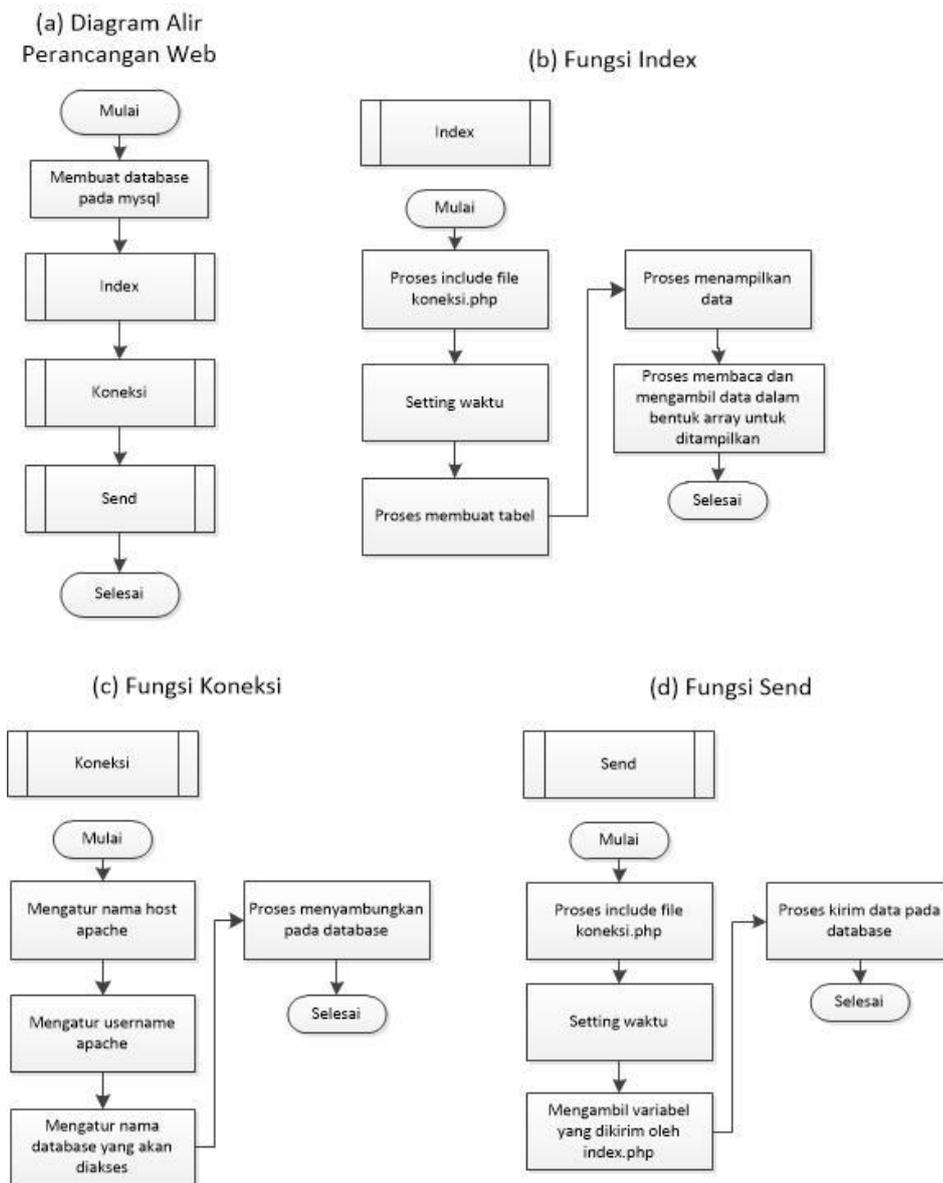
Gambar 5.6 Diagram Alir Node Sensor dan Node Koordinator

Pada diagram alir diatas menunjukkan alur jalannya node sensor dan node koordinator. Pada node sensor dimulai dari sensor api 5 channel mendeteksi keberadaan api, yang kemudian diolah oleh mikrokontroler Arduino dan menghasilkan data/informasi yang selanjutnya dikirim ke node koordinator. Alur pada node koordinator dimulai dari insialisasi untuk pengaturan IP dan *password*, selanjutnya proses deteksi api, hal ini dikarenakan pada node koordinator juga berfungsi untuk mendeteksi api. Dari proses deteksi api menghasilkan data/informasi. Selanjutnya proses terima data dari node

sensor. Data dari node sensor digabung dengan data/informasi sebelumnya. Proses selanjutnya adalah proses analisis parameter uji yaitu delay, throughput, dan jitter. Ketika proses analisis uji telah dilakukan maka akan menghasilkan data hasil dari analisis tersebut. Data hasil selanjutnya dikirim/disimpan pada database mysql.

5.1.1.3 Perancangan Perangkat Lunak Web dan Xampp

Perancangan perangkat lunak web merupakan proses pengaturan tampilan web dan pengaturan database dengan bantuan xampp. **Gambar 5.7** merupakan digram alir perancangan web dan xampp.



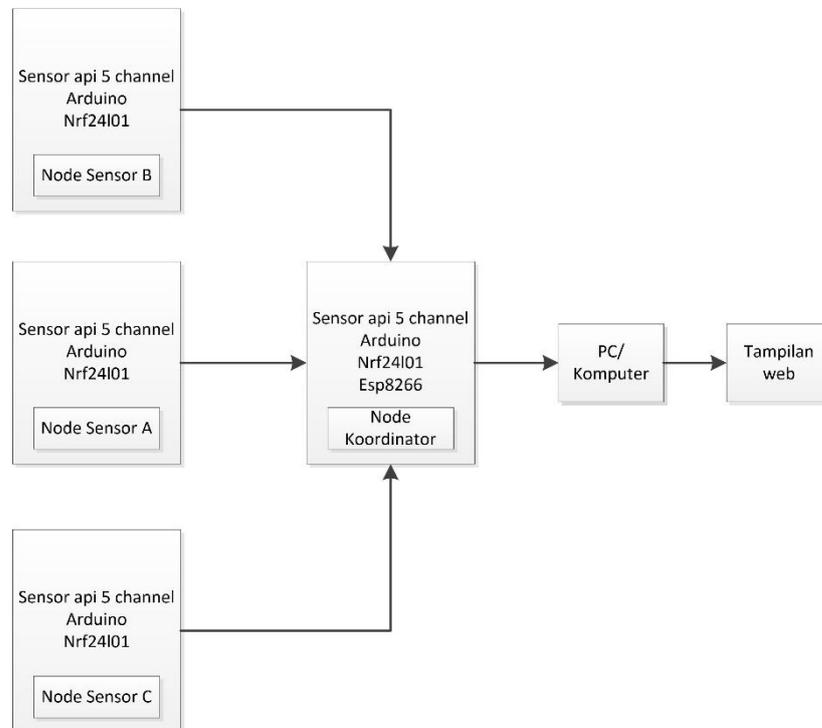
Gambar 5.7 Diagram Alir Perancangan Web

Awal perancangan ini dimulai dari membuat database mysql dengan bantuan xampp untuk menampung data hasil pengolahan. Proses supaya web dapat menampilkan hasil maka terdapat beberapa tahapan. Yang pertama tahapan

index. Pada index mengatur tampilan yang diinginkan pengguna pada web. Pada fungsi index mempunyai tahap-tahap untuk menjalankannya. Yang kedua adalah koneksi. Pada tahap ini dibuat sebuah koneksi supaya dapat tersambung antara data, database dan dengan web. Terdapat tahapan-tahapan membuat fungsi koneksi seperti terlihat pada gambar. Ketiga adalah send, yang membantu memasukkan data ke database khususnya tabel api selanjutnya database menampilkan di web. Tahapan pada fungsi send dapat dilihat pada gambar 5.7 (d).

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan perancangan sistem melibatkan perangkat keras seperti arduino, sensor, nrf24l01 dan sebagainya. Perancangan perangkat keras tersusun atas perancangan node pada transmitter, dan perancangan node pada receiver. Gambar 5.8 merupakan keseluruhan sistem perangkat keras pendeteksi kebakaran:



Gambar 5.8 Keseluruhan Sistem Perangkat Keras

Perancangan perangkat keras tersusun atas node sensor dan node koordinator. Node sensor berjumlah 3 node, dan 1 node koordinator. Data pada node sensor akan dikirim pada node coordinator dan akan diteruskan pada database dan selanjutnya dikirim ke komputer dan tampil pada web browser.

