BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian terdahulu sangat penting sebagai dasar pijakan dalam rangka penyusunan penelitian ini. Penelitian terdahulu adalah upaya peneliti untuk mencari perbandingan dan selanjutnya untuk menemukan inspirasi baru untuk penelitian selanjutnya di samping itu kajian terdahulu membantu penelitian dalam memposisikan penelitian serta menunjukkan orsinalitas dari penelitian.

Pada bagian ini peneliti mencantumkan berbagai hasil penelitian terdahulu yang terkait dengan penelitian yang hendak dilakukan, kemudian membuat ringkasannya, baik penelitian yang sudah terpublikasikan atau belum terpublikasikan (skripsi, tesis, disertasi dan sebagainya). Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

1. Asma Yanziah, Sopian Soim, Martinus Mujur Rose, 2020 dalam skripsinya yang berjudul **"**ANALISIS JARAK JANGKAUAN LORA DENGAN PARAMETER RSSI DAN *PACKET LOSS* PADA AREA URBAN**".** Penelitian ini berfokus pada pengujian *range chip* LoRa Rfm 95/96 pada frekuensi 920 MHz dan SF7 untuk area perkotaan. Parameter yang digunakan adalah *Received Signal Strength Indicator* (RSSI) dan *Packet loss*.
2. Rakhmadhany Primananda, 2019 dalam skripsinya yang berjudul **"**ANALISIS KINERJA LORA SX1278 MENGGUNAKAN TOPOLOGI STAR BERDASARKAN JARAK DAN BESAR DATA PADA WSN**".** Penelitian ini membahas tentang Bagaimana cara melakukan analisis kinerja dari modul LoRa dengan melihat hasil dari *Delay*, RSSI, *Throughput* dan nilai SNR yang dihasilkan saat transmisi data berlangsung.

2.2. Komunikasi LoRa

LoRa adalah kependekan dari *long range* adalah teknik modulasi *spread spectrum* yang berasal dari teknologi *chirp spread spectrum* (CSS). LoRa Semtech adalah platform nirkabel jarak jauh berdaya rendah yang telah menjadi platform nirkabel *de facto* Internet of Things (IoT). Perangkat dan jaringan LoRa seperti

LoRaWAN® memungkinkan aplikasi IoT pintar yang memecahkan beberapa tantangan saat ini, seperti manajemen energi, pengurangan sumber daya alam, pengendalian polusi, efisiensi infrastruktur, dan pencegahan bencana. Perangkat LoRa Semtech telah mengumpulkan beberapa ratus kasus penggunaan yang diketahui untuk kota pintar, rumah dan bangunan, komunitas, pengukuran, rantai pasokan dan logistik, pertanian, dan banyak lagi.

LoRa adalah standar nirkabel berdaya rendah 'Jarak Jauh' yang ditujukan untuk menyediakan jaringan komunikasi kecepatan data rendah gaya seluler. LoRa dikembangkan dengan tujuan untuk meraih pasar M2M dan IoT. LoRa sangat ideal untuk menyediakan konektivitas kecepatan data rendah yang terputus-putus pada jarak yang sangat jauh. Antarmuka radio telah dirancang untuk memungkinkan tingkat daya sinyal yang sangat rendah untuk diterima, dan sebagai hasilnya transmisi daya yang rendah pun dapat diterima pada rentang yang sangat jauh.

Modulasi LoRa dan antarmuka radio telah dirancang dan dioptimalkan untuk menyediakan jenis komunikasi yang tepat yang dibutuhkan untuk node IoT dan M2M jarak jauh.

