

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan membahas tentang landasan kepustakaan berisi kajian pustaka dan dasar teori yang menunjang skripsi tentang pengembangan lanjut penentuan posisi node pada jaringan sensor berdasarkan coverage area menggunakan algoritma genetika. Pada kajian pustaka akan membahas tentang jurnal-jurnal, artikel, buku dan dokumen lain yang berisikan penelitian sebelumnya yang berhubungan dengan topik skripsi dan menunjukkan persamaan maupun perbedaan dengan penelitian saat ini guna untuk menunjang penulisan skripsi ini. Dalam landasan kepustakaan juga terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan metode dan teori dalam penelitian.

2.1 Kajian Pustaka

Penelitian oleh Elis (2016) berisikan tentang cara menentukan posisi node terbaik pada jaringan sensor berdasarkan coverage area sensor menggunakan algoritma genetika pada sebuah bangunan agar mendapat hasil coverage area yang maksimal. Pada sistem penentuan posisi node ini menggunakan perhitungan total luas jangkauan sensor untuk menentukan nilai yang paling optimal dengan jumlah node yang dimiliki oleh user. Sistem pada penelitian ini akan diimplementasikan menggunakan multi-parameter sebagai nilai *input* yang nantinya akan menentukan kondisi *output*. Pencarian nilai fitness pada proses algoritma genetika dilakukan dengan menghitung luas area yang ter-cover oleh sensor dikurangi dengan luas area yang tidak ter-cover oleh sensor. Luas area ruangan yang tidak tercover oleh sensor dan luas daerah yang tercover sensor diukur sistem dengan menggunakan nilai pixel. Hasil dari perhitungan genetika disimpan dalam bentuk .png. Pada penelitian ini tidak menggunakan hardware sebagai implementasi sistem, hanya simulasi pada aplikasi MATLAB.

Selanjutnya penelitian Riri Fitri (2008), tentang melakukan perbandingan hasil simulasi dengan NS-2 dan implementasi pada *Testbed* dengan PDA untuk menganalisa kinerja protokol *routing ad hoc on-demand distance vector* (AODV) pada jaringan *ad hoc on-demand*. Penelitian ini dilakukan pengkajian protokol AODV dengan membangun suatu *testbed* menggunakan *personal computer* (PC), beberapa laptop (sistem operasi Linux Red Hat 9.0 dan Fedora Core 2), serta *personal digital assitant* (PDA). Penelitian ini juga membuat *package* yang lengkap dengan cara *cross compilation* untuk PDA iPAQ. Hasil yang didapat dari analisa simulasi protokol AODV dengan menggunakan *Network Simulator* NS-2 didapatkan rata-rata *packet delivery ratio*, *end-to-end delay*, dan *routing overhead*, yang kemudian hasil pengukuran simulasi dibandingkan dengan hasil pengukuran *testbed*.

Penelitian yang dilakukan oleh Dwi (2014) tentang perancangan sebuah sistem pemantauan suhu dan kelembaban di beberapa titik area yang berbeda. Sistem ini mendukung sensor di beberapa titik, yang hasil akuisinya akan dikirimkan ke komputer secara nirkabel. Sedangkan sistem monitoring alat ini ditampilkan pada halaman web menggunakan bahasa HTML (*Hypertext Markup Language*).

Komunikasi yang digunakan dalam sistem ini adalah jaringan sensor nirkabel menggunakan nRF24L01+. Pada sistem ini menggunakan modul nRF24L01+ karena harga dari nRF24L01+ lebih murah dibanding dengan modul komunikasi lainnya, juga karena kebutuhan jangkauan komunikasi yang diterapkan pada area tidak terlalu luas.

Pada penelitian ini akan membahas tentang analisis performa jaringan sensor nirkabel yang penentuan lokasi node nya telah diimplementasikan berdasarkan algoritma genetika. Penelitian ini bertujuan untuk melakukan analisis performa jaringan sensor nirkabel, sehingga pada penelitian ini nantinya dapat memberikan informasi tingkat *quality of service* (performa) dari jaringan sensor nirkabel ketika node sensornya diletakkan sesuai dengan metode algoritma genetika, dan juga untuk mendapatkan kevaliditasan dari data. Analisis pada sistem ini dilakukan dengan melihat nilai *quality of service* (QOS) dari komunikasi data sistem. Selanjutnya juga dengan melihat pola dari hasil pengujian keakuratan *coverage area* sensor. Pengujian dilakukan berulang-ulang dengan nilai yang berbeda-beda. Dari nilai yang berbeda-beda tersebut diambil rata-rata dari tiap parameter *quality of service*.

