BAB II

LANDASAN TEORI

Seperti tertulis pada penjelasan sistematika penulisan, dalam bab ini akan menjelaskan beberapa landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dengan perancangan dan pembuatan alat ini. Pada bab ini akan di bahas tentang dasar-dasar dari beberapa bagian terpenting dari sistem yang akan dibuat.

2.1 Kajian Pustaka

Penelitian yang akan dilaksanakan oleh penulis yaitu mengenai suatu sistem *tracking* pada wahana pengiriman barang menggunakan modul *Global Positioning System* (GPS) sehingga dapat ditelusuri keberadaan wahana pengirim barang tersebut yang akan ditampilkan pada suatu *map digital*. Selain itu, sistem ini juga menggunakan RFID yang berfungsi sebagai sistem pencatatan pada barang. Penulis melakukan kajian pustaka dari penelitian-penelitian sebelumnya. Selain itu penulis juga melakukan kajian pustaka dari system yang sudah dipakai dipasaran seperti JNE, J&T dan Sajira Mahardika.

Penelitian yang dilaksanakan oleh Mohammad A. Al-Khedher [2] membahas tentang sistem pelacakan kendaraan berbasis GPS dan GSM. Sistem ini memiliki dua modul utama, yaitu modul pelacakan perangkat yang dipasang pada mobil dan modem penerima GSM dan *server*. Modul pelacakan perangkat tersusun dari penerima GPS, mikrokontroler dan modem GSM. Penerima GPS mengambil informasi posisi dari satelit dalam bentuk bujur dan lintang dengan pembacaan *real time*. Mikrokontroler memiliki tiga tugas utama yaitu untuk membaca parameter

mesin tertentu dari port data mobil, untuk proses informasi GPS untuk mengambil nilai-nilai yang diinginkan dan untuk mengirimkan data ke *server* menggunakan GSM modem dengan SMS. Parameter mesin yang dipilih yaitu kecepatan putaran mesin, suhu mesin pendingin, kecepatan kendaraan, besar persen masuknya udara ke mesin pembakaran. Modul kedua terdiri dari modem penerima GSM dan *server*. Modem menerima SMS yang meliputi koordinat GPS dan parameter mesin. Program *visual basic* digunakan untuk memproses teks untuk mendapatkan parameter numerik yang disimpan dalam bentuk file excel dan membaca koordinat yang diterima dari GPS yang selanjutnya dikoreksi oleh Kalman filter. File excel dikonversi ke format *Keyhole Markup Language* (KML) untuk memasukkan informasi ini ke *google earth*. *Google earth* memproses file KML dan menunjukkan parameter lokasi mobil di peta.

Ika Tri Meilani dan Sudaryanto [3] *Prototype Tracking* Dengan Pemanfaatan *Geolocation* Sebagai GPS (*Global Positioning System*) berbasis *web mobile* pada jasa pengiriman JNE Semarang. Pada penelitiannya ini dikatakan penggunaan *geolocation* sebagai GPS dapat memberikan estimasi waktu yang sesuai pada sebuah pengiriman. Sedangkan sistem yang dibangun merupakan sistem *tracking* yang bertujuan untuk melacak keberadaan barang dan melakukan antisipasi terjadinya kriminalitas dijalanan.

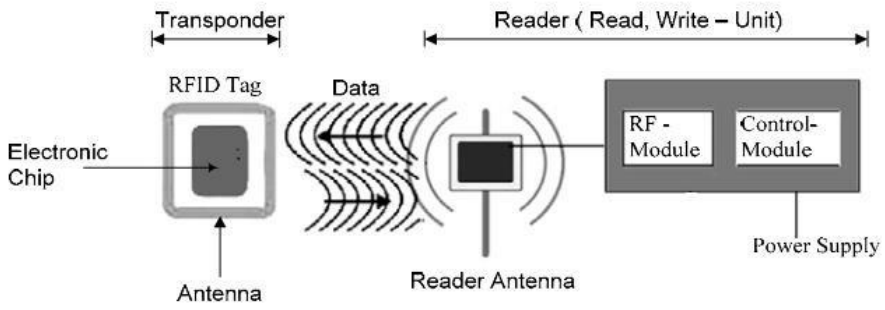
Setelah melakukan wawancara dengan karyawan JNE, di dapat data bahwa sistem yang digunakan oleh JNE yaitu menggunakan barcode sebagai identitas barang, mengenai monitoring barang perusahaan ini masih menggunakan sistem *warehouse,* dimana saat pengirim dan penerima barang ingin tahu keberadaan barang, hanya muncul keberadaan barang ketika di *warehouse* saja, selain itu pada

saat barang telah terkirim ke alamat yang dituju bukti penerimaannya hanya berupa tanda tangan penerima.

Sistem yang digunakan oleh perusahaan J&T pun hampir sama dengan JNE, perbedaannya hanya terletak pada saat barang telah terkirim ke alamat yang dituju, bukti penerimaannya menggunakan foto penerima barang yang di kirim ke pegawai yang berada di *warehouse.* Sedangkan perusahaan Sajira Mahardika pada prinsipnya memiliki sistem yang sama dengan JNE. Namun perbedaannya perusahaan ini tidak memiliki aplikasi untuk memonitoring barang.

