

BAB II

TEORI PENUNJANG

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. Adapun kemampuan seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT [7].

2.2 Smart Home

Smart Home atau Rumah Pintar merupakan bagian dari inovasi *Internet of Things*, dimana semua perangkat atau peralatan rumah sehari-hari yang akrab dengan kehidupan masyarakat di pintarkan karena integrasi teknologi dalam bentuk chip yang serba bisa. Dalam hal *smart home*, *Internet of Things* telah hadir dalam rupa barang-barang yang biasa ditemui di rumah orang kebanyakan. *Smart home* hadir untuk memudahkan para penghuni rumah dalam mengatur segala hal yang berhubungan dengan kenyamanan diri sebagai penghuni rumah, mulai dari soal keamanan hingga soal akses peralatan yang dibuat lebih interaktif dan bisa dikontrol melalui satu alat saja, yakni aplikasi pada *smartphone* atau perangkat lainnya [8].

2.3 Chatbot

Chatbot merupakan program komputer yang dirancang untuk dapat melakukan interaksi dengan manusia melalui pesan teks, maupun suara. *Chatbot* biasanya juga dibekali dengan kecerdasan buatan dan pemrosesan bahasa alami yang membuatnya menjadi program komputer yang cerdas dan dapat menjawab pertanyaan yang diberikan oleh manusia. *Chatbot* dibangun sesuai dengan topik yang sudah dimodelkan dalam basis pengetahuan. Banyak *Chatbot* yang sudah ada dibangun sesuai dengan topik dan permasalahan yang ingin dipecahkan oleh seseorang untuk keperluan pribadi ataupun keperluan bisnis, sebagai contoh adalah *chatbot* Veronika yang dibuat oleh Telkomsel untuk pelayanan pelanggan telkomsel. Di dalam *chatbot* tersebut telah ditanamkan model pengetahuan untuk menjawab pertanyaan-pertanyaan yang sesuai dengan konteks yang telah disusun [9].

Chatbot terdiri dari tiga kombinasi, di mana ketiga kombinasi inilah yang membentuk sebuah *Chatbot* (Guzman & Ines, 2016), di antaranya adalah:

a. *User Interface*

User interface dalam *Chatbot* ini sendiri adalah jembatan antara *Chatbot* dan *user* saling berinteraksi. Melalui aplikasi pesan berbasis text. *User interface* haruslah dapat memberikan pengalaman yang lebih baik kepada *user* ketika berinteraksi dengan *Chatbot*.

b. *Artificial Intelligence* (Kecerdasan Buatan)

AI atau *Artificial Intelligence* akan membuat *Chatbot* mengerti dan memahami setiap interaksi yang terjadi dengan *user*. *Chatbot* menangani pemecahan masalah melalui aturan yang telah ditentukan sebelumnya di pemodelan keputusan.

c. Integrasi

Integrasi dengan sistem lainnya akan menambah kekayaan fitur yang terdapat di dalam suatu *Chatbot*. Dengan mengintegrasikan *Chatbot* ke

sistem yang lain dapat menyediakan informasi tambahan. Dengan cara ini *Chatbot* mampu memberikan informasi yang lebih kaya kepada *user*, seperti

pada penelitian kali ini yang akan mengimplementasikan *Chatbot* pada *Smart Home*.

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari *user* bisa dipahami dengan baik oleh komputer [10].

Ada berbagai terapan aplikasi dari NLP. Diantaranya adalah *Chatbot* (aplikasi yang membuat *user* bisa seolah-olah melakukan komunikasi dengan manusia), *Stemming* atau *Lemmatization* (pemotongan kata dalam bahasa tertentu menjadi bentuk dasar pengenalan fungsi setiap kata dalam kalimat), *Summarization* (ringkasan dari bacaan), *Translation Tools* (menterjemahkan bahasa) dan aplikasi-aplikasi lain yang memungkinkan komputer mampu memahami instruksi bahasa yang diinputkan oleh *user*.

Pustejovsky dan Stubbs (2012) menjelaskan bahwa ada beberapa area utama penelitian pada field NLP, diantaranya:

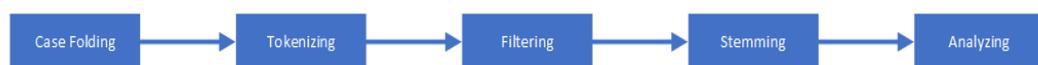
1. *Question Answering Systems (QAS)*. Kemampuan komputer untuk menjawab pertanyaan yang diberikan oleh *user*. Daripada memasukkan *keyword* ke dalam browser pencarian, dengan QAS, *user* bisa langsung bertanya dalam bahasa natural yang digunakannya, baik itu Inggris, Mandarin, ataupun Indonesia.
2. *Summarization*. Pembuatan ringkasan dari sekumpulan konten dokumen atau email. Dengan menggunakan aplikasi ini, *user* bisa dibantu untuk mengkonversikan dokumen teks yang besar ke dalam bentuk slide presentasi.
3. *Machine Translation*. Produk yang dihasilkan adalah aplikasi yang dapat memahami bahasa manusia dan menterjemahkannya ke dalam bahasa lain. Termasuk di dalamnya adalah *Google Translate* yang

apabila dicermati semakin membaik dalam penterjemahan bahasa. Contoh lain lagi adalah BabelFish yang menterjemahkan bahasa secara real time.

