BAB 2
TINJAUAN PUSTAKA

2.1. Landasan Teori

Landasan teori dari penelitian ini menguraikan proses analisis sistem serta mendukung proses pembangunan sistem absensi pada SMK Negeri 1 Kayuagung berbasis IoT menggunakan *Bluetooth Low Energy* (BLE).

2.1.1. Sistem Absensi

Sistem absensi menurut Setiawan (2015) adalah suatu pendataan kehadiran yang merupakan bagian dari aktifitas pelaporan yang ada pada suatu instansi. Absensi disusun dan diatur sehingga mudah untuk dicari dan dipergunakan saat diperlukan oleh pihak yang berkepentingan.

2.1.2. *Internet of Things (IoT)*

Menurut Efendi (dalam Arafat 2016:262-268), *Internet of Things* atau biasa dikenal juga dengan singkatan IoT merupakan sebuah konsep yang memanfaatkan konektifitas internet yang tersambung terus menerus sehingga memungkinkan untuk menghubungkan mesin, peralatan serta benda fisik lainnya.

Internet of Things adalah perkembangan keilmuan yang amat menjanjikan untuk memaksimalkan kehidupan yang didasarkan oleh sensor cerdas serta peralatan pintar yang bekerjasama melalui internet (Keoh, Kumar, & Tschofenig, 2014). Awal mula 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. Steve mann menciptakan *WearCam* pada tahun 1994. Pada Tahun 1999, Kevin Ashton menciptakan *The Internet of Things*, direktur eksekutif Auto IDCentre, MIT. Mereka juga menemukan peralatan berbasis RFID (*Radio Frequency Identification*) global yang sistem identifikasi pada tahun yang sama. Penemuan ini disebut sebagai sebuah lompatan besar dalam *commercializing* IoT.

Menurut beberapa penilitian, IoT sudah banyak diterapkan di beberapa industri dan bidang keilmuan, seperti dalam bidang ilmu informatika, kesehatan , geografis, dan beberapa bidang ilmu lai. Beberapa penelitian yang sudah

melakukan riset tentang *monitoring* kesehatan pasien menggunakan *wireless* sensor yang dipasangkan pada tubuh pasien. Beberapa hal yang dipantau adalah psikologi pasien, tekanan darah, detak jantung semua kegiatan tersebut dilakukan secara remote melalu peralatan yang terhubung ke internet dengan tetap memperhatikan kerahasiaan data pasien.

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 (Elkhodr, Shahrestani, & Cheung, 2012).

2.1.3. *Bluetooth Low Enery* (BLE)

Menurut Rizaldi (dalam Indrayana, 2018:2462-2472) *Bluetooth low Energy* (BLE) dirancang agar menggunakan daya yang jauh lebih rendah, jangkauan sinyal yang lebih luas, relative cepat, dan komunikasi tidak depengaruhi benda. Teknologi nirkabel baru ini dapat diimplementasikan di berbagai tempat dengan konsumsi energy yang rendah, berbeda dengan spesifikasi *Bluetooth* sebelumnya. Implmentasi perangkat akan memungkinkan untuk menjalankannya selama bertahun-tahun pada standar baterai *coin-cell.* Selain itu penggunaan *bluetooth* 4.0 ini memiliki jangkauan yang lebih luas, dapat dioeprasikan *multi-vendor* dan memakan biaya yang relativ terjangkau dengan konfigurasi yang cukup mudah.

Teknologi ini merupakan penyempurna kapabilitas *Bluetooth 3.0,* generasi *Bluetooth* sebelumnya. Saat ini, Bluetooth 4.0 dianggap sebagai teknologi yang paling cocok untuk digunakan sebagai media transfer jaringan komunikasi jarak dekat meskipun memiliki pesaing perusahaan teknologi nirkabel yang tidak kalah besarnya seperti WiBro, UWB (ultra wide band), dan WiFi.

2.1.4. BLE Beacon

BLE Beacon pada dasarnya adalah sebuah perangkat yang sangat sederhana berupa perangkat *wireless* kecil yang berbasiskan *Bluetooth Low Energy* yang mentransmisikan sinyal radio secara terus menerus yang berkaitan dengan ID dari beacon tersebut.