Beberapa elemen kunci dari teknologi LoRa adalah sebagai berikut:

- Jarak jauh: 15 - 20 km.
- Jutaan node
- Masa pakai baterai yang lama: lebih dari sepuluh tahun

Terdapat berbagai elemen pada teknologi LoRa yang menyediakan fungsionalitas dan konektivitas keseluruhan untuk sistem, diantaranya adalah:

- Antarmuka LoRa PHY / RF

  Lapisan fisik LoRa atau PHY adalah kunci untuk pengoperasian sistem. Ini mengatur aspek sinyal RF yang ditransmisikan antara node atau titik akhir, yaitu sensor dan gateway LoRa tempat sinyal diterima. Lapisan fisik atau antarmuka radio mengatur aspek sinyal termasuk frekuensi, format modulasi, tingkat daya, pensinyalan antara elemen pemancar dan penerima, dan topik terkait lainnya.

- LoRa Protocol Stac:

  Selain lapisan fisik LoRa, Aliansi LoRa juga telah menetapkan tumpukan protokol terbuka (*Protocol Stack*). Penciptaan tumpukan sumber terbuka telah

memungkinkan konsep LoRa berkembang karena semua jenis perusahaan yang berbeda yang terlibat dalam pengembangan, penggunaan, dan penyebaran LoRa telah mampu bersatu untuk menciptakan solusi konektivitas yang mudah digunakan dan berbiaya rendah untuk semua cara perangkat IoT yang terhubung.

- Arsitektur jaringan LoRa (LoRaWAN)

  Selain elemen RF dari sistem nirkabel LoRa, ada elemen lain dari arsitektur jaringan, termasuk arsitektur sistem secara keseluruhan, backhaul, server, dan komputer aplikasi. Arsitektur keseluruhan sering disebut sebagai LoRaWAN.

2.2.1. Aplikasi Khusus LoRa

Teknologi nirkabel LoRa ditempatkan secara ideal untuk digunakan dalam berbagai macam aplikasi.

- Daya rendah dan kemampuan jarak jauh berarti bahwa titik akhir dapat digunakan di berbagai tempat, di gedung dan di luar dan masih memiliki kemampuan untuk dapat berkomunikasi dengan *gateway*
- Dengan demikian, sistem ini mudah diterapkan dan dapat digunakan untuk sejumlah besar Internet of Things, IoT, dan aplikasi mesin ke mesin, M2M.
- Aplikasi untuk teknologi nirkabel LoRa meliputi: pengukuran cerdas; pelacakan inventaris, data dan pemantauan mesin penjual otomatis; Industri otomotif; aplikasi khusus, bahkan di mana pun di mana pelaporan dan kontrol data mungkin diperlukan.
- Teknologi LoRa sangat menarik untuk banyak aplikasi karena kemampuan jangka panjangnya. Node baru dapat dengan mudah dihubungkan dan diaktifkan dan cakupannya mudah disediakan.

2.2.2. Pita Frekuensi LoRa

Sistem nirkabel LoRa memanfaatkan frekuensi tanpa izin yang tersedia di seluruh dunia. Frekuensi/pita yang paling banyak digunakan adalah:

- Eropa : 868 MHz
- Amerika Utara : 915 MHz
- Asia : 433 MHz

Menggunakan frekuensi yang lebih rendah daripada pita ISM 2,4 atau 5,8 GHz memungkinkan cakupan yang jauh lebih baik untuk dicapai oleh modul dan perangkat nirkabel LoRa, terutama saat node berada di dalam gedung. Meskipun pita ISM sub-1GHz biasanya digunakan, teknologi ini pada dasarnya adalah agnostik frekuensi dan dapat digunakan pada sebagian besar frekuensi tanpa penyesuaian mendasar.

2.2.3. Modulasi LoRa

Lapisan fisik RF LoRa menggunakan bentuk modulasi spektrum tersebar (*spread spectrum*). Skema modulasi LoRa menggunakan pulsa termodulasi frekuensi linier pita lebar. Tingkat kenaikan atau penurunan frekuensi dari waktu ke waktu digunakan untuk mengkodekan data yang akan dikirimkan, yaitu suatu bentuk modulasi kicauan (*chirp modulation*).