Adapun perbedaan antara kajian pustaka yang digunakan dengan penelitian yang dilakukan, perbedaannya antara lain adalah pada penelitian yang dilakukan Elis (2016) hanya dilakukan proses simulasi dan belum melakukan implementasi untuk menguji validitas data, sedangkan pada penelitian ini dilakukan implementasi rancangan jaringan sensor nirkabel berdasarkan informasi letak posisi node dari sistem penentuan posisi node yang ada sebelumnya. Pada penelitian tersebut jari-jari node, serta luas ruangan peletakan node telah tertanam pada sistem dan tidak dapat diganti dengan ukuran lain, sehingga pada penelitian ini dibuat sistem yang lebih dinamis yaitu pengguna dapat memasukkan nilai jari-jari node dan luas ruangan tempat peletakan sesuai dengan kebutuhan.

Selanjutnya penelitian yang dilakukan oleh Dwi (2014), implementasi *wireless sensor network* menggunakan sensor suhu dan kelembabapan. Sedangkan pada penelitian ini menggunakan sensor api 5 channel. Penggunaan sensor api 5 channel ini didasarkan pada akhir-akhir ini semakin maraknya terjadi kebakaran dalam bangunan, sehingga perlu melakukan pendeteksian api. Selain itu juga sensor api 5 channel memiliki tingkat akurasi sudut yang lebih besar daripada sensor api lainnya yang terintegrasi dengan Arduino.

Penelitian yang dilakukan Riri (2008) adalah membandingkan hasil simulasi dengan hasil implementasi untuk menganalisa kinerja protokol routing AODV pada jaringan Ad Hoc Hybrid. Sedangkan pada penelitian sistem ini membandingkan hasil simulasi dengan hasil implementasi pada sistem penentuan posisi node yang sudah ada sebelumnya untuk menganalisa validitas data sistem,

2.2 Dasar Teori

Pada dasar teori berisikan teori-teori dan dasar-dasar penulis yang menjadi literatur dalam pembuatan sistem.

2.2.1 Algoritma Genetika

Algoritma genetika pertama kali diperkenalkan oleh John Holland dari Universitas Michigan pada tahun 1975. John Holland bersama murid-murid beserta rekan kerjanya mempublikasikan tulisannya berjudul “Adapted in Natural and Artificial Sistem”. Dalam tulisan tersebut dijelaskan bahwa algoritma genetika sangat cocok digunakan untuk memecahkan masalah optimasi kompleks dan juga untuk aplikasi yang membutuhkan pemecahan masalah adaptif. Sehingga dengan beberapa keunggulan tersebut, algoritma genetika diterima dalam berbagai kalangan dan telah diaplikasikan dalam berbagai bidang, hal ini diungkapkan pada penelitian oleh Elis (2016).

Penelitian yang dilakukan oleh Bina (2010), mengatakan bahwa algoritma Genetika atau *Genetic Algorithm (GA)* merupakan algoritma pencarian *heuristic* yang didasarkan atas mekanisme evolusi biologis, artinya pencarian solusi suatu masalah dengan algoritma genetik akan terus berevolusi. Inti dari algoritma genetika adalah secara bertahap mencari solusi terbaik (*survival of the fittest*) dari begitu banyak solusi yang ada. Pertama-tama algoritma genetika bekerja dengan membuat beberapa solusi acak, tentu saja dari tahapan pertama ini solusinya kemungkinan masih buruk. Solusi tersebut akan mengalami proses evolusi secara terus menerus, dan akan menghasilkan suatu solusi yang lebih baik. Setiap solusi yang terbentuk mewakili satu individu dan satu individu terdiri dari satu kromosom. Kumpulan dari individu-individu ini akan membentuk suatu populasi, dari populasi ini akan lahir populasi-populasi baru sampai dengan sejumlah generasi yang ditentukan.

Algoritma genetika banyak diaplikasikan untuk berbagai macam permasalahan, seperti: permasalahan optimasi, pemrograman otomatis, *machine learning*, model ekonomi, model sistem imunisasi, dan model ekologis. Algoritma genetika juga sangat berguna dan efisien untuk mengatasi masalah dengan karakteristik sebagai berikut: ruang masalah sangat besar, kompleks dan sulit dipahami, tidak tersedianya analisis matematika yang memadai, ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi, solusi yang diharapkan tidak harus paling optimal, tetapi cukup “bagus” atau bisa diterima, dan terdapat batasan waktu, misalnya dalam *real time sistem* atau sistem waktu nyata, hal tersebut diungkapkan pada penelitian Zuhri (2014).

2.2.1.1 Prosedur Algoritma Genetika

Hal-hal yang harus dilakukan dalam menggunakan algoritma genetika :

1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
2. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
3. Menentukan proses perkawinan silang.
4. Mutasi gen yang akan digunakan.

5. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik tidaknya sebuah individu atau baik tidaknya solusi yang didapatkan.
6. Menentukan proses seleksi yang akan digunakan.

