

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

Bangunan Gedung Mapolwiltabes Bandung yang bertempat di Jl. Merdeka No. 16, 18 dan 20 Bandung didirikan pada tahun 1866, dulunya berfungsi sebagai Sekolah Guru (*Kweekschool Voor Inlandsche Onderwijzers*) yang bangun atas inisiatif seorang Belanda bernama *K.F. Hole* sebagai Administratur Perkebunan Teh Waspada di Gunung Cikuray, Bayongbong, Garut. Di sekolah inilah pernah menimba ilmunya tokoh-tokoh nasional, seperti *Abdulharis Nasution*, *Otto Iskandardinata* dan yang lainnya.

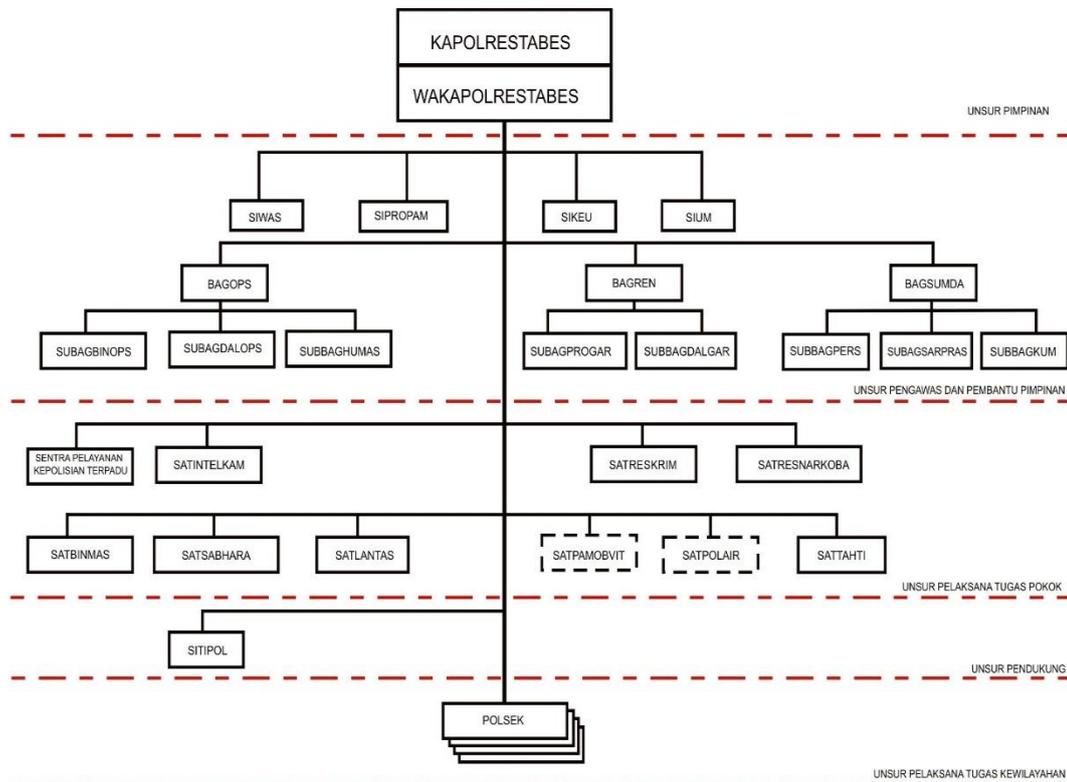
Dilihat dari sejarah berdirinya Polrestabes Bandung, dimulai pada tahun 1966, dimana belum adanya polsek-polsek jajaran seperti saat ini. Kepolisian di Bandung pada tahun tersebut, berdiri dengan nama "Komtabes-86 Bandung" dengan pembagian wilayah hukum pada saat itu, terdiri dari:

1. Seksi I di Jl. Dalam Kaum, Alun-alun Bandung
2. Seksi II di Jl. Sawung Galing Bandung
3. Seksi III di Jl. Pasirkaliki Bandung
4. Seksi IV di Jl. Asia Afrika (Simpang Lima) Bandung

Pada tahun 1970, nama Komtabes-86 Bandung diganti namanya menjadi "Poltabes Bandung" dengan pembagian wilayah hukum pada saat itu terdiri dari 16 (enam belas) Polsekta, yaitu : Bandung Kulon, Babakan Ciparay, Bojong Loa, Astana Anyar, Andir, Cicendo, Sukajadi, Sukasari, Cidadap, Cihapit, Coblong, Regol, Lengkong, Batununggal, Kiaracondong dan Cibeunying.

