

BAB II

TINJAUAN PUSTAKA

2.1 *Internet of Things (IoT)*

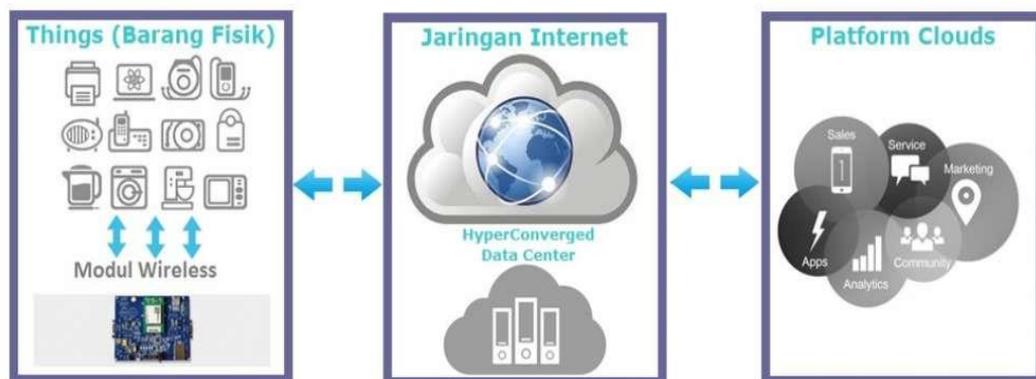
Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus^[1]. Dapat memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen^[2].

Internet Of Things (IoT) adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung, misalnya CCTV yang terpasang di sepanjang jalan dihubungkan dengan koneksi internet dan disatukan di rung kontrol yang jaraknya mungkin puluhan kilometer. atau sebuah rumah cerdas yang dapat dimanage lewat smartphone dengan bantuan koneksi internet. Pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komuniikasi, dan server sebagai pengumpul informasi yang diterima sensor dan untuk analisa.

Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami Internet of Things sebut saja Intel, Microsoft, Oracle, dan banyak lainnya. Banyak yang memprediksi bahwa pengaruh Internet of Things adalah “*The Next Big Thing*” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa digali. Contoh sederhana manfaat dan implementasi dari *Internet of Things* misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus distok lagi^[2].

2.1.1 Cara Kerja *Internet of Things* (IoT)

Konsep IoT ini sebetulnya cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yaitu barang fisik yang dilengkapi modul IoT, Perangkat Koneksi ke Internet seperti Modem dan *Router Wireless*, serta *Cloud Data Center* tempat untuk menyimpan aplikasi beserta data base.



Gambar 2.1 Konsep *Internet of Things* (IoT)^[2]

Dasar prinsip kerja perangkat IoT adalah benda di dunia nyata diberikan identitas unik dan dapat dikali di sistem komputer dan dapat di representasikan dalam bentuk data di sebuah sistem komputer. Pada awal-awal implementasi gagasan IoT pengenalan yang digunakan agar benda dapat diidentifikasi dan dibaca oleh komputer adalah dengan menggunakan kode batang (*Barcode*), Kode QR atau *QR Code* dan *Identifikasi Frekuensi Radio (RFID)*. Dalam perkembangannya sebuah benda dapat diberi pengenalan berupa IP address dan menggunakan jaringan internet untuk bisa berkomunikasi dengan benda lain yang memiliki pengenalan IP address. Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapa pun. Internetlah yang menjadi penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung^[2].