Bentuk modulasi ini memungkinkan sistem nirkabel LoRa mendemodulasi sinyal yang 20dB di bawah tingkat kebisingan saat demodulasi digabungkan dengan koreksi kesalahan maju, FEC. Ini berarti bahwa link budget untuk sistem LoRa dapat memberikan peningkatan lebih dari 25dB jika dibandingkan dengan sistem FSK.

Sebagai hasil dari fakta bahwa transmisi menyebar secara pseudo-acak, itu tampak seperti kebisingan dan mungkin sulit untuk dideteksi oleh pengguna non-Lora. Hal ini dapat membantu dalam keamanan sistem. Keuntungan lebih lanjut dari sistem ini adalah bahwa modulasi kicauan, dan sistem secara umum toleran terhadap offset frekuensi dan sebagai hasilnya dimungkinkan untuk menggunakan osilator kristal dasar dengan toleransi 20 - 30 ppm daripada osilator kompensasi suhu, TCXO. Ini dapat memberikan penghematan pantai yang baik dalam sirkuit elektronik.

2.2.4. Tautan Adaptif LoRa (*LoRa Adaptive Link*)

Fakta bahwa hanya kecepatan data rendah yang digunakan, dan tingkat transfer data keseluruhan yang rendah berarti diperlukan bandwidth yang rendah. Berbagai bandwidth yang tersedia. Yang utama adalah: 7,8 kHz; 10,4 kHz; 15,6 kHz; 20,8 kHz; 31,2 kHz; 41,7 kHz; 62,5 kHz; 125 kHz; 250 kHz; 500 kHz, meskipun tiga teratas cenderung menjadi yang paling umum digunakan.

Selain berbagai bandwidth yang tersedia, LoRa menggunakan enam faktor penyebaran, yang ditunjuk SF 7 hingga SF 12 untuk menyesuaikan kecepatan data dan rentang *trade-off*. Faktor penyebaran yang lebih tinggi memungkinkan jangkauan yang lebih jauh dengan mengorbankan kecepatan data yang lebih rendah, dan sebaliknya.

Kombinasi *bandwidth* dan faktor penyebaran dapat dipilih sesuai dengan kondisi link dan tingkat data yang akan dikirimkan. Faktor penyebaran yang lebih tinggi meningkatkan kinerja transmisi untuk *bandwidth* tertentu, tetapi juga meningkatkan waktu transmisi. Faktor penyebaran yang lebih panjang memberikan sensitivitas yang lebih tinggi tetapi waktu transmisi yang lebih lama dan oleh karena itu kecepatan data yang lebih rendah. Ini dapat bervariasi dari sedikitnya 18bps hingga 40Kbps.

Dimungkinkan juga untuk meningkatkan kekebalan kebisingan dengan menggunakan koreksi kesalahan maju. Namun, ini mengurangi throughput data aktual karena koreksi kesalahan itu sendiri menambahkan data koreksi tambahan untuk memungkinkan penerima memulihkan pesan yang sebenarnya jika ada kesalahan.

Tingkat daya yang digunakan dalam lapisan fisik RF nirkabel LoRa bersifat adaptif. Tingkat daya yang digunakan tergantung pada kecepatan data yang diperlukan, kondisi tautan, dll. Sebuah algoritma digunakan untuk menentukan tingkat daya yang diperlukan - daya yang ditransmisikan biasanya dicadangkan sedikit dari maksimum yang diperlukan untuk mendukung komunikasi yang cepat dan dengan cara ini baterai hidup dimaksimalkan dan kapasitas jaringan dipertahankan.

Lapisan fisik/antarmuka radio untuk sistem nirkabel LoRa menyediakan format yang sangat baik untuk komunikasi sinyal rendah daya rendah. Ini membuatnya ideal untuk kecepatan data rendah, daya rendah, dan komunikasi jarak jauh. Antarmuka RF memungkinkan node LoRa ditempatkan di banyak lokasi yang buruk dan tetap mempertahankan konektivitas.