2.1.1 Struktur Organisasi

Dalam menjalankan tugas yang telah dibebankan, Polrestabes Kota Bandung memiliki struktur organisasi sebagai berikut :



Gambar 2.1 Struktur Organisasi

2.1.2 Visi dan Misi Polrestabes Kota Bandung

Visi dan Misi SLB Negeri Cileunyi adalah sebagai berikut :

a) Visi

Terwujudnya pelayanan prima kepada masyarakat dalam upaya penegakan hukum dan terpeliharanya keamanan dan ketertiban masyarakat di wilayah hukum Polrestabes Bandung guna meningkatkan kepercayaan masyarakat

b) Misi

1. Meningkatkan Sumber Daya manusia resort Kota Bandung untuk tampil sebagai sosok pelayanan prima dalam penegakan hukum sesuai perkembangan dan tantangan yang di hadapi.
2. Melaksanakan pelayanan secara optimal sehingga dapat menimbulkan kepercayaan bagi masyarakat dalam upaya meningkatkan kesadaran hukum masyarakat dan mewujudkan kemitraan.
3. Memelihara soliditas institusi Polrestabes Bandung dari berbagai pengaruh yang merugikan organisasi.

4. Memelihara keamanan dan ketertiban masyarakat dengan tetap berpedoman kepada norma budaya masyarakat dan sekitarnya.
5. Melaksanakan penegakan hukum secara konsisten berkesinambungan, profesional, dan transparan dengan mengedepankan dan menjunjung tinggi HAM serta bekerja sama dengan unsur penegak hukum lainnya untuk memelihara kamtibmas dalam kehidupan bermasyarakat.
6. Menciptakan kondisi keamanan yang kondusif dengan peran serta masyarakat dan instansi terkait secara aktif dalam pengamanan hasil pemilu dan kebijakan pemerintah.
7. Melakukan pengendalian dan pengawasan secara berjenjang untuk mengurangi adanya penyimpangan pada sesuatu objek.
8. Mengelola secara profesional dalam bertugas, transparan dalam memilih, *akuntabel*, dan *modern* seluruh sumber daya Polri untuk mendukung operasional tugas polisi.

2.2 Penelitian Sebelumnya

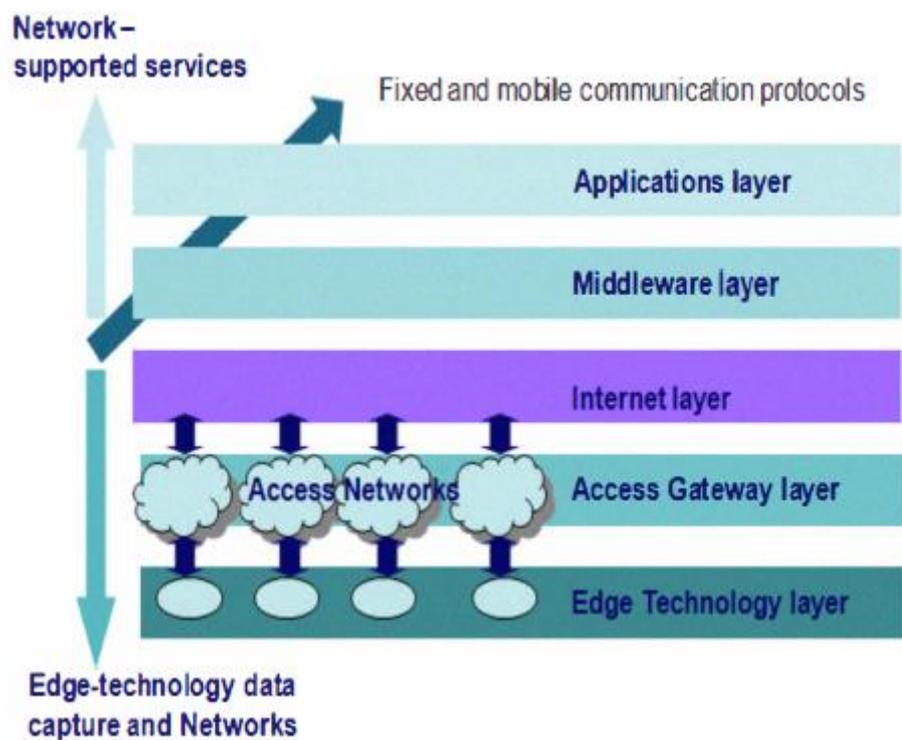
Sebelumnya sudah ada yang meneliti tentang pelanggaran lalu lintas pada marka garis jalan berbasis *Internet Of Things* (IOT). Penelitian sebelumnya dilakukan oleh Adi Abdurakhim di Universitas Komputer Indonesia (UNIKOM). Penelitian itu dilakukan di Dinas Perhubungan atau DISHUB kota Bandung. Penelitian itu menggunakan metodologi pembangunan perangkat lunak *prototype*. Karena metode pembangunan perangkat lunak berupa *prototype* maka alat-alat atau perangkat kerasnya pun berupa *prototype*, sehingga butuh perubahan dari alat agar dapat diimplementasikan ke jalan.[3]