2.2.1.2 Istilah dalam Algoritma Genetika

Penelitian yang dilakukan oleh Entin (2011) mengemukakan beberapa hal penting yang digunakan dalam perhitungan Algoritma Genetika, yaitu:

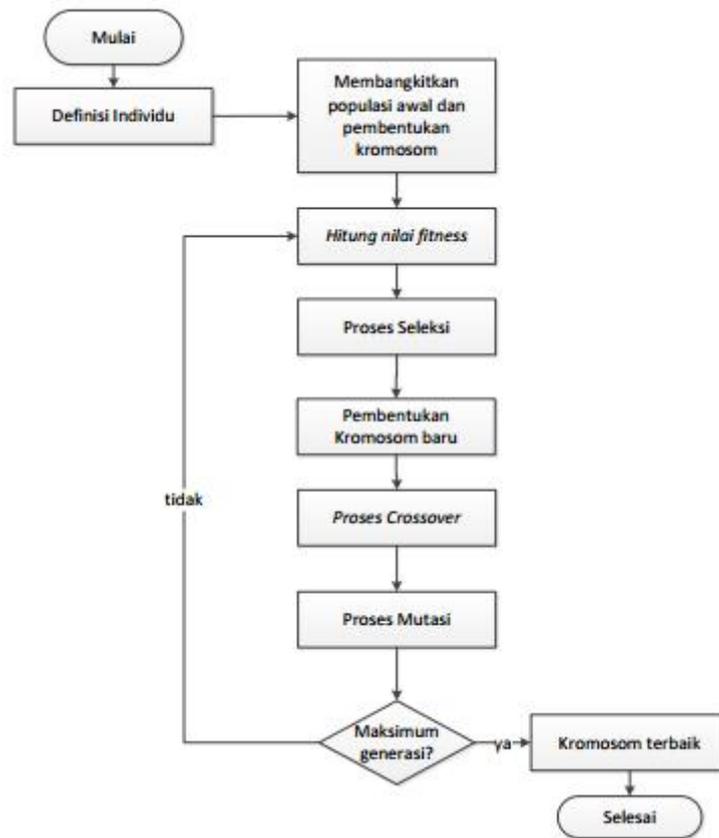
1. Individu, menyatakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat. Individu dapat dikatakan sama dengan kromosom, karena merupakan kumpulan gen. Pendefinisian individu atau disebut juga merepresentasikan kromosom yang akan diproses nanti, dilakukan dengan mendefinisikan jumlah dan tipe dari gen yang digunakan dan dapat mewakili solusi permasalahan yang diangkat.
2. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Gen dapat direpresentasikan dalam bentuk: bit, bilangan real, string, daftar aturan, gabungan dari beberapa kode, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Nilai dari gen disebut *Allele*.
3. Kromosom, merupakan gabungan gen-gen yang membentuk nilai tertentu. Kromosom merepresentasikan solusi dari permasalahan yang diangkat, sehingga apabila kromosom yang baik dihasilkan, maka diharapkan solusi yang baik dari permasalahan juga di dapatkan.
4. Populasi, adalah sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu atau kromosom secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan.
5. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma genetika.
6. Nilai Fitness, merupakan nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai fitness dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma genetika bertujuan untuk mencari individu dengan nilai fitness yang paling tinggi.
7. Proses Seleksi, operasi seleksi dilakukan dengan memperhatikan fitness dari tiap individu. Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses kawin silang dan mutasi. Seleksi digunakan untuk mendapatkan calon induk yang baik.
8. Pindah Silang (*Crossover*), merupakan operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Sebuah kromosom yang mengarah pada solusi yang baik dapat diperoleh dari proses memindah-silangkan dua buah kromosom. Pindah silang hanya bisa dilakukan dengan

suatu probabilitas *crossover*, artinya pindah silang bisa dilakukan hanya jika suatu bilangan random yang dibangkitkan kurang dari probabilitas *crossover* yang ditentukan. Pada umumnya probabilitas tersebut diset mendekati 1.

9. 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 inialisasi populasi.
10. *Offspring*, merupakan kromosom baru yang dihasilkan setelah melalui proses-proses sebelumnya. Kemudian pada *offspring* tersebut dihitung nilai fitnessnya telah optimal atau belum. Jika sudah optimal maka *offspring* tersebut merupakan solusi optimal, dan apabila *offspring* belum optimal maka akan diseleksi kembali hingga kriteria terpenuhi. Berikut kriteria berhenti yang sering digunakan :
 - a. Berhenti pada generasi tertentu.
 - b. Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah.
 - c. Berhenti bila n generasi berikut tidak didapatkan nilai fitness yang lebih tinggi.