2.1.2 Penerapan *Internet of Things* (IoT)

Menurut beberapa penelitian *Internet of Things* sudah banyak diterapkan di beberapa bidang keilmuan dan industri, seperti dalam bidang ilmu kesehatan, informatika, geografis dan beberapa bidang ilmu lain^[3]. Contoh kasus yang berkaitan dengan penerapan IoT yaitu mesin dibuat agar pekerjaan manusia menjadi lebih mudah, pada awalnya mesin dibuat hanya untuk membantu manusia dan dioperasikan secara manual, lambat laun mesin dapat beroperasi sendiri (otomatis), tetapi dalam perkembangannya pemanfaatan mesin sebagai alat dalam sebuah sistem akan menemui kendala jika sudah menyangkut jarak dan waktu. Dengan jarak yang begitu jauh maka mesin tidak akan bisa berinteraksi dengan mesin yang lain, untuk mengatasi hal inilah diterapkan gagasan *Internet of Things* dimana semua mesin dengan pengenalan IP address dapat menggunakan jaringan internet sebagai media komunikasi (saling bertukar data)^[2].

Perkembangan pada teknologi mobile juga ikut memberi sumbangsih kepada perkembangan *Internet of Things* yaitu dilakukannya penelitian tentang privasi di bidang pengamatan wilayah, mendeteksi lokasi berdasarkan Location Based Service sehingga seseorang bisa merasa nyaman menggunakan perangkat mobile tanpa harus terganggu privasi pribadi^[3].

2.2 Aplikasi Mobile

Aplikasi Mobile adalah perangkat lunak yang berjalan pada perangkat mobile seperti smartphone atau tablet PC. Aplikasi Mobile juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah fungsionalitas dari perangkat mobile itu sendiri. Untuk mendapatkan mobile application yang diinginkan, user dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. *Google Play* dan *iTunes* merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan^[4]. Maka aplikasi mobile dapat diartikan sebagai sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ke tempat yang

lain serta mempunyai ukuran yang kecil. Aplikasi mobile ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, smartpone, dan perangkat sejenisnya^[2].

2.2.1 Android

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *TMobile*, dan *Nvidia*. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler^[5].

Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java^[6].

Android memungkinkan penggunaanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti *Google Play*, *Amazon Appstore*, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga^[6].

2.3 *Global Positioning System (GPS)*

Global Positioning System (GPS) adalah sistem satelit navigasi dan penentuan posisi, dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa bergantung waktu dan cuaca, bagi banyak orang secara simultan. Saat ini GPS sudah banyak digunakan orang di seluruh dunia dalam berbagai bidang aplikasi yang menuntut informasi tentang posisi, kecepatan, percepatan ataupun waktu yang teliti. GPS dapat memberikan informasi posisi dengan ketelitian bervariasi dari beberapa millimeter (orde nol) sampai dengan puluhan meter^[7].

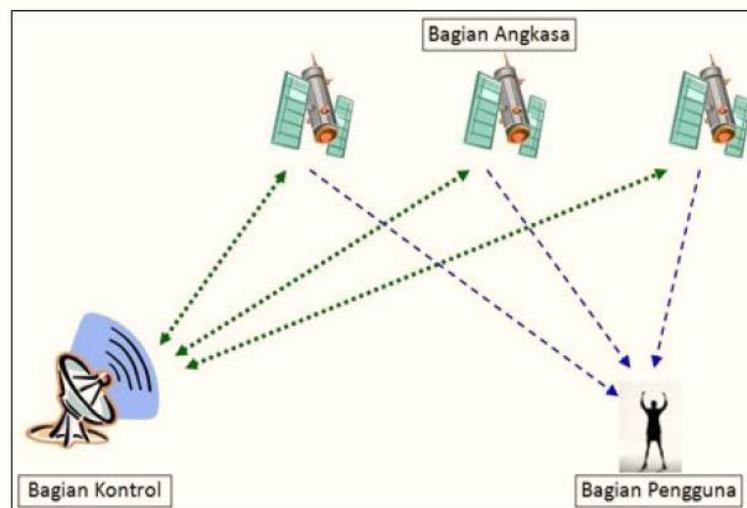
Global Positioning System yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit^[8].