2.3 *Internet of Things*

Berdasarkan referensi yang dikutip, *Things* artinya segala, artinya apapun yang terhubung ke internet termasuk dalam definisi internet of things (IoT). Artinya semua barang fisik yang dapat di-monitor dan dikendalikan dari jarak jauh menggunakan internet adalah IoT. Konsep IoT ini akan sangat mendorong perkembangan big data dan penggunaan data center di Indonesia, oleh karena itu

pemerintah Republik Indonesia sudah merencanakan membangun pusat data ter-sentralisasi. Konsep IoT ini sebetulnya cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yakni:

- a. Barang Fisik yang dilengkapi modul IoT,
- b. Perangkat Koneksi ke Internet seperti Modem Wifi atau jaringan internet yang ada di rumah anda.
- c. Cloud Data Center tempat untuk menyimpan aplikasi beserta data base yang telah terhubung dengan aplikasi yang dibangun.



Gambar 2.2 Arsitektur *Internet of Things*

2.3.1 Unsur-unsur Pembentuk IoT

Ada beberapa unsur pembentuk *IoT* yang mendasar termasuk kecerdasan buatan, konektivitas, sensor, keterlibatan aktif serta pemakaian perangkat berukuran kecil. Berikut, kami akan menjelaskan masing-masing unsur pembentuk tersebut dengan singkat[12]:

- a. **Kecerdasan Buatan (*Artificial Intelligence/AI*)** – *IoT* membuat hampir semua mesin yang ada menjadi "*Smart*". Ini berarti *IoT* bisa meningkatkan segala aspek kehidupan kita dengan pengembangan teknologi yang didasarkan pada *AI*. Jadi, pengembangan teknologi yang ada dilakukan dengan pengumpulan data, algoritma kecerdasan buatan, dan jaringan yang tersedia. Contohnya bisa jadi mesin yang tergolong sederhana semacam meningkatkan/mengembangkan lemari es/kulkas Anda sehingga bisa mendeteksi jika stok susu dan sereal favorit Anda sudah hampir habis, bahkan bisa juga membuat pesanan ke supermarket secara otomatis jika stok mau habis. Penerapan kecerdasan buatan ini memang sangatlah menarik.
- b. **Konektivitas** – Dalam *IoT*, ada kemungkinan untuk membuat/membuka jaringan baru, dan jaringan khusus *IoT*. Jadi, jaringan ini tak lagi terikat hanya dengan penyedia utamanya saja. Jaringannya tidak harus berskala besar dan mahal, bisa tersedia pada skala yang jauh lebih kecil dan lebih murah. *IoT* bisa menciptakan jaringan kecil tersebut di antara perangkat sistem.
- c. **Sensor** – Sensor ini merupakan pembeda yang membuat *IoT* unik dibanding mesin canggih lainnya. Sensor ini mampu mendefinisikan instrumen, yang mengubah *IoT* dari jaringan standar dan cenderung pasif dalam perangkat, hingga menjadi suatu sistem aktif yang sanggup diintegrasikan ke dunia nyata sehari-hari kita.
- d. **Keterlibatan Aktif (*Active Engagement*)** – *Engagement* yang sering diterapkan teknologi umumnya yang termasuk pasif. *IoT* ini mengenalkan paradigma yang baru bagi konten aktif, produk, maupun keterlibatan layanan.
- e. **Perangkat Berukuran Kecil** – Perangkat, seperti yang diperkirakan para pakar teknologi, memang menjadi semakin kecil, makin murah, dan lebih kuat dari masa ke masa. *IoT* memanfaatkan perangkat-perangkat kecil yang dibuat khusus ini agar menghasilkan ketepatan, skalabilitas, dan fleksibilitas yang baik. Media Pembelajaran.