2.2.1.3 Proses Algoritma Genetika

Algoritma genetika adalah algoritma yang didasarkan pada perkawinan seleksi gen secara alami untuk pencarian hasil yang terbaik. Kombinasi perkawinan ini dilakukan dengan proses acak (*random*). Dimana struktur gen hasil proses perkawinan akan menghasilkan sen inovatif untuk diseleksi. Dalam setiap generasi, ciptaan buatan yang baru (hasil perkawinan) diperoleh dari bit-bit dan bagian-bagian gen induk yang terbaik. Melakukan pengambilan gen induk terbaik diharapkan dapat diperoleh gen yang lebih baik lagi, meskipun kenyataannya tidak selalu tercipta gen anak yang lebih baik dari induknya. Pada penelitian yang dilakukan oleh Made (2012), penyusunan algoritma genetika memiliki beberapa tahapan proses seperti pada **Gambar 2.1**.



Gambar 2.1 Diagram Alir Proses Algoritma Genetika

Proses dalam melakukan pencarian hasil terbaik dengan menggunakan algoritma genetika dimulai dari mendefinisikan individu. Mendefinisikan individu atau merepresenstasikan kromosom yang akan diproses, dilakukan dengan mendefinisikan tipe dan jumlah dari gen yang digunakan yang mewakili solusi permasalahan yang diangkat.

Tahap selanjutnya ketika sudah melakukan definisi individu maka selanjutnya ke tahap membangkitkan populasi awal dan kromosom. Pada tahap ini membangkitkan sejumlah individu atau kromosom secara acak atau melalui prosedur tertentu, dengan ukuran populasi tergantung pada masalah yang diselesaikan dan jenis operator genetika yang diimplementasikan. Ketika sejumlah individu atau kromosom secara acak berkumpul, maka terdapat satu sekumpulan gen. dalam penelitian sistem ini, jumlah gen dapat ditentukan dengan mengalikan jumlah kromosom dengan jumlah node atau dapat ditulis seperti berikut:

$$\text{Jumlah Gen} = \text{Jumlah kromosom} * \text{Jumlah Node} \quad (2.1)$$

Selanjutnya tahap menghitung nilai fitness berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Setelah tahapan menghitung nilai fitness selesai, maka menuju tahap proses seleksi, dimana pada tahap ini melakukan operasi seleksi dengan memperhatikan nilai fitness dari tiap individu mana yang dapat digunakan untuk generasi selanjutnya. Semakin tinggi nilai fitnessnya maka

semakin besar kemungkinan individu tersebut terpilih. Setelah tahap seleksi maka terbentuklah kromosom baru.

Pada penelitian yang dilakukan Chandra (2015) mengatakan bahwa, apabila tahap pembentukan kromosom baru telah terbentuk, maka tahap selanjutnya adalah melakukan proses pindah silang (*crossover*), dimana tahap ini 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*). Pada sistem ini perhitungan *crossover* menggunakan *crossover aritmatika*. Perhitungan ini menggunakan teknik *crossover aritmatika*. *Crossover aritmatika* menghasilkan keturunan dengan melakukan kombinasi linear dari dua kromosom induk. Berikut persamaan dari *crossover aritmatika*:

$$\left. \begin{array}{l} O1 = \lambda1. x1 + \lambda2. x2 \\ O2 = \lambda1. x2 + \lambda2. x1 \end{array} \right\} \lambda1 + \lambda2 = 1 \quad (2.2)$$

$\lambda1$ dan $\lambda2$ merupakan bilangan random.

Penelitian tersebut juga mengungkapkan bahwa, setelah melakukan proses *crossover* maka melakukan proses mutasi. Pada tahap ini terjadi proses perubahan nilai dari satu atau beberapa gen dalam suatu kromosom. Proses inilah yang berperan menggantikan gen yang hilang dari populasi akibat seleksi yang memungkinkan muncul kembali gen yang tidak muncul pada inisialisasi populasi. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation_rate*

Setelah tahapan-tahapan diatas selesai dilakukan, apakah telah terbentuk generasi baru yang maksimum atau belum maksimum. Apabila telah terbentuk generasi yang maksimum berarti telah terbentuk kromosom yang terbaik. Sedangkan apabila generasi belum maksimum maka ulangi tahapan-tahapan sebelumnya dari menghitung nilai fitness dan seterusnya sampai terpenuhi kriteria berhenti dan terbentuk kromosom yang terbaik.

2.2.2 Koordinat

Menurut Kamus Besar Bahasa Indonesia (KBBI) (2017), koordinat merupakan bilangan yang dipakai untuk menunjukkan lokasi suatu titik pada garis, permukaan, atau pada ruang. Dalam koordinat, menentukan tiap titik dalam bidang yaitu dengan menggunakan dua bilangan yang biasa disebut koordinat x (absis) dan koordinat y (ordinat) dari titik tersebut. Koordinat adalah suatu titik yang didapatkan dari hasil perpotongan dari garis latitude (lintang) dengan garis bujur (longitude) sehingga akan menunjukkan lokasi pada suatu daerah.