2.2.5. Arsitektur Jaringan LoRa-LoRaWAN

Selain tautan nirkabel, LoRa memiliki arsitektur dan protokol jaringan dasar yang memungkinkan komunikasi simpul akhir untuk dirutekan ke tujuan yang diperlukan. Jaringan LoRa menggunakan jaringan telekomunikasi yang disebut LoRaWAN yang menyediakan perutean data dari node akhir melalui gateway LoRaWAN ke entitas yang diperlukan. LoRaWAN juga membatasi cara data dikirim ke seluruh jaringan, merinci respons *gateway* LoRaWAN, dan *server* jaringan LoRa. Seperti halnya jaringan seluler yang memiliki jaringan kabel atau jaringan inti, begitu pula LoRa yang sering disebut dengan LoRaWAN.

Jaringan LoRa terdiri dari beberapa elemen:

- Titik akhir (*end node*)

  End Node adalah elemen jaringan LoRa tempat penginderaan atau kontrol dilakukan. Mereka biasanya terletak jauh.

- Gateway LoRa

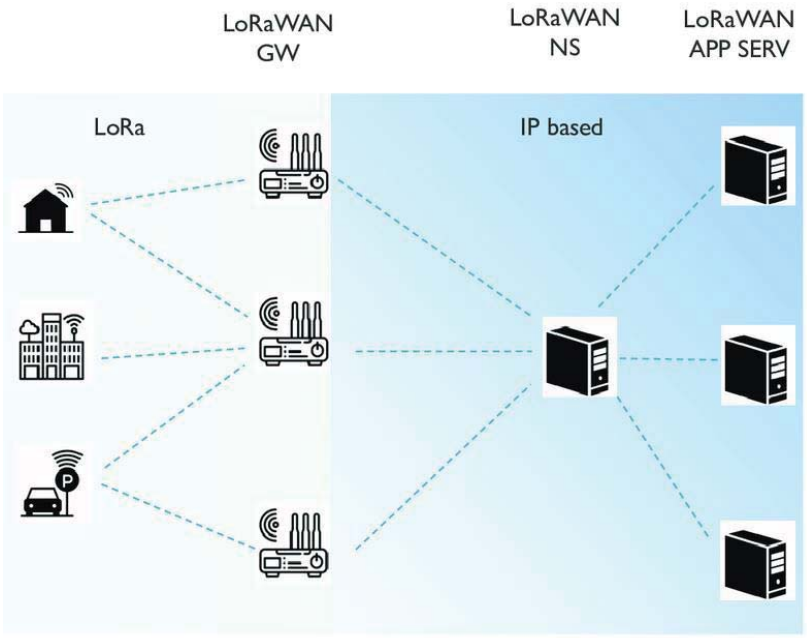
  Gateway menerima komunikasi dari titik akhir LoRa dan kemudian mentransfernya ke sistem backhaul. Bagian dari jaringan LoRa ini dapat berupa Ethernet, seluler, atau tautan telekomunikasi lainnya yang berkabel atau nirkabel. *Gateway* terhubung ke *server* jaringan menggunakan koneksi IP standar. Dengan cara ini data menggunakan protokol standar, tetapi dapat dihubungkan ke jaringan telekomunikasi apa pun, baik publik maupun pribadi. Mengingat kesamaan jaringan LoRa dengan jaringan seluler, *gateway* LoRaWAN mungkin sering ditempatkan bersama dengan stasiun pangkalan seluler. Dengan cara ini mereka dapat menggunakan kapasitas cadangan pada jaringan *backhaul*.

- Server Jaringan LoRa

  Server jaringan LoRa mengelola jaringan dan sebagai bagian dari fungsinya, ia bertindak untuk menghilangkan paket duplikat, menjadwalkan pengakuan, dan menyesuaikan kecepatan data. Mengingat cara penyebaran dan koneksinya, sangat mudah untuk menyebarkan jaringan LoRa.

- Komputer jarak jauh

  Komputer jarak jauh kemudian dapat mengontrol tindakan titik akhir atau mengumpulkan data dari *end node*.