2.4 Kendaraan Bermotor

Kendaraan bermotor adalah sebuah kendaraan gerak yang untuk pergerakannya menggunakan peralatan teknik dan digunakan untuk transportasi. Kendaraan bermotor biasanya menggunakan mesin pembakar dalam untuk pergerakannya, tetapi tidak hanya mesin pembakar dalam saja. Kendaraan bermotor ada juga yang bermesin listrik dan mesin-mesin lainnya.[4]

2.5 Diagram Konteks

Diagram konteks adalah level pertama dari DFD yang menampilkan gambaran tentang sebuah sistem informasi. Diagram konteks hanya menampilkan aliran data dari entitas luar ke sistem atau dari sistem ke entitas luar. Entitas di luar disini bisa orang, tempat, benda yang mengirim data atau menerima data dari sistem [5]. Simbol yang digunakan sama dengan yang ada di DFD kecuali tanpa simbol *data store*.

2.6 Data Flow Diagram

Data Flow Diagram (DFD) menampilkan gambaran mengenai sistem. DFD memperlihatkan komponen-komponen sistem. Juga menampilkan aliran data antara komponen dengan *sources*, *destinations* dan *data storage* [12]. Ada dua jenis DFD, yaitu *Pyshical Data Flow Diagram* dan *Logical Data Flow Diagram*. Dimana *Pyshical* menampilkan entitas internal sistem dan entitas luar serta aliran data di antara mereka. Pada *Pyshical*, entitas internal didefinisikan sebagai entitas baik orang, tempat atau benda di dalam sistem yang melakukan operasi terhadap data. Sehingga *Pyshical* memperlihatkan bagaimana atau dimana atau oleh siapa sebuah proses di dalam sistem dikerjakan. Kemudian jenis lain DFD yaitu *Logical* menampilkan proses yang ada di sistem dan aliran data antara proses. *Logical* menekankan pada kebutuhan fungsional yang akan dilakukan oleh sistem. Contoh DFD jenis ini bisa dilihat pada Gambar berikut [5].

2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah model teknik pendekatan yang menyatakan atau mengGambarkan hubungan suatu model. Didalam hubungan ini

tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada Entity berikutnya. Proses memungkinkan analisis menghasilkan struktur basis data dapat disimpan dan diambil secara efisien [5]. Pada penelitian ini konsep ERD digunakan untuk merancang database yang akan digunakan dalam sistem. Simbol-simbol dalam *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

- a. Entitas, suatu yang nyata atau bastrak yang mempunyai karakteristik dimana kita akan menyimpan data.
- b. Atribut, ciri umum semua atau sebagian besar instansi pada entitas tertentu.
- c. Relasi, hubungan alamiah yang terjadi antara satu atau lebih entitas.
- d. Link, garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Hubungan kardinalitas relasi dalam *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

- a. Satu ke satu (*One to One*)
Setiap elemen dari Entitas A berhubungan paling banyak dengan elemen pada Entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada Entitas A.
- b. Satu ke banyak (*One to Many*)
Setiap elemen dari Entitas A berhubungan dengan maksimal banyak elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan paling banyak satu elemen di Entitas A.
- c. Banyak ke satu (*Many to One*)
Setiap elemen dari Entitas A berhubungan paling banyak dengan satu elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan maksimal banyak elemen di entitas A.
- d. Banyak ke banyak (*Many to Many*)
Setiap elemen dari Entitas A berhubungan maksimal banyak elemen pada Entitas B demikian sebaliknya.

