

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian ini merujuk pada beberapa penelitian dan beberapa jurnal lainnya. Adapun beberapa penelitian sebelumnya beserta keterangan penelitiannya secara singkat seperti pada penelitian Bukti Gultom dengan judul Deteksi Suhu Ruang *Server* dan Penggerak Kipas Berbasis Arduino Uno Dengan *Report SMS* Berisi tentang alat Deteksi suhu ruang *server* dengan menggunakan 4 sensor LM 35 sebagai sensor pendeteksi suhu. Jika terjadi kenaikan suhu melebihi batas yang tentukan maka modul GSM SIM900 akan mengirimkan notifikasi kepada administrator ruang *server* melalui sms. Penggunaan kipas pada penelitian ini sebagai penurun saat suhu naik melebihi batas yang ditentukan, kelemahan dari sistem yang dibuat adalah saat modul GSM SIM900 yang sudah diberi sim card tidak mendapatkan sinyal yang baik maka sistem yang dirancang tidak bisa mengirimkan notifikasi melalui sms, sehingga pada penelitian ini modul GSM SIM900 tidak lagi digunakan tetapi digantikan dengan menggunakan jaringan *internet* yang notabene disetiap ruang *server* selalu memiliki koneksi *internet* [1].

Perancangan Sistem Monitoring Suhu Gudang Berbasis *Internet Of Things* (IOT) oleh Irsandi Satria Wicaksana menjelaskan bahwa Suhu saat proses penyimpanan barang dapat mempengaruhi kualitas barang tersebut, pengaturan suhu pada ruangan penyimpanan dapat menjaga kualitas barang sampai waktu tertentu. Pengaturan tersebut perlu dilakukan pemantauan agar setiap saat kondisi suhu serta aktuatornya dapat diketahui, sehingga dibutuhkan *system* pemantau yang *realtime* dan dimanapun dapat diketahui. Salah satu alternatifnya melalui pendekatan *Internet of Things* (IoT). Tulisan ini bertujuan memberikan gambaran model pemantau pengendali suhu ruang berbasis IoT. Sensor LM35 diimplementasikan untuk pembacaan data suhu ruang dan actuator berupa kipas untuk pendinginan dan *heater* untuk pemanasan. Sistem yang dikembangkan menggunakan Node MCU sebagai pemroses data dan mengirimkan secara nirkabel menggunakan ESP 8266 menuju *Internet*. Hasil pengujian menunjukkan data suhu

yang terdeteksi mendekati suhu ruang yang terukur menggunakan thermometer standar yang dapat dipantau secara *online* menggunakan *smartphone*. Demikian halnya status kedua actuator dapat diketahui apakah sedang aktif (on) atau non aktif (off), penggunaan sensor LM35 hanya dapat membaca suhu, sedangkan pada penelitian tugas akhir ini parameter yang dimonitoring adalah suhu dan kelembaban oleh karena itu pada penelitian ini menggunakan sensor DHT 11, dikarenakan sensor DHT 11 dapat membaca 2 parameter sekaligus yaitu suhu dan kelembaban[3].

Pada penelitian Hendra Budianto, Slamet Winardi dengan judul Rancang Bangun dan Web Monitoring Pengukur Temperatur Suhu untuk Peringatan pada Ruang *Server* Menggunakan Sensor Dht 11 dengan Modul Arduino Uno, penelitian ini menyatakan bahwa suhu didalam ruang *server* harus dalam keadaan tetap dengan batasan yang telah ditentukan. Administrator ditugaskan untuk selalu memantau keadaan suhu didalam ruang *server*, agar tercipta kondisi ruangan yang kondusif dan ideal. Yang dikawatirkan saat suhu naik tetapi administrator tidak berada di ruang *server*. Permasalahan dapat diminimalisir dengan pembuatan alat yang dapat merekam kondisi suhu didalam ruang *server* serta memberikan *signal alarm* saat pembacaan suhu yang meningkat naik melebihi batas yang ditentukan. Penambahan kipas pada penelitian ini bertujuan untuk mengalirkan suhu panas keluar ruangan. Monitoring dapat dilakukan dengan mengakses web monitoring. Pada penelitian Hendra Budianto dan Slamet Winardi penggunaan modul Arduino uno dan harus menambahkan modul *wifi* sedangkan pada penelitian tugas akhir ini mikrokontroler menggunakan NodeMcu Esp 8266 yang sudah terintegrasi dengan modul *wifi* didalamnya sehingga NodeMcu 8266 dinilai lebih baik[4].