1. *Speech Recognition*. Field ini merupakan cabang ilmu NLP yang cukup sulit. Proses pembangunan model untuk digunakan telpon/komputer dalam mengenali bahasa yang diucapkan sudah banyak dikerjakan. Bahasa yang sering digunakan adalah berupa pertanyaan dan perintah.
2. *Document classification*. Sedangkan aplikasi ini adalah merupakan area penelitian NLP Yang paling sukses. Pekerjaan yang dilakukan aplikasi ini adalah menentukan dimana tempat terbaik dokumen yang baru diinputkan ke dalam sistem. Hal ini sangat berguna pada aplikasi *spam filtering*, *news article classification*, dan *movie review*.

2.5 Text Mining

Text Mining merupakan proses pencarian pola atau penggalian informasi baru. Tujuan dari *text mining* adalah menemukan informasi yang penting dari teks dengan mengubah teks menjadi data yang dapat digunakan untuk analisis yang lebih lanjut [11]. Secara umum tahapan yang dilakukan pada *text mining* dapat dilihat pada Gambar II-1.



Gambar II-1 Alur Text Mining

2.5.1 Case Folding

Case Folding merupakan proses perubahan kalimat menjadi huruf kecil (*lower case*). karakter yang dirubah dari huruf 'a' sampai 'z'.

2.5.2 Tokenizing

Tahap *Tokenizing* adalah tahap pemotongan tiap kata dalam kalimat atau parsing dengan menggunakan spasi sebagai delimiter yang akan menghasilkan token

berupa kata. Proses selanjutnya adalah penguraian, proses penguraian yang dimaksud adalah membagi text menjadi kumpulan kata tanpa memperhatikan keterhubungan antara kata satu dengan kata lain serta peran dan posisinya pada kalimat.

2.5.3 Filtering

Tahap *Filtering* merupakan tahap pengambilan kata-kata penting dari hasil proses *tokenizing*. Pada tahap *filtering* terdapat dua algoritma yang dapat digunakan yaitu algoritma *stoplist/stopword* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). Contoh *stopwords* adalah “apa”, “di”, pada” dan seterusnya. Jika terdapat kata stoplist dalam kalimat maka kata-kata tersebut akan di-*remove* dari kalimat sehingga kata-kata yang tersisa di dalam deskripsi di anggap sebagai kata-kata penting atau *keywords*.

2.5.4 Stemming

Tahap *stemming* adalah tahap mengembalikan kata-kata yang diperoleh dari hasil *Filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufix*) sehingga didapat kata dasar.

2.5.5 Analyzing

Proses *analyzing* adalah proses analisa dari hasil proses *stemming* sehingga diketahui seberapa jauh tingkat keterhubungan antar kata-kata dan antar dokumen yang ada. Untuk melakukan analisa pada tahap *analyzing* dapat digunakan Algoritma *TF/IDF* (*Term Frequency-Inverse Documents Frequency*) dan Algoritma *Vector Space Model*.

2.6 Android

Android adalah sistem operasi pada smartphone yang berbasis Linux dan bersifat *open source*, sehingga android dapat dikembangkan secara bebas oleh para pengembangnya. Android merupakan sistem operasi pada *smartphone* yang

memiliki kekuatan besar di lebih dari 190 negara di seluruh dunia yang terus tumbuh dengan pesat setiap harinya dengan banyaknya inovasi dari para pengembang berupa aplikasi, game dan konten digital lainnya yang didukung oleh platform Android [12].

Google menggambarkan android sebagai sistem operasi terbuka berbasis perangkat lunak Linux yang terdiri dari komponen-komponen sistem yang terintegrasi membentuk sebuah arsitektur seperti sebuah tumpukan [13]. Setiap lapisan dari tumpukan ini terdiri dari beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Gambar dari arsitektur android dapat dilihat pada Gambar II-2.