5.1.2.1 Perancangan Node pada *Transmitter* (Node Sensor)

Pada gambaran umum sistem sebelumnya telah diketahui bahwa node *transmitter* (node sensor) terdiri dari 3 node. Satu node tersusun atas satu sensor api 5 channel, satu arduino uno, dan satu nrf24l01. Sensor api 5 channel sebagai pendeteksi api, arduino uno sebagai mikrokontroler pemroses data, nrf24l01

sebagai modul *transceiver* yaitu pengirim data ke *receiver*. Tiga node transmitter memiliki rancangan sistem yang sama. Perancangan node transmitter terdiri dari beberapa tahapan perancangan seperti berikut.

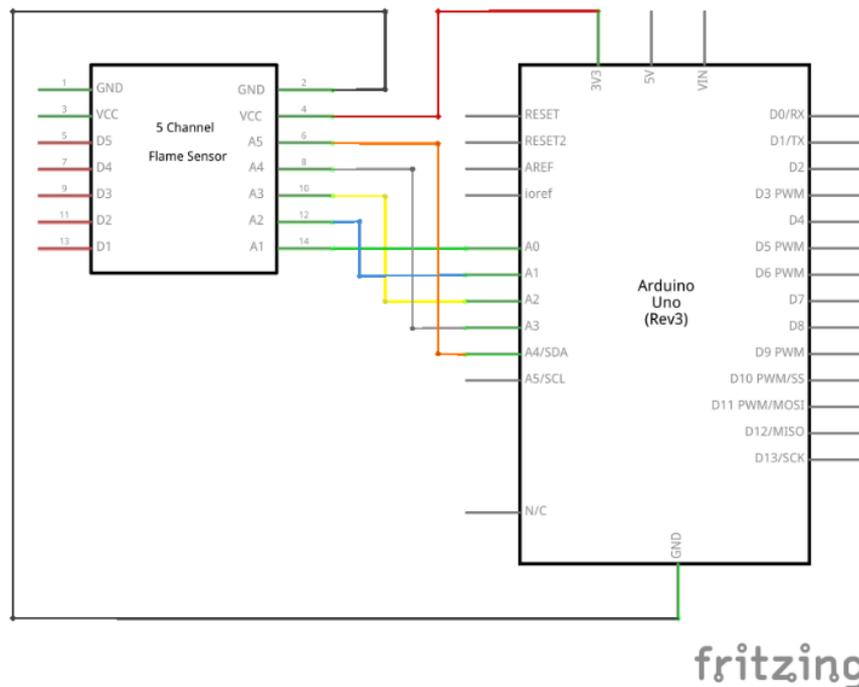
1. Perancangan sensor api 5 channel.

Sistem ini menggunakan sensor api 5 channel sebagai masukan data untuk menganalisis kinerja sistem. Pin analog sensor api 5 channel akan dihubungkan ke pin analog arduino uno. Pada arduino uno terdapat ADC (*analog digital converter*) sehingga data dari sensor api 5 channel yang berupa data analog akan diubah (*converted*) menjadi data digital. **Tabel 5.7** merupakan konfigurasi pin sensor api 5 channel dengan arduino uno.

Tabel 5.7 Konfigurasi Pin Sensor Api 5 Channel pada Node Sensor

Pin Sensor Api 5 Channel	Pin Arduino Uno
Vcc	5 V
A1	A0
A2	A1
A3	A2
A4	A3
A5	A4
GND	GND

Skematik perancangan sensor api 5 channel dapat dilihat pada gambar 5.9.



Gambar 5.9 Skematik Perancangan Sensor Api 5 Channel

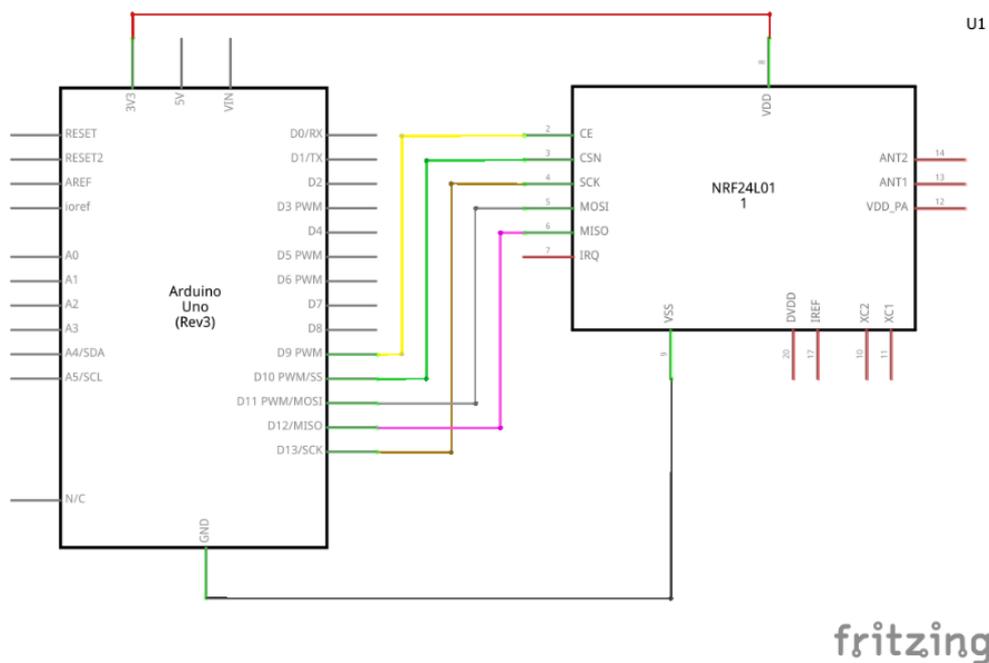
2. Perancangan modul nrf24l01

Perancangan modul nrf24l01 dengan menghubungkan pin-pin modul nrf24l01 dengan pin-pin pada arduino uno. Modul nrf24l01 akan membantu proses transceiver data. Tabel 5.8 merupakan konfigurasi pin nrf24l01 dengan pin arduino uno.

Tabel 5.8 Konfigurasi Pin nRF24l01 pada Node Sensor

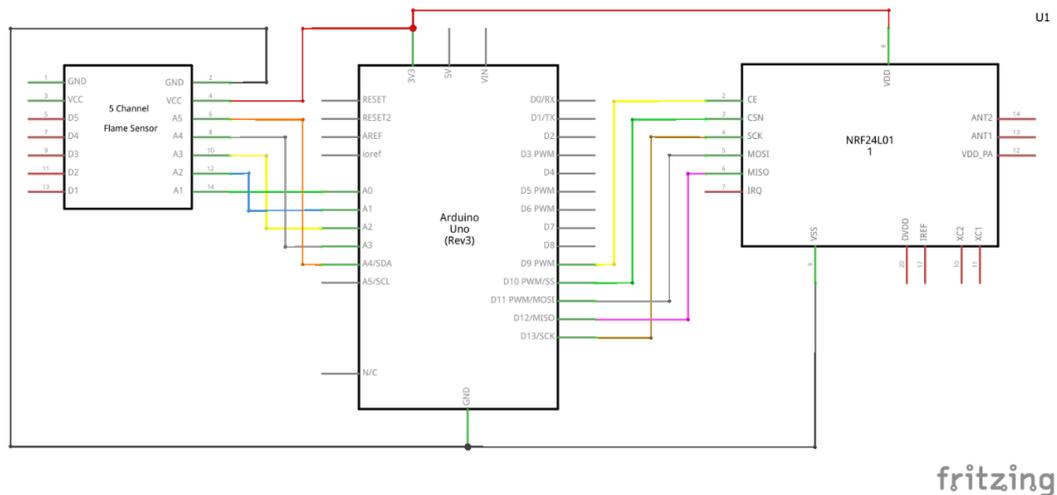
Pin nRF24l01	Pin Arduino Uno
VCC	3.3V
GND	GND
CE	9
CSN	10
MOSI	11
SCK	13
MISO	12
IRQ	-

Gambar 5.10 merupakan skematik dari perancangan modul nRF24l01.



Gambar 5.10 Skematik Perancangan nRF24l01

Dari perancangan sensor api 5 channel dan perancangan modul nRF24l01 tersebut, maka tersusunlah node transmitter (node sensor) dari sistem. Secara keseluruhan perancangan node transmitter (node sensor) dapat dilihat pada **Gambar 5.11**.