Cara kerja GPS secara logik ada 5 langkah yaitu :

1. Memakai perhitungan "*triangulation*" dari satelit.
2. Untuk perhitungan "*triangulation*", GPS mengukur jarak menggunakan *travel time* sinyal radio.
3. Untuk mengukur *travel time*, GPS memerlukan memerlukan akurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya.
5. Terakhir harus mengoreksi *delay* sinyal waktu perjalanan di atmosfer sampai diterima receiver.

GPS terdiri dari tiga bagian yaitu *space segment* (luar angkasa), *ground segment* (bumi) dan *pengguna segment* (pengguna). Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS. 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 lainlain. 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.



Gambar 2.2 Skema GPS^[17]

Dari beberapa pemakaian GPS di atas dikategorikan menjadi:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dari latitude, longitude, altitude.
3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
5. Komulasi data. GPS *receiver* dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya.

6. *Tracking*. Membantu untuk memonitoring pergerakan obyek. Membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat^[8].

2.4 *Google maps*

Menurut istilah *Google Help*, pengertian *Google maps* adalah layanan gratis peta dan pemetaan digital yang bisa dimanfaatkan untuk mengamati peta dunia melalui browser web. *Google maps* memanfaatkan teknologi *digital imaging*. Seperti foto satelit sehingga kita dapat melihat bagaimana *landscape* planet bumi apabila dilihat dari luar angkasa. Layanan ini gratis dan dapat ditemukan di <http://maps.google.com>^[9].

2.4.1 *Googlemaps API*

Google maps adalah penyedia layanan pemetaan dan kartografi berbasis web. *Google maps* juga menampilkan peta secara tiled map dan menyediakan layanan *script Application Program Interface* yang kaya dan bisa dikembangkan dengan mudah. *Google maps API* merupakan layanan untuk mengintegrasikan *Google maps* pada halaman situs yang dikembangkan secara mandiri. API ini menyediakan fungsifungsi untuk memanipulasi peta dan menambahkan konten pada peta. Layanan ini dikembangkan dalam beberapa versi seperti Javascript dan Flash^[10].

1. API : API adalah kependekan dari *Application programming interface*. Dengan bahasa yang lebih sederhana, API adalah fungsi fungsi pemrograman yang disediakan oleh aplikasi atau layanan agar layanan tersebut bisa diintegrasikan dengan aplikasi yang kita buat.
2. Google MAP API : fungsi fungsi pemrograman yang disediakan oleh *Google maps* agar *Google maps* bisa diintegrasikan kedalam Web atau aplikasi yang sedang buat^[9].