Gambar II-2 Arsitektur Android

Berikut merupakan penjelasan mengenai setiap lapisan pada arsitektur Android:

1. Linux Kernel

Bagian dasar atau fondasi dari arsitektur Android adalah Linux Kernel. Pada lapisan ini menyediakan *driver* layar, kamera, audio, *bluetooth*, *shared memory*, *keypad*, USB, WiFi, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2. *Hardware Abstraction Layer* (HAL)

Hardware abstraction layer menyediakan *interface* standar yang mengekspos kemampuan perangkat keras ke tingkat lebih tinggi, Java API framework. HAL terdiri dari beberapa modul *library*, yang masing-masing mengimplementasikan antarmuka untuk jenis tertentu dari komponen *hardware*, seperti kamera, atau modul Bluetooth. Ketika kerangka API melakukan panggilan untuk mengakses perangkat keras, sistem Android akan memuat modul *library* untuk komponen *hardware*.

3. *Android Runtime* (ART)

Android Runtime ditulis untuk menjalankan beberapa mesin virtual pada perangkat *low-memory* dengan mengeksekusi *file* DEX, format bytecode yang dirancang khusus untuk Android yang dioptimalkan untuk jejak memori yang optimal.

4. *Native C/C++ Libraries*

Banyak komponen sistem dan *service* dasar Android seperti *Android Runtime* (ART) dan *Hardware Abstraction Layer* (HAL) dibangun dari *native code* yang membutuhkan *native libraries* yang ditulis dalam C dan C++. Platform Android menyediakan Java Framework API yang dapat digunakan untuk mengakses beberapa fungsi dari *library* ini ke aplikasi. Misalnya kita dapat mengakses OpenGL ES melalui kerangka Android Java OpenGL API untuk menambahkan dukungan dalam menggambar dan memanipulasi 2D, 3D grafis pada aplikasi.

1. Java API Framework

Seluruh fitur-set dari sistem operasi Android tersedia untuk kita melalui API yang ditulis dalam bahasa Java. API ini membentuk blok bangunan yang dibutuhkan untuk membangun aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan jasa sistem modular, yang meliputi:

- a. *A rich and extensible View System*, dapat kita gunakan untuk membangun *User Interface*, termasuk seperti *lists view*, *grids view*, *button*, *text box* dan *embeddable web browser*.
- b. *A Resource Manager*, menyediakan akses ke sumber daya *non-code* seperti *localized string*, grafis dan *file layout*
- c. *A Notification Manager*, yang membuat semua aplikasi dapat menampilkan pesan pada status bar.
- d. *An Activity Manager*, yang mengelola siklus hidup dari aplikasi dan menyediakan navigasi umum pada tumpukan *activity*.
- e. *Content Providers*, Jawa API yang membuat aplikasi dapat mengakses data dari aplikasi lain, seperti aplikasi *contact*, atau untuk membagikan data yang ada pada aplikasi.

2.7 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek (PBO) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya [14].

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung PBO mengklaim bahwa PBO lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan PBO lebih mudah dikembangkan dan dirawat. Konsep dasar dari pemrograman berorientasi objek menekankan konsep sebagai berikut:

1. Kelas

Kelas merupakan kumpulan atas definisi data dan fungsi-fungsi untuk suatu tujuan tertentu. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen. Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. Objek

Objek berfungsi membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program berorientasi objek.

3. Abstraksi

Abstraksi adalah pengabstrakan atau melakukan seleksi terhadap aspek-aspek tertentu suatu masalah. Abstraksi digunakan untuk penyembunyian kerumitan dari suatu proses. Sebagai contoh, dalam membuat suatu sistem, ada tombol-tombol yang dapat digunakan. Operator atau pengguna tidak perlu berpikir tentang pembuatan tombol tersebut, tetapi yang penting mereka dapat menggunakannya.

4. Enkapsulasi

Pembungkusan sering pula disebut pengkapsulan. Artinya, proses membuat paket (memaketkan) data objek bersama dengan metode-metodenya. Berdasarkan kode program, proses memisahkan aspek-aspek objek dilakukan dengan pembungkusan. Proses pembungkusan itu sendiri merupakan cara atau mekanisme untuk melakukan abstraksi.

5. Polimorpisme

Polimorfisme adalah suatu kejadian ketika objek dapat mengungkap banyak hal melalui satu cara yang sama.

1. Pewarisan

Pewarisan adalah memberikan atau mewariskan sesuatu kepada keturunan berikutnya. Misalnya, seorang anak pasti akan mewarisi beberapa sifat atau perilaku yang dimiliki oleh ibu/bapaknya. Dalam konteks ini, suatu kelas dalam program dapat diturunkan menjadi kelas-kelas baru lainnya yang akan mewarisi beberapa sifat atau perilaku dari kelas induknya

2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan suatu cara untuk menyelesaikan suatu masalah dengan mendeskripsikan yang telah menjadi standar dalam dunia industri untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. Kita dapat membangun model untuk segala bentuk dan jenis sistem perangkat lunak dengan sistem operasi dan jaringan dengan menggunakan UML. UML juga merupakan salah satu *tool* / model untuk merancang pengembangan *software* yang berbasis objek (*object oriented*). UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan *object oriented database* [15].