Gambar 5.11 Skematik Node Transmitter (Node Sensor)

5.1.2.2 Perancangan Node pada Receiver (Node Koordinator)

Perancangan node coordinator tersusun atas 3 perancangan, yaitu perancangan sensor api 5 channel, perancangan modul nRF24I01, dan perancangan modul WIFI esp8266. Modul WIFI esp8266 yang nantinya akan menjembatani antara data dari node koordinator ke PC koordinator, sehingga PC koordinator dapat dengan mudah mengakses data dari node koordinator.

1. Perancangan Sensor Api 5 Channel

Sama seperti pada node *transmitter* (node sensor), perancangan sensor api 5 channel dilakukan dengan menghubungkan pin-pin pada sensor api 5 channel dengan pin pada arduino uno. Tata letak pin-pin sensor api 5 channel sama dengan tata letak pin-pin pada node *transmitter*. Tabel 5.9 merupakan konfigurasi pin sensor api 5 channel dengan arduino uno pada node *receiver* (node koordinator).

Tabel 5.9 Konfigurasi Pin Sensor Api 5 Channel pada Node Koordinator

Pin Sensor Api 5 Channel	Pin Arduino Uno
Vcc	5 V
A1	A0
A2	A1
A3	A2
A4	A3
A5	A4
GND	GND

2. Perancangan Modul nRF24I01

Perancangan modul nRF24I01 pada node *receiver* sama dengan perancangan modul nRF24I01 pada node *transmitter* (node sensor). Pin-pin pada modul nRF24I01 dihubungkan dengan pin-pin pada arduino uno. Tabel 5.10 merupakan konfigurasi pin modul nRF24I01 dengan arduino uno pada node *receiver* (node koordinator).

Tabel 5.10 Konfigurasi Pin nRF24I01 pada Node Koordinator

Pin nRF24I01	Pin Arduino Uno
VCC	3.3V
GND	GND
CE	9
CSN	10
MOSI	11
SCK	13
MISO	12
IRQ	-

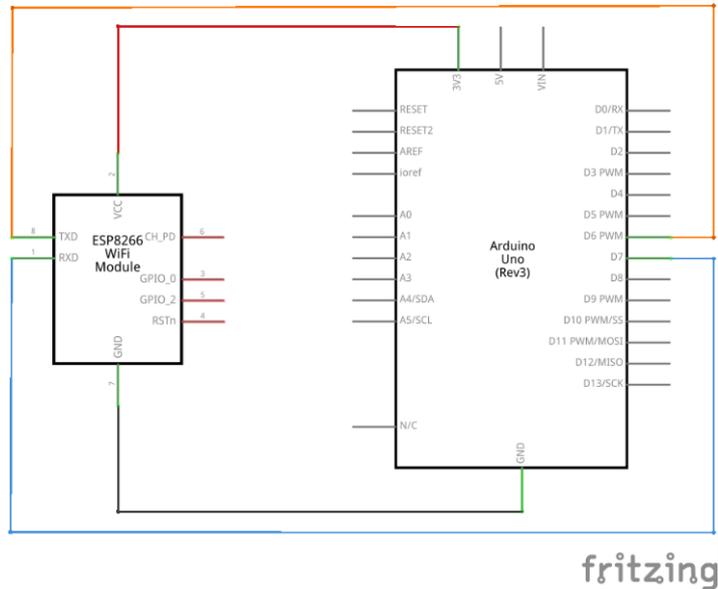
3. Perancangan Modul WIFI ESP8266

Pada perancangan node *receiver* (node coordinator) terdapat perancangan modul WIFI esp8266. Modul ESP8266 merupakan SOC mandiri terintegrasi dengan TCP/IP stack protokol yang dapat memberikan akses mikrokontroler ke jaringan WIFI. Sehingga akan membantu memudahkan PC koordinator untuk memantau informasi yang dikirim dari node koordinator. Perancangan modul WIFI esp8266 dengan menghubungkan pin-pin esp8266 ke pin-pin arduino uno. Tabel 5.11 merupakan konfigurasi pin modul WIFI esp8266 dengan arduino uno.

Tabel 5.11 Konfigurasi Pin Modul WIFI ESP8266 dengan Arduino Uno

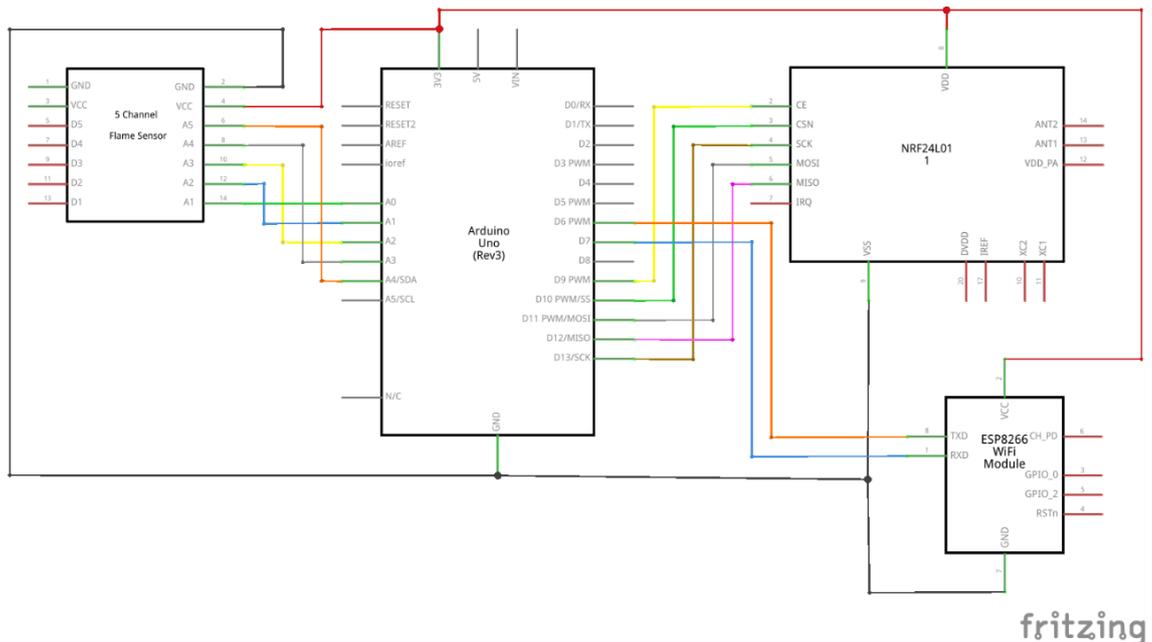
Pin Modul WIFI ESP8266	Pin Arduino Uno
VCC	VCC
RX	7
RST	-
GPIO0	-
CH_PD	-
GPIO2	-
TX	6
GND	GND

Tabel diatas merupakan konfigurasi antara pin modul wifi esp8266 dengna arduino. **Gambar 5.12** merupakan skematik dari perancangan modul *wifi* esp8266 dengan arduino pada node *receiver* (node koordinator).



Gambar 5.12 Skematik Perancangan Modul *WIFI* ESP8266

Dari perancangan-perancangan diatas, maka dapat dirancanglah sebuah node *receiver* atau node koordinator yang akan mengirimkan informasi ke *pc/computer* coordinator dengan bantuan modul *wifi* esp8266 sehingga keadaan ruangan dapat dipantau dengan mudah. **Gambar 5.13** merupakan skematik dari perancangan node *receiver*/node koordintor.



Gambar 5.13 Skematik Perancangan Node *Receiver* (Node Koordinator)

5.2 Implementasi Sistem

Implementasi system berisikan proses menerapkan/menjalankan semua rancangan yang telah dibuat sebelumnya. Pada system ini, implementasi dibagi berdasarkan proses perancangan yang ada yaitu implementasi perangkat lunak dan perangkat keras. Dalam perangkat lunak terdiri dari 2 implementasi yaitu implementasi perangkat lunak pada matlab dan implementasi pernakgkat lunak pada xampp.

5.2.1 Implementasi Perangkat Lunak Matlab

Ketika proses perancangan perangkat lunak pada matlab yaitu system untuk menentukan posisi node dengan menggunakan algoritma genetika adalah sebagai berikut:

5.2.1.1 Program Input Data

Pada program *input* data ini berisikan program untuk membuat sebuah variable yang mana nilai variable tersebut didapat berupa dari *input* pengguna dan penentuan secara awal.