2.8 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis yang berlisensi General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan pemakaian GPL. Pada penelitian ini MySQL digunakan sebagai platform penyimpanan data dan implementasi model rancangan database yang sudah di buat. Gambar 2.8 menunjukkan logo *MySQL*. [6]



Gambar 2.3 Logo MySQL

MySQL memiliki banyak fitur. fitur dimiliki MySQL sebagai berikut :

- a. *Relational Database System*, Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
- b. *Arsitektur Client-Server*, *MySQL* memiliki arsitektur client-server dimana *server database MySQL* terinstal di *server*. Client *MySQL* dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan *internet*.
- c. Mengenal perintah *SQL* standar, *SQL (Structured Query Language)* merupakan suatu bahasa standar yang berlaku di hampir semua software database. *MySQL* mendukung *SQL* versi *SQL:2003*.
- d. *Performace Tuning*, *MySQL* mempunyai kecepatan yang cukup baik dalam menangani *query-query* sederhana, serta mampu memproses lebih banyak *SQL* per satuan waktu.
- e. *Sub Select*.
- f. *Views*.
- g. *Stored Prosedured (SP)*.

- h. *Triggers*.
- i. *Replication*.
- j. *Foreign Key*.
- k. *Security* yang baik.

2.9 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 saat ini dikembangkan dan dikelola oleh tim relawan yang besar dan tersedia secara gratis dari *Python Software Foundation*. *Python* termasuk dari jajaran bahasa pemrograman tingkat tinggi seperti bahasa pemrograman C, C++, Java, Perl dan Pascal, dikenal juga bahasa pemrograman tingkat rendah, yang dikenal sebagai bahasa mesin yaitu bahasa pemrograman *assembly*, Bahasa pemrograman tingkat rendah hanya dapat berjalan di satu jenis arsitektur komputer dan harus ditulis ulang untuk menjalankannya di lain mesin, hal ini diakrenakan karena perbedaan urutan register dan services – servicesnya. Pada penelitian ini bahasa pemrograman python digunakan sebagai komponen utama untuk pembangunan website dan konfigurasi *raspberry pi*. Gambar 2.4 menunjukkan logo *python*. [2]



Gambar 2.4 Logo Python

Python digunakan di berbagai bidang pengembangan. Berikut beberapa implementasi bahasa python yang paling populer:

1. Website

Bahasa pemrograman python dapat digunakan sebagai server side yang diintegrasikan dengan berbagai internet *protocol* misalnya JSON, *Email*

Processing, FTP, dan IMAP. Selain itu python juga mempunyai *library* pendukung untuk pengembangan website seperti *Django*, *Flask*, *Pyramid* dan *Bottle*.

2. Penelitian

Python dapat digunakan juga untuk melakukan riset ilmiah untuk mempermudah perhitungan numerik. Misalnya penerapan algoritma KNN, *Naïve Bayes* dan lain-lain. Beberapa *library* yang sering digunakan untuk riset seperti *Pandas*, *Numpy*, *Mathplotlib* dan lain-lain.

3. Media Pembelajaran

Python dapat digunakan sebagai media pembelajaran di universitas. Python sangat mudah dan hemat untuk dipelajari sebagai *Object Oriented Programming* dibandingkan bahasa lainnya seperti *MATLAB*, *C++*, dan *C#*.

4. Graphical User Interface (GUI)

Python dapat digunakan untuk membangun interface sebuah aplikasi. Tersedia *library* untuk membuat GUI menggunakan python, misalnya *Qt*, *win32extension*, dan *GTK+*.

5. Internet Of Things (IoT)

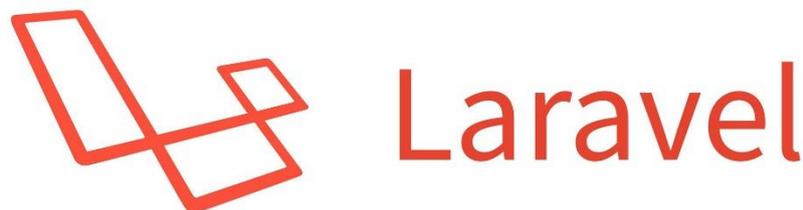
Bahasa pemrograman *Python* mendukung ekosistem *Internet of Things* (IOT) dengan sangat baik. *Internet Of Things* (IOT) merupakan sebuah teknologi yang menghubungkan benda-benda di sekitar kita ke dalam sebuah jaring-jaring yang saling terhubung.