Pada penelitian Rachmad Andri Atmoko dengan judul Sistem Monitoring dan Pengendalian Suhu dan Kelembaban Ruang pada Rumah Walet Berbasis Android, Web, dan SMS membahas tentang Sarang walet yang menjadi komoditi yang mempunyai nilai ekonomis tinggi dan salah satu hal yang sangat diperhatikan adalah kondisi ideal rumah walet yang harus menyerupai habitat aslinya agar kualitas sarang yang dihasilkan baik. Penelitian ini membahas sistem monitoring dan pengendalian suhu serta kelembaban rumah walet yang dapat diakses dari perangkat berbasis android, web, dan sms. Pada penelitian ini dibuat sebuah model

menyerupai rumah walet dengan aktuator terdiri dari pompa air untuk sistem *cooler* dan *hair dryer* untuk *heater*. Aktuator bekerja otomatis ketika ada instruksi yang dikirimkan dari PC *server* bertujuan untuk menyesuaikan kondisi suhu dan kelembaban dan mode manual bekerja berdasarkan instruksi yang dikirim dari PC *Server*. Perangkat berbasis android, web, dan sms mengakses aplikasi monitoring dan pengendalian yang ditanamkan pada PC *server*. Monitoring data suhu dan kelembaban dapat dipantau melalui web secara *realtime*. Tujuannya yaitu saat pengecekan suhu dan kelembaban rumah wallet tidak dilakuka secara manual, dengan cara masuk kedalam rumah walet. Dengan kondisi rumah walet yang selalu dipantau suhu dan kelembabanya dan diatur agar selalu ideal maka sarang walet yang dihasilkan akan meningkat nilai jualnya dan memiliki kualitas yang lebih baik. Kelemahan dari sistem yang dibuat terkendala pada kualitas sinyal dikarenakan pemberitahuan dilakukan melalui sms, saat kualitas sinyal buruk maka pemberitahuan tidak dapat dikirimkan[5].

Pada penelitian Muhammad Fahmi Awaj, Adian Fatchur Rochim dan Eko Didik Widiyanto dengan judul Sistem Pengukur Suhu Dan Kelembaban Ruang Server membahas tentang perancangan sistem pengukur suhu dan kelembaban ruang *server* menggunakan Arduino sebagai mikrokontrolernya. Sistem terbentuk terdiri dari sensor DHT 11 yang berfungsi sebagai pendeteksi suhu dan kelembaban pada ruang *server*, kemudian akan ditampilkan pada LCD serta Data suhu dan kelembaban ditampilkan juga ke dalam *website* dengan menggunakan *IP address* tertentu. Saat sensor DHT 11 membaca Suhu yang melebihi 27⁰C maka menyebabkan relay dan kipas aktif, serta sistem akan mengirim peringatan sms secara otomatis kepada administrator ruang *server* melalui sistem sms secara periodik dan akan berhenti saat suhu yang dibaca kurang dari 27⁰C[6]. Kelemahan pada penelitian penelitian Muhammad Fahmi Awaj, Adian Fatchur Rochim dan Eko Didik Widiyanto adalah penggunaan *website* untuk mengakses *website* yang dituju memerlukan waktu dan pembuatan *website* memakan biaya sedangkan pada penelitian tugas akhir ini semua data suhu dan kelembaban ditampilkan pada aplikasi android dan pembuatan aplikasi tidak memakan biaya, serta lebih efisien menggunakan aplikasi android.

2.2 RUANG SERVER

Ruang *Server* adalah sebuah ruangan yang digunakan untuk menyimpan data, alat-alat seperti halnya *router*, *mikrotik*, *hub*, *server* dan perangkat lainnya yang terkait dengan operasional sistem sehari-hari seperti *Uninterruptible Power Supply* (UPS). Pada umumnya ruang *server* harus memiliki standarisasi keamanan yang mumpuni agar dapat melindungi perangkat-perangkat yang tersimpan di dalamnya dari mulai kondisi suhu udara, dan kondisi kelembaban. Ruang *server* adalah aset bagi sebuah perusahaan atau instansi dikarenakan pada ruang server terdapat data-data perusahaan atau instansi yang sangat berharga, oleh karena itu ruangan ini harus selalu terjaga dengan baik, baik dalam keamanan dari pihak yang tidak bertanggung jawab dan kelayakan pada standar perawatan ruang *server* itu sendiri[7].

