

BAB II

TINJAUAN PUSTAKA

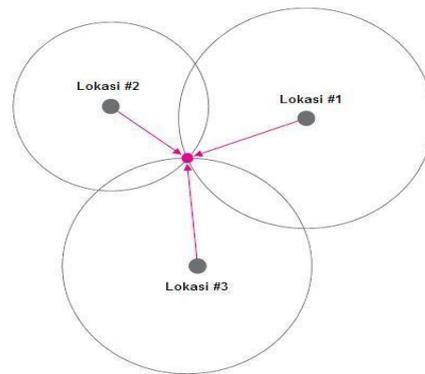
2.1 Global Positioning System (GPS)

Global Positioning System (GPS) merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana dia berada (secara global) dipermukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital.

2.1.1 Definisi Global Positioning System (GPS)

GPS (*Global Positioning System*) adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik.

GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh di dunia saat ini (*undergraduate thesis* Wildan Habibi, ITS, Surabaya Januari : 2011). GPS menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah GPS *receiver* menerima informasi dari tiga atau lebih satelit tersebut seperti terlihat dalam Gambar 2.1 dibawah, untuk menentukan posisi. GPS *receiver* harus berada dalam *line-of sight* (LoS) terhadap ketiga satelit tersebut untuk menentukan posisi, sehingga GPS hanya ideal untuk digunakan dalam *outdoor positioning*.



Gambar 2.1 Trilaterasi Dalam Global Positioning System (GPS)

(Sumber : *Global Positioning System Overview*, Andi Sunyoto, 2013)

Aplikasi yang berada disisi target (*client*) setelah mendapatkan *request* dari pelacak (*server*) maka *client* akan meminta koordinat posisinya pada GPS (*Global Positioning System*), yang kemudian akan dikirimkan ke pelacak (*server*).

Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Meskipun satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis. Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau *area coverage* yang lebih luas. Satelit-satelit ini akan selalu berada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya blank spot (area yang tidak terjangkau oleh satelit).

Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi Anda di atas permukaan bumi. GPS *reciever* sendiri berisi beberapa *integrated circuit* (IC) sehingga murah dan teknologinya mudah untuk di gunakan oleh semua orang. GPS dapat digunakan untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan di integrasikan dengan komputer maupun laptop.



Gambar 2.2 Macam-Macam Perangkat GPS

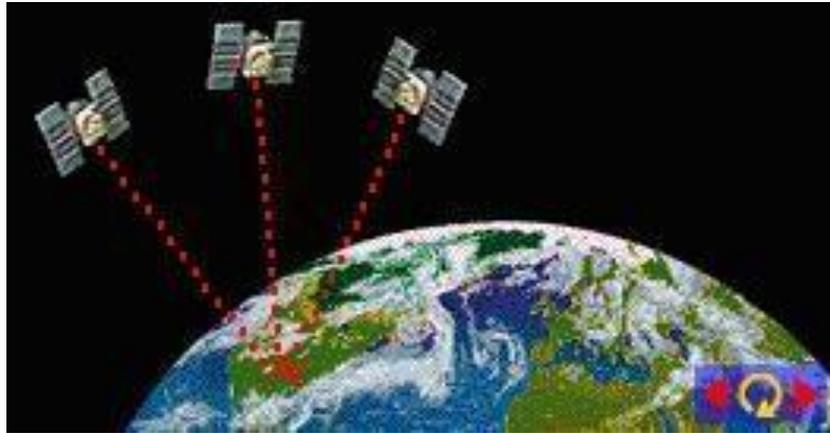
(Sumber : *Global Positioning System Overview*, Andi Sunyoto, 2013)

2.1.2 Cara Kerja *Global Positioning System* (GPS)

Setiap daerah di atas permukaan bumi ini minimal terjangkau oleh 3-4 satelit. Pada prakteknya, setiap GPS terbaru bisa menerima sampai dengan 12 channel satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat GPS dapat dengan mudah menangkap sinyal yang dikirimkan oleh satelit. Semakin banyak satelit yang diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi.

Cara kerja GPS secara sederhana ada 5 langkah, yaitu :

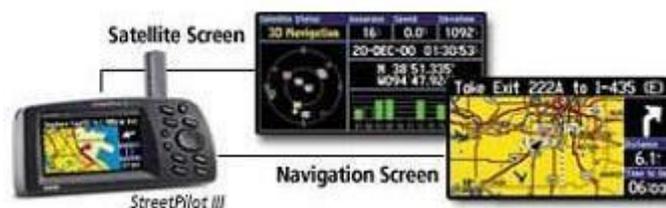
- a. Memakai perhitungan “triangulation” dari satelit.
- b. Untuk perhitungan “triangulation”, GPS mengukur jarak menggunakan travel time sinyal radio.
- c. Untuk mengukur travel time, GPS memerlukan memerlukan akurasi waktu yang tinggi.
- d. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya.
- e. Terakhir harus mengoreksi delay sinyal waktu perjalanan di atmosfer sampai diterima receiver.