UML terdiri dari beberapa diagram, yaitu:

1. Diagram Use Case

Diagram *Use Case* merupakan diagram yang digunakan untuk menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pangamatan luar. Persoalan yang ada pada diagram *use case* itu apa yang dilakukan bukan bagaimana melakukannya. Diagram *Use Case* berguna dalam 3 hal, yakni:

- a. Menjelaskan fasilitas atau fitur yang ada
- b. Komunikasi dengan klien
- c. Membuat pengujian dari kasus-kasus secara umum

2. Diagram Class

Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelas dan hubungannya. Diagram *class* bersifat statis, dengan menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Diagram *Class* mempunyai 3 jenis hubungan, yakni:

a. Association

Association merupakan hubungan bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan.

b. Aggregation

Aggregation terjadi jika salah satu kelasnya merupakan bagian dari suatu kumpulan kelas. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian.

c. Generalization

Generalization merupakan hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *superclass* (kelas super) dari kelas lain. *Generalization* memiliki tingkatan yang berpusat pada *superclass*.

1. Diagram Package

Diagram *Package* digunakan untuk mengatur pengorganisasian diagram *Class* yang kompleks, dapat dilakukan pengelompokan kelas-kelas berupa package (paket-paket).

2. Diagram Sequence

Diagram *Sequence* merupakan salah satu diagram interaksi yang menjelaskan bagaimana suatu operasi itu dilakukan, pesan apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu, dan objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

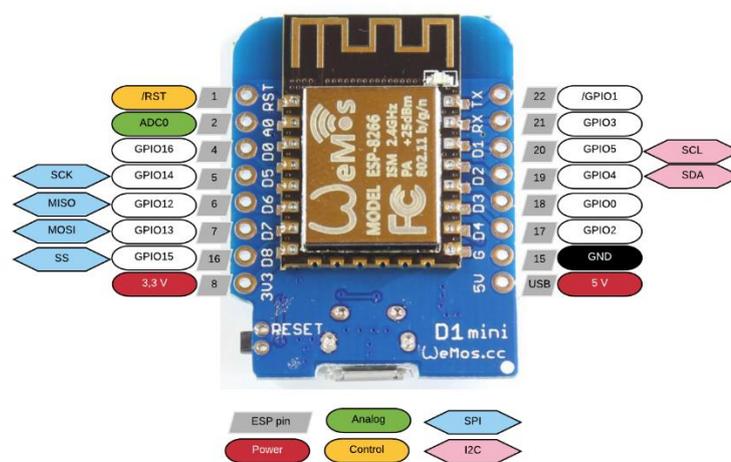
1. Diagram Activity

Pada dasarnya diagram *activity* sering digunakan oleh *flowchart*. Diagram *activity* berfokus pada aktifitas-aktifitas yang terjadi dan terkait dalam suatu proses tunggal. Jadi diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

2.9 Modul Wifi ESP8266

ESP8266 adalah modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP [16].

Modul ini membutuhkan tegangan 3.3V. ESP8266 memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini dilengkapi dengan procecor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang digunakan. Modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler. Perangkat ini dapat diprogram menggunakan Arduino IDE dengan cara menambahkan *library* ESP8266 pada *board manager*. Pada proyek akhir ini jenis modul wifi yang digunakan adalah Wemos D1 Mini. Gambar dari Wemos D1 Mini dapat dilihat pada Gambar II-3.



Gambar II-3 Wemos D1 Mini

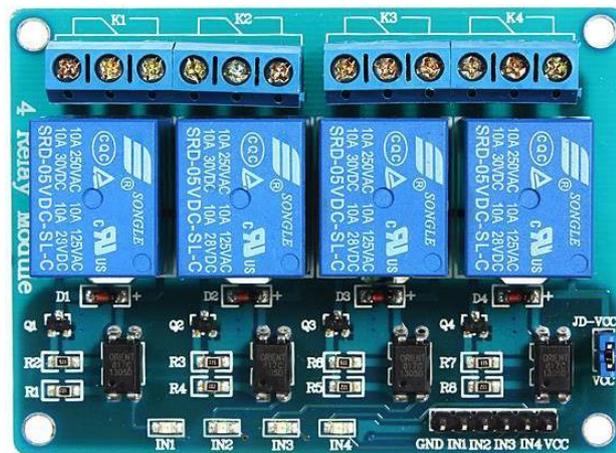
Wemos D1 Mini memiliki 9 pin input/output digital dan 1 pin *input* analog, pin serial RX/TX, led indikator, sambungan mini USB, dan tombol reset. Spesifikasi lengkap Wemos D1 Mini dapat dilihat pada Tabel II-1.