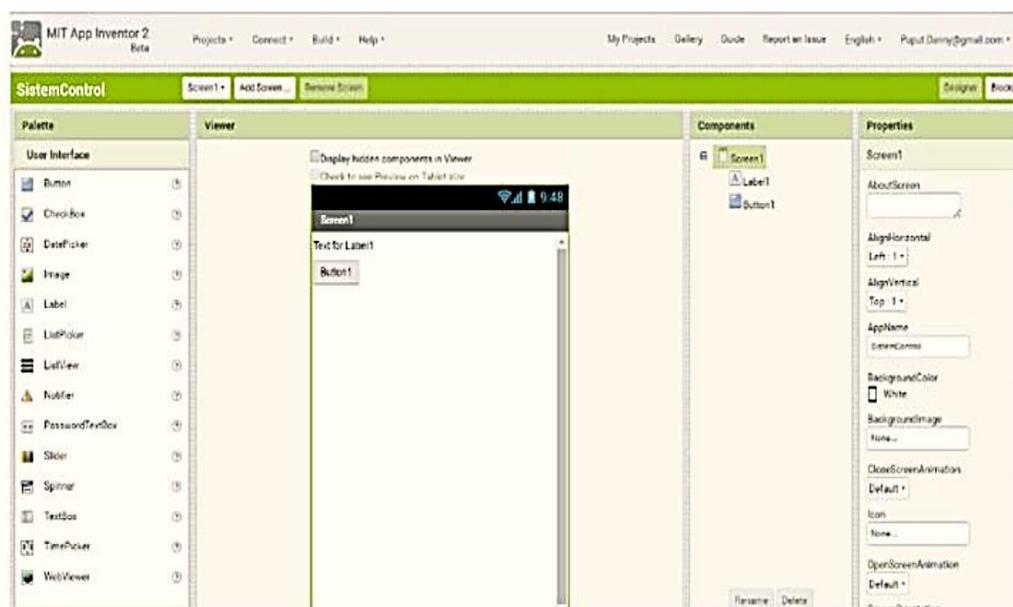
2.5 *Tracking*

Tracking secara harfiah memiliki arti mengikuti jalan, atau dalam arti bebasnya adalah suatu kegiatan untuk mengikuti jejak suatu obyek. Pengertian

tracking atau pemantauan dalam hal ini adalah kegiatan untuk memantau keberadaan mobil berdasarkan posisi yang didapatkan dari peralatan *tracking*^[11]. Pada sistem *tracking* kendaraan, teknologi GPS digunakan untuk memperoleh koordinat kendaraan yang dilengkapi perangkat GPS dan memperbaharui datanya secara real time^[10].

2.6 MIT App Inventor

App Inventor adalah sebuah tool untuk membuat aplikasi android yang berbasis visual *block programming*, sehingga pengguna bisa membuat aplikasi tanpa melakukan *coding*. Visual *block programming* maksudnya adalah dalam penggunaannya user akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi –event handler tertentu dalam membuat aplikasi, dan secara sederhana bisa disebut tanpa menuliskan kode program. Aplikasi *App Inventor* ini pada dasarnya adalah aplikasi yang disediakan oleh google dan sekarang di-maintenance oleh *Massachusetts Institute of Technology* (MIT). Aplikasi ini selesai dibuat pada 12 juli 2010 dan dirilis untuk public pada 31 Desember 2011. App Inventor sekarang dipegang oleh MIT *Centre for Mobile Learning* dengan nama MIT App Inventor^[12].



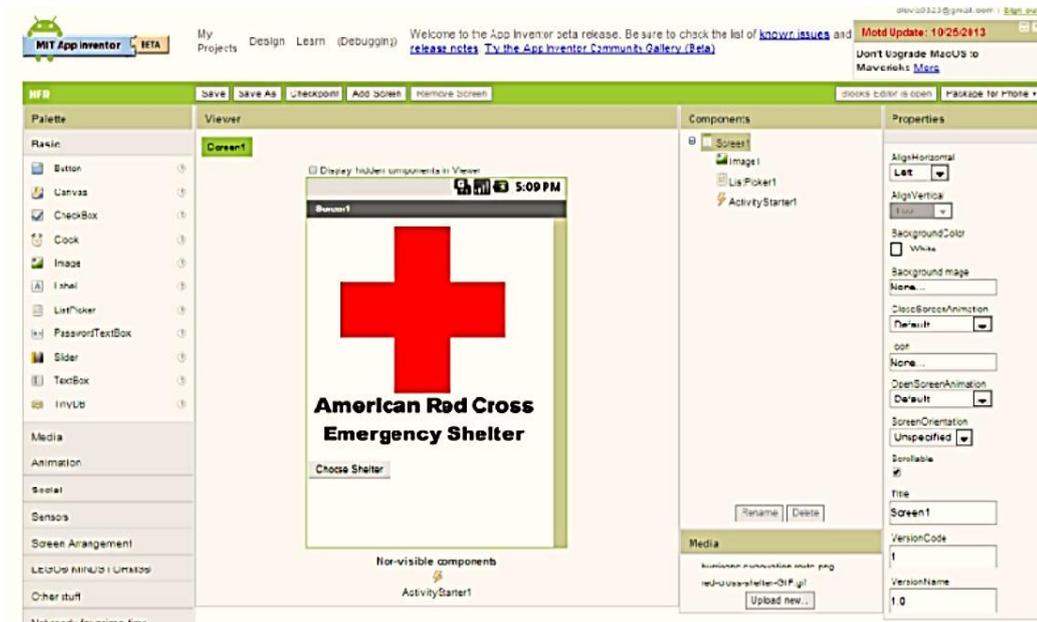
Gambar 2.3 Tampilan MIT App Inventor^[14]