2.2 *Radio Frequency Identification* (RFID)

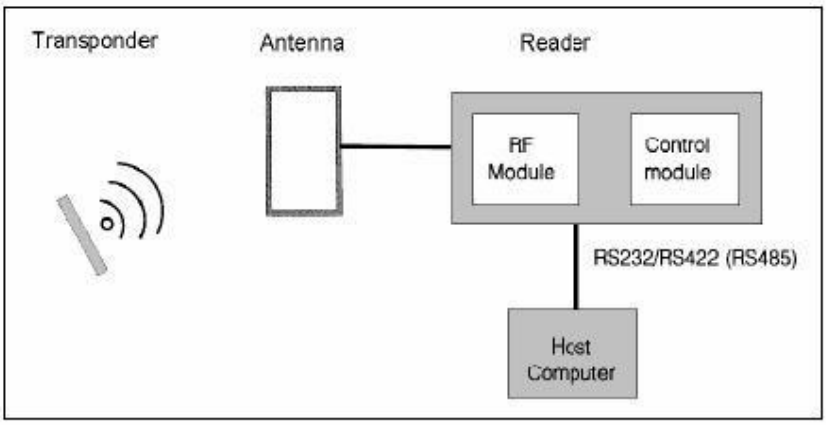
Radio Frequency Identification atau yang lebih dikenal sebagai RFID merupakan suatu metoda identifikasi objek yang menggunakan gelombang radio. Proses identifikasi dilakukan oleh RFID *reader* dan RFID *transponder* (RFID tag) [4]. RFID tag terdiri dari sebuah *microchip* yang tertanam pada sebuah antena gelombang radio yang ditempelkan. *Chip* tersebut dapat menyimpan data seukuran 2 kilobyte. Untuk mengambil data yang tersimpan pada RFID tag, dibutuhkan sebuah *reader*. *Reader* adalah sebuah alat yang memiliki satu atau banyak antena yang memancarkan gelombang radio dan menerima sinyal kembali dari tag reader kemudian melewatkan sebuah informasi dalam bentuk digital.



Gambar 2.1 Sistem *Radio Frequency Identification* (RFID) [5]

2.2.1 Komponen utama RFID

Sebuah sistem RFID terdiri atas empat komponen utama, yaitu *tag*, antena, *reader* dan basis data. Mekanisme kerja yang terjadi dalam sebuah sistem RFID adalah bahwa sebuah reader frekuensi radio melakukan *scanning* terhadap data yang tersimpan dalam tag, kemudian tag mengirimkan informasi tersebut secara nirkabel melalui antena ke sebuah basis data yang menyimpan data yang terkandung dalam tag tersebut.



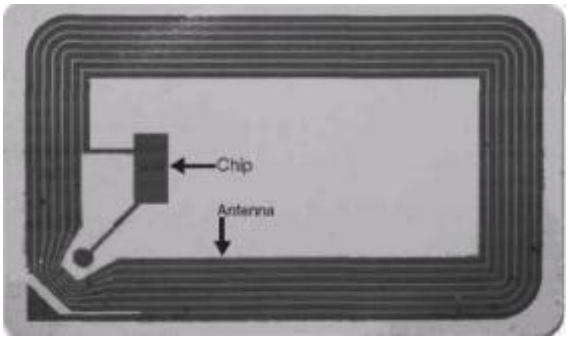
Gambar 2.2 Komponen utama *Radio Frequency Identification* (RFID) [6]

Sistem RFID merupakan suatu tipe sistem identifikasi otomatis yang bertujuan untuk memungkinkan data ditransmisikan oleh peralatan *portable* yang di sebut tag, yang di baca oleh suatu *reader* RFID dan di proses menurut kebutuhan dari aplikasi tertentu. Data yang ditransmisikan oleh *tag* dapat menyediakan informasi identifikasi atau lokasi, atau hal-hal khusus tentang produk-produk yang di beri tag, seperti harga, warna, tanggal pembelian dan lain-lain. Penggunaan RFID dalam aplikasi-aplikasi pelacakan dan akses pertama kali muncul pada tahun 1980an. RFID mendapat perhatian karena kemampuannya untuk melacak objek-objek bergerak. Seiring semakin canggihnya teknologi, maka semakin luas pula penggunaan tag RFID.

Sebuah tag RFID atau transponder, terdiri atas sebuah mikro (microchip) dan sebuah antena. Chip mikro itu sendiri dapat berukuran sekecil butiran pasir, seukuran 0.4 mm. Chip tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya. Tipe memori itu sendiri *dapat read-only*, *read-write*, atau *write-once read-many*. Antena yang terpasang pada chip mikro mengirimkan informasi dari chip ke reader. Biasanya rentang pembacaan diindikasikan dengan besarnya antena. Antena yang lebih besar mengindikasikan rentang pembacaan yang lebih jauh. Tag tersebut terpasang atau tertanam dalam objek yang akan diidentifikasi. *Tag* dapat di-*scan* dengan *reader* menggunakan gelombang radio.

2.2.2 Tag RFID

Tag RFID atau *transponder* terdiri dari sebuah chip dan antena yang dibuat dari bahan silikon. Chip tersebut dapat menyimpan sebuah nomor seri atau informasi berdasarkan jenis memori yang terdapat pada label, dalam hal ini dapat bersifat *read only*, *read write*, atau sekali pemyimpanan data saja. Antena yang melekat pada chip digunakan untuk mentransmisikan informasi dalam chip ke *reader* RFID. Antena yang semakin besar akan memudahkan pembacaan data dalam jangkuan yang cukup luas. Antena akan memancarkan sinyal padafrekuensi radio.



Gambar 2.3 RFID *Tag*[6]

Tag RFID sangat bervariasi dalam hal bentuk dan ukuran. Sebagian tag mudah ditandai, misalnya tag anti-pencurian yang terbuat dari plastik keras yang di pasang pada barang-barang di toko. Tag untuk tracking hewan yang ditanam di bawah kulit berukuran tidak lebih besar dari bagian lancip dari ujung pensil. Bahkan ada tag yang lebih kecil lagi yang telah dikembangkan untuk ditanam didalam serat kertas uang.

Tabel 2.1 Karakteristik umum *Tag* Radio Frequency Identification (RFID)

	***Tag* pasif**	***Tag* semi pasif**	***Tag* aktif**
Catu daya	Eksternal	Internal	Internal
Tipe memori	Umumnya *read-Only*	*Read – write*	*Read – write*
Usia tag	Dapat mencapai 20 tahun	2 sampai 7 tahun	5 sampai 10 tahun

Tag pasif dapat beroperasi pada frekuensi rendah (*low frequency*), frekuensi tinggi (*high frequency*), frekuensi ultra tinggi (*ultrahigh frequency*), atau gelombang mikro (*microwave*). Harga tag pasif lebih murah dibandingkan harga tag jenis lainnya. Perkembangan tag murah ini telah menciptakan revolusi dalam adopsi RFID dan memungkinkan penggunaannya dalam skala yang luas baik oleh organisasi-organisasi pemerintah maupun industri.