Tabel II-1 Spesifikasi Wemos D1 Mini

Microcontroller	ESP-8266EX
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1(Max input: 3.2V)
Clock Speed	80MHz/160MHz
Flash	4M bytes
Length	34.2mm
Width	25.6mm
Weight	3g

2.10 Modul Relay

Relay adalah suatu perangkat yang berfungsi sebagai saklar elektrik dengan prinsip kerja elektromagnetik untuk menggerakkan kontak saklar apabila *coil* di dalamnya dialiri listrik. Dengan prinsip ini *relay* dapat digunakan sebagai saklar elektronik yang dikendalikan oleh arus listrik yang kecil dari mikrokontroler untuk menghubungkan antara beban dengan sumber listrik tegangan tinggi [17]. Gambar dari Modul Relay dapat dilihat pada Gambar II-5.



Gambar II-4 Modul Relay

Pada umumnya relay memiliki dua kondisi yaitu:

1. Bila kumparan dialiri arus listrik maka kontaknya akan menutup dan disebut sebagai kontak *Normally Open* (NO).
2. Bila kumparan dialiri arus listrik maka kontaknya akan membuka dan disebut dengan kontak *Normally Close* (NC).

2.11 Aplikasi Pendukung

2.11.1 Java

Java merupakan bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems, suatu perusahaan yang terkenal dengan *Workstation* UNIX high-end. Java dibuat setelah C++, dirancang lebih kecil, *source* dan *binary* level-nya sederhana dan *portable* untuk *platform* dan sistem operasi apapun. Hal ini berarti program Java (Applets dan Aplikasi) dapat dijalankan pada mesin apa saja asalkan mempunyai fasilitas JVM (*Java Virtual Machine*). Java sebagai bahasa pemrograman memiliki keuntungan yang lebih bila dibandingkan dengan bahasa dan *environment* lain, sehingga Java cocok untuk pemrograman apa saja.

Platform Java terdiri dari dua komponen, yakni platform Java VM (*Virtual Machine*) dan Java API (*Application Programming Interface*). Java VM adalah

mesin virtual yang menjadi transportasi bagi bahasa pemrograman Java sehingga dapat dimengerti oleh mesin/ komputer. Sedangkan Java API adalah komponen-komponen siap pakai yang dikelompokkan ke dalam *library-library* yang disebut dengan *packages*. Java API dapat digunakan jika terdapat JDK (*Java Development Kit*) pada sistem operasi. Selain untuk mengakses Java API, JDK juga dibutuhkan dan berfungsi untuk membuat aplikasi dengan bahasa pemrograman Java [18].

Java juga memiliki fitur lain, yakni *Garbage Collection (GC)*. *Garbage Collection* adalah sebuah *thread (background process)* yang bertanggung jawab untuk melakukan pembersihan memori yang mungkin untuk dibersihkan. Fitur ini akan berjalan secara otomatis saat program sedang berjalan (*lifetime*). Dengan demikian para pengembang tidak perlu lagi untuk melakukan realokasi memori yang tidak perlu.

Dibawah ini akan disebutkan beberapa keuntungan dari Java antara lain:

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program dengan Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek.

2. Aman

Aplikasi-aplikasi yang dibangun dengan bahasa Java sangat handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat secure, yaitu kasus-kasus seperti buffer overflow yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan keamanannya yang bagus

3. Interaktif

Java memiliki kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya.

1. Multiplatform

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan diatas operating system Microsoft Windows, Linux, dan Mac OS.

2.11.2 Python

Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan Guido Monty Python's Flying Circus. Van Rossum mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif dan memiliki pustaka yang luas.

Python adalah pemrograman berorientasi objek (OOP). Data dalam Python adalah sebuah objek yang dibuat dari kelas (*class*). Pemrograman berorientasi objek merupakan alat yang mudah untuk mengembangkan perangkat lunak yang dapat digunakan kembali [19]. Python saat ini dikembangkan dan dikelola oleh tim relawan yang besar dan tersedia secara gratis dari Python Software Foundation

2.11.3 PHP

PHP (PHP: *Hypertext Preprocessor*) merupakan bahasa pemrograman *open source* yang berjalan pada *server-side*, dan dapat ditanamkan pada HTML sehingga cocok untuk membangun sebuah website yang dinamis dan interaktif. Karena berjalan pada sisi *server*, *source code* PHP tidak dapat dilihat pada halaman klien. PHP mulai dirilis di bawah lisensi Perl-stlye. Pada dasarnya lisensi PHP memberikan kebebasan pada pengembangnya untuk menggunakan, mendistribusikan dan memodifikasi PHP sebanyak yang diinginkan, baik untuk kepentingan komersial ataupun bukan komersial. Pengembang hanya perlu

menjelaskan kepada para pengguna aplikasi bahwa apa yang telah didistribusikan mengandung PHP [20].