2.10 Laravel

Laravel pertama kali dirilis pada 9 Juni 2011. Menurut Laravel “s pencipta Taylor Otwell, alasan untuk penciptaan kerangka adalah kurangnya beberapa fungsi penting, seperti otentikasi pengguna dalam kerangka *CodeIgniter* [7]. Laravel adalah kerangka *open source*. Ini memiliki seperangkat fitur yang sangat kaya yang akan meningkatkan kecepatan pengembangan web. Jika Anda terbiasa dengan *PHP* inti dan *Advanced PHP*, Laravel akan membuat tugas Anda lebih mudah.

Laravel dibangun di atas bahu raksasa. Menggunakan komponen dari waktu lain diuji kerangka seperti *symfony*. Jika mengisi adalah sesuatu untuk pergi dengan,

Laravel MVC memiliki saham Lion “ s di pangsa pasar kerangka kerja PHP MVC. Menurut sebuah survei yang dilakukan oleh *SitePoint*, *Laravel MVC Framework* adalah Framework yang paling banyak digunakan di kedua proyek pribadi dan di tempat kerja. Karena itu berhasil melakukan semua tugas penting mulai dari web melayani dan manajemen *database* hak untuk generasi HTML, Laravel disebut kerangka tumpukan penuh.[8].



Gambar 2.5 Logo Laravel [8]

Fitur LARAVEL :

1. Mendukung pengembangan aplikasi cepat (RAD)
2. Mendukung *Blade template*: ini adalah *built in template* mesin dibangun di fitur untuk tugas aplikasi web umum seperti otentikasi, routing, mengelola database, mengirim email kelas otomatis *Loading*.
3. Mendukung pengendali RESTful yang memungkinkan Anda untuk mengambil keuntungan dari kata kerja HTTP seperti *GET*, *POST*, *PUT*, dan *DELETE* dll.
4. Mendukung *unit testing* yang dibangun pada fitur pengujian unit yang dapat dijalankan dari baris perintah Artisan.

2.11 *Java Script Object Notation (JSON)*

Format data *JavaScript Object Notation (JSON)* memungkinkan aplikasi untuk berkomunikasi melalui jaringan, biasanya melalui RESTful APIs. JSON adalah teknologi-agnostik, nonproprietary, dan porTabel. Semua bahasa modern

(mis., Jawa, JavaScript, Ruby, C #, PHP, Python, dan Groovy) dan platform memberikan dukungan yang sangat baik untuk memproduksi (membuat serial) dan mengonsumsi data JSON (deserializing). JSON sederhana: terdiri dari konstruksi ramah-pengembang seperti *Objects*, *Array*, dan pasangan nama / nilai. JSON tidak terbatas pada Representational State Transfer (REST).

RESTful Web Services sebagai tidak standar, tetapi (seperti HTTP) JSON sebenarnya merupakan standar. Baik *Internet Engineering Task Force* (IETF) dan *Ecma International* (sebelumnya *European Computer Manufacturers Association*, atau ECMA) telah mengakui JSON sebagai standar. Douglas Crockford awalnya menciptakan JSON pada tahun 2001, dan awalnya membakukannya pada tahun 2006 di bawah RFC 4627 melalui IETF; lihat spesifikasi JSON. Pada musim gugur 2013, Ecma International juga menstandarisasi JSON di bawah ECMA 404; lihat spesifikasi JSON mereka. Dengan pengakuan Ecma, JSON sekarang dianggap sebagai standar pemrosesan data internasional formal [8]. Pada penelitian ini JSON digunakan sebagai API untuk mengirim data sensor dari *raspberry pi* ke *website*.



Gambar 2.6 Logo JSON [9]

2.12 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B merupakan generasi ketiga dari keluarga *raspberry pi*. *raspberry pi 3* memiliki RAM 1GB dan grafis *Broadcom VideoCore IV* pada frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3* juga memiliki 4 *USB port*, 40 *pin GPIO*, *Full HDMI port*, *Port Ethernet*, *Combined 3.5mm audio jack and composite video*, *Camera interface* (CSI), *Display interface* (DSI), slot kartu *Micro SD* (Sistem tekan-tarik, berbeda dari yang sebelum nya ditekan-tekan), dan *VideoCore IV 3D graphics core* . Pada

penelitian ini raspberry pi digunakan sebagai mini pc pengontrolan pengambilan gambar serta tempat hasil sensor disimpan sebelum dikirimkan ke *website*. Gambar 2.7 adalah bentuk *raspberry pi 3 model b*. [10]



Gambar 2.7 Raspberry Pi 3 Model B [10]

2.12.1 GPIO Raspberry Pi 3 Model B

Raspberry pi 3 model B terdiri dari 40 buah pin. Pin pada raspberry pi 3 model B terdiri dari beberapa bagian yaitu bagian VCC, GND, dan GPIO (General Purpose Input/Output) terdapat 3 pin VCC dan 8 pin GND [10]. Pin GPIO mulai dari GPIO 2 hingga GPIO 27, pada pin GPIO terdapat fungsi lain yang dapat dilihat pada Gambar 2.8.