Ruang *server* sangat rentan terhadap temperatur yang tinggi. Oleh sebab itu penggunaan sensor suhu yang diletakkan di rak *server* menjadi sebuah solusi untuk mengendalikan suhu. Selain memperhatikan panas pada *server*, yang perlu diperhatikan adalah suhu ruangan. Untuk itu diperlukan sistem pendingin yang baik. Sejak mulai awal pembangunan ruang *server* hendaknya sudah diperhitungkan berapa kapasitas yang diperlukan untuk membuat ruangan tetap dingin, sehingga tidak kesulitan dalam menghitung listrik yang dibutuhkan. Meningkatnya suhu dapat diatasi dengan penambahan AC, namun akan dapat menimbulkan masalah karena membutuhkan listrik yang cukup besar[8]. Ruangan yang terlalu lembab juga dapat mengakibatkan rusaknya komponen. Pengaturan AC untuk *ruang server* khusus untuk kelembaban sebaiknya 40% - 55% dan suhu yang baik berkisar antara 20 – 25 °C sesuai pada *Telecommunications Infrastructure Standard for Data Centers*[2].

2.3 INTERNET OF THINGS.

Menurut (Burange&Misalkar,2015) *Internet of Things* adalah struktur dimana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. *Internet of*

Things merupakan perkembangan keilmuan yang sangat menjanjikan untuk mengoptimalkan kehidupan berdasarkan sensor cerdas dan peralatan pintar yang bekerjasama melalui jaringan internet (Keoh, Kumar &Tschofenig,2014). Sejak mulai dikenalnya *internet* pada tahun 1989, mulai banyak hal kegiatan melalui internet, Pada tahun 1990 John Romkey menciptakan 'perangkat', pemanggang roti yang bisa dinyalakan dan dimatikan melalui *Internet. Wear Cam* diciptakan pada tahun 1994 oleh Steve Mann. Pada tahun 1997 Paul Saffo memberikan penjelasan singkat pertama tentang sensor dan masa depan. Tahun 1999 Kevin Ashton menciptakan *The Internet of Things*, direktur *eksekutif Auto ID Centre*, MIT. Mereka juga menemukan peralatanberbasis RFID (*Radio Frequency Identification*) global yang sistem identifikasi pada tahun yang sama. Penemuan ini disebut sebagai sebuah lompatan besar dalam *commercializing IoT*[9].

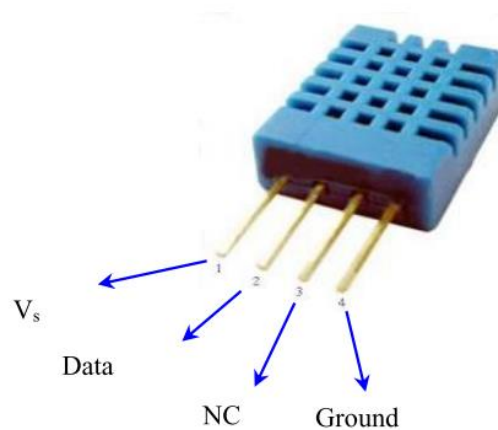
Tahun 2000 LG mengumumkan rencananya menciptakan kulkas pintar yang akan menentukan sendiri apakah bisa atau tidak makanan yang tersimpan di dalamnya diisi ulang. Pada ahun 2003 RFID mulai ditempatkan pada tingkat besar-besaran di militer AS di Program Savi mereka. Pada tahun yang sama melihat raksasa ritel Walmart untuk menyebarkan RFID di semua toko - toko di seluruh dunia untuk lebih besar batas. Pada tahun 2005 arus publikasi utama seperti *The Guardian*, Amerika ilmiah dan BostonGlobe mengutip banyak artikel tentang IOT. Pada tahun 2008 kelompok perusahaan meluncurkan *IPSO Alliance* untuk mempromosikan penggunaan *Internet Protocol* (IP) dalam jaringan dari "*Smart object*" dan untuk mengaktifkan *Internet of Things*. Pada tahun 2008 FCC menyetujui Penggunaan "*whitespacespectrum*". Akhirnya peluncuran IPv6 di tahun 2011 memicu pertumbuhan besar dibidang *Internet of Things*, perkembangan ini didukung oleh perusahaan raksasa seperti *Cisco, IBM, Ericson* mengambil inisiatif banyak dari pendidikan dan komersial dengan IOT teknologi dapat hanya dijelaskan sebagai hubungan antara manusia dan komputer.

Perkembangan *Internet of Things*, semua peralatan yang kita gunakan dalam kehidupan kita sehari-hari dapat dikendalikan dipantau menggunakan IOT. Mayoritas proses dilakukan dengan bantuan sensor di IOT. Sensor dikerahkan dimana mana dan sensor ini mengkonversi data fisik mentah menjadi sinyal digital dan mengirimkan mereka ke pusat kontrol. Dengan cara ini kita bisa memonitor

perubahan lingkungan jarak jauh dari setiap bagian dari dunia melalui *internet*. Arsitektur sistem ini akan didasarkan pada konteks operasi dan proses dalam skenario *real-time*[9].