2.11.4 Bahasa C

Bahasa C dikembangkan pada Lab Bell pada tahun 1978, oleh Dennis Ritchi dan Brian W. Kernighan. Pada tahun 1983 dibuat standar C yaitu standar ANSI (American National Standards Institute), yang digunakan sebagai referensi dari berbagai versi C yang beredar dewasa ini termasuk Turbo C.

Dalam beberapa literature, bahasa C digolongkan bahasa level menengah karena bahasa C mengkombinasikan elemen bahasa tinggi dan elemen Bahasa rendah. Kemudahan dalam level rendah merupakan tujuan diwujudkan bahasa C. Kemudahan dalam level rendah merupakan tujuan diwujudkan Bahasa C. pada tahun 1985 lahirlah pengembangan ANSI C yang dikenal dengan C++ (diciptakan oleh Bjarne Stroustrup dari AT & T Lab). Bahasa C++ adalah pengembangan dari bahasa C. Bahasa C++ mendukung konsep pemrograman berorientasi objek dan pemrograman berbasis windows. Sampai sekarang Bahasa C++ terus berkembang dan hasil perkembangannya muncul bahasa baru pada tahun 1995 (merupakan keluarga C dan C++ yang dinamakan java). Istilah prosedur dan fungsi dianggap sama dan disebut dengan fungsi saja. Hal ini karena di C++s sebuah prosedur pada dasarnya adalah sebuah fungsi yang tidak memiliki tipe data kembalian (*void*) [21]. Hingga kini bahasa ini masih populer dan penggunaannya tersebar di berbagai platform dari windows sampai linux dan dari PC hingga *mainframe*. Ada pun struktur dari bahasa C sebagai berikut:

1. Program bahasa C tersusun atas sejumlah blok fungsi.
2. Setiap fungsi terdiri dari satu atau beberapa pernyataan untuk melakukan suatu proses tertentu.
3. Tidak ada perbedaan antara prosedur dan fungsi.
4. Setiap program bahasa C mempunyai suatu fungsi dengan nama "*main*" (Program Utama).
5. Fungsi bisa diletakkan diatas atau dibawah fungsinya "*main*".

1. Setiap statemen diakhiri dengan semicolon (titik koma).

2.11.5 Android Studio

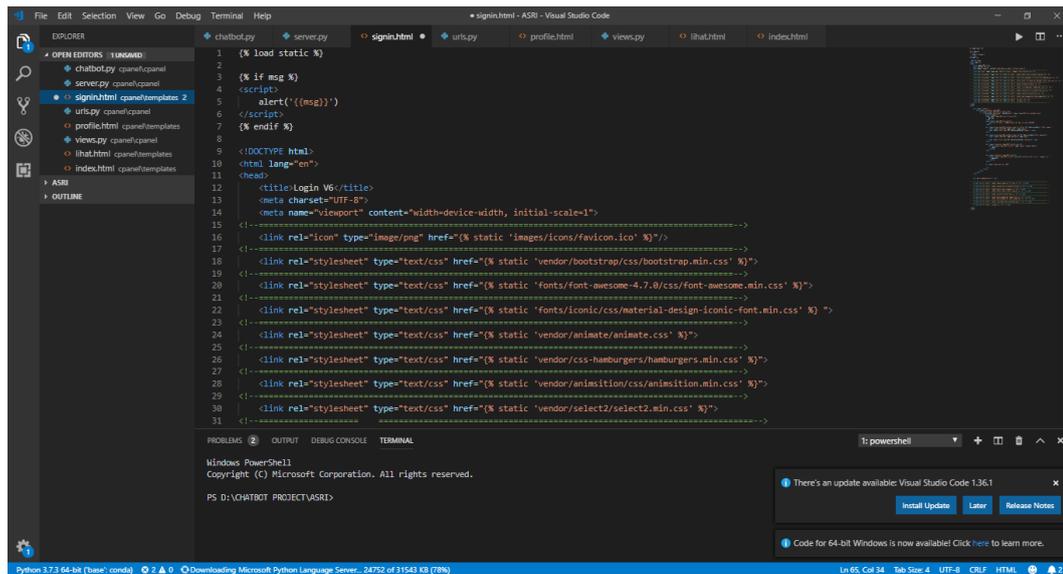
Android Studio merupakan tools aplikasi atau IDE yang digunakan untuk membuat atau pengembangan sebuah aplikasi berbasis Android. Android Studio dibangun oleh Google untuk memudahkan pengembang dalam membangun aplikasi Android dengan memanfaatkan berbagai *library* [22]. Android Studio menawarkan fitur yang lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