Gambar 2.3 Cara Satelit menentukan Posisi

(Sumber : *Global Positioning System Overview*, Andi Sunyoto, 2013)

Satelit GPS berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dia dan mengirimkan sinyal informasi ke bumi. GPS *reciever* mengambil informasi itu dan dengan menggunakan perhitungan “*triangulation*” menghitung lokasi *user* dengan tepat. GPS *reciever* membandingkan waktu sinyal di kirim dengan waktu sinyal tersebut di terima. Dari informasi itu didapat diketahui berapa jarak satelit. Dengan perhitungan jarak GPS *reciever* dapat melakukan perhitungan dan menentukan posisi user dan menampilkan dalam peta elektronik.



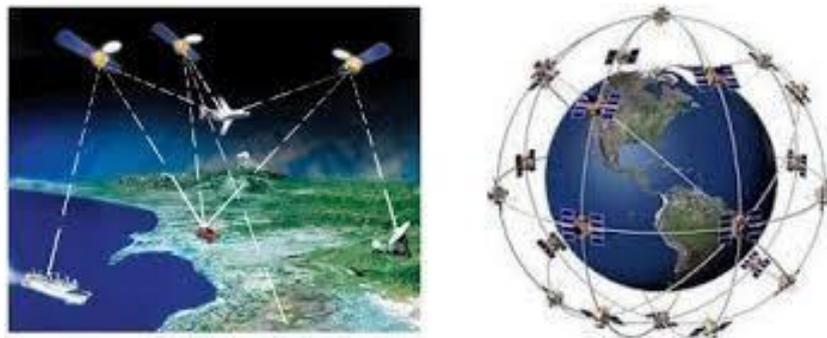
Gambar 2.4 Tampilan GPS Reciever

Sumber : *Global Positioning System Overview*, Andi Sunyoto, 2013

Sebuah GPS receiver harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Jika GPS receiver dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude*, *longitude* dan *altitude*). Jika sudah dapat menentukan posisi *user*, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS dalam mengirim informasi waktu sangat presisi karena satelit tersebut memakai jam atom. Jam atom yang ada pada satelit ialah dengan partikel atom yang di isolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi kita. Selain itu semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data yang diterima karena ketiga satelit

mengirim *pseudo-random code* dan waktu yang sama. Ketinggian itu menimbulkan keuntungan dalam mendukung proses kerja GPS, bagi kita karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit dan orbit yang cocok dan perhitungan matematika yang cocok. Satelit harus tepat pada posisi yang tepat sehingga stasiun di bumi harus terus memonitor setiap pergerakan satelit, dengan bantuan radar yang presisi selalu di cek tentang altitude, position dan kecepatannya.



Gambar 2.5 Cara Satelit Menentukan Posisi Lokasi

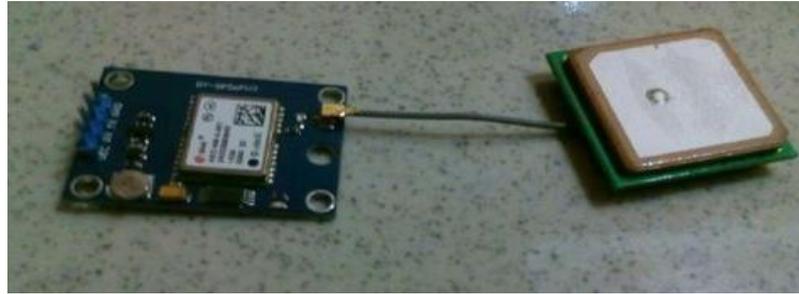
(Sumber : *Global Positioning System Overview*, Andi Sunyoto, 2013)

Sinyal yang dikirimkan oleh satelit ke GPS akan digunakan untuk menghitung waktu perjalanan (*travel time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival* (TOA). Sesuai dengan prinsip fisika, bahwa untuk mengukur jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal. Maka, jarak antara satelit dengan GPS juga dapat diperoleh dari prinsip fisika tersebut. Setiap sinyal yang dikirimkan oleh satelit akan juga berisi informasi yang sangat detail, seperti orbit satelit, waktu, dan hambatan di atmosfer.

Satelit menggunakan jam atom yang merupakan satuan waktu paling presisi. Untuk dapat menentukan posisi dari sebuah GPS secara dua dimensi (jarak), dibutuhkan minimal tiga buah satelit. Empat buah satelit akan dibutuhkan agar didapatkan lokasi ketinggian (secara tiga dimensi). Setiap satelit akan memancarkan sinyal yang akan diterima oleh GPS *receiver*. Sinyal ini akan dibutuhkan untuk menghitung jarak dari masing-masing satelit ke GPS. Dari jarak tersebut, akan diperoleh jari-jari lingkaran jangkauan setiap satelit. Lewat perhitungan matematika yang cukup rumit, interseksi (perpotongan) setiap lingkaran jangkauan satelit tadi akan dapat digunakan untuk menentukan lokasi dari GPS di permukaan bumi^[3]