Tabel 5.12 Program Input Data

No	Program
1	iterasi = str2double(get(handles.edit1,'String'));
2	jml_populasi = str2double(get(handles.edit2,'String'));
3	if jml_populasi < 2
4	msgbox('jumlah populasi harus lebih besar dari 1');
5	return;
6	end
7	node1 = str2double(get(handles.edit3,'String'));
8	node2 = str2double(get(handles.edit4,'String'));
9	jaril = str2double(get(handles.edit5,'String'));
10	jaril2 = str2double(get(handles.edit6,'String'));
11	panjang = str2double(get(handles.edit11,'String'));
12	lebar = str2double(get(handles.edit12,'String'));
13	pesan = 'total node harus kurang atau sama dengan %d.';
14	jml_node = (panjang+1)*(lebar+1);
15	str = sprintf(pesan,jml_node);
16	
17	if node1+node2 > (panjang+1)*(lebar+1)
18	msgbox(str);
19	return;
20	end
21	tic;
22	n_kromosom = jml_populasi;
23	p_kromosom = (node1+node2)*2;
24	ulang = iterasi;
25	gen = zeros(n_kromosom,p_kromosom);
26	gen_total = zeros(2*n_kromosom+2,p_kromosom);
27	gen_max = zeros(p_kromosom/2,2);
28	jarak = zeros(p_kromosom/2,p_kromosom/2);

Program baris 1-2, menunjukkan variable yang nilainya berasal dari masukan/*input* pengguna. Baris 3-6 merupakan kondisi dimana pengguna

memasukkan jumlah populasi pada kolom populasi < 2 maka akan muncul kondisi “jumlah populasi harus lebih besar dari 1”. Baris 14-28 membuat variable serta mendeklarasikan variable tersebut sesuai dengan kondisi yang akan dibuat sistem.

5.2.1.2 Inisialisasi Nilai Gen Awal

Pada program inisialisasi awal ini dibuat perulangan dalam perulangan dengan menggunakan fungsi for.

Tabel 5.13 Program Inisialisasi Nilai Gen Awal

No	Program
1	for i=1 : n_kromosom
2	for j= 1: p_kromosom
3	if mod(j,2)~= 0
4	gen(i,j)= floor(rand*panjang);
5	else
6	gen(i,j)= floor(rand*lebar);
7	end
8	end
9	end

Baris 1 menunjukkan i dimana nilai i sebanyak 1 sampai dengan n_kromosom. Baris 2 menunjukkan j dengan nilai 1 sampai dengan p_kromosom. Variable i merupakan baris, j merupakan kolom. Pada inisialisasi nilai gen awal ini, membangkitkan bilangan secara acak dengan nilai x memiliki interval sampai dengan panjang, dan nilai y interval nilai sampai lebar.

5.2.1.3 Crossover

Program crossover berisikan program untuk perhitungan crossover sesuai pada perhitungan algoritma genetika. Crossover pada proses ini menggunakan crossover aritmatika. Pada program crossover terdiri dari beberapa tahapan.

Tabel 5.14 Program Crossover

No	Program
1	function gen_anak=crossover(n_kromosom,p_kromosom,gen)
	gen_anak =zeros(2,p_kromosom);
2	ind(1)=randi(n_kromosom);
3	ind(2)=randi(n_kromosom);
4	while ind(2)==ind(1)
5	ind(2)=randi(n_kromosom);
6	end
7	pc=0.5;
8	n_pc=p_kromosom*pc;
9	% memilih kolom yang akan dicrossover
10	gen_c=zeros(n_pc);
11	gen_c(1)=randi(p_kromosom);
12	a=2;
13	while a<= n_pc
14	gen_c(a)=randi(p_kromosom);
15	for i=1:a-1
16	if gen_c(a)== gen_c(i)
17	a=a-1;
18	end

```

19     end
20     a=a+1;
21 end
22 % proses crossover
23 lamb(1)=rand;
24 lamb(2)=1-lamb(1);
25 for i=1:2
26     for p=1: p_kromosom
27         gen_anak(i,p)=gen(ind(i),p);
28     end
29 end
30 for i=1:n_pc
31     gen_anak(1,gen_c(i))=floor(lamb(1)*gen(ind(1),gen_c(i))+lam
32     b(2)*gen(ind(2),gen_c(i)));
33     gen_anak(2,gen_c(i))=floor(lamb(1)*gen(ind(2),gen_c(i))+lam
34     b(2)*gen(ind(1),gen_c(i)));
35 end

```

Baris 1-7 mendeklarasikan variable. Baris 8 mendeklarasikan variable “pc” yaitu probabilitas crossover sebesar 0.5. Baris 8 menunjukkan bahwa terdapat 50% gen dalam satu kromosom yang akan dipilih sebagai induk untuk di *crossover*. Baris 10-22 merupakan proses untuk memilih induk yang akan di *crossover*. Selanjutnya setelah memilih induk maka proses *crossover*. Perhitungan menggunakan *crossover* aritmatika, dijelaskan pada program baris 31-34.

5.2.1.4 Mutasi

Program mutasi merupakan program tahapan-tahapan proses mutasi pada algoritma genetika.

Tabel 5.15 Program Mutasi

No	Program
1	Function
2	gen_mutasi=mutasi(n_kromosom,p_kromosom,gen,panjang,lebar)
3	pm=0.2;
4	gen_mutasi=zeros(n_kromosom,p_kromosom);
5	for i=1:n_kromosom
6	for j=1:p_kromosom
7	gen_mutasi(i,j)=rand;
8	if gen_mutasi(i,j)<pm
9	if mod(j,2)~= 0
10	gen_mutasi(i,j)= floor(rand*panjang);
11	else
12	gen_mutasi(i,j)= floor(rand*lebar);
13	end
14	else
15	gen_mutasi(i,j)=gen(i,j);
16	end
17	end

Baris 2 menentukan terlebih dahulu probabilitas dari mutasi yaitu 0.2. Baris 3 membuat variable *gen_mutasi* dengan membuat matriks yang semua datanya bernilai 0 dengan perintah “zeros”. Selanjutnya mengisi variable *gen_mutasi*

dengan nilai acak/*random* seperti terlihat pada baris 4-6. Setelah membangkitkan nilai acak, memilih nilai yang kurang dari probabilitas mutasi pada baris 7. Baris 8-17 mengganti nilai < pm dengan nilai acak x sampai dengan nilai panjang, dan y sampai dengan nilai lebar.

5.2.1.5 Nilai Fitness

Program nilai fitness berisikan tahapan-tahapan perhitungan nilai fitness algoritma genetika.

Tabel 5.16 Program Mencari Nilai Fitness

No	Program
1	function
2	fitness=fit(total_kromosom,p_kromosom,gen_total,n1,jari1,ja
3	ri2,panjang,lebar)
4	fitness=zeros(total_kromosom,1);
5	for i=1:total_kromosom
6	for j=1:p_kromosom/2
7	if j<=n1
8	x=gen_total(i,j*2-1);
9	y=gen_total(i,j*2);
10	gambar(x,y,jari1,panjang,lebar);
11	else
12	x=gen_total(i,j*2-1);
13	y=gen_total(i,j*2);
14	gambar(x,y,jari2,panjang,lebar);
15	end
16	end
17	print('coba','-dpng','-r0','-nou');
18	A=imread('coba.png');
19	fitness(i)=luas(A);
20	cla;
21	end;
22	% operasi fungsi untuk menggambar lingkaran hasil
23	function gambar(x,y,jari2,panjang,lebar)
24	pos = [x-jari2 y-jari2 jari2*2 jari2*2];
25	rectangle('Position',pos,'Curvature',[1
	1],'facecolor','k');
	axis([0 panjang 0 lebar]);

Proses perhitungan nilai fitness dibuat pada fungsi fitness. Nilai fitness diperoleh dari menyimpan hasil perhitungan genetika dalam bentuk .png (gambar). Selanjutnya ditampilkan pada denah gambar, daerah yang tercover sensor dikurangi dengan luas yang tidak tercover oleh sensor.

5.2.1.6 Seleksi

Program seleksi menunjukkan proses atau tahapan-tahapan untuk melakukan seleksi pada algoritma genetika. Proses seleksi dengan memilih satu yang mempunyai nilai fitness paling tinggi.