Gambar 2. 1 Arsitektur Jaringan LoRa

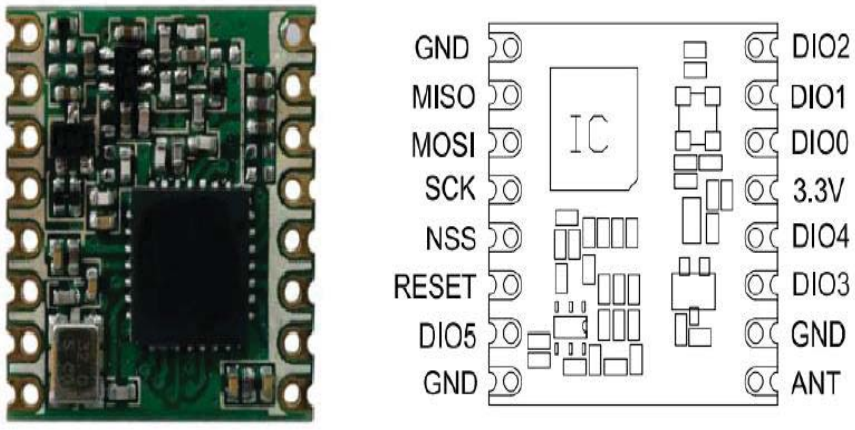
Dalam hal arsitektur sebenarnya untuk jaringan LoRa, node biasanya dalam topologi bintang-bintang dengan *gateway* membentuk jembatan transparan. Ini menyampaikan pesan antara perangkat akhir dan *server* jaringan pusat di *backend*. Komunikasi ke node titik akhir umumnya dua arah, tetapi juga memungkinkan untuk mendukung operasi *multicast*, dan ini berguna untuk fitur seperti peningkatan perangkat lunak dan sejenisnya atau pesan distribusi massal lainnya. Jaringan LoRaWAN menyediakan konektivitas yang diperlukan, memungkinkan titik akhir LoRa untuk melewatkan data melalui antarmuka nirkabel ke *gateway* dan kemudian ke tujuan yang diperlukan

2.3. Spesifikasi Teknis LoRa SX1276

LoRa memungkinkan transmisi jarak jauh dengan konsumsi daya yang rendah. Teknologi ini mencakup lapisan fisik, sedangkan teknologi dan protokol lain seperti LoRaWAN (Jaringan Area Luas Jangkauan Panjang) menutupi lapisan atas. Ini dapat mencapai kecepatan data dari 27 Kbps hingga 0,3 Kbps tergantung pada faktor penyebaran (*Spreading Factor*, SF).

Fitur-fitur yang tersedia di LoRa adalah:

1. *Geolocation*, fungsi ini memungkinkan kita dapat mendeteksi lokasi keberadaan suatu benda tanpa biaya alias gratis.
2. Biaya rendah, dapat mengurangi biaya dengan 3 cara: mengurangi biaya infrastruktur, biaya operasional dan sensor-sensor yang mempunyai jaringanya sendiri.
3. Terstandar, dibuat agar dapat berinteraksi dan berfungsi dengan produk atau platform lain, sehingga dapat cepat beradaptasi dengan jaringan dan aplikasi IoT.
4. Daya rendah, dengan konsumsi daya yang dibutuhkan hanya berkisar dari 13mA hingga 15mA, satu unit LoRa dapat memancarkan hingga 3km.
5. Aman, tertanam *end to end* enkripsi AES128.
6. Kapasitas tinggi, Mendukung jutaan pesan per *base station*, ideal untuk operator jaringan publik yang melayani banyak pelanggan.



Gambar 2. 2 LoRa Semtech SX1276

2.4. Parameter LoRa

2.3.1 *Spreading Factor* (SF)

Spreading Factor (faktor penyebaran) adalah menunjukkan seberapa banyak *chirp* yang dipakai untuk mewakili satu simbol. SF ini dapat mengambil nilai 6 hingga 12, maka semakin besar nilai SF maka semakin besar *noise*.