- a. Sistem pembuatan berbasis Gradle yang fleksibel.
- b. *Emulator* yang cepat dan kaya fitur.
- c. Lingkungan yang menyatu untuk pengembangan semua perangkat Android
- d. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- e. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.
- f. Dukungan C++ dan NDK.
- g. Alat penguji dan perangkat kerja yang ekstensif.
- h. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.

2.11.6 Visual Studio Code

Visual Studio Code adalah *editor source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary. Muncul dengan built-in dukungan untuk JavaScript, Node.js dan

memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C#, Python, dan PHP. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi cross-platform dari Atom komponen kode-editing, berdasarkan JavaScript dan HTML5 [23]. Berikut adalah tampilan dari Visual Studio Code yang dapat dilihat pada Gambar II-5.



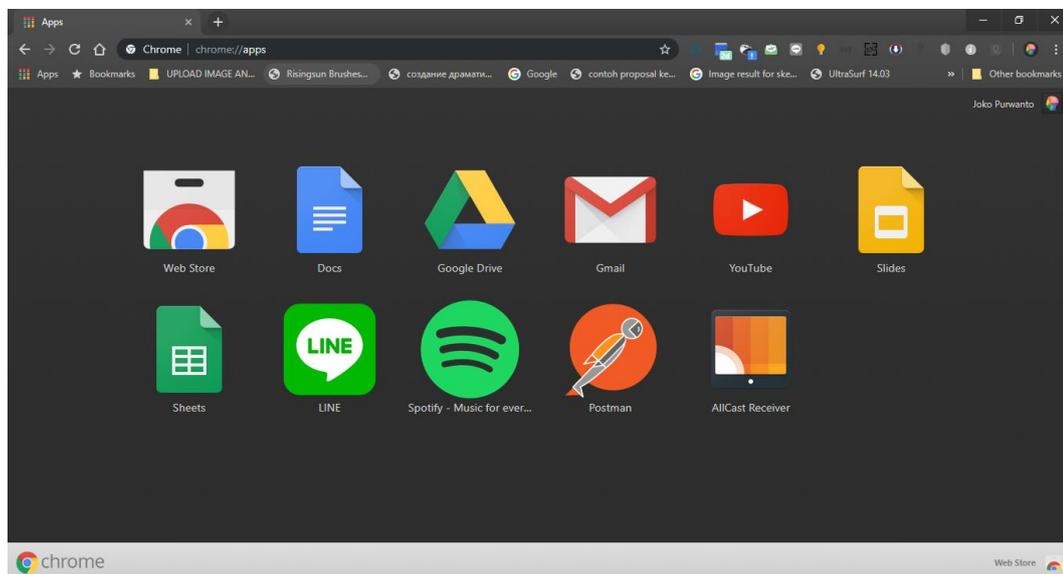
Gambar II-5 Tampilan Visual Studio Code

Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online.

2.11.7 Google Chrome

Google Chrome adalah web browser gratis yang dikembangkan oleh Google. Chrome pertama kali dirilis pada bulan September 2008, untuk Microsoft Windows, dan kemudian dikirim ke Linux, macOS, iOS dan Android. Google Chrome juga merupakan komponen utama Chrome OS, yang berfungsi sebagai

platform untuk menjalankan aplikasi web [24]. Berikut adalah tampilan dari Google Chrome yang dapat dilihat pada Gambar II-6.



Gambar II-6 Tampilan Browser Google Chrome

Google merilis sebagian besar sumber kodenya sebagai proyek open source. Salah satu komponen yang bukan open-source adalah Adobe Flash Player bawaan (yang telah dinonaktifkan Chrome secara default sejak September 2016). Chrome menggunakan WebKit layout engine sampai versi 27. Seperti versi 28, semua port Chrome kecuali port iOS menggunakan Blink.

Pada tahun 2018, StatCounter memperkirakan bahwa Google Chrome memiliki pangsa penggunaan browser web sebesar 66% di seluruh dunia sebagai browser desktop. Chrome juga memiliki pangsa pasar 56% di semua platform yang digabungkan, karena memiliki lebih dari 50% pangsa pada smartphone. Keberhasilannya telah menyebabkan Google mengembangkan nama merek "Chrome" di berbagai produk lainnya seperti Chromecast, Chromebook, Chromebit, Chromebox dan Chromebase.