2.1.3 GPS NEO-6M Module

Modul GPS (*Global Positioning System*) UBLOX NEO-6M berukuran 25x35mm untuk modul, 25x25mm untuk antenna. Modul GPS UBLOX NEO-6M berfungsi sebagai penerima GPS (*Global Positioning System Receiver*) yang dapat mendeteksi lokasi dengan menangkap dan memproses sinyal dari satelit navigasi. Aplikasi dari modul ini melingkupi sistem navigasi, sistem keamanan terhadap kemalingan pada kendaraan / perangkat bergerak, akuisisi data pada sistem pemetaan medan, penjejak lokasi / *location tracking*, dan lainnya.^[2]



Gambar 2.6 Modul UBLOX Neo-6M

(Sumber : *Datasheet Modul GPS UBLOX NEO-6M* , 2017)

Modul ini kompatibel dengan APM2 dan APM2.5 dengan EEPROM (*Electrically Erasable Programmable Read Only Memory*) terpadu yang dapat digunakan untuk menyimpan data konfigurasi. Komunikasi antarmuka menggunakan serial TTL (*Transistor Transistor Logic*) (RX/TX) yang dapat diakses dari

mikrokontroler yang memiliki fungsi UART (*Universal Asynchronous Receiver Transmitter*) atau emulasi serial TTL (pada Arduino dapat menggunakan pustaka komunikasi serial / *serial communication library* yang sudah tersedia dalam paket Arduino IDE). Baud rate diset secara *default* pada 9600 bps.

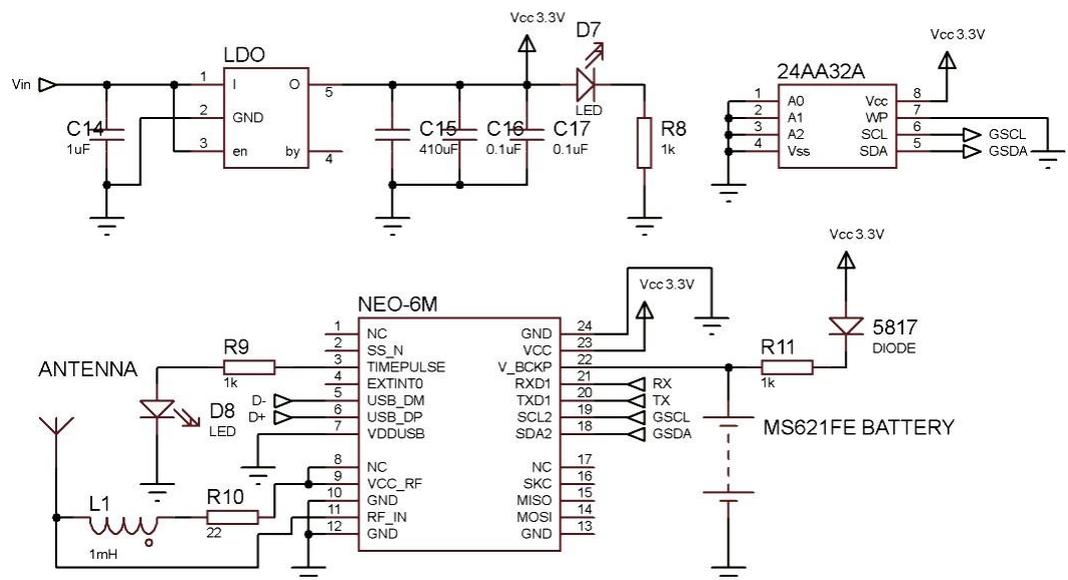
GPS Processor dari modul ini menggunakan u-blox NEO-6 GPS Module Modul ini dapat memproses hingga 50 kanal sinyal secara cepat dengan waktu Cold TTFF (*Cold-Start Time-To-First-Fix*, waktu yang diperlukan untuk menentukan posisi dari kondisi mati total) kurang dari 27 detik. ^[4]

Spesifikasi Modul u-blox NEO-6M

- a. Tipe penerima: 50 channel, GPS L1 frekuensi, C/A Code. SBAS (Satellite Based Augmentation System): WAAS (Wide Area Augmentation System), EGNOS (European Geostationary Navigation Overlay Service), MSAS (Multi-functional Satellite Augmentation System).
- b. Sensitivitas penjejak & navigasi: -161 dBm (reakuisisi dari blank-spot: -160 dBm)
- c. Sensitivitas saat baru memulai: -147 dBm pada cold-start, -156 dBm pada

hot start

- d. Kecepatan pembaharuan data / navigation update rate: 5 Hz
- e. Akurasi penetapan lokasi GPS secara horisontal: 2,5 meter (SBAS = 2m)
- f. Rentang frekuensi pulsa waktu yang dapat disetel: 0,25 Hz hingga 1 kHz
- g. Akurasi sinyal pulsa waktu: RMS 30 ns (99% dalam kurang dari 60 ns) dengan granularitas 21 ns atau 15 ns saat terkompensasi
- h. Akurasi kecepatan: 0,1 meter / detik
- i. Akurasi arah (heading accuracy): $0,5^\circ$
- j. Batasan operasi: daya tarik maksimum 4x gravitasi, ketinggian maksimum 50 Km, kecepatan maksimum 500 meter / detik (1800 km/jam). red: dengan limit seperti ini, modul ini bahkan dapat digunakan di pesawat jet super-cepat sekalipun.