Beacon/pemancar BLE dapat dideteksi oleh perangkat yang berkemampuan BLE. *Beacon* memiliki daya pancar sampai 100 meter. Setiap perangkat keras yang mendukung *beacon*/pemancar BLE pasti mendukung satu atau beberapa spesifikasi *beacon* berikut diantaranya iBeacon, AltBeacon, dan EddyStone/UriBeacon. *Beacon* memiliki beberapa parameter di antaranya.

1. *Universally Unique Identifier* (UUID)

   UUID berfungsi sebagai indentifikasi pemilik *beacon*/perangkat tersebut. UUID memiliki panjang 32-digit heksadesimal, salah satu contoh UUID adalah ec602aff-1b0c-4fb4-abf4-e6520d99ff25. Dalam penelitian ini, UUID digunakan untuk mengindentifikasi perangkat siswa.

2. *Receive Signal Strength Indicator* (RSSI)

   RSSI adalah indikator untuk mengukur kekuatan dari sinyal yang diterima. RSSI yang dipancarkan oleh *beacon* berkisar antara -40 sampai -100, semakin besar nilai RSSI maka semakin kuat sinyal yang diterima.

3. *Major Value*

   Major value merupakan berupa integer berukuran 16-bit bernilai antara 0 sampai 65535 yang berfungsi untuk mengidentifikasi sebuah grup *beacon*.

4. *Minor Value*

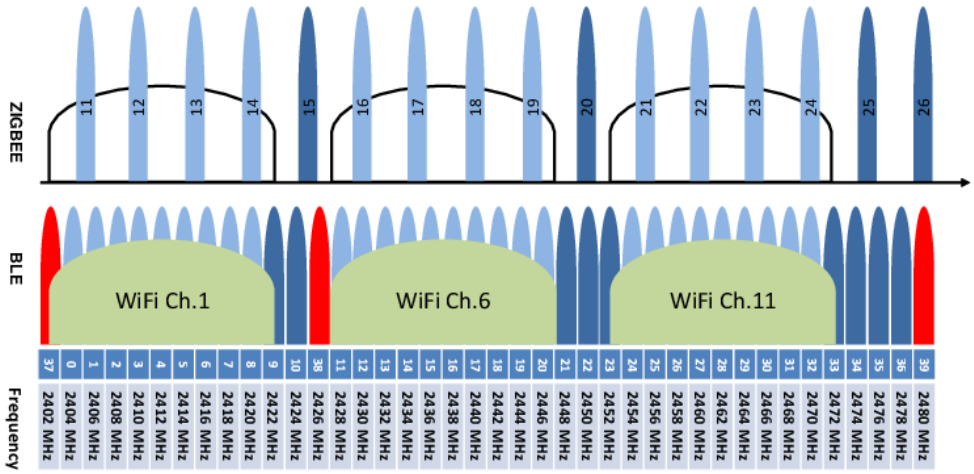
   Minor value merupakan berupa integer berukuran 16-bit bernilai antara 0 sampai 65535 yang berfungsi untuk mengidentifikasi sebuah *beacon* secara individual.

BLE sangat mudah dibaca dan dideteksi. Pesatnya perkembangan *smartphone android* saat ini sangat memungkinkan untuk *smarthphone anroid* di setting menjadi *beacon.* Bahkan saat ini dengan berkembangnya *Bluetooth 5.0*, menurut Bluetooth SIG, dapat menjangkau 4 kali lipat dibandingkan dengan *Bluetooth* 4.0. Selain itu dari segi *low energy*, teknologi ini menciptakan interaksi seamless yang tidak mengonsumsi energy batere (secara teori, dengan batere 3 volt dapat bertahan selama 2 tahun). Selain itu karena sistem yang tidak kompleks, teknologi beacon ini sangat memudahkan developer dalam pengembangannya.

2.1.5. *Advertising Packet* dan *Scan Response* Data pada BLE

Advertising *Packet* selalu dikirimkan secara terus menerus yang dikirimkan dari perangkat phiperal agar dapat terdeteksi oleh perangkat lain. Saat perangkat lain menerima data ini, perangkat tersebut dapat meminta data tambahan dari perangkat periferal yang kemudian mengirim Scan Response Data.