2.2.3 Quality of Service (QoS)

Menurut Tiphon (1999) *Quality of Service* merupakan sekumpulan parameter yang menunjukkan kualitas jaringan yang memungkinkan sebuah aplikasi

beroperasi sesuai yang diharapkan. QoS bertujuan untuk memberikan layanan yang lebih baik pada sebuah aplikasi dengan meningkatkan kualitas pada suatu jaringan. Kualitas layanan pada sebuah aplikasi pada jaringan dapat diketahui dari beberapa parameter untuk menentukannya yaitu *delay*, *jitter*, dan *throughput*.

1. Delay

Delay merupakan lamanya waktu atau keterlambatan didalam proses transmisi paket data ke penerima. Factor yang mempengaruhi delay transmisi adalah jumlah paket data yang dikirim (bit) dan laju kecepatan pengiriman paket data untuk setiap detik.

$$\text{Delay transmisi} = \frac{\text{panjang paket (byte)}}{\text{bandwidth (Kbps)}} \quad (2.3)$$

Menurut Tiphon kategori delay adalah sebagai berikut:

Tabel 2.1 Kategori Delay

Kategori	Delay	Index
Sangat Bagus	<150ms	4
Bagus	150 ms s/d 300ms	3
Sedang	300ms s/d 450 ms	2
Jelek	>450 ms	1

2. Jitter

Menurut Angie (2014) Jitter merupakan variasi delay antar paket yang terjadi pada jaringan *Internet Protocol* (IP). Semakin besar beban trafik yang ada pada jaringan, maka peluang terjadinya tumbukan antar paket/perlambatan (*congestion*) semakin besar, dengan demikian nilai jitter akan semakin besar. Semakin besar nilai jitter maka nilai QoS akan semakin turun dan apabila semakin kecil nilai jitter maka nilai QoS akan semakin baik.

$$\text{Jitter} = \frac{\text{total variasi delay}}{\text{total paket yang diterima}-1} \quad (2.4)$$

$$\text{Variasi delay} = \text{delay akhir} - \text{delay sebelumnya} \quad (2.5)$$

Menurut Tiphon (2002), kategori jitter adalah:

Tabel 2.2 Kategori Jitter

Kategori	Jitter	Index
Sangat Bagus	0 ms	4
Bagus	0 ms s/d 75 ms	3
Sedang	75 ms s/d 125 ms	2
Jelek	125 ms s/d 225 ms	1

3. Throughput

Throughput menunjukkan kecepatan rata-rata yang diterima dalam selang waktu pengamatan tertentu, throughput disebut juga sebagai bandwidth yang sebenarnya. Pengukuran throughput dilakukan dalam satuan *bits per second* (bps).

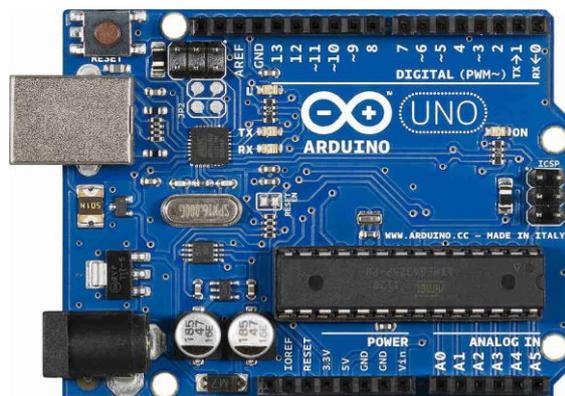
$$\text{Throughput} = \frac{\text{total data yang diterima (bits)}}{\text{waktu terima-waktu kirim(second)}} \quad (2.6)$$

Tabel 2.3 Kategori Throughput

Kategori	Throughput	Index
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	<25	1

2.2.4 Mikrokontroler Arduino Uno

Arduino Uno merupakan papan mikrokontroler berbasis Atmega328P bersifat *open source*. Arduino Uno memiliki 14 digital pin *input/ output*, 6 pin dapat digunakan sebagai output PWM, terdapat 6 input analog, kristal kuarsa 16 MHz. Terdapat juga USB untuk terhubung ke komputer, jack listrik, header ICSP dan tombol reset. Secara keseluruhan Arduino memiliki 20 pin I/O, yaitu 6 pin input analog dan 14 pin digital *input/output*. 6 pin analog pada arduino dapat digunakan sebagai output digital jika diperlukan, dengan mengubah konfigurasi pin pada program. Bahasa pemrograman pada Arduino menggunakan bahasa C yang sudah disederhanakan *code* bahasa pemrogramannya, hal ini dikutip berdasarkan web Arduino (2017). **Gambar 2.2** merupakan tampilan dari Arduino Uno.