Tabel 5.17 Program Seleksi

No	Program
1	cek_fitness=fitness;
2	for k=1:n_kromosom
	[maksimum,indeks]=max(cek_fitness(1:total_kromosom-k+1));
3	gen(k,:)=gen_total(indeks,:);
4	if k==1
5	kromosom_max=indeks;
6	fit_max=maksimum;
7	end
8	cek_fitness(indeks)=fitness(total_kromosom-k+1);
9	end

Program seleksi menunjukkan perintah untuk menyeleksi nilai fitness terbesar yang nantinya menjadi individu baru. Baris 2-7 menunjukkan proses seleksi pemilihan nilai fitness terbesar.

5.2.1.7 Iterasi

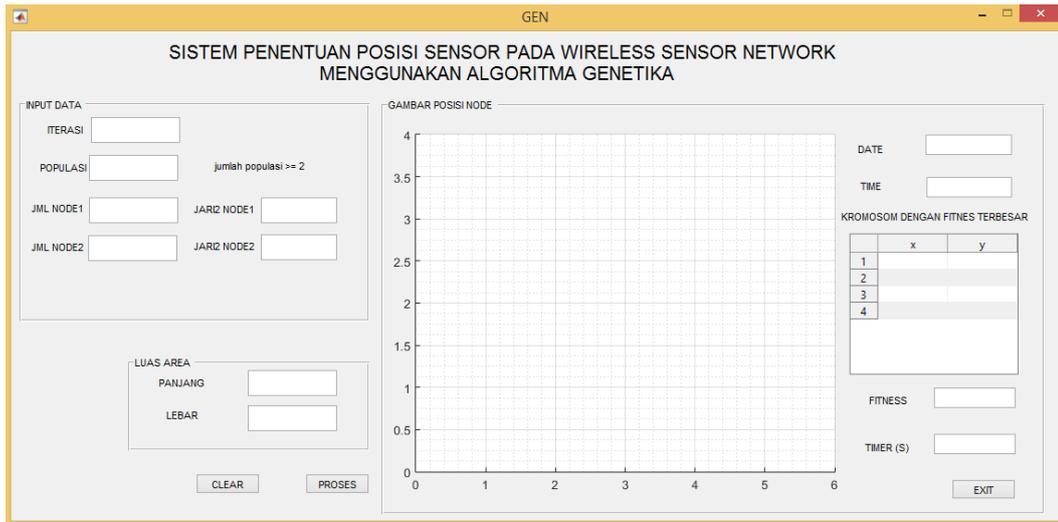
Pada sistem ini iterasi merupakan sebuah perulangan dimana ketika suatu proses selesai maka akan terus melakukan proses tersebut sebanyak jumlah yang telah ditentukan. Pada sistem ini iterasi ditentukan oleh pengguna sebelumnya ketika memulai sistem.

Tabel 5.18 Program Iterasi

No	Program
1	ite=1;
2	while ite <= ulang gen0=gen;
3	total_kromosom = n_kromosom;
4	gen_total(1:total_kromosom,:)=gen0;
5	gen_anak=crossover(n_kromosom,p_kromosom,gen0);
6	total_kromosom = total_kromosom+2;
7	gen_total(n_kromosom+1:total_kromosom,:)=gen_anak;
9	gen_mutasi=mutasi(n_kromosom,p_kromosom,gen0,panjang,
10	lebar);
11	total_kromosom = total_kromosom+n_kromosom;
12	gen_total(n_kromosom+3:total_kromosom,:)=gen_mutasi;
14	fitness=fit(total_kromosom,p_kromosom,gen_total,nodel,jari1, jari2,panjang,lebar);

Pada baris 2-14 merupakan perintah untuk mengulang semua proses-proses sebelumnya dari proses pemilihan induk, crossover, mutasi, mencari nilai fitness, dan seleksi.

Ketika sistem telah dijalankan, Gambar 5.14 merupakan tampilan awal dari sistem:



Gambar 5.14 Tampilan Sistem Perangkat Lunak Matlab

Pada tampilan awal system ini terbagi menjadi 2 bagian, yang pertama masukan/*input* terletak pada bagian kiri gambar dan keluaran/*output* pada bagian kanan.

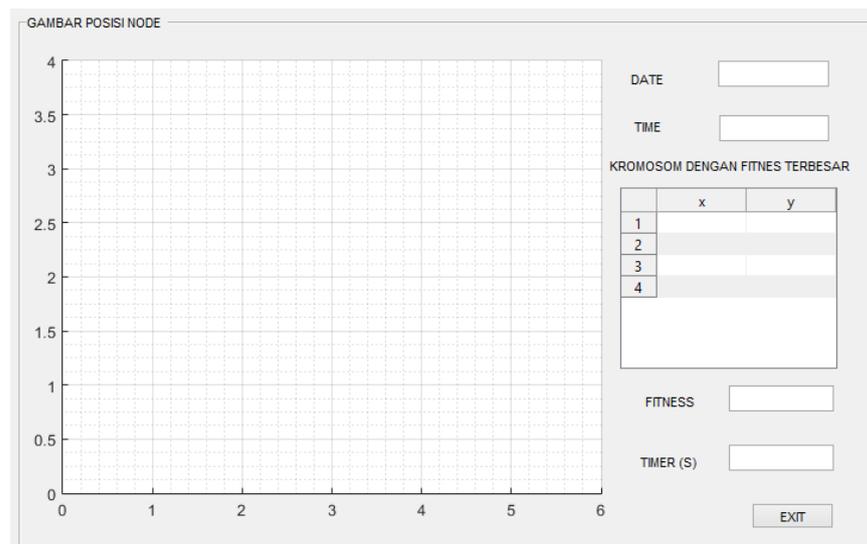
Gambar 5.15 merupakan tampilan masukan/*input* data:

Gambar 5.15 Bagian Masukan/*Input* Data Matlab

- Kolom iterasi diisi oleh pengguna dengan menambahkan angka jumlah iterasi yang diinginkan.

- Kolom populasi diisi jumlah populasi yang diinginkan oleh pengguna. Jumlah populasi harus lebih besar atau sama dengan 2, dikarenakan agar dapat memulai perhitungan genetika maka populasi harus lebih besar atau sama dengan 2.
- Kolom jml node 1 dan jml node 2. Kolom ini diisi dengan jumlah node yang diinginkan atau yang akan digunakan oleh *user*. Jumlah node dibuat 2 kolom karena pada system ini dibuat 2 jenis node yang setiap satu jenis node memiliki jari-jari sensor berbeda. Jumlah node berdasarkan pada jumlah jenis sensor berbeda.
- Kolom jari2 node 1 dan kolom jari2 node 2. Kolom ini diisi dengan ukuran jari-jari sensor. Pada system ini jumlah jenis sensor dibatasi maksimal 2 jenis sensor. Satuan jari-jari sensor adalah meter (m).
- Kolom Panjang. Kolom panjang terdapat pada bagian luas area. Luas area ini merupakan luas area ruangan atau bangunan yang merupakan tempat implementasi. Kolom panjang berisikan panjang ruangan dalam satuan meter (m).
- Kolom Lebar. Berisikan lebar ruangan atau bangunan tempat dilakukan implementasi dalam satuan meter (m). Dapat diisi dengan angka decimal.
- Botton “clear”. Merupakan tombol hapus atau membersihkan semua data *input* dan *output* yang terdapat pada tampilan matlab.
- Botton “proses”. Merupakan tombol untuk memproses keluaran/*input* yang telah diisikan oleh pengguna.