Raspberry Pi 3 GPIO Header					
Pin#	NAME		NAME	Pin#	
01	3.3v DC Power	⬇	DC Power 5v	02	
03	GPIO02 (SDA1 , I ² C)	⬇	DC Power 5v	04	
05	GPIO03 (SCL1 , I ² C)	⬇	Ground	06	
07	GPIO04 (GPIO_GCLK)	⬇	(TXD0) GPIO14	08	
09	Ground	⬇	(RXD0) GPIO15	10	
11	GPIO17 (GPIO_GEN0)	⬇	(GPIO_GEN1) GPIO18	12	
13	GPIO27 (GPIO_GEN2)	⬇	Ground	14	
15	GPIO22 (GPIO_GEN3)	⬇	(GPIO_GEN4) GPIO23	16	
17	3.3v DC Power	⬇	(GPIO_GEN5) GPIO24	18	
19	GPIO10 (SPI_MOSI)	⬇	Ground	20	
21	GPIO09 (SPI_MISO)	⬇	(GPIO_GEN6) GPIO25	22	
23	GPIO11 (SPI_CLK)	⬇	(SPI_CE0_N) GPIO08	24	
25	Ground	⬇	(SPI_CE1_N) GPIO07	26	
27	ID_SD (I ² C ID EEPROM)	⬇	(I ² C ID EEPROM) ID_SC	28	
29	GPIO05	⬇	Ground	30	
31	GPIO06	⬇	GPIO12	32	
33	GPIO13	⬇	Ground	34	
35	GPIO19	⬇	GPIO16	36	
37	GPIO26	⬇	GPIO20	38	
39	Ground	⬇	GPIO21	40	

Rev. 2
29/02/2016
www.element14.com/RaspberryPi

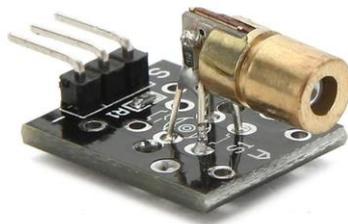
Gambar 2. 1 Pin GPIO Raspberry Pi 3 Model B [10]

2.13 Sensor

Sensor adalah suatu perangkat yang mendeteksi perubahan energi yang berada di alam seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Di dalam sebuah sensor terdapat transduser yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik [10].

2.13.1 Sensor Laser KY-008 dan *Laser Detector*

Kata Laser merupakan akronim dari “*Light Amplification by Stimulated Emission of Radiations*”, yang berarti penguatan cahaya dalam radiasi gelombang elektromagnetik oleh emisi radiasi terangsang[10]. Biasanya dalam bentuk cahaya yang tidak dapat dilihat secara langsung oleh mata melalui proses pancaran terstimulasi. Laser merupakan pembentukan dari cahaya dengan unsur – unsur tertentu baik yang dapat dilihat secara kasat mata maupun tidak. Perambatan cahaya di ruang bebas dilakukan oleh gelombang elektromagnetik. berikut adalah gambar Sensor *Laser KY-008* dan *Laser Detector* pada Gambar 2.9 dan 2.10.



Gambar 2. 2 Sensor *Laser KY-008*



Gambar 2. 3 Sensor *Laser Detector*

2.13.2 Webcam USB

Sebuah webcam adalah kamera video yang *feed* atau *Stream* gambarnya secara real time ke atau melalui komputer ke jaringan komputer. Ketika "ditangkap" oleh komputer, aliran video dapat disimpan, dilihat atau dikirim ke jaringan lain melalui sistem seperti internet, dan diemail sebagai lampiran. Ketika dikirim ke lokasi terpencil, aliran video mungkin akan disimpan, dilihat atau dikirim ke sana. Tidak seperti kamera IP, ini terhubung menggunakan Ethernet atau Wi-Fi. Sebuah webcam umumnya dihubungkan dengan kabel USB, atau kabel serupa, atau dibangun ke dalam perangkat keras komputer, seperti laptop. Kamera web terhubung ke web terus-menerus untuk waktu yang tidak terbatas dan umumnya persediaan pandangan bagi siapa saja yang mengunjungi halaman web melalui internet. Beberapa dari mereka, misalnya seperti lalu lintas online kamera dll. Dalam implementasi ini memainkan peran utama daripada digunakan di ATM sekarang hari..[11]