Gambar 2.7 Rangkaian Modul GPS UBLOX NEO-6M

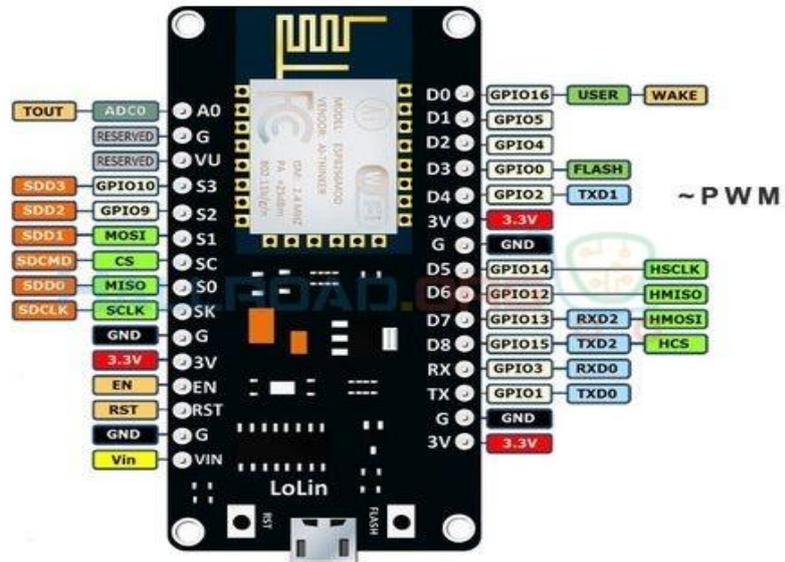
(Sumber :*Datasheet Modul GPS UBLOX NEO-6M*, 2017)

2.2 NodeMCU ESP8266

NodeMCU merupakan papan pengembangan produk Internet of Things (IoT) yang berbasisan Firmware eLua dan System on a Chip (SoC) ESP8266-12E. ESP8266 sendiri merupakan chip WiFi dengan protocol stack TCP/IP yang lengkap. NodeMCU dapat dianalogikan sebagai board arduino-nya ESP8266. Program ESP8266 sedikit merepotkan karena diperlukan beberapa teknik wiring serta tambahan modul USB to serial untuk mengunduh program.^[5]

Namun NodeMCU telah me-package ESP8266 ke dalam sebuah board yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap Wifi juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan charging smarphone. Alasan penulis memilihan NodeMCU ESP8266 ialah karena mudah diprogram dan memiliki pin I/O yang memadai dan dapat mengakses jaringan Internet untuk mengirim atau mengambil data melalui koneksi WiFi. Spesifikasi dari NodeMCU sebagai berikut :

1. 10 port pin GPIO
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 Wire
5. ADC



Gambar 2.8 NodeMCU EPS8266 dan Skema Pin

(Sumber : arduino-esp8266.readthedocs.io)

Gambar diatas merupakan kaki pin yang ada pada NodeMCU. Berikut penjelasan dari pin – pin NodeMCU tersebut.

1. ADC: Analog Digital Converter. Rentang tegangan masukan 0-1v, dengan skup nilai digital 0-1024
2. RST : berfungsi mereset modul
3. EN: Chip Enable, Active High
4. IO16 :GPIO16, dapat digunakan untuk membangunkan chipset dari mode deep sleep
5. IO14 : GPIO14; HSPI_CLK
6. IO12 : GPIO12: HSPI_MISO
7. IO13: GPIO13; HSPI_MOSI; UART0_CTS
8. VCC: Catu daya 3.3V (VDD)

9. CS0 :Chip selection
10. MISO : Slave output, Main input.
11. IO9 : GPIO9
12. IO10 GBIO10
13. MOSI: Main output slave input
14. SCLK: Clock
15. GND: Ground
16. IO15: GPIO15; MTDO; HSPICS; UART0_RTS
17. IO2 : GPIO2;UART1_TXD
18. IO0 : GPIO0
19. IO4 : GPIO4
20. IO5 : GPIO5
21. RXD : UART0_RXD; GPIO3
22. TXD : UART0_TXD; GPIO1

Untuk tegangan kerja ESP8266 menggunakan standar tegangan JEDEC (tegangan 3.3V) untuk bisa berfungsi. Tidak seperti mikrokontroler AVR dan sebagian besar board Arduino yang memiliki tegangan TTL 5 volt. Meskipun begitu, NodeMCU masih bisa terhubung dengan 5V namun melalui port micro USB atau pin Vin yang disediakan oleh board-nya. Namun karena semua pin pada ESP8266 tidak toleran terhadap masukan 5V. Maka jangan sekali – kali langsung mencatunya dengan tegangan TTL jika tidak ingin merusak board anda. Anda bisa menggunakan Level Logic Converter untuk mengubah tegangan ke nilai aman 3.3v.