2.4 Sensor Suhu DHT 11

DHT11 adalah salah satu sensor dapat mengukur dua parameter sekaligus yaitu suhu dan kelembaban udara. Keluaran sinyal pada DHT 11 adalah sinyal digital yang dikalibrasi dengan sensor suhu dan kelembaban, hal ini yang menyebabkan stabilitas kinerja sensor menjadi sangat baik dan responsif. Selain memiliki kualitas yang sangat baik, sensor ini memiliki respon cepat, kemampuan anti-gangguan dan keuntungan biaya dikarenakan satu sensor tetapi dapat mengukur dua parameter sekaligus tidak seperti beberapa sensor yang hanya dapat membaca suhu saja.



Gambar 2.1 Sensor DHT 11[10]

Sensor ini mempunyai dua sensor didalamnya yaitu sensor thermistor tipe NTC (*Negative Temperature Coefficient*) untuk mengukur suhu udara, dan sensor kelembaban tipe resistif untuk mengukur kelembaban udara. Selain terdapat dua sensor di dalamnya, terdapat pula sebuah mikrokontroler kecil 8 bit di dalamnya, yang mengolah data kedua sensornya, dan mengirim hasilnya ke pin output dengan tipe *single wire bidirectional*. Sistem *single wire bidirectional* ini membuat penggunaan menjadi cepat dan mudah. Sensor ini merupakan sensor yang cukup kompleks karena mempunyai tiga sistem di dalamnya. Cara pengambilan data sensor DHT11, yaitu dengan cara menyambungkan pin *output* dari sensor tersebut.

Ukuran yang kecil, daya rendah, sinyal transmisi jarak hingga dua puluh meter merupakan beberapa kelebihan dari sensor ini. Kelebihan - kelebihan ini membuat sensor ini sering dipakai pada berbagai aplikasi. Empat kaki sensor DHT11 yang diperlihatkan pada gambar diatas adalah kaki Vs, Data, NC dan *Ground*. Kaki Vs digunakan sebagai tegangan sumber sensor ini. Tegangan sumber yang diperkenankan adalah diantara rentang 3V sampai 5.5V. Pada kaki data digunakan untuk mengambil data suhu dan kelembaban udara yang telah diukur oleh sensor DHT11[10]. Kaki NC yang merupakan singkatan dari *Not Connected*, adalah kaki yang tidak dihubungkan dengan apa-apa. Jadi dalam prakteknya, kaki ini tidak boleh dihubungkan dengan rangkaian apapun. Kemudian kaki Ground disambung dengan Ground tegangan sumber Beberapa spesifikasi dari DHT11 yaitu, pasokan *voltage* sebesar 5 V dengan rentang temperature 0 – 50 °C dengan kesalahan ± 2 celsius, pembacaan kelembaban dengan 20-90% RH ± 5% RH error.

Pengukuran akurasi sensor DHT 11 menggunakan perhitungan manual yaitu dengan mencari galat absolut dan galat relatif. Galat absolut dari pengukuran didefinisikan sebagai selisih antara nilai sebenarnya (*true value*) dengan nilai hasil pengukuran (*measured value*). Nilai absolut dari galat ini ditentukan melalui rumusan berikut[11]:

$$E_a = x_i - x_p \dots \dots \dots [1]$$

Sedangkan galat relatif ditentukan dari perbandingan antara galat absolut tersebut terhadap nilai sebenarnya. Dalam persentase, ini dirumuskan:

$$E_r = \frac{E_a}{x_p} \times 100 \dots \dots \dots [2]$$

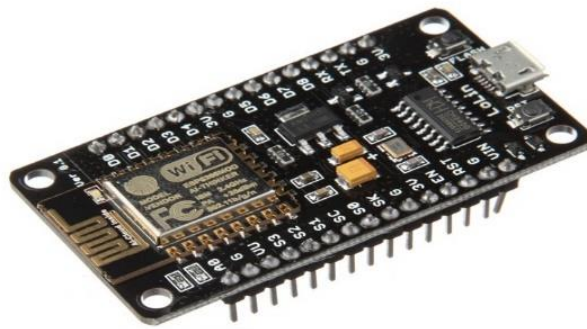
dengan:

- a. Er = galat relatif (%)
- b. Ea = galat absolut
- c. xi = nilai pengukuran
- d. xp = nilai sejati

2.5 NodeMcu Esp 8266.

NodeMcu pada dasarnya adalah pengembangan dari ESP 8266 dengan *firmware* berbasis e-Lua. NodeMcu dilengkapi dengan micro usb port yang berfungsi untuk pemrograman maupun *power supply*. Selain itu juga pada NodeMcu di lengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash*. NodeMcu

menggunakan bahasa pemrograman Lua yang merupakan *package* dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan bahasa C hanya berbeda *syntax*. Jika menggunakan bahasa Lua maka dapat menggunakan *tool Lua loader* maupun *Lua uploader*. NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan *firmware* berbasis e-Lua. Pada NodeMcu dilengkapi dengan micro USB *port* yang berfungsi untuk pemrograman maupun *power supply*[12]. Lua mengandalkan Bahasa C untuk melakukan tugas tersebut. Oleh karena itu, Lua mempunyai lingkungan yang aman dan manajemen memori otomatis[13].