Menurut Shofi(dalam spesifikasi Bluetooth Core 4.2), sebuah sebuah perangkat Bluetooth Low Energy pada umumnya bekerja menggunakan 3 mode yang berbeda hanya ketika berada pada kondisi "*discovery*", kondisi ini yaitu *advertising*, *scanning*, dan kondisi *inisialisasi*. *Advertiser* adalah perangkat BLE yang berada dalam kondisi *advertising*, dimana pada kondisi ini, perangjat *advertiser* mengirimkan sebuah informasi data berupa *advertising data* menggakan saluran *edvertising* yaitu kanal 37, 38, dan 39, kemudian perangkat advertiser ini menunggu respon data dari perangkat lainnya. *Scanner* atau bisa disebut juga dengan perangkat inisiator, adalah sebuah perangkat yang berada pada kondisi *scanning* dimana pada kondisi ini, perangkat *scanner* secara periodic melakukan proses dengan mendengarkan paket *advertising data* yang dikirimkan dari saluran *advertising* yaitu saluran 37, 38, dan 39. Saluran frekuiensi yang digunakan pada proses ini ditunjukkan pada gambar dibawah ini dimana warna merah menunjukkan saluran *advertising* BLE 37, 38, dan 39; biru muda menunjukkan saluran data BLE dan ZigBee yang tumpeng tindih dengan saluran WiFi dan biru tua adalah saluran data BLE dan ZigBee yang tidak tumpang tindih dengan saluran WiFi (Bebas dari gangguan WiFi). saluran *Advertising* 37, 38, dan 39 digunakan dengan tujuan terhindar adanya interferensi antar jaringan seperti jaringan WiFi.



Gambar 2.1 Kanal *Advertising* pada *Bluetooth Low energy*

2.1.6. Koneksi pada BLE

Koneksi yaitu pertukaran data secara permanen dan berkala antara dua perangakat. Menurut T'Jonck (2021:3) dalam koneksi antar dua perangkat BLE ada dua peran yang didefinisikan dalam *Generic Access Protocol (GAP),* lapisan yang bertanggung jawab untuk menangani koneksi ini yaitu

1. Master (*Central*/Perangkat Pusat*)* adalah perangkat yang menemukan perangkat *peripheral* dengan mendengarkan iklan dari perangkat tersebut.
2. *Slave (Peripheral*/Perangkat periferal*)* adalah perangkat yang menyiarkan paket iklan untuk membuat dirinya dapat ditemukan. Perangkat master (*central*) dapat merespon iklan ini untuk memulai koneksi dengan perangkat ini.

Setelah koneksi terbentuk, perangkat pusat mengatur waktu dan memulai pertukaran data secara berkala.

2.1.7. Mikrokontroler ESP32

Menurut Muliadi (2020:74), ESP32 adalah penerus dari mikrokontroler ESP8266 yang buat oleh Espressif System. Pada mikrokontroler ini sudah tersedia modul WiFi dan *Bluetooth Low Energy* (BLE) sehingga untuk membuat alat yang memerlukan adanya WiFi dan *bluetooth* tidak membutuhkan komponen tambahan

sehingga tidak memakai banyak ruang dan lebih hemat biaya. Berikut adalah spesifikasi lengkap dari mikrokonroler ESP32.

Tabel 2.1 Spesifikasi ESP32

Kategori	**Item**	**Spesifikasi**
Sertifikasi	Sertifikasi RF	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Sertifikasi Wi-Fi	Wi-Fi *Alliance*
	Sertifikasi Bluetooth	BQB
	Sertifikasi Green	RoHS/REACH
Wi-Fi	Protokol	802.11 b/g/n (802.11n *up to* 150 Mbps)
	Frekuensi	2.4 GHz ~ 2.5 GHz
Bluetooth	Protokol	Bluetooth v4.2 BR/EDR dan BLE
	Radio	NZIF receiver sampai dengan sensitivitas –97 dBm
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I 2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®, compatible with ISO11898-1)
	On-chip sensor	Hall sensor
	SPI Flash	4 MB
	Power Supply	3.0 V ~ 3.6 V