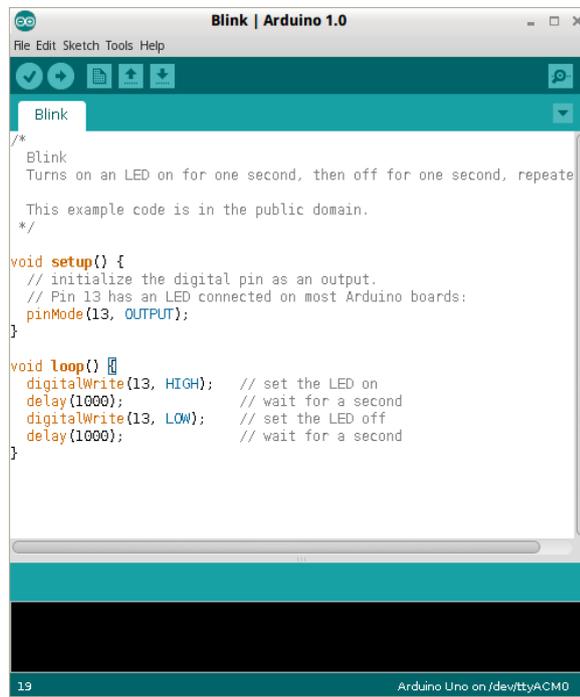
2.3 *Integrated Development Environment (IDE) Arduino*

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*. *Integrated Development Environment (IDE) Arduino* terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah toolbar dengan tombol-tombol untuk fungsi yang umum dan beberapa menu. *Integrated Development*

Environment (IDE) Arduino terhubung ke arduino board untuk meng-upload program dan juga untuk berkomunikasi dengan arduino board.^[6]

Perangkat lunak (*software*) yang ditulis menggunakan *Integrated Development Environment (IDE)* Arduino disebut *sketch*. *Sketch* ditulis pada *editor teks*. *Sketch* disimpan dengan file berekstensi *.ino*. area pesan memberikan informasi dan pesan *error* ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan output teks dari *Integrated Development Environment (IDE)* Arduino dan juga menampilkan pesan *error* ketika kita mengkompilasi *sketch*. Pada sudut kanan bawah jendela *Integrated Development Environment (IDE)* Arduino menunjukkan jenis *board* dan *port serial* yang sedang digunakan. Tombol toolbar digunakan untuk mengecek dan meng-upload *sketch*, membuat, membuka, atau menyimpan *sketch*, dan menampilkan serial monitor. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
2. Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan Arduino.



Gambar 2.9 Tampilan *Software Compiler* Arduino

(Sumber : www.arduino.cc, 2014)

Di bawah ini merupakan tombol-tombol toolbar serta fungsinya yang terdapat pada IDE Arduino, diantaranya:



Verify : berfungsi untuk mengecek error pada kode program



Upload : berfungsi untuk meng-*compile* dan meng-*upload* program ke Arduino board.



New : berfungsi untuk membuat *sketch* baru



Open : berfungsi untuk menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam *sketchbook*.



Save : berfungsi untuk menyimpan *sketch*.

2.4 Android

Android adalah sistem operasi *mobile* yang didasari oleh Linux yang telah dimodifikasi. Awalnya Android dikembangkan oleh perusahaan *startup* yang bernama Android, Inc. Pada tahun 2005, sebagai awal memasuki dunia *mobile*, Google membeli Android dan mengambil alih pengembangan. Google menginginkan Android menjadi terbuka dan bebas. Pengembangan Android yang sederhana membuat Android sangat interaktif.

Untuk mengembangkan Android dibentuklah Open Handset Alliance yang terdiri dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Terdapat dua jenis distributor sistem operasi Android. Pertama, Google Mail Service yang mendapat dukungan penuh dari Google. Kemudian yang kedua adalah Open Handset Distribution (OHD) yang distribusinya tanpa dukungan langsung Google.

Keuntungan dari Android adalah memberikan kemudahan pada pengembangan Android. *Developer* hanya perlu mengembangkan aplikasi pada *platform* Android, dan aplikasi tersebut akan dapat berjalan pada berbagai perangkat, yang tentunya mempunyai sistem operasi Android. Selain itu Android merupakan platform yang lengkap, baik dari sistem operasi, aplikasi serta tools untuk mengembangkan aplikasi.^[7]

2.4.1 Fitur Pada Android

Android merupakan sistem operasi *open source* dan bebas untuk di-*customize*. Oleh karena itu, tidak ada konfigurasi standar untuk *software* dan *hardware* pada perangkat berbasis Android, namun, Android mendukung beberapa fitur dasar yaitu: (Lee, 2012, pp. 3)

- a. Storage – menggunakan SQLite, karena SQLite merupakan
- b. database yang ringan untuk penyimpanan data.