Dengan menggunakan App Inventor ini, ada beberapa aplikasi yang dapat dibuat diantaranya yaitu:

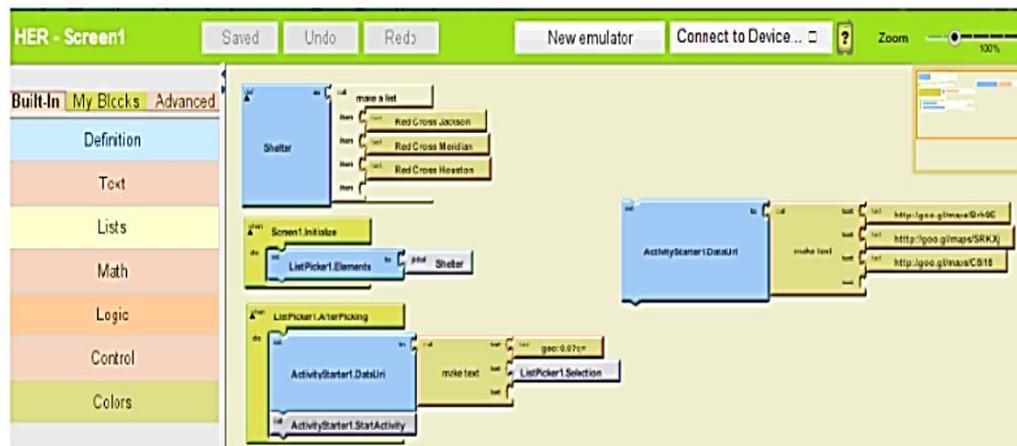
1. Aplikasi game
2. Aplikasi edukasi
3. Aplikasi berbasis *tracking* lokasi
4. Aplikasi SMS
5. Aplikasi berbasis web
6. Aplikasi kompleks^[12]

Untuk membuat aplikasi mobile merupakan tantangan bagi siapa saja tanpa pengalaman pemrograman sebelumnya. Beberapa bahasa pemrograman visual telah dikembangkan untuk mengatasi tantangan ini. MIT App Inventor memungkinkan nonprogrammers untuk membuat aplikasi mobile untuk perangkat yang menjalankan sistem operasi Android dan mampu mengembangkan semua jenis aplikasi, tidak hanya animasi^[13].

MIT App Inventor memiliki dua jendela utama: *Designer* Komponen (Gambar 2.4) untuk membangun antar muka pengguna dan Editor Blok (Gambar 2.5) untuk menentukan perilaku aplikasi. Perancang Komponen memungkinkan pemilihan "komponen" (objek dari kelas tertentu) yang akan ditambahkan, dengan menyeretnya ke tata letak layar, sebagai bagian dari aplikasi. Setelah bagian ditambahkan ke aplikasi, propertinya dapat ditetapkan. Editor Blok menggunakan teka-teki "kerangka" gambar (metode kelas terkait) yang digabungkan untuk menentukan perilaku aplikasi sebagai respons terhadap berbagai peristiwa. Pengguna dapat menguji aplikasi mereka secara langsung di ponsel Android atau emulator dari ponsel yang berjalan di komputer. *MIT App Inventor* server juga dapat menyimpan semua proyek serta memungkinkan pengguna bisa masuk ke akun mereka untuk melanjutkan pekerjaan kapan dan dimana saja. MIT App Inventor sangat kaya fitur, memiliki komponen untuk menerima teks SMS, mengirim pesan, membaca sensor GPS, dan berkomunikasi dengan sumber web eksternal.