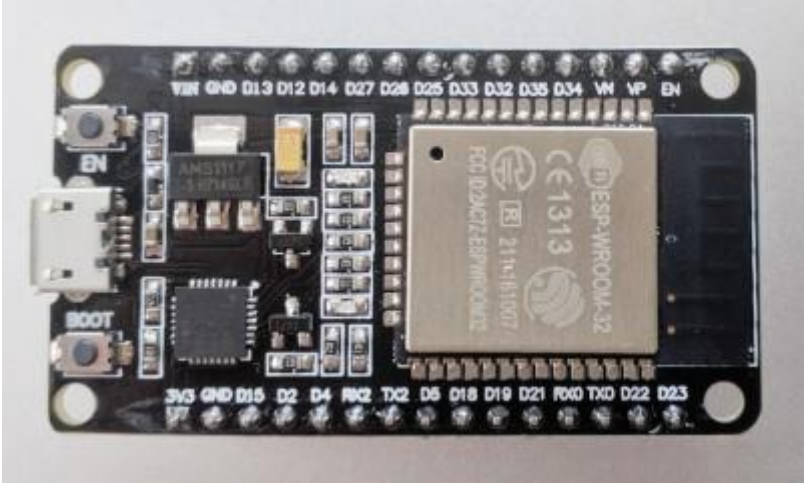
Mikrokontroler ESP32 juga lebih lengkap jika dibandingkan dengan mokrokontroler lain seperti Arduino ataupun nodeMCU EPS8266. Perbandingan mikrokontroler ESP32 dengan yang lain dapat dilihat pada tabel berikut.

Tabel 2.2 Perbedaan ESP32 dengan mikrokontroler lain

	Arduino Uno	**NodeMCU(ESP8266)**	**ESP32**
Tegangan	5 volt	3.3 volt	3.3 volt
CPU	ATmega 328-16 MHz	Xtensa single core L106 - 60 MHz	Xtens dual core LX6 - 160M Hz
Arsitektur	8 bit	32 bit	32 bit
Flash Memory	32KB	16MB	16MB
SRAM	2KB	160KB	512KB
GPIO Pin (ADC/DAC)	14 (6/-)	17 (1/-)	36(18/2)
Bluetooth	Tidak ada	Tidak ada	Ada

WiFi	Tidak ada	Ada	Ada
SPI/12C/UART	1/1/1	2/1/2	4/2/2

Terlihat perbedaan yang menjadi keunggulan mikrokontroler ESP32 dibanding dengan mikrokontroler yang lain, mulai dari pin out yang lebih banyak, pin analog lebih banyak, memori yang lebih besar, terdapat *Bluetooth Low Energy* serta tersedia WiFi yang memungkinkan untuk mengaplikasikan *Internet of Things* dengan mikokontroler ESP32. Berikut gambar fisik dari mikrokontroler ESP32.



Gambar 2.2 Mikrokontroler ESP32

2.1.8. Metode Perancangan Sistem

Teknik pemprograman yang memasukkan *property* dan *method* kedalam beberapa objek biasanya dinamakan teknik pemprograman berbasis objek atau *Object Oriented Programming* (OOP). *Property* ini lebih dikenal dengan variabel yang bersifat *public* dan *method* merupaka *function.* Dalam OOP ada yang namanya *inheritance* atau turunan yaitu membentuk *class* baru untuk memiliki bagian bagian dari *class* yang sudah ada sebelumnya.

Dengan pengelompokan-pengelompokan ini maka *programming* akan terstruktur dengan baik dan mudah dikelola. Bahasa *programming* yang sudah mengimplementasikan OOP adalah Java, C, dll. Pemrograman Python juga dikembangkan menjadi bahasa pemrograman berbasis OOP *(Object Orieented Programming)* dengan dibuatnya framework berbasis Python.

2.1.8.1. *Object Oriented Programming* (OOP)

Menurut Rais (2019:96) *Object Oriented Programming* (OOP) atau pemrograman berbasis objek adalah suatu strategi atau cara baru untuk membuat program atau merancang sistem dengan memperhatikan objek. Metode berorientasi objek sudah banyak digunakan karena metodologi yang lama banyak menimbulkan masalah seperti kesulitan dalam mentransformasikan hasil dari satu tahap pengembang ketahap berikutnya, misalnya pada contoh pendekatan terstruktur. Adapun keuntungan penggunaan OOP dalam membangun sebuah sistem, diantaranya.