2.3.2 *Coding Rate* (CR)

Code Rate adalah implementasi modulasi LoRa juga menambahkan *forward error correction* (FEC), dengan menggunakan enkode 4 bit data dengan redundansi menjadi 5, 6, 7 dan 8 bit. Menggunakan redundansi ini membuat sinyal LoRa lebih tahap terhadap interferensi singkat, nilai *Code Rate* perlu diatur sesuai dengan kondisi kanal yang dipakai, jika terdapat banyak interferensi sebaiknya nilai CR ditingkatkan. Namun perlu diperhatikan bahwa kenaikan nilai CR juga meningkatkan waktu transmisi.

2.3.3 *Bandwidth* (BW)

Bandwidth adalah lebar frekuensi yang dipakai untuk memodulasi data sinyal LoRa memungkinkan penggunaan tiga *bandwidth* berbeda dari 125 kHz, 250 kHz, dan 500 kHz. Semakin lebih tinggi nilai bandwidth akan mengurangi waktu jangkauan transmisi.

2.3.4 *Symbol Rate* (Rs)

Symbol Rate merupakan kumpulan bit data yang berbentuk code. *Symbol Rate* (Rs) adalah laju perubahan symbol tiap satuan waktu. Rumus mencari Rs adalah :

$$Rs = \frac{BW}{2^{SF}}\ Symbol/sec \quad \text{.....(2.1)}$$

Sedangkan rumus durasi *symbol* (Ts) adalah :

$$Ts = \frac{1}{Rs}\ se \quad \text{.....(2.2)}$$

2.3.5 *Bit Rate* (Rb)

Bit Rate merupakan satuan dasar informasi digital dan bersifat biner (1/0, *high/low*). *Bit Rate* (Rb) adalah laju perubahan bit tiap satuan waktu. Rumus menentukan *Bit Rate* adalah :

$$Rb = \frac{(SF \times RS \times CR)}{1000}\ kbps \quad \text{.....(2.3)}$$

2.3.6 PLE (*Path Loss Exponent*)

PLE merupakan parameter *n* yang berpengaruh dalam menentukan batas kritis dari cakupan wilayah dari sistem selular. Parameter ini dapat di tentukan dari

nilai *pathloss* dan level daya, sehingga nilai PLE sangat tergantung pada kondisi lingkungan.

2.3.7 RSSI (*Receive Signal Strength Indicator*)

RSSI adalah parameter untuk mengukur indikator kekuatan sinyal yang diterima. Nilai RSSI sangat bergantung pada kondisi lingkungan yaitu jarak dan penghalang, semakin jauh dan semakin banyak penghalangnya maka nilai RSSI akan menurun. Hal ini dapat mempengaruhi kinerja dari LoRa. Rumus untuk menghitung rata – rata RSSI adalah :

$$RSSI\ (dBm) = \frac{Total\ Nilai\ RSSI\ yang\ dipenerima}{Jumlah\ Sampel\ RSSI\ yang\ dikirim} \text{.....(2.4)}$$

2.3.8 PDR (*Packet Data Ratio*)

PDR (*Packet Data Ratio*) adalah persentasi keberhasilan paket data yang diterima. Rumus rata – rata untuk menghitung PDR adalah :

$$PDR\ (\%) = \left(\frac{paket\ terima}{paket\ kirim}\right) * 100\ \% \text{ .....(2.5)}$$

2.3.9 SNR (*Signal Noise Ratio*)

SNR adalah paket data yang diterima dari pengirim yang sinyal terganggu oleh gangguan *noise*. Nilai SNR mempresentasikan gangguan *noise* selama proses transmisi data terhadap kualitas data. Semakin besar nilai SNR menunjukan kualitas data yang diterima bagus. Rumus rata – rata untuk menghitung SNR adalah :

$$SNR\ (dB) = \frac{Total\ Nilai\ SNR\ yang\ diterima}{Jumlah\ Sampel\ SNR\ yang\ Terkirim} \text{.....(2.6)}$$