Gambar 2.4 Design Komponen Pada MIT App Inventor^[13]



Gambar 2.5 Block Editor Pada MIT App Inventor^[13]

App Inventor memungkinkan pengguna baru untuk memprogram komputer guna membentuk software pada sistem operasi Android^[13]. App Inventor memakai graphical interface, seperti semacam *user interface* pada *Scratch* dan *StarLogo TNG*, yang memungkinkan pengguna untuk mendrag and drop subjek visual guna membentuk aplikasi yang dapat dioperasikan peranti Android. Dalam membentuk App Inventor, Google telah mengadakan penelitian yang berkaitan terhadap komputasi edukasional serta menyelesaikan lingkungan

pengembangan *online Google*. Pada *App Inventor* ini ada beberapa komponen yang terdiri atas:

1. *Komponen Desainer*

Komponen *desainer* terdiri atas 5 bagian, diantaranya *palette*, *viewer*, *component*, *media* dan *properties*, serta beroperasi di *browser* yang dipakai untuk menentukan komponen yang diperlukan juga mengatur propertinya.

2. *Block Editor*

Block Editor berjalan di luar *browser* dan digunakan untuk membuat dan mengatur *behaviour* dari komponen-komponen yang kita pilih dari komponen *desainer*.

3. *Emulator* yang digunakan untuk menjalankan dan menguji *project* yang telah dibuat^[14].

App inventor berbasis tarik visual tidak mengharuskan pengguna untuk menghafal atau mengingat kembali instruksi atau kode program, dan komponen blok acara benar-benar tersedia sehingga pengguna dapat menggunakan aplikasi dengan mudah. Sebagai suatu objek, pengguna dapat mengumpulkannya dengan komponen-komponen dan blok-blok yang saling mendukung fungsi tersebut. Pengguna harus meletakkannya seperti *puzzle* dan mengubah properti, misalnya memberikan nomor untuk mengatur timer, dll. Penangan kejadian membantu menangani setiap peristiwa dalam proses.

Sebelum memulai pembuatan disini ada beberapa menu yang perlu diketahui :

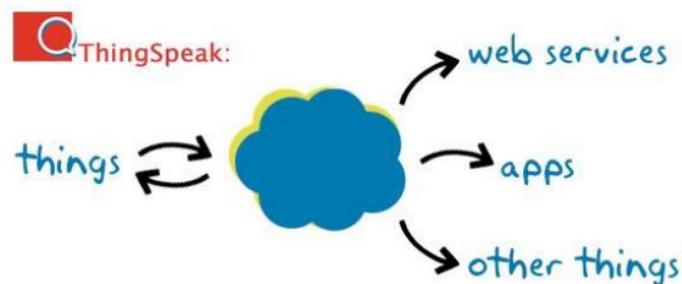
1. *Project* adalah menu awal berisi antara lain *start new project*, *save project*, *Delete Project*.

2. *Connect* adalah menu untuk menghubungkan project kita dengan menggunakan media antara lain. *MIT App Inventor Companiaon* (bisa didownload di *Google Playstore*). Berfungsi sebagai *emulator* untuk *project* kita. Jadi kita bisa langsung mencoba aplikasi yang kita buat tanpa harus mendownloadnya terlebih dahulu.

3. *Build* adalah menu untuk *download* aplikasi kita. Dengan *save project* kita ke computer atau melalui scan barcode yang akan menuju ke link download yang bisa langsung di download dari *Smartphone*.
4. *Add Screen* adalah untuk menambah screen jika membutuhkan beberapa *screen*.
5. *Menu Designer* untuk mendisain aplikasi kita dalam pengaturan layout, tombol, gambar dan lain-lain.
6. *Block* akan berisi logika dari aplikasi kita.