Selanjutnya bagian keluaran/*output* yang terletak pada bagian kanan system. **Gambar 5.16** merupakan tampilan keluaran/*output* data:



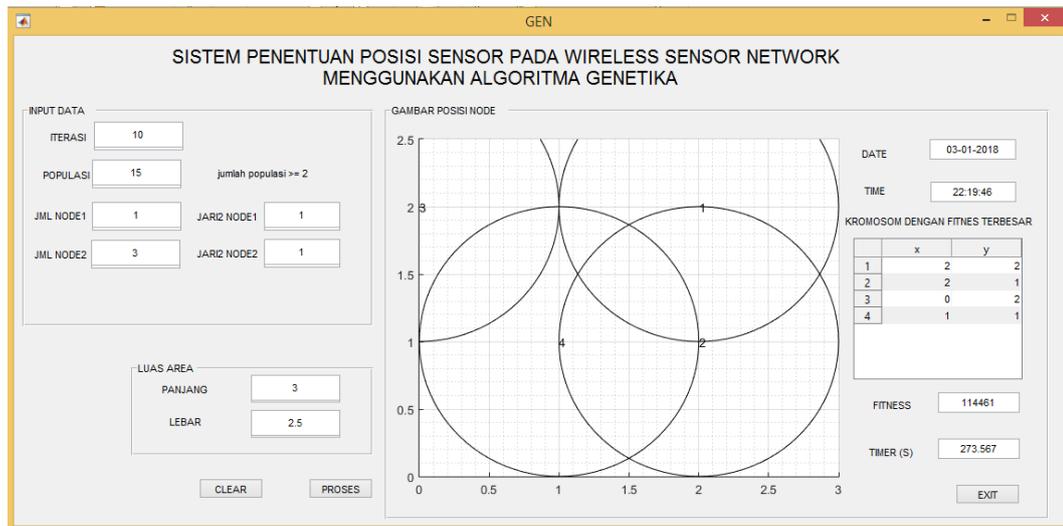
Gambar 5.16 Bagian Keluaran/Output Matlab

- Gambar ruang dengan absis dan ordinat, merupakan perumpamaan dari ruangan atau bangunan yang akan digunakan implementasi. Pada absis

merupakan panjang ruang, dan ordinat merupakan lebar ruang. Angka pada absis dan ordinat tergantung pada angka yang dimasukkan oleh pengguna pada bagian *input* luas area.

- Kolom date dan time. Pada kolom ini akan menampilkan hari dan waktu ketika kita memproses/menjalankan system.
- Tabel kromosom dengan fitness terbesar, merupakan table yang berisikan koordinat-koordinat kromosom yang terpilih menjadi individu baru dari seleksi fitness terbesar atau merupakan hasil akhir dari proses pemilihan posisi node.
- Kolom fitness, menunjukkan nilai fitness yang terpilih.
- Kolom timer (s), merupakan kolom waktu pembacaan/proses system dalam satuan second (detik).
- Tombol exit, merupakan tombol keluar system.

Pada implementasi system ini memakai 4 node sensor, 1 node sebagai koordinator dan 3 lainnya sebagai node sensor. Dengan nilai jari-jari ke empat sensor sama yaitu 1 m. jumlah iterasi sebanyak 10 kali. Populasi berjumlah 10. Luas area untuk penempatan node adalah 6.5 meter persegi dengan panjang ruang 3m dan lebar ruang 2.5m. **Gambar 5.17** merupakan hasil simulasi pada matlab:



Gambar 5.17 Hasil Simulasi Matlab

Dari hasil simulasi diatas nampak hasil berupa lingkaran pada gambar denah. Satu lingkaran memiliki jari-jari 1 meter. Terdapat 4 lingkaran yang menempati koordinat-koordinat tertentu, dengan nilai fitness sebesar 114461. Titik koordinat lingkaran dapat dilihat pada table kromosom dengan nilai fitness terbesar di sebelah kanan. Berikut merupakan koordinat-koordinat dari node:

1. Node 1 (2, 2). Nilai X terletak pada posisi 2 dan Y pada posisi 2.
2. Node 2 (2, 1). Nilai X terletak pada posisi 2 dan Y pada posisi 1.

3. Node 3 (0, 2). Nilai X terletak pada posisi 0 dan Y pada posisi 2.
4. Node 4 (1, 1). Nilai X terletak pada posisi 1 dan Y pada posisi 1.

5.2.2 Implementasi Perangkat Lunak dan Perangkat Keras Arduino

Implementasi perangkat lunak dan perangkat keras pada Arduino ditujukan untuk merancang system. Menggabungkan seluruh perangkat keras dan menghubungkan satu sama lain dengan sebuah program sehingga terhubung menjadi sebuah system. Perancangan dilakukan pada Arduino IDE.

5.2.2.1 Implementasi Sensor Api 5 Channel

Program ini menunjukkan proses untuk mengaktifkan sensor api 5 channel sehingga sensor api dapat mendeteksi keberadaan api, ketika untuk pengujian *coverage* sensor.

Tabel 5.19 Program Sensor Api 5 Channel pada Uji Coverage Sensor

No	Program
1	<code>int flame[5];</code>
2	<code>void setup() {</code>
3	<code> Serial.begin(9600);</code>
4	<code>}</code>
5	<code>void loop() {</code>
6	<code> flame[0] = analogRead(A0);</code>
7	<code> flame[1] = analogRead(A1);</code>
8	<code> flame[2] = analogRead(A2);</code>
9	<code> flame[3] = analogRead(A3);</code>
10	<code> flame[4] = analogRead(A4);</code>
11	<code> Serial.print("0 = ");</code>
12	<code> Serial.print(flame[0]);</code>
13	<code> Serial.print(" 1 = ");</code>
14	<code> Serial.print(flame[1]);</code>
15	<code> Serial.print(" 2 = ");</code>
16	<code> Serial.print(flame[2]);</code>
17	<code> Serial.print(" 3 = ");</code>
18	<code> Serial.print(flame[3]);</code>
19	<code> Serial.print(" 4 = ");</code>
20	<code> Serial.print(flame[4]);</code>
21	<code> Serial.println();</code>
22	<code> delay(1);</code>
23	<code>}</code>

Pada program diatas berisikan perintah untuk mengaktifkan sensor api 5 chanel dengan olah data menggunakan mikrokontroler Arduino. Program tersebut berisikan deklarasi variable, fungsi void setup untuk mengatur baudrate, fungsi void loop berisikan perintah-perintah untuk mendapatkan nilai analog dari sensor yang selanjutnya ditampilkan pada serial monitor.

5.2.2.2 Implementasi Node Sensor

Node sensor terdiri dari 3 node, yaitu node A, B, dan node C. **Tabel 5.20** merupakan program dari setiap node sensor:

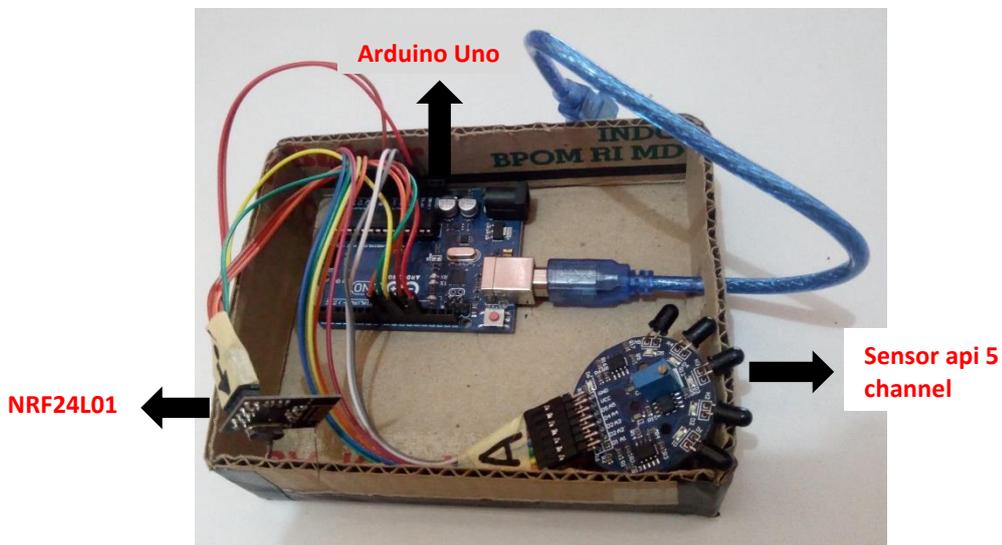
Tabel 5.20 Program Node Sensor

No	Program
1	#include <SPI.h>
2	#include <nRF24L01.h>
3	#include <RF24.h>
4	#define CE_PIN 9
5	#define CSN_PIN 10
6	const uint64_t pipe = 0xE8E8F0F0E1LL;//0xE8E8F0F0E1LL
7	RF24 radio(CE_PIN, CSN_PIN);
8	int data[6];
9	const int analogInPin0 = A0;
10	const int analogInPin1 = A1;
11	const int analogInPin2 = A2;
12	const int analogInPin3 = A3;
13	const int analogInPin4 = A4;
14	int sensorValue1, sensorValue2, sensorValue3, sensorValue4,
15	sensorValue5;
16	int out1, out2, out3, out4, out5;
17	int data1, data2;
18	void setup()
19	{
20	Serial.begin(9600);
21	Serial.println("NRF_24L01 transmitter Starting....");
22	delay(1000);
23	radio.begin();
24	radio.openWritingPipe(pipe);
25	}
26	void loop()
27	{
28	sensorValue1 = analogRead(analogInPin0);
29	sensorValue2 = analogRead(analogInPin1);
30	sensorValue3 = analogRead(analogInPin2);
31	sensorValue4 = analogRead(analogInPin3);
32	sensorValue5 = analogRead(analogInPin4);
33	out1 = map(sensorValue1, 0, 1023, 0, 255);
34	out2 = map(sensorValue2, 0, 1023, 0, 255);
35	out3 = map(sensorValue3, 0, 1023, 0, 255);
36	out4 = map(sensorValue4, 0, 1023, 0, 255);
37	out5 = map(sensorValue5, 0, 1023, 0, 255);
38	if (out1 >= 200)
39	{
40	data1 = 1;
41	}
42	else if (out2 >= 200)
43	{
44	data1 = 1;
45	}
46	else if (out3 >= 200)
47	{
48	data1 = 1;
49	}
50	else if (out4 >= 200)
	{

```
51     data1 = 1;
52     }
53     else if (out5 >= 200)
54     {
55         data1 = 1;
56     }
57     else{data1 = 0;}
58     data2 = (out1+out2+out3+out4+out5)/5;
59     data[0] = 'A';
60     data[1] = data1;
61     data[2] = data2;
62     radio.write( data, sizeof(data) );
63 }
```

Program diatas merupakan program dari node sensor A. Untuk program node sensor B dan C, maka mengganti program baris ke 59 sesuai dengan nama node sensor masing-masing. Dimulai dari proses pembacaan sensor dengan kondisi apabila nilai yang didapat sensor ≥ 200 maka akan tersimpan pada data1=1 yang berarti terdeteksi api. Jika kondisi lainnya atau dibawah 200 maka data1=0 yang berarti tidak terdeteksi. Selanjutnya proses akumulasi nilai hasil pembacaan dengan pengambilan rata-rata dari nilai yang didapat sensor. Dan terakhir proses pengiriman data ke node koordinator.

Ketika perangkat lunak dan perangkat keras telah terhubung, maka node sensor siap untuk dioperasikan. **Gambar 5.18** merupakan tampilan (*interface*) dari node sensor.



Gambar 5.18 Tampilan Node Sensor

5.2.2.3 Implementasi Node Koordinator

Node koordinator terdiri dari 1 node yaitu node ke 4. Tersusun atas beberapa perintah. **Tabel 5.21** merupakan program proses inialisasi dan deklarasi variabel.

Tabel 5.21 Inisialisasi dan Deklarasi variabel

No	Program
1	#include <SoftwareSerial.h>
2	SoftwareSerial esp(6, 7); // RX, TX
3	#define SSID "woww"
4	#define PASS "12345qwert"
5	#define IP "192.168.43.236"
6	String GET = "GET http://192.168.43.236/a/api/send.php?";
7	int tes, a, b, c;
8	String data;
9	#include <SPI.h>
10	#include <nRF24L01.h>
11	#include <RF24.h>
12	#define CE_PIN 9
13	#define CSN_PIN 10
14	const uint64_t pipe = 0xE8E8F0F0E1LL;
15	RF24 radio(CE_PIN, CSN_PIN);
16	int Data[60];
17	const int analogInPin0 = A0;
18	const int analogInPin1 = A1;
19	const int analogInPin2 = A2;
20	const int analogInPin3 = A3;
21	const int analogInPin4 = A4;
22	int sensorValue1, sensorValue2, sensorValue3, sensorValue4, sensorValue5;
23	int outputValue1, outputValue2, outputValue3, outputValue4, outputValue5;
24	String Status, status1, status2, status3;
25	String status1A, status2B, status3C;
26	int teg, teg1, teg2, teg3;
27	float tegA, teg1A, teg2B, teg3C;
28	float delay_, jitter_, throughput;
29	int panjang_paket;
30	float delay_akhir, delay_sebelumnya, lama_pengiriman, bandwidth=22.4;
41	long waktu_sekarang, waktu_lampau;

Pada tabel diatas berisikan variable-variabel yang digunakan dalam proses koordinasi. Selanjutnya juga terdapat inisialisasi modul nrf24l01 dan juga inisialisasi alamat *IP* dan *password* untuk kerja modul wifi esp8266. Dan juga inisialisasi terhadap sensor api. Selain itu berisikan deklarasi variable dengan berbagai tipe data sesuai kegunaan. Pada baris include merupakan penyertaan library dari masing-masing modul yang digunakan.

Selain proses inisialisasi dan deklarasi variable, dalam menjalankan node koordinator terdapat fungsi void setup yang merupakan fungsi yang hanya

dijalankan sekali dalam proses tersebut. **Tabel 5.22** merupakan fungsi void setup pada node koordinator.

Tabel 5.22 Fungsi Void Setup Node Koordinator

No	Program
1	void setup()
2	{
3	Serial.begin(9600);
4	Serial.println("NRF_24L01 Receiver Starting...");
5	radio.begin();
6	radio.openReadingPipe(0, pipe);
7	radio.startListening();
8	esp.begin(9600);
9	sendDebug("AT+RST");
10	delay(2000);
11	connectWiFi();
12	}

Dalam fungsi void setup berisikan instruksi-instruksi yang hanya satu kali eksekusi. Seperti pengaturan baudrate yang digunakan. Selanjutnya instruksi prosedur mulai pembacaan modul nrf24l01. Dan juga proses untuk menjalankan esp8266 yaitu pengaturan *software* serial esp8266, perintah reset esp8266, pengaturan delay, perintah terhubung ke *wifi*.

Setelah fungsi void setup pada node koordinator maka terdapat fungsi void loop. Void loop merupakan fungsi yang digunakan untuk menjalankan suatu siklus program yang dilakukan terus menerus hingga Arduino mati/reset. **Table 5.23** merupakan fungsi void loop pada node koordinator.

Tabel 5.23 Void Loop Node Koordinator

No	Program
1	void loop(){
2	get_data();
3	rumus();
4	}

Pada void loop terdapat perintah get data dan rumus. Sehingga akan menjalankan fungsi get data dan fungsi rumus. **Tabel 5.24** merupakan fungsi get data.

Tabel 5.24 Void Get Data

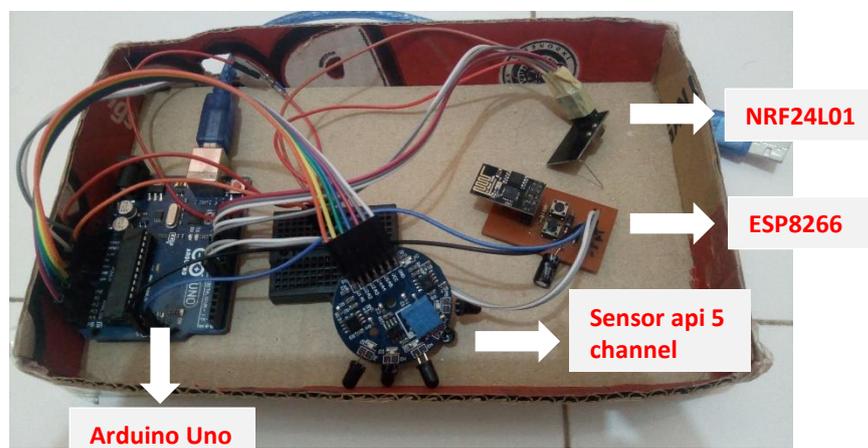
No	Program
1	void get_data()
2	{
3	sensorValue1 = analogRead(analogInPin0);
4	sensorValue2 = analogRead(analogInPin1);
5	sensorValue3 = analogRead(analogInPin2);
6	sensorValue4 = analogRead(analogInPin3);


```

9   esp.println("AT+CWMODE=1");
10  delay(1000);
11  String cmd = "AT+CWJAP=\"";
12  cmd += SSID;
13  cmd += "\",\"";
14  cmd += PASS;
15  cmd += "\"";
16  sendDebug(cmd);
17  delay(1000);
18  if (esp.find("OK"))
19  {
20    Serial.println("RECEIVED: OK");
21    return true;
22  }
23  else
24  {
25    Serial.println("RECEIVED: Error");
26    connectWiFi();
27    return false;
28  }
29  }

```

Pada void sendDebug ini berisikan proses pengaturan mengaktifkan modul esp8266, berupa perintah mengirim data dan pengaturan sambungan ke *wifi*. Perintah “AT-CWMODE=1” berarti merubah mode modul esp8266 ke mode 1 yaitu station mode. Perintah “AT-CWJAP” untuk menyambungkan ke access point. Ketika perancangan perangkat lunak dan perangkat keras telah selesai dan terhubung, maka akan menampilkan node koordinator seperti pada **gambar 5.19**.