Tag semi pasif adalah jenis tag yang memiliki catu daya sendiri (baterai) tetapi bukan untuk membantu meningkatkan komunikasi dengan reader. Dalam hal ini baterai digunakan oleh tag sebagai catu daya untuk melakukan fungsi yang lain seperti pemantauan keadaan lingkungan dan mensuply bagian elektronik internal tag, serta untuk memfasilitasi penyimpanan informasi. Tag jenis ini tidak secara aktif memancarkan sinyal ke *reader*. Sebagian tag semi pasif tetap menerima sinyal dari *reader*.

Tag aktif adalah tag yang selain memiliki antena dan chip juga memiliki catu daya dan pemancar serta mengirimkan sinyal kontinyu. Tag jenis ini biasanya memiliki kemampuan *read* dan *write*, dalam hal ini data tag dapat ditulis ulang

dan dimodifikasi. Tag aktif dapat meningkatkan kecepatan komunikasi dan dapat berkomunikasi pada jarak yang lebih jauh.

Tag juga memiliki berbagai tipe memori, seperti *read-only*, *read-write*, dan *write-once read-many*. *Tag read-only* memiliki kapasitas yang cukup rendah dan mengandung data yang tidak dapat di ubah. *Tag read-write* memungkinkan data untuk diubah (diperbaharui) ketika diperlukan, akan tetapi *tag* ini akan membutuhkan memori yang cukup besar dan biaya yang cukup besar. *Tag writeonce read-many* memungkinkan data untuk di simpan sekali, dan dapat di baca seterusnya.

2.2.3 Pembaca RFID (RFID Reader)

Pembaca RFID merupakan komponen *scanning* yang mampu membaca *tag*, dan kemudian membandingkan hasilnya dengan basis data. Pembaca memiliki sebuah antena untuk berkomunikasi dengan *tag*. Ketika sebuah pembaca memancarkan gelombang radio, semua *tag* akan merespon terhadap frekuensinya. Selain itu, pembaca RFID memiliki kemampuan untuk berkomunikasi dengan *tag* tanpa harus terlihat *tag* nya,bergantung pada frekuensi radio yang dipancarkan dan tipe dari *tag* nya (aktif, pasif, atau semi pasif).

Pembaca dapat memroses banyak data sekaligus, yang memungkinkan untuk menambah waktu dalam pembacaan dan pemrosesan. Pembaca ini juga dapat di bawa kemana-mana. Pembaca RFID dapat dibedakan berdasarkan kapasitas penyimpanannya, kemampuan pemrosesan, dan frekuensi yang dapat dibacanya.

2.2.4 Basis data RFID

Basis data RFID adalah sebuah sistem informasi terakhir yang melacak dan mengandung informasi-informasi dari benda dan/atau barang yang di-*tag*. Informasi yang disimpan pada basis data ini bisa berisi identifikasi benda, deskripsi, pembuatan, dan lokasi dari benda tersebut. Dalam hal ini, tiap informasi yang di simpan bergantung dari penggunaan RFID. Basis data ini juga dapat dihubungkan dengan jaringan-jaringan lainnya seperti jaringan lokal yang bisa menghubungkan dengan basis data RFID melalui jaringan internet. Koneksi ini juga memungkinkan untuk penyebaran data antar basis data lokal.

2.2.5 Frekuensi radio RFID

Pemilihan frekuensi yang digunakan sangat memengaruhi karakteristik dari sistem RFID. Secara umum frekuensi sangat menentukan kecepatan komunikasi dan jarak *tag* bisa di baca oleh pembaca RFID. Semakin besar frekuensi, maka semakin jauh keterbacaannya. Dalam pemilihan frekuensi juga sangat berpengaruh karena tiap frekuensi memiliki cirri khas masing-masing. Contohnya, untuk gelombang frekuensi rendah dapat menembus dinding lebih baik dari pada frekuensi tinggi, tetapi frekuensi tinggi memiliki kecepatan penyaluran data yang lebih cepat.

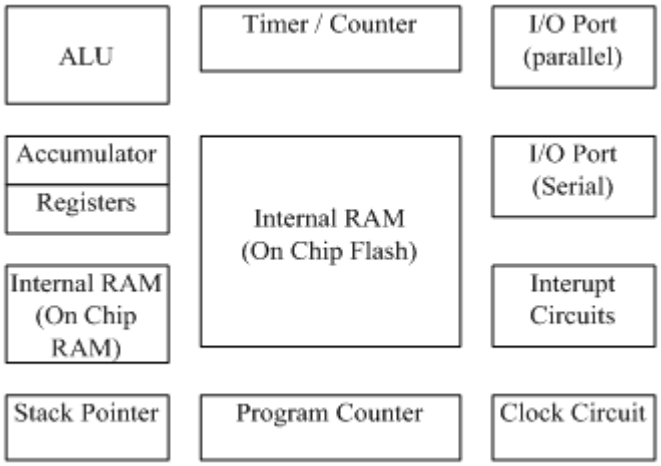
Tabel 2.2 Frekuensi RFID

Gelombang	Frekuensi	Jarak baca
LF	125 KHz	45 cm dan kecepatan pembacaan rendah
HF	13.56MHz	91 cm dan kecepatan pembacaan sedang
UHF	860-930 MHz	Hingga 457cm dan kecepatan pembacaan tinggi
Gelombang mikro	2.45/5.8 GHz	91cm dan kecepatan pembacaan tinggi

Secara umum, ada empat frekuensi utama yang digunakan untuk system RFID, yaitu frekuensi rendah, tinggi, sangat tinggi, dan gelombang mikro. Frekuensi rendah memiliki lebar pita dari 125 KHz hingga 134 KHz. Frekuensi ini cocok digunakan untuk jarak pendek. Frekuensi tinggi beroperasi pada 13.56 megahertz (MHz). RFID dengan frekuensi tinggi dapat menstabilkan pentransmisian pada jarak yang cukup jauh, dan mengurangi resiko ketidaktepatan pembacaan tag. Frekuensi yang sangat tinggi bekerja pada frekuensi 900 MHz yang memungkinkan untuk pembacaan yang lebih jauh lagi dan lebih sensitive terhadap faktor lingkungan. Frekuensi pada ambang gelombang mikro memiliki frekuensi antara 2.45 dan 5.8 gigahertz (GHz), yang memungkinkan pemantulan gelombang radio sehingga pembacaan semakin baik.