Gambar 2.2 Tampilan Arduino Uno

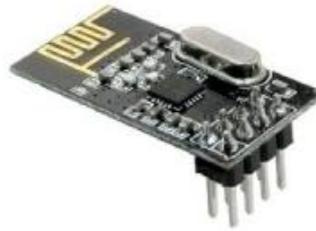
Tabel 2.4 merupakan tabel spesifikasi teknis dari Arduino Uno.

Tabel 2.4 Spesifikasi Teknis Arduino Uno

Mikrokontroler	Atmega328P
Tegangan Operasi	5 V
Input Voltage (dianjurkan)	7-12V
Input Voltage (batas)	6-20V
Digital I/O Pins	14 (yang mana 6 memberikan <i>output</i> PWM)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (Atmega328P) yang mana 0.5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (Atmega328P)
EEPROM	1 KB (Atmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25

2.2.5 Modul nRF24L01

Modul nRF24L01 adalah sebuah modul transceiver yang memanfaatkan gelombang RF 2.4GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan antarmuka SPI (*Serial Peripheral Interface*) untuk komunikasinya. Modul ini mengintegrasikan pengirim lengkap 2.4GHz RF, RF pengumpul dan akselerator protokol berupa *Enhanced Shockburst* yang mendukung antarmuka SPI kecepatan tinggi untuk kontroler aplikasi. nRF24L01 memiliki daya ultra rendah yang memungkinkan daya tahan baterai berbulan-bulan, hal tersebut dikemukakan pada penelitian oleh Dwi (2014). **Gambar 2.3** merupakan tampilan dari nRF24L01+.



Gambar 2.3 Tampilan Modul nRF24L01

Tabel 2.5 adalah pin-pin beserta fungsi dari nRF24L01.

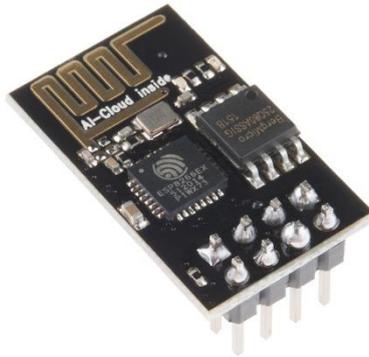
Tabel 2.5 Fungsi pin-pin nRF24L01

Pin	Name	Fungsi Pin	Keterangan
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin. Active low
7	XC2	Analog Output	Crystal Pin 2
8	XC1	Analog Input	Crystal Pin 1
9	VDD_PA	Power Output	Power Supply Output (+1.8V) for the internal nRF24L01+ Power Amplifier. Must be connected to ANT1 and ANT2
10	ANT1	RF	Antenna interface 1
11	ANT2	RF	Antenna interface 2
12	IREF	Analog Input	Reference current. Connect a 22k Ω resistor to ground
13	VSS	Power	Ground (0V)
14	VDD	Power	Power Supply (+1.9V - +3.6V DC)
15	DVDD	Power Output	Internal digital supply output for de-coupling purposes

Sumber: (Semiconductor, 2017)

2.2.6 Modul WIFI ESP8266

Modul ESP8266 merupakan SOC mandiri terintegrasi dengan TCP/IP stack protokol yang dapat memberikan akses mikrokontroler ke jaringan WIFI. ESP8266 mampu dengan baik melakukan pembongkaran semua fungsi jaringan Wi-Fi dari prosesor aplikasi lain. ESP8266 diprogram dengan perintah AT set firmware atau hanya dapat menghubungkan hal ini ke perangkat Arduino. Modul WIFI ESP8266 ini adalah board dengan biaya efektif yang sedang berkembang. **Gambar 2.4** merupakan tampilan dari modul WIFI ESP8266.



Gambar 2.4 Tampilan Modul WIFI ESP8266

Espressif telah merilis *Software Development Kit (SDK)* baru, sehingga memungkinkan banyak developer melakukan pengembangan pada modul ini. Saat ini esp8266 dapat diprogram menggunakan compiler C dan SDK dari Espressif, bahasa *wiring* menggunakan arduino IDE, BASIC menggunakan ESP8266 Basic, python menggunakan microPhyton dan bahasa LUA menggunakan NodeMCU. Spesifikasi pada modul WIFI ESP8266 adalah sebagai berikut:

1. 802.11 b/g/n
2. Integrated low power 32-bit MCU
3. Terintegrasi 10 bit ADC
4. Terintegrasi TCP/IP protocol stack
5. Integrated TR switch, balun, LNA, power amplifier and matching network
6. Integrated PLL, regulators, and power management units
7. Support STA/AP/STA+AP operation modes
8. Support Smart Link Function for both Android and ios devices
9. Support antenna diversity
10. WIFI 2.4 GHz, support WPA/WPA2
11. SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
12. STBC, 1X1 MIMO, 2X1 MIMO
13. A-MPDU & A-MSDU aggregation & 0.4s guard interval