Gambar 2.2 NodeMCU esp 8266.[12]

Tabel 2.1 Spesifikasi NodeMcu V3[12]

SPESIFIKASI	NodeMcu V3
Mikrokontroler	ESP 8266
Ukuran Board	57mm x 30 mm
Tegangan Input	3.3 – 5 V
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 PIN
Flash Memory	4MB
Clock Speed	40/26/24 MHZ
Wifi	IEEE 802.11 b/g/n
Frekuensi	2.4 Ghz – 22.5 Ghz
USB Port	Micro USB
Card Reader	Tdak ada
USB Serial Converter	CH3040G

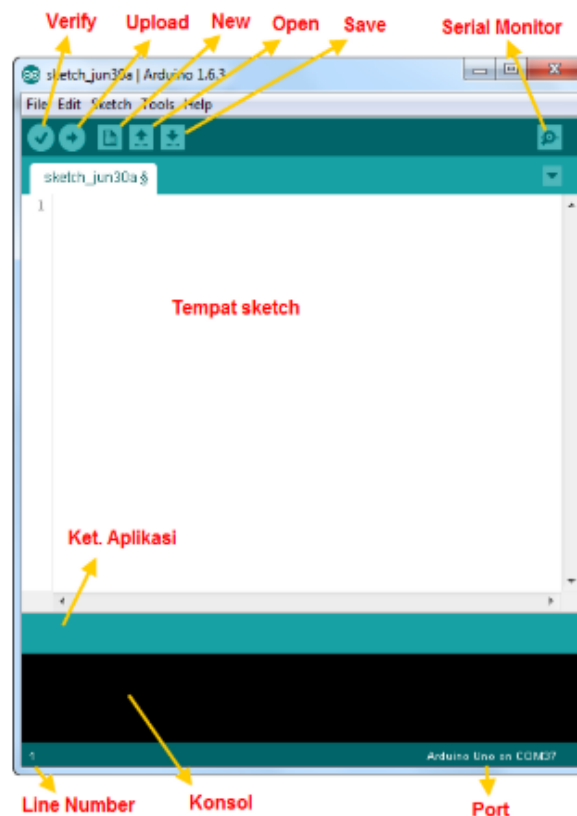
Pada tabel 2.1 menjelaskan spesifikasi pada NodeMcu V3 dengan ukuran *board* 57mm x 30 mm, tegangan *input* sebesar 3.3 – 5 volt, dengan jumlah pin GPIO sebanyak 13 pin, kanal PWM sebanyak 10 kanal, 10 bit *analog digital converter* sebanyak 1 pin, *flash memory* sebesar 4 MB dengan *clock speed* 40/26/24 MHZ, *wifi* menggunakan standarisasi IEEE 802.11 b/g/n dengan frekuensi sebesar 2.4 Ghz – 22.5 Ghz, USB *port* menggunakan jenis micro USB dan USB serial converter tipe CH3040G.



Gambar 2.3 Skematik posisi Pin NodeMcu Dev Kit v3[12]

2.6 SOFTWARE ARDUINO IDE

Untuk memprogram board nodemcu esp 8266, menggunakan aplikasi IDE (*Integrated Development Environment*) bawaan dari Arduino. Aplikasi ini berguna untuk membuat, membuka, dan mengedit *source code* arduino (*Sketches*), *source code arduino* bisasa disebut ("*sketches*"). Selanjutnya, *source code* yang ditulis untuk Arduino, bisa disebut "*sketch*". *Sketch* merupakan *source code* yang berisi logika dan algoritma yang akan diupload ke dalam IC mikrokontroller (Arduino)[14].