2.3 Mikrokontroller

Mikrokontroller merupakan sebuah personal computer yang di tanam dalam sebuah chip [8]. Desainnya menggabungkan sebuah fitur yang terdapat dalam sebuah CPU mikroprosesor seperti ALU, stack pointer, dan register. Juga terdapat fitur lain yang membuatnya seperti komputer keseluruhan seperti RAM, ROM, I/O serial, *counters*, dan *clock-circuit*[8]. Mikrokontroller juga memiliki fungsi untuk mengambil data, melakukan komputasi, dan mengontrol keseluruhan data di dalam lingkungannya. Penggunaan utama mikrokonroller yaitu untuk mengontrol pengoprasian mesin menggunakan program tetap yang di simpan di dalam ROM dan tidak berubah selama sistem tersebut masih di pakai [8]. Gambar 2.15 merupakan blok diagram mikrokontroller.



Gambar 2.4 Blok diagram mikrokontroller [8]

Banyak sekali varian mikrokontroller yang di jual dipasaran, terdiri dari keluarga pabrikan seperti AVR, MCS51, dan PIC. Beberapa keluarga mikrokontroller tersebut memiliki keunggulan dan kelemahan masing-masing. Namun untuk penilitian kali ini, penulis menggunaka mikrokontroller yang sudah

tergabung dengan modul WIFI. Mikrokontroller tersebut sudah memiliki sistem minimum sendiri, atau lebih di kenal sebagai NodeMCU.

2.3.1 NodeMCU ESP8266

Sejarah lahirnya NodeMCU berdekatan dengan rilis ESP8266 pada 30 Desember 2013, Espressif Systems selaku pembuat ESP8266 memulai produksi ESP8266 yang merupakan SoC WIFI yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan NodeMCU dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama nodemcu-firmware ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-*commit* file dari board ESP8266, yang diberi nama devkit v.0.9 [9].

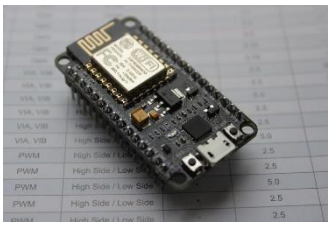
Berikutnya di bulan yang sama. Tuan PM memporting pustaka *client* MQTT dari Contiki ke platform SOC ESP8266 dan di-*commit* ke project NodeMCU yang membuatnya mendukung protokol IOT MQTT melalui Lua. Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika Devsaurus memporting u8glib ke project NodeMCU yang memungkinkan NodeMCU bisa mendrive display LCD, OLED, hingga VGA. Demikianlah, project NodeMCU terus berkembang hingga kini berkat komunitas open source dibaliknya, pada musim panas 2016 NodeMCU sudah terdiri memiliki 40 modul fungsionalitas yang bisa digunakan sesuai kebutuhan developer[9].

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, node mcu 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.

Beberapa pengguna awal masih cukup bingung dengan beberapa kehadiran *board* NodeMCU. Karena sifatnya yang *open source* tentu akan banyak produsen yang memproduksinya dan mengembangkannya. Secara umum ada tiga produsen NodeMCU yang produknya kini beredar di pasaran: Amica, DOIT, dan Lolin/WeMos. Dengan beberapa varian board yang diproduksi yakni V1, V2 dan V3. Tabel 2.3 merupakan tabel produk NodeMCU.

Tabel 2.3 Jenis NodeMCU[10]

Versi NodeMCU	**Generasi**	**Spesifikasi**	**Gambar**
NodeMCU V1 / board v.0.9	Generasi Pertama	• Berdimensi 47mm x 31mm • Flash memory 4MB • IC Serial CHG340 • Chip ESP12	
NodeMCU V2/ board v 1.0 (official)	Generasi Kedua	• Berdimensi 47mm x 24mm • Flash memory 4MB • IC Serial CP2102 • Chip ESP12E	

NodeMCU V3/ board v 1.0 (unofficial)	Generasi Ketiga	• Berdimensi 32mm x 25mm • Flash memory 4MB • IC Serial CH340G • Chip ESP12E	

Spesifikasi teknis dari setiap NodeMCU dengan Mikrokontroller yang berbeda pasti akan berbeda antara satu sama lain, perbedaannya terletak seperti pada I/O, memori, clock dan yang lainnya. Contohnya ukuran yang berbeda antara ketiga NodeMCU tersebut, perbedaannya pun terlihat dari spesifikasi NodeMCU tersebut.

2.3.2 Arduino IDE

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 *microcontroller* Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan *microcontroller*.

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.

Program yang ditulis dengan menggunaan Arduino Software (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi ***.ino***. Teks editor pada Arduino Software memiliki fitur seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan kamu dalam menulis kode program.

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Software* Arduino IDE, menunjukan *board* yang terkonfigurasi beserta COM *Ports* yang digunakan.



Gambar 2.5 Tampilan Awal Arduino IDE

2.4 *Global Positioning System* (GPS)

GPS merupakan sebuah sistem navigasi dan posisi yang berbasis satelit yang mengorbit pada bumi dengan ketinggian sekitar 11.000 mil dari bumi. Satelit yang mengitari bumi pada orbit pendek ini terdiri 24 susunan satelit, dimana 21 satelit digunakan secara aktif dan 3 buah satelit digunakan sebagai cadangan apabila terdapat satu atau lebih satelit yang gagal berfungsi.