14. Deep sleep power <10uA, Power down leakage current <5uA
15. Wake up and transmit packets in < 2ms
16. Standby power consumption of < 1.0mW (DTIM3)

2.2.7 Sensor Api 5 Channel

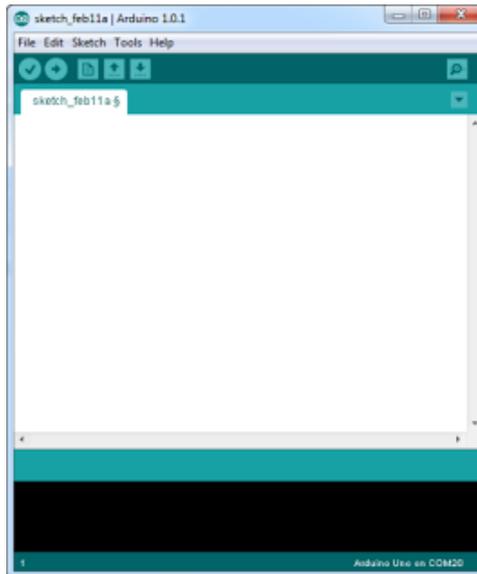
Sensor merupakan komponen atau alat elektronika untuk mengubah dari besaran fisis ke besaran listrik dapat berupa tegangan, arus, dan resistansi. Sensor Api merupakan komponen atau alat untuk mendeteksi keberadaan api. Berbagai macam alat alarm kebakaran memanfaatkan sensor api sebagai pendeteksi api. Sensor api memiliki beberapa jenis salah satunya adalah sensor api 5 channel. Pada sensor api 5 *channel* ini memanfaatkan 5 buah IR *receiver* yang memiliki kepekaan terhadap gelombang inframerah (IR). Rentang dari sensor ini adalah 700-1100nm yang berasal dari sumber api. **Gambar 2.5** merupakan tampilan sensor api 5 *channel*:



Gambar 2.5 Sensor Api 5 *Channel*

2.2.8 Arduino IDE

Menurut Agung (2014) *integrated Development Environment* (IDE) adalah lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Arduino IDE merupakan lingkungan *open source* arduino yang memudahkan untuk menulis kode dan meng-*upload* ke *board* arduino. IDE dapat dijalankan pada banya sistem yang berbeda. Arduino IDE terdiri dari editor, compiler, loader, dan monitor serial. **Gambar 2.6** adalah antarmuka Arduino IDE.



Gambar 2.6 Antarmuka Arduino IDE

2.2.9 Matlab

Matlab merupakan singkatan dari Matrix Laboratory. Dikenalkan pertama kali pada tahun 1970 oleh University of Mexico dan University of Stanford. Matlab adalah bahasa tingkat tinggi dan interaktif yang memungkinkan melakukan komputasi secara intensif. Matlab dikembangkan oleh MathWorks khusus untuk komputasi numerik, pemrograman, dan visualisasi. Matlab dapat digunakan untuk analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Fungsi-fungsi, *tools built-in*, dan bahasa yang dibuat memudahkan pengguna untuk mendalami berbagai penelitian. Matlab menggunakan konsep array/matrik sebagai standar variabel elemennya tanpa harus ada pendeklarasian array seperti dalam bahasa lainnya. Selain itu juga matlab dapat dikolaborasikan dengan aplikasi dan bahasa pemrograman eksternal seperti C, Java, .Net, dan Microsoft® Excel®. Menurut Kristalina (2015) Berikut merupakan tabel beberapa penjelasan karakter pada Matlab:

Tabel 2.6 Definisi Karakter Matlab

Karakter	Definisi
+	Penjumlahan (skalar, vektor, matriks)
-	Pengurangan (skalar, vektor, matriks)
*	Perkalian (skalar, vektor, matriks)
/	Pembagian (skalar)
^	Eksponensial (skalar, matriks kuadrat)
.*	Perkalian element by element
.^	Eksponensial element by element

;	Menyembunyikan hasil <i>output</i> proses
:	Menyatakan nilai kenaikan
'	Transpose Konyugasi (transpose untuk vektor real, matriks)
...	Lanjutan <i>command</i> (jika satu baris tidak cukup)
%	<i>Command</i> yang tidak diproses
==	Logika untuk kesamaan
	Logika untuk OR
&	Logika untuk AND
~ =	Logika untuk NOT

Pada matlab, selain fungsi-fungsi diatas terdapat pula fungsi trigonometri. Seperti fungsi **sqrt()** untuk akar kuadrat, **exp()** untuk fungsi eksponensial dan **abs()** untuk nilai absolut. Ketika sebuah fungsi telah diaplikasikan pada sebuah matriks, maka pada setiap elemen matriks tersebut selalu menggunakan fungsi yang berlaku tersebut. Berikut beberapa contoh fungsi dan simbol yang dipakai pada matlab.