- c. Connectivity – mendukung GSM/EDGE, IDEN, CDMA, EV- DO, UMTS, Bluetooth, WiFi, LTE, WiMAX
- d. Messaging – mendukung SMS dan MMS
- e. Web Browser - berdasarkan pada open source webkit, bersama dengan chrome's v8 Javascript engine
- f. Media Support - mendukung beberapa media seperti AAC, HE-AAC, MPEG-4 SP, AMR, AMR-WB, WAV, MP3, JPG, PNG
- g. Hardware Support - accelerometer sensor, kamera, digital kompas, proximity sensor, GPS
- h. Multi-touch - mendukung multi-touch screens
- i. Multi-tasking – mendukung aplikasi multi-tasking
- j. Flash support – Android 2.3 mendukung flash 10.1
- k. Tethering - mendukung untuk sharing internet.

2.4.2 Versi Android

Dari awal dimunculkan Android terdapat perubahan yang terus menerus dilakukan. Hingga saat ini versi Android terbaru bernama Marshmallow. Berikut versi Android dari awal hingga saat ini:

Tabel 2.1 Tabel Versi Android

(Sumber : www.Android.com)

Versi Android	Kode Nama
Android 1.5	Cupcake
Android 1.6	Donut
Android 2.0	Eclair
Android 2.2	Froyo (Frozen Yoghurt)
Android 2.3	Gingerbread
Android 3.0	Hpneycomb
Android 4.0	Ice Cream Sandwich
Android 4.1	Jelly Bean
Android 4.4	KitKat
Android 5.0	Lollipop
Android 6.0	Marshmallow
Android 7.0	Nougat

2.4.3 Arsitektur Android

Android terbagi menjadi lima bagian dari empat lapisan utama yaitu :

a. *Linux Kernel*

Linux kernel merupakan dasar dari Android. Lapisan ini semua *low level device driver* untuk berbagai komponen perangkat keras dari perangkat Android.

b. *Libraries*

Lapisan ini mengandung seluruh kode yang menyediakan fitur utama dari sistem operasi Android. Sebagai contoh, SQLite library menyediakan dukungan *database* sehingga aplikasi dapat menggunakannya sebagai tempat penyimpanan.

c. *Android Runtime*

Android *Runtime* terdapat pada lapisan yang sama dengan *libraries*, yang menyediakan kumpulan *libraries* utama yang dapat digunakan pengembang untuk membuat aplikasi pada Android menggunakan bahasa pemrograman Java. Android *Runtime* juga termasuk Dalvik virtual machine, yang memungkinkan aplikasi Android berjalan dengan proses tersendiri.

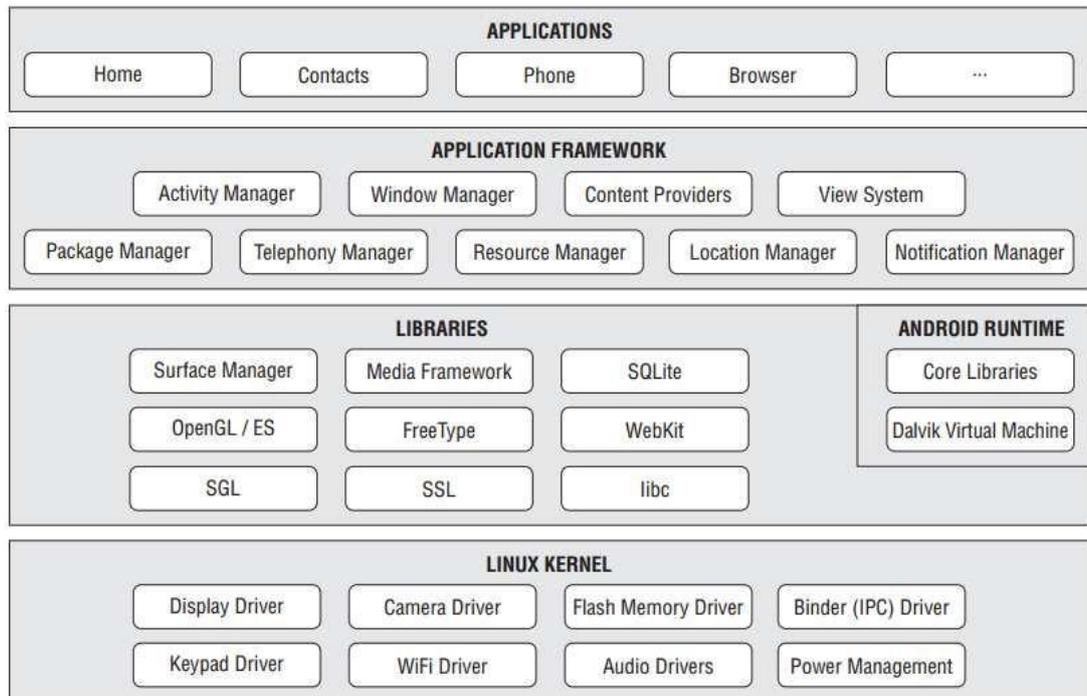
d. *Application Framework*

Memperlihatkan berbagai kemampuan dari sistem operasi Android untuk pengembang aplikasi Android sehingga mereka dapat menggunakannya dalam membuat aplikasi.