GPS yang dimaksudkan pada perancangan alat ini adalah GPS *receiver* yang mana dapat menitik posisikan manapun di bumi, sehingga dapat melacak posisi keberadaan sekarang. GPS dapat beroperasi selama 24 jam dan dapat digunakan di seluruh wilayah muka bumi ini, karena GPS menggunakan satelit yang selalu mengorbitkan pada bumi selama 24 jam.

Sistem GPS menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima. Ada tiga bagian penting dari sistem ini, yaitu bagian kontrol, bagian angkasa dan bagian pengguna. Bagian *space segment* (luar angkasa) yaitu satelit, terdapat 24 satelit aktif, 6 *orbital planes* dengan inklinasi (sudut antara bidang yang menjadi acuan dengan bidang yang diukur kemiringannya) sebesar 55°, dengan lama waktu 12 jam periode orbital, tinggi 20.000 km, dengan kecepatan aproksimasi satelit sebesar 4 km/detik [11].

Pesawat penerima GPS menggunakan sinyal satelit untuk melakukan triangulasi posisi yang hendak ditentukan dengan cara mengukur lama perjalanan waktu sinyal dikirimkan dari satelit, kemudian mengalikannya dengan kecepatan cahaya untuk menentukan secara tepat berapa jauh pesawat penerima GPS dari setiap satelit. Dengan mengunci sinyal yang ditransmit oleh satelit minimum 3

sinyal dari satelit yang berbeda, pesawat penerima GPS dapat menghitung posisi tetap sebuah titik yaitu posisi lintang dan bujur bumi (*Latitude* & *Longitude*) atau sering disebut dengan 2D *fix*.

Penguncian sinyal satelit yang keempat membuat pesawat penerima GPS dapat menghitung posisi ketinggian titik tersebut terhadap muka laut rata-rata (*Mean Sea/Level*) atau disebut 3D *fix* dan keadaan ini yang ideal untuk melakukan navigasi. Faktor – faktor yang dapat menurunkan sinyal GPS dan mempengaruhi akurasi antara lain yaitu:

1. Kesalahan orbital, merupakan kesalahan di mana posisi orbit satelit yang dilaporkan oleh pengontrol satelit tidak sama dengan posisi orbit satelit yang sebenarnya.
2. Penundaan dari ionosfer dan troposfer. Sinyal satelit melambat saat melewati atmosfer. Sistem GPS menggunakan model yang sudah terpasang yang menghitung jumlah rata – rata keterlambatan dan mengoreksi kesalahan.
3. *Multipath* merupakan suatu keadaan saat sinyal yang dikirimkan dari satelit diterima oleh antena pada GPS melalui lebih dari satu lintasan yang berbeda. Hal ini terjadi dikarenakan sinyal GPS yang diterima oleh antena GPS mengalami pantulan dari suatu objek seperti gedung – gedung yang sangat tinggi sebelum sampai di penerima Bangunan – bangunan besar, interferensi dari alat – alat elektronik dan juga pepohonan yang rimbun dapat menghambat diterimanya sinyal oleh antena GPS sehingga mengakibatkan kekeliruan penentuan posisi atau mungkin tidak dapat menentukan posisi sama sekali. GPS rata – rata tidak dapat berfungsi dengan baik di dalam ruangan, di bawah tanah

maupun di bawah laut. Hal ini menyebabkan meningkatnya waktu perjalanan sinyal dari satelit sehingga dapat mengakibatkan kesalahan penentuan posisi.

4. Kesalahan jam satelit dan penerima. Ketelitian data ukuran jarak antara satelit dengan penerima akan terpengaruh oleh ketelitian jam satelit dan penerima.
5. Jumlah satelit terlihat. Semakin banyak sinyal satelit yang dapat diterima oleh GPS maka akurasi akan semakin baik. Hal ini dikarenakan semakin banyak sampel yang dilakukan untuk penentuan posisi penerima [11].

2.4.1 Prinsip Kerja *Global Positioning System (GPS)*

Masing – masing wilayah diatas permukaan bumi minimal dapat terjangkau oleh 3 sampai 4 satelit. Setiap GPS dengan teknologi terbaru dapat menerima sampai dengan 12 kanal satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat GPS dapat dengan mudah menerima sinyal yang dikirimkan oleh satelit. Semakin banyak satelit yang diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi. Pesawat penerima GPS menggunakan sinyal satelit untuk melakukan triangulasi posisi yang hendak ditentukan dengan cara mengukur lama perjalanan waktu sinyal dikirimkan dari satelit, kemudian mengalikannya dengan kecepatan cahaya yaitu sebesar $3x10^8$ meter per detik untuk menentukan secara tepat berapa jauh pesawat penerima GPS dari setiap satelit, dengan menggunakan sinyal yang dikirim oleh satelit minimal tiga sinyal dari satelit yang berbeda, pesawat penerima GPS dapat menghitung posisi tetap sebuah titik yaitu posisi lintang (*latitude*) dan bujur bumi (*longitude*). Penggunaan sinyal satelit yang keempat membuat pesawat penerima GPS dapat menghitung posisi

ketinggian titik tersebut terhadap rata – rata permukaan laut dan keadaan ini yang ideal untuk melakukan navigasi [11].

2.4.2 Penentuan Posisi *Global Positioning System* (GPS)

Satelit GPS berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dan mengirimkan sinyal informasi ke bumi. Penerima GPS menerima sinyal informasi tersebut dan menggunakan perhitungan triangulasi untuk menghitung lokasi pengguna dengan tepat. Penerima GPS membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima. Setelah informasi posisi didapatkan sehingga diketahui berapa jarak dari satelit dan posisi pengguna. *Pseudorange* merupakan suatu kode pengamatan yang menggunakan perbedaan antara waktu transmisi di satelit dan waktu penerimaan pada GPS *receiver* di bumi [12].

Persamaan pengamatan untuk *pseudorange* adalah,

$$p = r + c\,(dt - dT) \quad \text{(2.1)}$$

dimana,

p adalah *pseudorange*

r adalah jarak antara satelit GPS dengan *receiver* GPS

c adalah kecepatan cahaya yaitu sebesar $3x10^8$ meter/detik

dt adalah waktu pengiriman sinyal satelit GPS dT adalah waktu penerimaan GPS *receiver*.