2.11.8 IDE Arduino

“*Integrated Development Environment* (IDE) yaitu berupa software processing yang digunakan untuk menulis program kedalam mikrokontroler seperti Arduino Uno, merupakan penggabungan antara bahasa C++ dan Java” [25].

Berdasarkan definisi di atas, maka dapat disimpulkan *Integrated Development Environment* (IDE) adalah software atau program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan aplikasi mikrokontroler mulai dari menuliskan source program, kompilasi, upload hasil kompilasi dan uji coba secara terminal serial. IDE Arduino dapat diinstall di berbagai sistem operasi seperti Linux, Mac OS, Windows. Software IDE Arduino terdiri dari tiga bagian yaitu:

1. Editor Program

Untuk menulis dan mengedit program dalam bahasa processing. Listing program pada Arduino disebut Sketch.

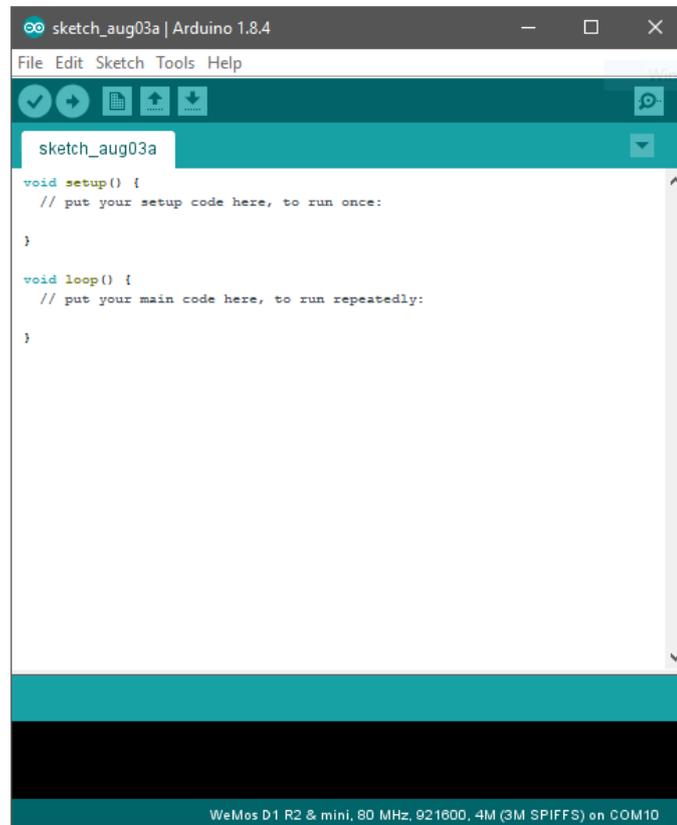
2. Compiler

Modul yang berfungsi mengubah bahasa processing (kode program) ke dalam kode biner, karena kode biner adalah bahasa satu-satunya bahasa program yang dipahami oleh Mikrokontroler.

3. Uploader

Modul yang berfungsi memasukan kode biner kedalam memori Mikrokontroller.

Struktur perintah pada Arduino secara garis besar terdiri dari dua bagian yaitu *void setup* dan *void loop*. *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan sedangkan *void loop* berisi perintah yang akan di eksekusi berulang-ulang selama Arduino dinyalakan. Berikut adalah tampilan dari IDE Arduino yang dapat dilihat pada Gambar II-7.



Gambar II-7 IDE Arduino

2.11.9 Firebase

Firebase memiliki produk utama, yaitu menyediakan *database realtime* dan *backend* sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* Firebase ini. Firebase menyediakan *library* untuk berbagai client platform yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service*) dengan konsep *realtime*. Firebase digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh developer [26].

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.

2.11.10 Microsoft Visio

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), brainstorm, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya [27].

Visio aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakuisisi oleh Microsoft pada tahun 2000. Versi yang telah menggunakan nama Microsoft Visio adalah Visio 2002, Visio 2003, Visio 2007, Visio 2013 dan Visio 2016 yang merupakan versiter baru. Visio 2016 *Standard* dan *Professional* menawarkan antarmuka pengguna yang sama, tapi seri *Professional* menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (*layout*). Disini penulis menggunakan Visio 2016 *Professional* untuk pembuatan diagram UML. Berikut ini merupakan beberapa *format file* yang digunakan dalam Microsoft Visio dapat dilihat pada Tabel II-2.

Tabel II-2 Format File Microsoft Visio

No	Format	File
1	VSD	Diagram
2	VSS	Stencil
3	VST	Template
4	VSW	Web
5	VDX	XML Daigarm
6	VSX	XML Stencil
7	VTX	XML Template
8	VSL	Add-on