Gambar 2.4 Interface Arduino IDE[14]

Interface Arduino IDE tampak seperti gambar 1.6.1 Dari kiri ke kanan dan atas ke bawah, bagian-bagian IDE Arduino terdiri dari *Verify*, pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi diupload ke *board* Arduino, biasanya untuk memverifikasi terlebih dahulu *sketch* yang dibuat, jika ada kesalahan pada *sketch*, maka akan muncul *error*. Proses *Verify / Compile* mengubah *sketch* ke *binary code* untuk diupload ke mikrokontroller, selanjutnya *upload*. *Upload* berfungsi untuk mengupload *sketch* ke *board* NodeMcu Esp8266. Jika tidak memilih menu *verify*, maka *sketch* akan di-*compile*, kemudian akan diupload ke

board. Berbeda dengan tombol *verify* yang hanya berfungsi untuk memverifikasi *source code* saja. Pada menu *New Sketch*, digunakan untuk membuka *window* dan membuat *sketch* baru. *Open Sketch* digunakan untuk membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file *.ino* dan untuk menyimpan digunakan menu *Save Sketch*. Serial monitor berfungsi untuk membuka *interface* untuk komunikasi serial

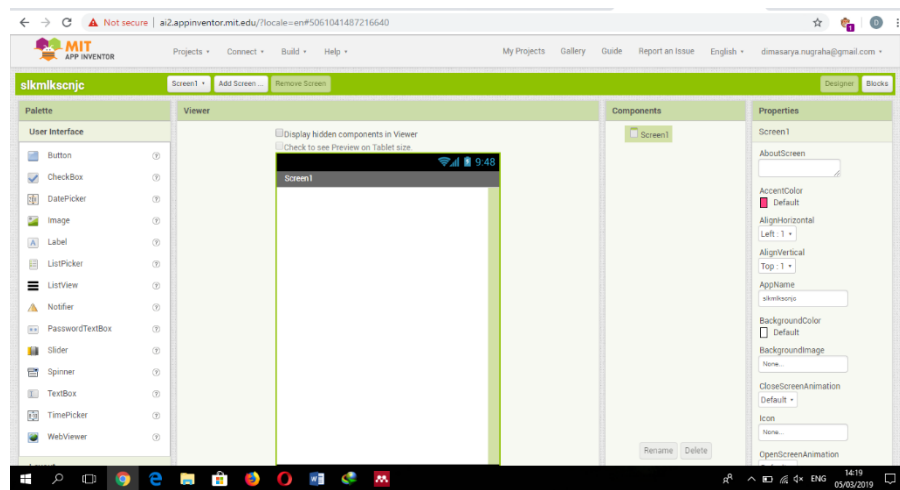
2.7 MIT APP INVENTOR.

MIT App Inventor adalah tool pemrograman berbasis *block* yang memungkinkan semua orang, bahkan pemula, untuk memulai pemrograman dan membangun aplikasi yang berfungsi penuh untuk perangkat Android. Mit App inventor adalah sebuah pemrograman visual yang digunakan untuk mengembangkan aplikasi berbasis android dengan dukungan fitur *drag-drop tool*.

MIT App Inventor dapat mendesain *user interface* dari sebuah aplikasi dengan menggunakan web GUI(*Graphical User Interface*) *builder*, kemudian pengguna dapat menspesifikan *behavior* aplikasi dengan memasang *block* yang sesuai seperti bermain *puzzle*[15]. Bahasa yang digunakan berbasis teks yang lebih tradisional (mudah). Awalnya dikembangkan oleh Profesor Hal Abelson dan timnya dari *Google Education* pada saat sedang cuti di Google.

App Inventor berjalan sebagai layanan Web yang dikelola oleh staf di *MIT Center for Mobile Learning* - sebuah kolaborasi dari *MIT Computer Science and Artificial Intelligence Laboratory* (CSAIL) dan MIT Media Lab. MIT App Inventor mendukung komunitas di seluruh dunia hampir dua juta pengguna yang mewakili 195 negara di seluruh dunia. Lebih dari 85 ribu pengguna aktif setiap minggunya.

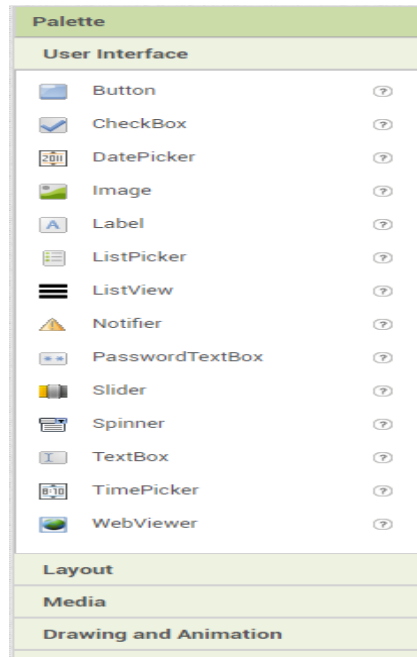
Sebuah *tool open source* untuk pemrograman sekaligus penciptaan aplikasi yang dapat diakses berbagai orang dari seluruh dunia.[10]