Persamaan jarak antara satelit GPS dengan penerima yaitu:

$r = \sqrt{(x_s - x)^2 + (y_s - y)^2}$……..……...…….(2.2)

dimana, r adalah jarak antara satelit GPS dengan penerima

(x_s, y_s) adalah koordinat satelit GPS

(x, y) adalah koordinat penerima [12].

2.4.3 Jarak Antara Dua Titik Koordinat *Latitude* dan *Longitude*

Jarak antara dua titik koordinat *latitude* dan *longitude* dapat dihitung melalui persamaan berikut [13].

1. Radian sudut koordinat = nilai *latitude* / *longitude* x π/180……...…….(2.3)
   dimana nilai $\pi = {}^{22}/_{7}$.
2. Cos (d) = sin(radian *latitude* 1) x sin(radian *latitude* 2) + cos(radian *latitude* 1) x cos(radian *latitude* 2) x cos(radian *longitude* 1 – radian *longitude* 2)…...(2.4)
   dimana d adalah jarak antara dua titik.
3. Radian sudut jarak antara dua titik (d) =arc cos(cos(d))…….....(2.5)
   Jarak antara dua tempat (r) = jari – jari bumi x radian sudut (d).....(2.6)
   dimana nilai dari jari – jari bumi = 6.378,137 km.

2.4.4 Mengukur Jarak Dua Titik dengan Titik Koordinat

Jarak merupakan selisih dari posisi awal dan posisi akhir. Pada kasus ini akan dilakukan pengukuran jarak dengan hanya mengetahui koordinat bumi yang

disebut kordinat lintang dan bujur dengan menggunakan metode *Euclidean*. Nilai koordinat bumi yang digunakan metode *Euclidean* merupakan bentuk Decimal Degree yang dituliskan seperti berikut:

• Latitude : -7.275515 • Longitute: 112.794895

Formula *Euclidean* adalah sebagai berikut:

Jarak = $\sqrt{(lat2 - Lat1)2 + (long2 - Long1)2}$ *x 111.319 Km*

Keterangan :

$Lat1$ = Latitude Referensi

$Long1$ = Longitude Referensi

$Lat2$ = Pembacaan Latitude modul GPS

$Long2$ = Pembacaan Longitude modul GPS

2.5 Android

Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan atau penghubung antar piranti *(device)* dan penggunanya, sehingga pengguna dapat berinteraksi dengan *device*nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device.*

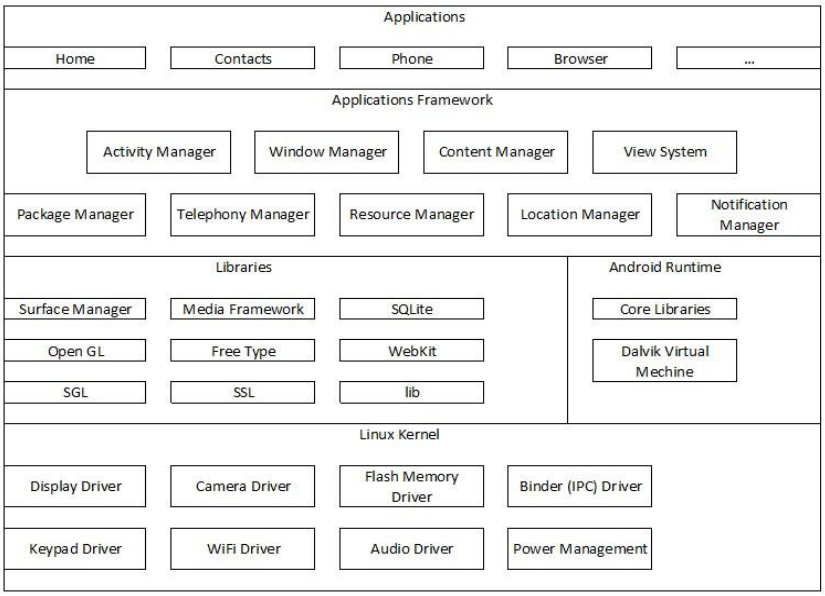
Android merupakan sistem operasi yang bersifat *open source*. Disebut *open source* karena *source code* dari sistem operasi Android dapat dilihat, di-*download* dan dimodifikasi secara bebas. Paragdigma *open source* ini memudahkan pengembangan teknologi Android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi [14].

2.5.1 Arsitektur Android

Tentunya Android mempunyai arsitektur tersendiri, arsitektur Android dapat digambarkan seperti pada Gambar 2.6. Arsitektur Android dapat dijelaskan sebagai berikut.

a. Application dan Widgets

Application dan *Widgets* merupakan *layer* yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien *email*, kalender, *browser*, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java.



Gambar 2.6 Arsitektur Android [15]

b. Applications Frameworks

Applications Framework merupakan *layer* dimana para pembuat aplikasi menggunakan komponen-komponen yang ada untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk didalam

Applications Framework adalah sebagai berikut.

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notivication Manager*
5. *Activity Manager*

c. Libraries

Libraries merupakan *layer* tempat fitur-fitur Android berada. Pada umumnya *libraries* diakses untuk menjalankan aplikasi. Beberapa *library* yang terdapat pada Android diantaranya adalah *libraries* media untuk memutar media *video* atau *audio*, *libraries* untuk menjalankan tampilan, *libraries graphic*, *libraries* SQLite untuk dukungan database.

d. Android *RunTime*

Android *RunTime* merupakan *layer* yang membuat aplikasi Android dapat dijalankan. Android *RunTime* dibagi menjadi dua bagian.

1. *Core Libraries*: berfungsi untuk menerjemahkan bahasa Java/C.
2. *Dalvik Birtual Machine*: sebuah mesin virtual yang dioptimalkan untuk menjalankan fungsi-fungsi pada Android secara efisien.

e. Linux Kernel

Linux Kernel adalah layer dimana inti dari sistem operasi Android itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi Android lainnya. Linux Kernel ini berfungsi sebagai jembatan penghubung antara perintah-perintah yang dibuat terhadap hardware yang digunakan pada Android [15].