2.3.10 ToA (*Time on Air*)

ToA adalah parameter untuk mengukur selisih waktu antara paket data diterima dan paket data yang dikirim berikut ilustrasi *Time on Air* yang diukur. Berikut merupakan gambar ilustrasi pengukuran *Time on Air*, dimana sinyal dikirim dari pengirim dengan waktu tertentu sebelum penerima menerima sinyal ini.

$$Rata - Rata\ ToA\ (Sec) = \frac{Total\ Nilai\ ToA\ yang\ diterima}{Jumlah\ Sampel\ ToA\ yang\ dikirim} \text{.....(2.7)}$$

2.5. Arduino UNO

Arduino UNO adalah board mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol *reset*.

Gambar 2. 3 Arduino UNO

Gambar 2. 4 Kabel USB Board Arduino UNO

2.4.1 Bagian-bagian *Arduino Uno*

Adapun bagian-bagian dari arduino UNO ditampilkan pada tabel berikut:

Tabel 2. 1 Deskripsi Arduino UNO

Mikroprosesor	ATmega 328
Tengangan Pengoperasian	5 V
Tegangan input yang disarankan	7 – 12 V
Batasan Tegangan Input	6 – 20 V
Jumlah Pin I/O Digital	14 pn Digital (6 diantaranya menyediakan keluaran PWM)
Jumlah Pin Input Analog	6 Pin
Arus DC Tiap Pin I/O	40Ma
Arus DC untuk Pin 3,3 V	50Ma
Memory Flash	32 KB(ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
Clock Speed	16MHz

2.4.2 Arduino IDE

Arduino IDE itu merupakan kependekan dari *Integrated Developtment Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah *arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman arduino (*sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC *mikrokontroler arduino* telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler arduino* dengan *mikrokontroler*. *Arduino* IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. *Arduino* IDE ini dikembangkan dari

software Processing yang dirombak menjadi *arduino* IDE khusus untuk pemrograman dengan *arduino*. Berikut adalah gambar tampilan *arduino* IDE yang ditampilkan pada Gambar 2.5 sebagai berikut:

Gambar 2. 5 Tampilan Arduino IDE

Pemrograman suatu aplikasi menggunakan *software* Arduino IDE mengikuti struktur kode yaitu berisi fungsi *library, value, setup(), loop() dan function*.

1. ***Library***

   Biasanya Dalam *Library* bisa digunakan untuk module *Bluetooth, Ethernet Shield*, GSM, GPS, LCD, Wifi ESP8266, Servo, dan masih banyak lagi. Ketika akan menggunakan module tersebut anda hanya perlu Include Library nya kedalam Software Arduino IDE, dan setelah itu module tersebut dapat di gunakan. Untuk mencari *Library* yang di inginkan anda padat mencari di google.com atau di github.com.

2. ***Value***

   Berisi variabel atau konstanta sesuai dengan tipe data yang didukung oleh Arduino.

3. ***Setup()***

   Fungsi ini dipanggil pertama kali ketika menjalankan *sketch*. digunakan sebagai **tempat** inisialisai *variable*, *pin mode*, penggunaan *library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di *reset.*

4. ***Loop()***

   Setelah membuat fungsi *setup()* sebagai tempat inisialisai variabel dan menetapkan nilai maka selanjutnya fungsi *loop()* seperti namanya fungsi ini akan melakukan perulangan berturu-turut, memungkina program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board* Arduino.

5. ***Function***

   Segmentasi kode ke fungsi memungkinkan programmer untuk membuat potongan-potongan modular kode yang melakukan tugas yang terdefinisi dan kemudian kembali ke asal kode dari mana fungsi itu “dipanggil”. Umumnya menggunakan fungsi adalah ketika salah satu kebutuhan untuk melakukan tindakan yang sama beberapa kali dalam sebuah program.