2.7 Thingspeak

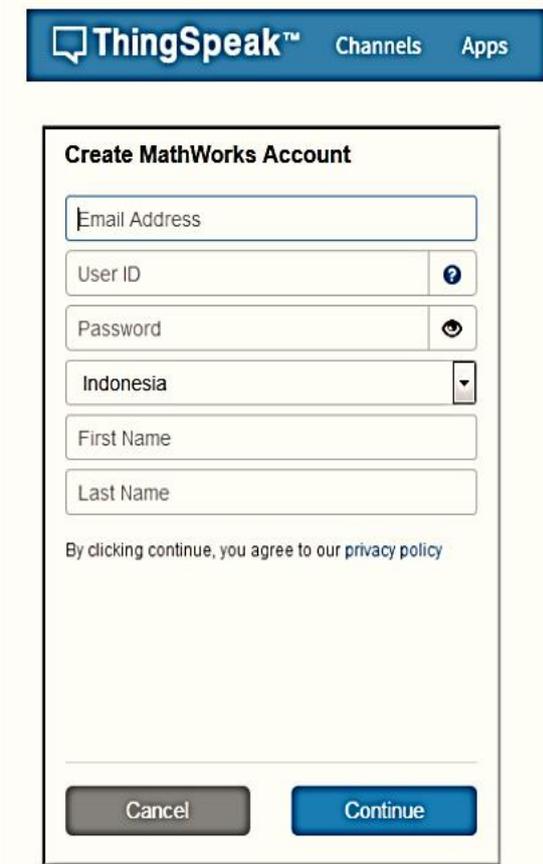
Thingspeak merupakan sebuah layanan internet yang menyediakan layanan untuk pengaplikasian "*Internet of Things*". *Thingspeak* merupakan layanan yang berisi aplikasi dan API yang bersifat *open source* untuk menyimpan dan mengambil data dari berbagai perangkat yang menggunakan *Hypertext Transfer Protocol* (HTTP) melalui Internet atau melalui *Local Area Network* (LAN). Dengan menggunakan *Thingspeak*, seseorang dapat membuat aplikasi *logging sensor*, aplikasi pelacakan lokasi, dan jaringan sosial dari segala sesuatu yang terhubung ke internet dengan pembaruan status^[15].



Gambar 2.6 Tampilan *Thingspeak* Sebagai *Cloud Server* ^[15]

Thingspeak merupakan server *Internet of Things* yang cukup diminati karena konfigurasinya yang mudah, untuk memulai *Thingspeak* pertama-tama kita harus membuat akun pada *server* tersebut dengan alamat email yang kita miliki. Setelah mendapat akun pada *server* tersebut, maka langkah selanjutnya ialah

mengatur *dashboard* atau tampilan awalnya dengan perangkat kita yang akan digunakan untuk melakukan pengiriman data. Gambar berikut menunjukkan halaman ketika kita memulai menggunakan *Thingspeak*, dan akan diarahkan untuk membuat akun jika belum terdaftar sebagai pengguna *Thingspeak*^[16].

The image shows a screenshot of the Thingspeak website's account creation page. At the top, there is a blue navigation bar with the Thingspeak logo and links for 'Channels' and 'Apps'. Below this is a white box titled 'Create MathWorks Account'. The form contains several input fields: 'Email Address', 'User ID' (with a help icon), 'Password' (with an eye icon for visibility), a country dropdown menu currently set to 'Indonesia', 'First Name', and 'Last Name'. Below the fields is a line of text: 'By clicking continue, you agree to our privacy policy'. At the bottom of the form are two buttons: a grey 'Cancel' button and a blue 'Continue' button.

Gambar 2.7 Pembuatan Akun Pada *Thingspeak*^[16]

Pada Gambar 2.7 setelah akun selesai dibuat, pada tampilan awal *Thingspeak* akan diarahkan untuk membuat *channel* untuk perangkat yang akan kita hubungkan dengan server. Gambar 2.8 menunjukkan pembuatan *channel* yang menggunakan *field* sebagai parameter pengiriman data yang dikirimkan melalui Modul WiFi ESP8266 menggunakan HTTP POST. Kode pengiriman WiFi dapat dilihat pada penggalan kode dari Gambar 2.9.