2.5.2 Komponen Android

Aplikasi Android dibangun menggunakan bahasa pemograman Java. Untuk membuat aplikasi Android terdapat beberapa komponen utama didalamnya. Ada 4 jenis komponen pada aplikasi Android, setiap jenis komponen memiliki tujuan yang berbeda dan memiliki siklus yang berbeda pula. Dibawah ini adalah komponen Android yang dimaksud.

1. Activites

   Merupakan suatu tampilan *(User Interface)* untuk pengguna melakukan interaksi dengan aplikasi Android. Sebuah aplikasi Android biasanya memiliki lebih dari satu *activity* tergantung tujuan dan desain aplikasi tersebut. Untuk berpindah dari suatu *activity* ke *activity* lain dapat dilakukan dengan suatu *event* seperti klik tombol.

2. Service

   Service ini tidak memiliki *Graphic User Interface* (GUI) seperti komponen *activity* tetapi *service* berjalan secara *background.* Maksudnya jika kita sedang menjalankan satu aplikasi, maka kita bisa mumbuka aplikasi yang lain selama aplikasi yang pertama masih berjalan.

3. Broadcast Receiver

   Broadcast Receiver adalah komponen aplikasi yang menanggapi *system-wide broadcast announcements.* Contohnya memunculkan notifikasi ketika adanya aplikasi baru yang ter-*install*. *Broadcast Receiver* juga tidak memiliki tampilan, tetapi dapat menjalankan suatu *activity* atau menampilkan notifikasi di Notifikasi Bar.

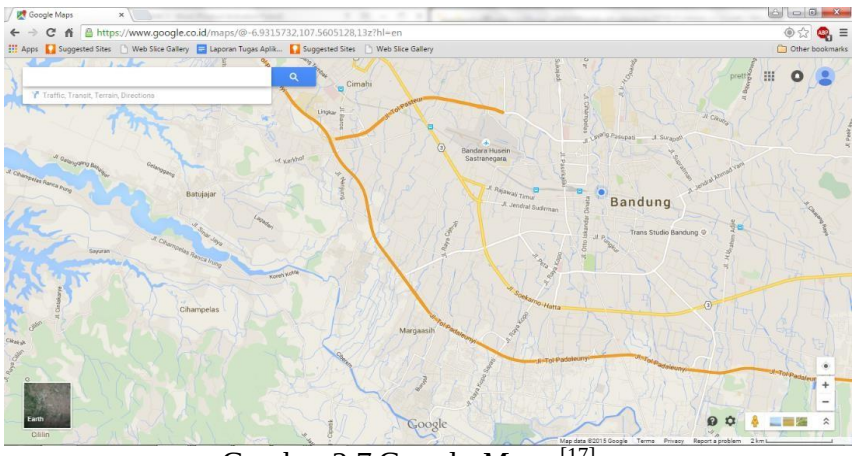
4. Intent

Intent merupakan pesan *Asynchronous* yang memungkinkan aplikasi untuk memberi *request* secara fungsional dari komponen yang berbedabeda didalam sistem Android [16].

2.6 *Google Maps*

Google *Maps* merupakan sebuah layanan gratis dari Google mengenai berbagai macam teknologi pemetaan. Google *Maps* menampilkan peta dunia secara digital sehingga dapat dengan mudah mengetahui suatu lokasi tertentu. Google *Maps* menyediakan beberapa keunggulan seperti peta jalan, menunjukan rute untuk suatu perjalanan yang diinginkan baik itu melalui jalur mobil sampai pejalan kaki.

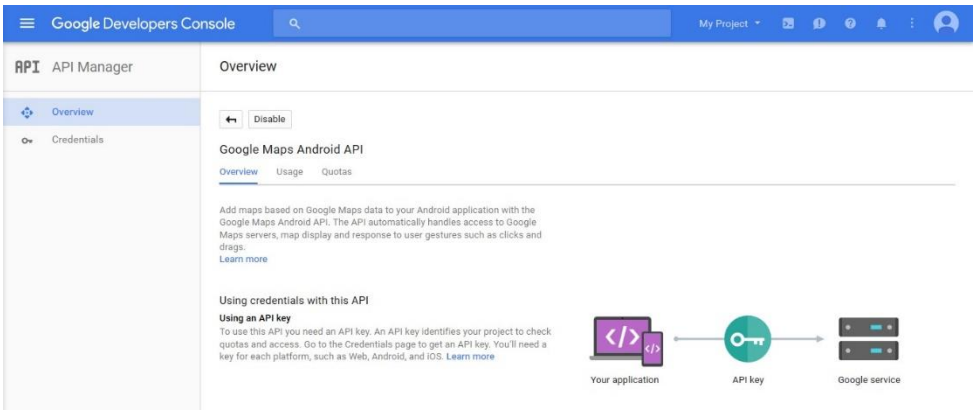
Google *Maps* dibuat dengan menggunakan kombinasi dari gambar peta, *database*, serta objek-objek interaktif yang dibuat menggunakan bahasa pemograman *Hyper Text Markup Language* (HTML), Java dan beberapa bahasa pemograman lainnya. Gambar-gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan *database* pada *web server* Google untuk menampilkan gambar yang diminta [17].



Gambar 2.7 Google *Maps* [17]

2.7 *Google Maps Application Programing Interface*

Google *Maps* API merupakan sebuah layanan yang memungkinkan Google *Maps* agar dapat diintegrasikan kedalam aplikasi yang memang membutuhkan fitur dari Google *Maps* ini. Dengan menggunakan Google *Maps* API ini memudahkan para pengembang dalam membuat aplikasi yang membutuhkan teknologi peta didalamnya. Dengan menggunakan Google *Maps* API ini sudah tersedia semua yang dibutuhkan terkait teknologi peta *digital* [17].