1. Meningkatkan produktifitas

   Karena kelas atau objek yang sudah dibuat untuk menyelesaikan masalah tertentu akan dapat digunakan lagi atau dimodifikasi untuk menyelesaikan masalah lainnya (reusable).
2. Kecepatan pengembangan

   Dapat mengurangi kesalahan pada pengodean sehingga dapat menambah kecepatan dalam pengembangan.
3. Kemudahan pemeliharaan

   Karena model objek yang tersturktur menyebabkan pola-pola yang cenderung stabil dan tetap dapat dipisahkan.
4. Adanya konsistensi

   Karena ada sifat *inheritance* atau pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Sistem yang berorientasi objek ini, memiliki fungsi serta data-data dikelompokkan dalam sebuah komponen yang dibungkus (enkapsulasi) ke dalam bentuk objek, sehingga setiap objek dapat mewariskan sifatnya atau setiap objek yang berbeda. Kumpulan dari objek-objek itu akan berinteraksi satu sama lainnya

untuk menghasikan output yang diinginkan. Metodologi pengembangan sistem yang berorientasi objek memiliki beberapa konsep dasar yang harus dipahami, meliputi:

1. Kelas (*Class*)

   Kelas adalah kumpulan objek-objek dengan karateristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama mungkin lahir atau diciptakan oleh kelas tersebut. Suatu kelas dapat diturunkan kekelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Sebuah kelas mempunyai sifat (atribut), kelakuan (operasi/*method*), hubungan *(relationship)*.

2. Objek (*Object*)

   Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan. Untuk lebih memahami bisa juga anggap objek sebagai sesuatu yang mewakili dunia nyata seperti manusia, benda, tempat, status, kejadian, struktur, atau hal-hal lain yang bersifat abstrak.

3. Metode (*Method*)

   Metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau yang dilakukan oleh objek.

4. Atribut (*Attribute*)

   Variabel global yang terdapat dalam kelas disebut atribut. Atribut dapat berupa nilai data yang dimiliki oleh objek. Atribut sebaiknya bersifat privat untuk menjaga enkasulapsi. Contoh dari atribut misalnya jenis, berat, nama, dan sebagainya.

5. Abstraksi (*Abstraction*)

   Prinsip untuk mempresentasikan dunia nyata yang komplek menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*Encapsulation*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*Inheritance*)

   Pewarisan adalah mewarisi sebagian atau seluruh defenisi dan objek lain sebagai bagian dari dirinya.

8. Antarmuka (*Interface*)

   Antarmuka atau interface biasa digunakan agar kelas lain tidak mengakses langsung ke suatu kelas, mengangkses antarmukanya. Antarmuka atau interface sangat mirip dengan kelas, tapi antara antribut kelas tanpa memiliki metode yang dideklarasikan tanpa isi.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefenisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khusus (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antar Objek

    Komunikasi antara objek dilakukan lewat pesan yang dikirim satu objek ke objek yang lainnya.

12. *Polimorpisme (Polymorphism)*

    Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas bernama sama disimpan dalam package yang berbeda.

2.1.8.2. *Unfield Modelling Language* (UML)

Tujuan utama dibuatnya UML adalah membantu tim dalam membangun sebuah proyek mengeksplorasi potensi desain, berkomunikasi, dan memvalidasi desain arsitektur dari perangkat lunak yang dibangun (Haviluddin, 2011:1). Terdapat 3 komponen UML yaitu *Object-Oriented Software Engineering* (OOSE), *Object Modelling Technique* (OMT), dan *Object-Oriented Design* (OOD).

1. *Class* Diagram

   Class diagram menggambarkan hubungan antar kelas, atribut, dan operasi dan menggambarkan struktur statis dari kelas dalam sistem. Selama dalam tahap design peran dari *class* diagram adalah untuk menangkap struktur dari semua kelas yang membantuk arsitektur sistem. *Class* diagram memiliki 3 area pokok: nama, atribut, dan metode.

2. *Use Case* Diagram

   Use case digambarkan sebagai elips horizontal dalam suatu diagram UML. *Use case* diagram adalah diagram yang menggambarkan *use case, actor,* dan relasi urutan atau tindakan yang memberikan nilai terukur untuk aktor. *Use case* memiliki dua istilah yaitu interaksi dengan sistem dan interaksi bisnis dengan dunia nyata atau konsumen.