Gambar 2.5 Tampilan Project di Mit App Inventor Online [10]

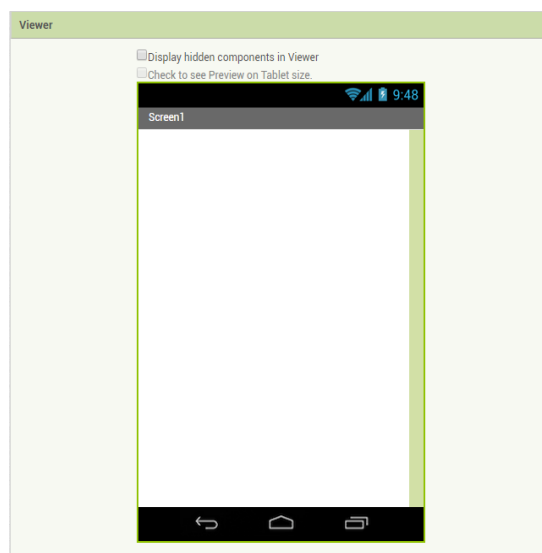
2 langkah utama dalam pembuatan aplikasi android pada Mit App Inventor, langkah pertama yaitu menggunakan *Design View* untuk menambahkan komponen-komponen kedalam proyek. Beberapa komponen seperti tampilan, tombol-tombol, dan sebuah tempat untuk memasukkan *text (text field)*. Dari tampilan ini dapat mendesain *User Interface* atau bagaimana cara aplikasi ini dapat berinteraksi dengan pengguna. Beberapa macam komponen yang tidak terlihat tapi sangat berguna, seperti database dan pengaturan layar. Dalam proses selanjutnya akan diarahkan pada bagaimana cara mendesain sebuah tampilan. *Design View* terdiri dari lima komponen dasar yaitu yang pertama adalah *Palette*, *Viewer*, *Component*, *Media Kolom* dan *Properties*.

Palette terdiri dari berbagai objek (komponen) yang dapat digunakan pada aplikasi android yang dibuat. *Palette* terdiri dari beberapa grup, semuanya dikelompokkan kedalam satu grup yang memiliki tema/fungsi yang sama. Contohnya *User Interface* yang memiliki fungsi digunakan untuk mengatur interaksi aplikasi dengan pengguna yang terdiri dari *button*, *check box*, *clock*, *image*, *label*, dan lain-lain. Cara untuk menampilkan atau menyembunyikan anggota dari suatu kelompok perlu mengeklik pada nama kelompok itu, seperti yang tertera pada gambar 2.6.



Gambar 2.6 Tampilan palette pada *Design View* [10]

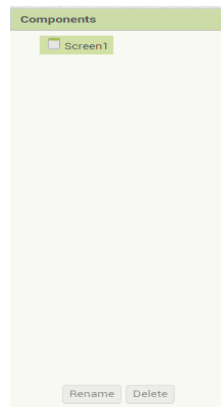
Viewer Merupakan tempat menempatkan komponen-komponen yang akan ditampilkan pada layer perangkat android dan komponen-komponen yang bisa di atur posisi layoutnya. Pada *viewer*, juga bisa melihat komponen yang tidak bisa dilihat dengan handphone, seperti yang tertera pada gambar 2.7.



Gambar 2.7 Tampilan viewer pada *Design View* [10]

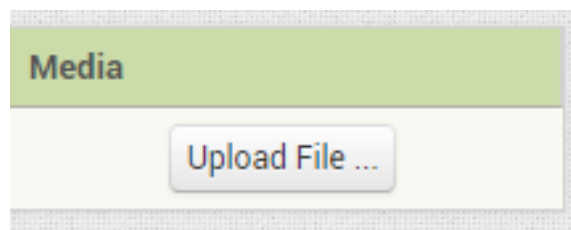
Component Terdiri dari daftar komponen apa saja yang telah ditambahkan ke dalam proyek baik yang bersifat terlihat maupun tidak terlihat dalam perangkat android. Tampilannya berupa susunan atau daftar (*seperti root explorer*) yang

memudahkan untuk mengatur komponen atau melihat apa saja yang berbentuk seperti direktori seperti yang tertera pada gambar 2.8.