Tabel 2.7 Fungsi dan Simbol Matlab

Fungsi	Keterangan
Phi	Π
i	$\sqrt{-1}$
x=zeros(N,1)	x adalah sebuah matriks vektor Nx1 berisi bilangan "0" semua
x=ones(N,1)	x adalah sebuah matriks vektor Nx1 berisi bilangan "1" semua
x=eye(N,N)	x adalah matriks identitas dengan ukuran NxN
x=rand(N,1)	Pembangkitan bilangan random uniform dalam matriks Nx1
x=randn(N,1)	Pembangkitan bilangan random Gaussian dalam matriks Nx1
rand('state',0)	Menginisialisasi pembangkit bilangan random Uniform
randn('state',0)	Menginisialisasi pembangkit bilangan random Gaussian
M=length(x)	Nilai M sama dengan panjang baris dari matriks x (mis: <i>m</i>)
M=size(x)	Nilai M sama dengan ukuran dari matriks x (mis: <i>m*n</i>)
Whos	Menampilkan semua variabel dan atributnya pada workspace

Help	Menyediakan menu pertolongan tentang sebuah command, mis: <i>help sqrt</i>
------	--

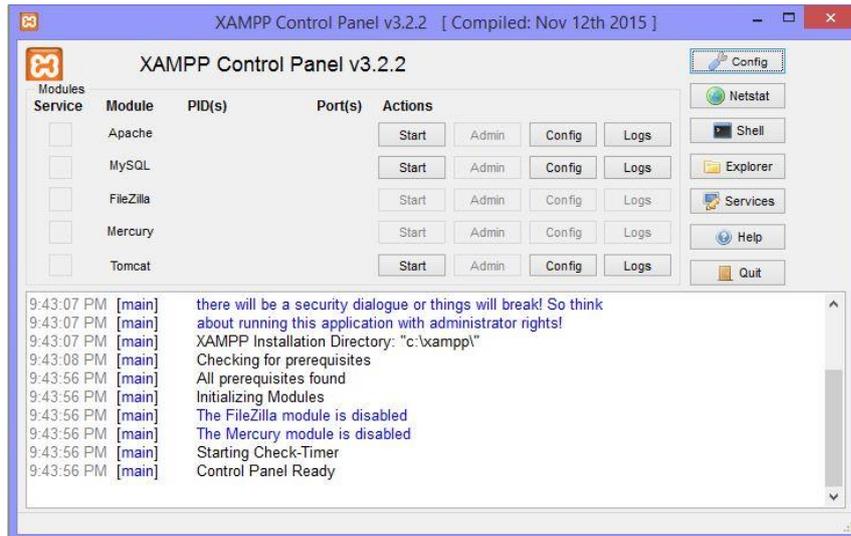
Selain fungsi-fungsi tersebut diatas Matlab juga dapat digunakan untuk menyelesaikan permasalahan statistika. Beberapa simbol atau fungsi untuk menyelesaikan permasalahan statistika telah disediakan. Berikut adalah beberapa fungsi statistika tersebut:

Tabel 2.8 Fungsi Statistika Pada Matlab

Fungsi	Keterangan
Mean(X)	Mendapatkan nilai rata-rata dari sebuah himpunan elemen X
Var(X,0,1)	Mendapatkan nilai ragam (varians) setiap kolom dari sebuah himpunan elemen X (1 merujuk kolom, 2 merujuk baris)
Std(X,0,1)	Mendapatkan nilai simpangan baku (standard deviasi) setiap kolom dari sebuah himpunan elemen X (1 merujuk kolom, 2 merujuk baris), dimana $std(X)=\sqrt{var(X)}$
Median(X,1)	Mendapatkan nilai tengah dari setiap kolom himpunan elemen X(1 merujuk kolom, 2 merujuk baris)
Unique(X)	Mendapatkan nilai yang paling banyak keluar (modus) dari himpunan elemen X
Cov(X,Y)	Mendapatkan hubungan varians dari N buah variabel X terhadap N buah variabel Y
Corrcoef(X,Y)	Mendapatkan koefisien kekuatan hubungan antara N buah variabel X dan N buah variabel Y
Min(X)	Mendapatkan elemen terkecil dari setiap kolom himpunan X
Max(X)	Mendapatkan elemen terbesar dari setiap kolom himpunan X

2.2.10 XAMPP

Menurut Riyanto (2010:1) XAMPP merupakan paket PHP dan MySQL berbasis *open source* yang digunakan sebagai alat (*tool*) yang membantu dalam pengembangan aplikasi berbasis PHP. XAMPP terdiri dari kombinasi beberapa paket perangkat lunak berbeda tersusun dalam satu paket. Gambar 4.7 merupakan tampilan control panel dari XAMPP.



Gambar 2.7 Control Panel XAMPP