e. *Applications*

Applications merupakan lapisan paling atas, yang memungkinkan para pengembang untuk menemukan aplikasi bawaan yang terdapat pada perangkat Android. Contohnya saja adalah Phone, Contacts, Browser,

dan lainnya. Selain itu juga aplikasi yang dapat di unduh dan di-install dari Android Market. Setiap aplikasi yang dibuat terdapat pada lapisan ini.



Gambar 2.10 Arsitektur Android

(Sumber : www.Android.com)

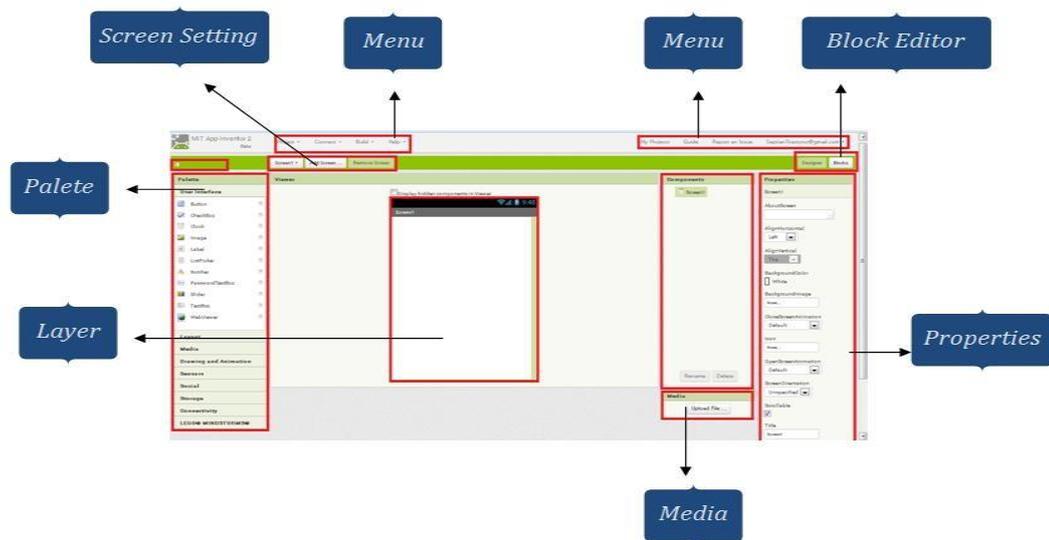
2.5 MIT App Inventor 2

App Inventor for Android adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-maintenance oleh Massachusetts Institute of Technology (MIT). App Inventor menggunakan bahasa Kawa Language Framework dan Kawa's dialect yang dikembangkan oleh Per Bothner. Kedua aplikasi tersebut dijadikan sebagai compiler dan menerjemahkan Visual Block Programming. ^[8]

2.5.1 Area kerja MIT App Inventor 2

Untuk dapat menggunakan App Inventor 2 diperlukan pengenalan

tentang area kerja dari App Inventor 2 tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 2.11.



Gambar 2.11. Area Kerja App Inventor 2

(Sumber : Membuat Aplikasi untuk Android. Mulyadi. 2010)

Berikut ini penjelasan dari masing-masing elemen yang terdapat pada area kerja App Inventor 2:

1. *Screen Setting* merupakan sebuah kelompok yang berguna untuk mengatur layar, menambah layar, dan menghapus layar.
2. *Palette* adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain.
3. *Menu* merupakan sekelompok menu yang berguna dalam membuat project baru, proses debugging, konversi file apk, dll.
4. *Block Editor* adalah suatu tombol untuk masuk ke halaman kode blok untuk proses pengkode-an.

5. *Properties* : untuk mengatur komponen yang telah di buat menjadi desain di layer.
6. *Media* : Tempat dimana untuk meng-upload file.
7. *Layer* : Area untuk men-desain.