Channel Settings

Percentage complete 50%

Channel ID 159192

Name

Description

Field 1

Field 2

Gambar 2.8 Pembuatan *Channel* Pada *Thingspeak*^[16]

```

void server1(){
    String apiKey =
    "PY8YG6RKBS85E7I9";
    kiriman = "&field1="; kiriman
    += suhu;
    kiriman += "&field2="; kiriman
    += "detak";

    const char* server =
    "thingspeak.com";
    if (client.connect(server,80))
    {
        String postStr =apiKey;
        postStr += kiriman;
        postStr +=
        "\r\n\r\n";
        client.print("POST /update
        HTTP/1.1\n");
        client.print("Host:
        thingspeak.com\n");
        client.print("Connection:
        close\n");
        client.print("X-
        THINGSPEAKAPIKEY: "+apiKey+"\n");
        client.print("Content-Type:
        application/x-www-form-
    
```

Gambar 2.9 Kode Program Pengiriman Data ke *Field Thingspeak*^[16]

Pada Gambar 2.10 pembuatan *channel* dilakukan dengan melakukan pengaturan untuk membuat apakah *channel* yang akan digunakan bersifat public atau private. Tujuan pembuatan ini ialah membuat hak pengaksesan terhadap data yang diolah. Pada tugas akhir ini *channel* dibuat publik supaya dapat diakses oleh server lain untuk melakukan embed terhadap grafik dari *Thingspeak*. Setelah *channel* berhasil dibuat maka *Application Programming Interface Key* akan otomatis tergenerate untuk dilakukan sebagai penghubung antara *Thingspeak* dan modul *WiFi*. Jika ada permasalahan mengenai koneksi antara perangkat dengan *API Key* kita bisa melakukan generate ulang *API Key*. Tampilan halaman *API Key* dapat dilihat pada Gambar berikut.

Write API Key

Key

PY8YG6RKBS85E7I9

Generate New Write API Key

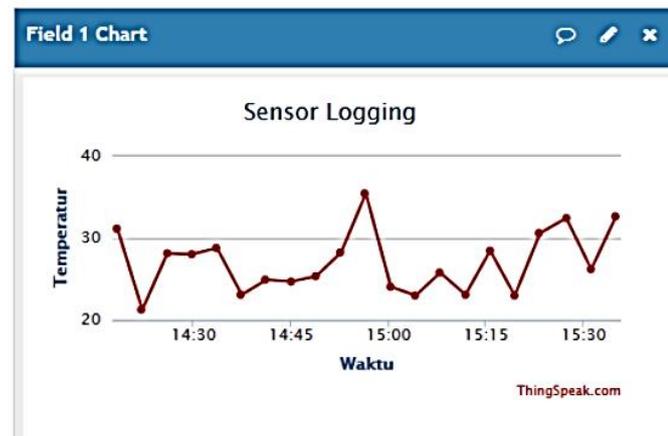
Read API Keys

Key

PHLIJEU03IKA6XZ0

Gambar 2.10 *API Key* Pada *Thingspeak*^[16]

Pada *API Key* terdapat *write API Key* dan *Read API Key*, yang digunakan dalam sistem ini ialah *write API Key* karena kita akan menuliskan data pada server. Setelah *API Key* sudah tergenerate maka selanjutnya kita dapat menempatkannya pada kode proram di atas. Tampilan awal grafik pada *My Channel* sebelum data dikirim akan terlihat kosong, setelah data dikirimkan dari perangkat, maka grafik akan terlihat seperti Gambar 2.11.



Gambar 2.11 Tampilan Grafik Pada *Thingspeak*^[16]

Grafik yang dapat ditampilkan dapat diatur sedemikian hingga terlihat seperti yang kita inginkan, tampilan grafik yang penulis gunakan adalah tampilan grafik dinamis dengan 10 data per tampilan dengan skala otomatis^[16].