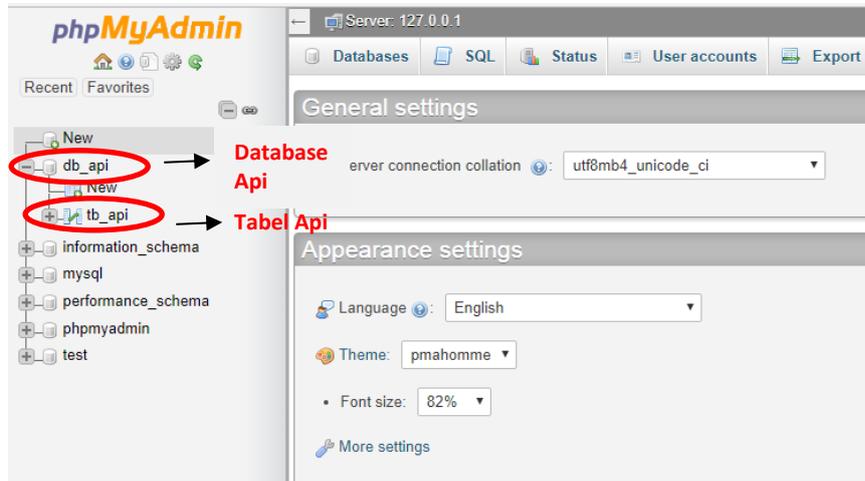
Gambar 5.19 Tampilan Node Koordinator

Node koordinator berjumlah 1 node tersusun atas mikrokontroler Arduino uno, modul nrf24l01, modul wifi esp8266, dan sensor api 5 channel.

5.2.3 Implementasi Web dan Xampp

Beberapa tahapan dalam perancangan web pada xampp. Dengan membuat database terlebih dahulu pada mysql dengan bantuan xampp. Selanjutnya proses index, proses menyambungkan dan proses mengirim data. Berikut tampilan

pembuatan database pada mysql, sebelumnya mengaktifkan apache dan mysql pada xampp. Setelah mengaktifkan apache dan mysql selanjutnya membuat database dengan nama api.



Gambar 5.20 Tampilan PhpMyAdmin

Pada gambar diatas menunjukkan bahwa telah dibuat database dengan nama “api”, yang didalamnya terdapat table api. Selanjutnya proses index. **Tabel 5.28** merupakan program index.

Tabel 5.28 Program Index

No	Program
1	<?php
2	include ("koneksi.php");
3	date_default_timezone_set('Asia/Jakarta');
4	\$date = date("d-m-y");
5	\$time = date("H:i:s");
6	?>
7	<center>
8	MENAMPILKAN DATA
9	
10	
11	<!Script untuk membuat tabel
12	<table border='2' Width='1000'>
13	<tr>
14	<th> V-1 </th>
15	<th> Api-1 </th>
16	
17	<th> V-2 </th>
18	<th> Api-2 </th>
19	
20	<th> V-3 </th>
21	<th> Api-3 </th>
22	
23	<th> V-4 </th>
24	<th> Api-4 </th>
25	<th> Waktu </th>
26	</tr>
27	<?php

26	<code>\$queri="Select * From tb_api ORDER BY waktu DESC LIMIT 20" ;</code>
27	<code>\$hasil=MySQL_query (\$queri);</code>
28	<code>echo "\t Waktu: \$time \$date";</code>
29	<code>echo "<p>";</code>
30	<code>while (\$data = mysql_fetch_array (\$hasil)){</code>
31	<code>echo "</code>
32	<code><tr></code>
33	<code><td>".\$data['v1']. "</td></code>
34	<code><td>".\$data['api1']. "</td></code>
35	<code><td>".\$data['v2']. "</td></code>
36	<code><td>".\$data['api2']. "</td></code>
37	<code><td>".\$data['v3']. "</td></code>
38	<code><td>".\$data['api3']. "</td></code>
39	<code><td>".\$data['v4']. "</td></code>
40	<code><td>".\$data['api4']. "</td></code>
41	<code><td>".\$data['waktu']. "</td></code>
42	<code></tr></code>
43	<code>";</code>
44	<code>//\$page = \$_SERVER['PHP_SELF'];</code>
45	<code>//\$sec = "5";</code>
46	<code>//header("Refresh: \$sec; url=\$page");</code>
47	<code>}</code>
48	<code>?></code>
49	<code></table></code>

Pada table inx berisikan instuksi untuk membuat tampilan awal pada web. Instruksi pembuatan table, dan instruksi untuk memasukkan data. Sebelumnya perintah untuk menyambungkan pada file koneksi.php. Baris 2 merupakan perintah untuk membuat koneksi/menyambungkan ke database. Baris 3-5 pengaturan waktu. Baris 7-10 menampilkan tulisan “menampilkan data”. Baris 11-23 perintah untuk membuat table yang nanti tampil pada web. Baris 24-27 perintah untuk menampilkan data yang telah tersimpan di database. Baris 30-49 perintah untuk membaca dan mengambil data dalam bentuk array. Setelah proses index maka selanjutnya adalah proses penyambungan/koneksi. **Tabel 5.29** merupakan program proses koneksi.

Tabel 5.29 Program Penyambungan/Koneksi Database

No	Program
1	<code><?php</code>
2	<code>error_reporting(E_ALL ^ E_DEPRECATED);</code>
3	<code>\$host="localhost";</code>
4	<code>\$user="root";</code>
5	<code>\$db="db_api";</code>
6	<code>\$pass="";</code>
7	<code>\$conn=mysql_connect (\$host, \$user, \$pass);</code>
8	<code>mysql_select_db (\$db, \$conn);</code>
9	<code>if (\$conn) {</code>
10	<code>echo "";</code>
11	<code>} else {</code>
12	<code>echo "Koneksi Gagal";</code>
13	<code>}</code>
14	<code>?></code>

Pada baris 3 diisi dengan nama host apache. Baris 4 diisi *username* apache. Baris 5 merupakan nama database. Baris 7-13 proses menyambungkan ke database. Ketika proses penyambungan ke database tidak berhasil maka muncul tulisan “Koneksi Gagal”. Selanjutnya adalah proses mengirim data. **Table 5.30** merupakan program proses pengiriman data untuk ditampilkan pada web.

Tabel 5.30 Program Pengiriman Data/Send

No	Program
1	<?php
2	include "koneksi.php";
3	date_default_timezone_set('Asia/Jakarta');
4	\$date = date("y-m-d H:i:s");
5	\$v1=\$_GET['v1'];
6	\$api1=\$_GET['api1'];
7	\$v2=\$_GET['v2'];
8	\$api2=\$_GET['api2'];
9	\$v3=\$_GET['v3'];
10	\$api3=\$_GET['api3'];
11	\$v4=\$_GET['v4'];
12	\$api4=\$_GET['api4'];
13	\$waktu=\$date;
14	\$delay=\$_GET['delay'];
15	\$jitter=\$_GET['jitter'];
16	\$throughput=\$_GET['throughput'];
17	\$query="insert into tb_api values ('','\$v1','\$api1','\$v2','\$api2','\$v3','\$api3','\$v4',' \$api4','\$waktu','\$delay','\$jitter','\$throughput)"; \$exe=mysql_query(\$query);
18	if(\$exe){
19	echo " data masuk";
20	}
21	else{
22	echo " data gagal masuk";
23	}
24	?>

Baris 2-4 memanggil file koneksi.php yang telah dibuat. File koneksi.php merupakan file penyambungan ke database. Baris 5-16 perintah untuk mengambil variable yang dikirim oleh index.php. Baris 17-24 memasukkan data pada web, yaitu pada table api.

Tahapan-tahapan implementasi dari perangkat lunak dan perangkat keras telah selesai. **Gambar 5.21** merupakan tampilan sistem keseluruhan pada web.