3. *Activity* diagram

   Activity diagram adalah kegiatan alur kerja yang menggambarkan perilaku sistem untuk aktifitas, objek, *event,* dan transisi *state*.

4. *Sequence* diagram

   Sequence diagram merupakan konsep langkah demi langkah dan urutan perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

2.1.9. Pengujian Perangkat Lunak

Untuk menemukan kesalahan atau kekurangan pada sistem yang dibangun dibutuhkan pengujian sistem agar dapat mengetahui apakah sistem yang dibangun sudah layak digunakan atau tidak. Pada pembangunan sistem metode pengujian

yang diguanakan yaitu metode *black box*. Pada metode ini terdapat 2 tahapan pengujian yaitu tahap pengujian alpha dan tahap pengujian beta.

2.1.9.1. Pengujian *Alpha*

Pengujian alpha adalah tahapan pengujian fungsional yang dilaksanakan dilingkungan pengembangan oleh sekumpulan *user* yang akan menggunakan perangkat lunaknya, pihak dari pengembang sistem mendampingi sambal mencatan kesalahan yang dirasakan oleh pengguna.

2.1.9.2. Pengujian Beta

Pengujian beta merupakan tahapan pengujian yang dilakukan langsung dilapangan untuk mengetahui kepuasan *user* mulai dari kebutuhan dari awal pembangunan, fungsional, dan tampilan antarmuka dari sistem yang dikembangkan.

2.1.10. Perangkat Lunak Pendukung

Adapun perangkat lunak yang digunakan untuk mempermudah dalam pembangunan sistem. Berikut beberapa perangkat lunak yang dibutuhkan dalam pembangunan sistem.

2.1.10.1. MySQL

Menurut Yuliansyah (2014:827) MySQL adalah sebuah *Database Management System* (DBMS) yang memiliki fungsi sebagai *Relational Database Manajemen System* (RDBMS). MySQL memiliki sifat *open source* dan mudah digunakan.

2.1.10.2. XAMPP

XAMPP merupakan perangkat lunak *open source* dan dapat dijalankan diberbagai sistem operasi dan memiliki fungsi sebagai server yang dapat berdiri sendiri. Perangkat lunak XAMPP dapat menjalankan beberapa program di antaranya Apache HTTP Server, MySQL *database*, dan lain-lain.

2.1.10.3. Visual Studio Code

Visual studio code merupaka editor teks atau *source code* yang dibuat oleh Microsoft untuk berbagai platform seperti Windows, Linux, dan Mac OS. Banyak fitur yang disediakan dari text editor tersebut diantaranya *GIT Control*, *debungging*,

penyesuaian *syntax,* dan pengguna dapat mengubah tema, *shortcut keyboard,* dan preferansi lainnya.

2.1.10.4. Android Studio

Menurut Juansyah (2015:2-3) Android studio adalah IDE (*Integrated Development Environment*). Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio mengantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android studio resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis.

Beberapa fitur dari android studio yaitu projek berbasis pada *Gradle Build*, *refactory* dan perbaikan *bug* yang cepat, *tools* baru yang bernama "*Lint*" dapat memonitor kegunaan, kecepatan, serta kompatibilitas aplikasi dengan cepat, mendukung *Proguard and App-signing* untuk keamanan, memiliki GUI mudah dipahami, didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan dan masih banyak lagi fitur-fitur yang disediakan.

2.1.10.5. Arduino IDE

Arduino IDE adalah sebuah perangkat lunak yang dibuat menggunakan bahasa java berfungsi digunakan untuk mengupload kode program ke dalam *hardware* Arduino ataupun sejenisnya. Arduino IDE terdiri dari 3 fitur utama yaitu editor pogram, *compiler,* dan *uploader.*

Ada beberapa menu pilihan pada Arduino IDE yaitu *verify* yang berfungsi untuk melakukan pengecekan dan kompilasi kode. Upload berfungsi untuk upload kode ke dalam *board/*mikrokontroler dan *serial monitor* berfungsi untuk melihat komunikasi yang terjadi antara komputer dan mikrokontroler dengan cara membuka serial port monitor.