2.6 ThingSpeak

ThingSpeak adalah sebuah wadah open source berbentuk website yang menyediakan layanan untuk kebutuhan IoT dan dapat menerima data menggunakan protocol HTTP melalui internet. ThingSpeak memungkinkan pembuatan aplikasi sensor logging, aplikasi lokasi pelacakan, dan jaringan sosial hal dengan update status . ThingSpeak awalnya diluncurkan oleh ioBridge pada tahun 2010 sebagai layanan untuk mendukung aplikasi IOT. ThingSpeak telah terintegrasi dukungan dari numerik komputasi perangkat lunak MATLAB dari MathWorks. Memungkinkan ThingSpeak pengguna untuk menganalisis dan memvisualisasikan data yang diunggah menggunakan Matlab tanpa memerlukan pembelian lisensi Matlab dari MathWorks. ThingSpeak memiliki hubungan dekat dengan MathWorks, Inc. Bahkan, semua dokumentasi ThingSpeak dimasukkan ke situs dokumentasi matlab yang MathWorks dan bahkan memungkinkan terdaftar MathWorks akun pengguna login sebagai valid di situs ThingSpeak. Persyaratan layanan dan kebijakan privasi dari ThingSpeak.com adalah antara pengguna setuju dan MathWorks, Inc^[9]

Fitur :

1. MATLAB Analyze dan visualisasi
2. Thingspeak App. Thingspeak App berfungsi untuk menyertakan code tambahan sesuai kebutuhan IoT yang ingin dikembangkan seperti menambahkan sebuah fungsi atau prosedur kedalam mikrokontroler yang terhubung.
3. Chart & Channel API yang interaktif untuk menampilkan hasil analisis data

4. Banyak referensi

2.7 Firebase Cloud Firestore

Menurut Google (2018), Firebase Cloud Firestore adalah database yang fleksibel dan skalable untuk pengembangan seluler, web, dan server di Firebase dan Google Cloud Platform. Sama seperti Firebase Realtime Database, Cloud Firestore membuat data tetap terhubung di aplikasi klien melalui listener realtime dan menawarkan dukungan secara offline untuk seluler dan web. Dengan begitu, aplikasi yang dibuat akan responsive dan mampu bekerja tanpa bergantung pada koneksi internet.

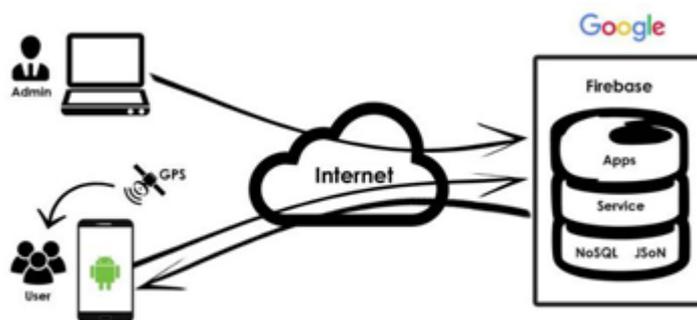
Cloud Firestore adalah database NoSQL yang dihosting di cloud dan dapat diakses langsung melalui SDK asli oleh iOS, Android, dan aplikasi web. Selain REST dan RPC API, Cloud Firestore juga tersedia di Node.js, Java, Python, dan Go SDK yang asli. Kemampuan utama dari Firebase Cloud Firestore adalah sebagai berikut:

1. Fleksibilitas Model data Cloud Firestore mendukung struktur data yang hierarkis dan fleksibel. Data disimpan dalam dokumen yang tersusun dalam koleksi. Selain subkoleksi, dokumen dapat berisi objek bertingkat yang kompleks.
2. Pembuatan Kueri yang Ekspresif Di Cloud Firestore, kueri dapat digunakan untuk mengambil masing-masing dokumen tertentu atau semua dokumen dalam koleksi yang sesuai dengan parameter kueri. Kueri dapat meliputi beberapa filter berantai dan menggabungkan penyaringan dan pengurutan. Kueri juga diindek secara default sehingga performa kueri sebanding dengan ukuran kumpulan hasil, bukan kumpulan data.
3. Update Realtime Seperti Realtime Database, Cloud Firestore menggunakan sinkronisasi data untuk mengupdate data pada perangkat yang terhubung.

Namun, Cloud Firestore juga dirancang untuk membuat kueri pengambilan satu kali yang sederhana secara efisien.

4. Dukungan Offline Cloud Firestore menyimpan data yang digunakan secara aktif oleh aplikasi, sehingga aplikasi dapat menulis, membaca, mendeteksi, dan melakukan kueri data meskipun perangkat sedang offline. Saat perangkat kembali online, Cloud Firestore akan menyinkronkan semua perubahan local kembali ke Cloud Firestore.
5. Dirancang Sesuai Paket Cloud Firestore menghadirkan infrastruktur Google Cloud Platform yang terbaik, yaitu replikasi data multi-region secara otomatis, jaminan konsistensi yang kuat, batch operasi yang bersifat menyeluruh, dan dukungan transaksi nyata. Cloud Firestore dirancang untuk menangani beban kerja database yang tersulit dari aplikasi terbesar di dunia.

Dalam Gambar 2. ditunjukkan contoh arsitektur sistem Firebase dengan Android.



Gambar 2.12 Arsitektur Sistem Firebase

(Sumber : Firebase Manual Google)

Semua data Firebase Realtime Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di awan. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa

diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu Anda menulis lebih banyak kode yang bisa dipertahankan.^[10]