Gambar 2.8 Tampilan *Component* pada *Design View* [10]

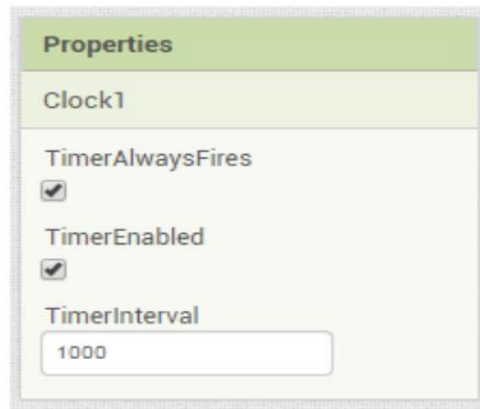
Pada media, kolom media terletak di bawah dari kolom *Component*. Kolom ini digunakan untuk mengatur (menambah dan menghapus) semua media komponen untuk mendukung aplikasi yang telah dibuat. Tipe media yang dapat ditambahkan ke dalam kolom media adalah gambar, *clip art*, musik, dan film. Disini juga dapat menambahkan media secara langsung kedalam kolom properti. Media yang anda tambahkan ke dalam App Inventor diambil dari komputer dan diupload ke dalam Mit App Inventor. Semua media yang anda tambahkan ke dalam sebuah aplikasi Android tidak boleh melebihi 5 MB. Di dalam kolom media anda juga bisa menghapus atau mendownload media yang telah ditambahkan dengan meng-klik nama mediana sehingga nanti akan muncul pilihan *delete* untuk perintah menghapus atau klik *download to my computer* untuk perintah mengunduh ke komputer seperti yang tertera pada gambar 2.9[16].



Gambar 2.9 Tampilan Media pada *Design View* [10]

Properties Setiap komponen yang ditambahkan ke dalam proyek, dapat diatur agar dapat berinteraksi dengan pengguna maupun dengan komponen lain, atau

bagaimana tampilannya. Setiap komponen pada App Inventor memiliki kolom *properties* yang berbeda-beda[16].



Gambar 2.10 Tampilan *Properties Clock1* pada *Design View*[16]

2.8 RELAY

Dalam dunia elektronika, *relay* dikenal sebagai komponen yang dapat mengimplementasikan logika *switching*. Sebelum tahun 1970an, *relay* merupakan “otak” dari rangkaian pengendali. Baru setelah itu muncul PLC yang mulai menggantikan posisi *relay*. *Relay* yang paling sederhana ialah *relay* elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik[17]. *Relay* yaitu alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar, kemudian saklar yang digerakkan (secara mekanis) oleh daya/energi listrik [4].



Gambar 2.11 Relay [4]

2.9 KIPAS

Kipas digunakan untuk menurunkan suhu ruang server apabila suhu sudah *overheat*[1], yang pada kali ini maksimal suhu sebesar 25 °C sehingga saat suhu

melebihi batas yang sudah ditentukan maka kipas berada pada kondisi *on* atau menyala untuk menurunkan suhu ruangan.

2.10 FIREBASE

Firebase adalah salah satu layanan yang menyediakan *realtime* database yang mudah diakses dan digunakan oleh web developer, database sendiri yaitu sekumpulan informasi dan data yang tersimpan dan tersusun dalam computer secara sistematis yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program computer guna mendapatkan data atau informasi dari basis data tersebut[18]. *Firebase* bisa dipasangkan kedalam aplikasi melalui kode *java script client* yang dapat dengan mudah dikonfigurasi dan dimodifikasi sendiri[19]. *Realtime* database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau bisa disebut aturan keamanan *Firebase*. *Realtime* database adalah database NoSQL, yang memiliki perbedaan fungsional dan pengoptimalan dari database terkait. *Firebase* menggunakan sinkronisasi data setiap data berubah, semua perangkat yang terhubung akan menerima *update* dalam waktu milidetik, dan saat *offline* aplikasi *Firebase* tetap responsive karena SDK *Firebase Realtime Database* menyimpan data ke disk. Setelah konektivitas pulih, perangkat klien akan menerima update perubahan data terbaru dan akan melakukan sinkronisasi pada status server kali ini[19].

2.11 QUALITY OF SERVICE.

Quality of Service merupakan kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *jitter* dan *delay*. Tujuan *Quality of Service* adalah untuk menyediakan layanan yang berbeda-beda untuk beragam kebutuhan akan layanan di dalam suatu jaringan. Pada penelitian tugas akhir ini parameter *Quality Of Service* yaitu *delay*, *throughput*, dan *packet loss*[20].

2.12 Software Wireshark.

Aplikasi *Wireshark* Adalah Sebuah *Network Protocol Analyzer* untuk berbagai Kebutuhan analisis unjuk kerja jaringan. *Wireshark* difungsikan untuk menangkap (*capturing*) setiap data yang lewat melalui salah *satu network interface* dari komputer di mana aplikasi tersebut terpasang. Aplikasi ini memiliki fungsi yang sama dengan aplikasi *tcpdump* pada sistem operasi linux[21].