Gambar 2.8 Google *Maps* API [17]

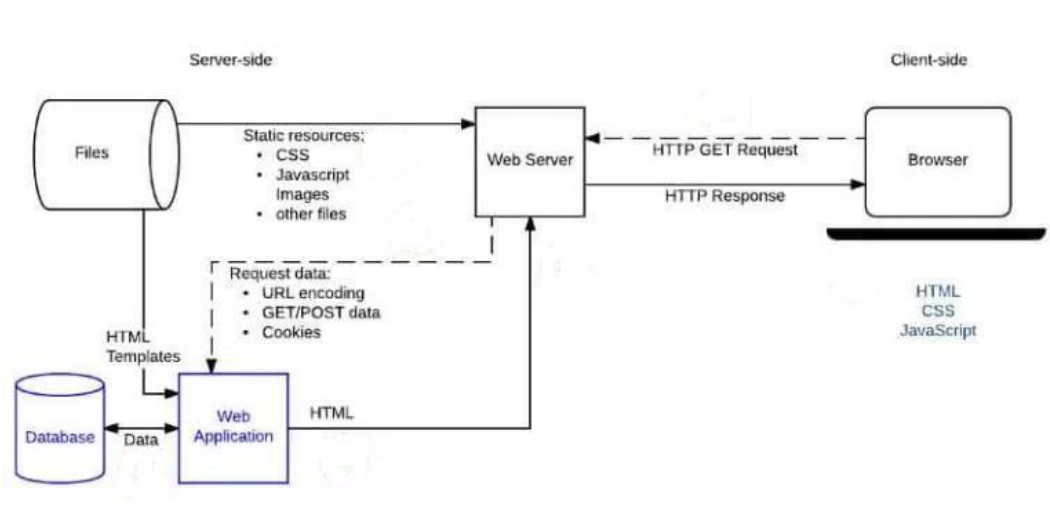
2.8 Web Server

Fungsi utama Web *server* adalah untuk melakukan atau mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sesuai dengan keinginan. Berkas yang dikirimkan bisa dalam berntuk teks, video, gambar, *file* dan banyak lagi. *Webserver* juga bisa dapat berfungsi menerima permintaan dari HTTP atau HTTPS biasanya dikenal dengan nama *web browser* (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML.

Pengguna internet atau *user* dapat dengan mudah membaca dari dokumen ke dokumen yang lainnya hanya dengan cara klik beberapa bagian dari beberapa

halaman dokumen *web* tersebut. Proses yang dimulai dari permintaan *web client* atau *browser*, akan diterima oleh *web server*, selanjutnya diproses, dan kemudian dikembalikan hasil proses tadi oleh *web server* kepada web *client* lagi hal ini dilakukan secara cepat sekali dan transparan. Secara garis besar web *server* hanya akan memproses semua permintaan yang diperolehnya dari web *client*. namun tentunya dengan kecepatan yang sangat tinggi hingga kita tidak tahu bagaimana proses sebenarnya terjadi.

Cara kerja dari *webserver* sendiri seperti dijelaskan sebelumnya *webserver* bekerja sebagai penyedia protokol atau jalur agar dapat mentransfer data ke *database*, untuk lebih jelasnya dapat diamati pada gambar di bawah ini.



Gambar 2.9 Blok Sistem Kinerja Webserver [18]

Webserver sebagai penyedia layanan protokol dimana webserver dapat menerima HTTP/HTTPS request dari browser kemudian Webserver akan menampilkan hasil dari request yang diinginkan oleh browser, menjadi data berupa HTML, di dalam HTML terdapat tampilan seperti berupa gambar, teks, javascript dan file lainnya.

2.9 Firebase Sebagai Web Server

Firebase adalah suatu layanan yang disediakan oleh google untuk mempermudah para pengembang aplikasi berbasis web server untuk mengembangkan aplikasinya.



Gambar 2.10 Logo Firebase

Dengan adanya Firebase, pengembang aplikasi dapat fokus mengembangan aplikasinya tanpa harus memberikan usaha yang besar dalam urusan database berbasis web server. Beberapa fitur yang dimiliki oleh Firebase adalah sebagai berikut :

1. Firebase Analytics
2. Firebase Cloud Messaging and Notification
3. Firebase Authentication
4. Firebase Remote Config
5. Firebase Real Time Database
6. Firebase Crash Reporting

Dari banyaknya fitur yang disediakan oleh Firebase, penulis akan fokus hanya pada fitur Real Time Database karena pada sistem yang akan dibuat hanya menggunakan Fitur Real Time Database yang terdapat pada Firebase. Real Time Database merupakan database yang di-*host* di cloud. Data akan disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika

membuat aplikasi lintas platform dengan SDK Android, iOS, dan JavaScript, akan berbagi instance Real Time Database dan menerima update data terbaru secara otomatis.

Real Time Database ini memiliki sistem kerja seperti protokol HTTP yang bekerja apabila ada permintaan. Pada Real Time Database ini sinkronisasi data setiap kali data berubah sebagai permintaan. Ketika data berubah maka perangkat yang terhubung ke database akan menerima update dengan waktu tunda hingga mili detik.

2.10 Perangkat Lunak Thunkable

Thunkable adalah platform dimana setiap orang dapat membuat aplikasi handphone sesuai dengan keinginan pengguna. Thunkable bekerja sama dengan Firebase, selain itu pada perangkat lunak ini tersedia fitur google maps yang dibutuhkan oleh penulis. Berikut logo thinkable dapat dilihat pada gambar 2.11.



Gambar 2.11 Logo Thunkable

Thunkable menyediakan pembuatan aplikasi berplatform Android dan iOS. Interface yang disediakan oleh Thunkable sangat *user-friendly* ini dikarenakan kita tidak perlu memasukan code untuk memprogram aplikasi yang akan kita buat. Kita hanya perlu menghubungkan blok-blok program dengan cara *drag and drop* blok-blok program yang disediakan oleh thunkable. Thunkable sendiri merupakan versi yang lebih lengkap dari MIT App Inventor maupun Apply Builder (platform sejenis Thunkable). Thunkable ini lebih difokuskan kepada pengembang yang akan

membuat aplikasi untuk kebutuhan sendiri atau bersifat komersil. Selain itu thunkable menyediakan fitur yang lebih lengkap dibandingkan platform yang lainnya.