Gambar 2. 4 Webcam USB

2.14 Open ALPR

Komputerisasi pelat pengakuan (ALPR) adalah gadget laptop yang robot mengenali kode registrasi dari foto digital yaitu. Dari foto yang diambil dari mobil. Gadget ini melibatkan beragam strategi bersama dengan menangkap gambar, lokalisasi nomor *Plate*, segmentasi karakter dan *Optical Character Recognition* (OCR) dari karakter alfanumerik. Tujuan utama dari perangkat ini adalah untuk tata letak dan memperluas strategi pengolahan gambar yang efisien dan algoritma untuk lokalisasi pelat di gambar yang diambil, segmentasi karakter dari kode registrasi

dan reputasi masing-masing karakter dari Bagian tersegmentasi menggunakan *Open laptop Vision Library*. Perangkat ini telah diterapkan dalam komputer papan disebut sebagai Beagle Bone Black dengan bantuan bahasa pemrograman OpenCV dan Python. Banyak program dapat memajukan penggunaan perangkat ini yang mencakup perlindungan, Deteksi kecepatan jalan tol, pelanggaran ringan, identitas teks tulisan tangan, mendeteksi mobil curian, dan sistem seri tol otomatis[2]. Rumus *Open Alpr* dapat dilihat pada Gambar 2.12.



Gambar 2. 12 Logo *Open ALPR*

2.15 Metode Pengujian

Pengujian perangkat lunak merupakan proses eksekusi program atau perangkat lunak dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi atribut dan kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Ada berbagai macam metode pengujian, teknik black box dan teknik white box merupakan metode pengujian yang telah dikenal dan banyak digunakan oleh pengembang perangkat lunak.

2.15.1 *Black Box Testing*

Black box testing merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut. Struktur program dan kode-kode yang ada di dalamnya tidak termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah mudah dan sederhana. Namun, pengujian dengan metode ini tidak dapat mendeteksi kekurangefektifan pengkodean dalam suatu program. Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut. Struktur program dan kode-kode yang ada di dalamnya tidak termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah mudah dan sederhana. Namun, pengujian dengan metode ini tidak dapat mendeteksi kekurangefektifan pengkodean dalam suatu program. Ciri-ciri *black box testing* adalah sebagai berikut:

1. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. Merupakan pendekatan pelengkap dalam mencangkup error dengan kelas yang berbeda dari metode white box testing.
3. Melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai *behavioural testing*, *specification-based testing*, *input/output testing* atau *functional testing*.
4. Terdapat jenis test yang dapat dipilih berdasarkan pada tipe testing yang digunakan.
5. Kategori error yang akan diketahuai melalui *black box testing* seperti fungsi yang hilang atau tidak benar, *error* dari antar-muka, *error* dari struktur data atau akses *eksternal database*, *error* dari kinerja dan *error* dari inisialisasi.

Equivalence class partitioning adalah sebuah metode *black box* terarah yang meningkatkan efisiensi dari pengujian dan meningkatkan *coverage* dari

error yang potensial. Sebuah *equivalence class* adalah sebuah kumpulan dari nilai *variable input* yang memproduksi *output* yang sama. Selanjutnya *output correctness test* merupakan pengujian yang memakan sumber daya paling besar dari pengujian. Pada kasus yang sering terjadi dimana hanya *output correctness test* yang dilakukan, maka sumber daya pengujian akan digunakan semua. Implementasi dari kelas-kelas pengujian lain tergantung dari sifat produk *software* dan pengguna selanjutnya dan juga prosedur dan keputusan pengembang. *Output correctness test* mengaplikasikan konsep dari *test case*. Pemilihan *test case* yang baik dapat dicapai dengan efisiensi dari penggunaan *equivalence class partitioning*. Jenis-jenis *test case* sebagai berikut:

1. *Availability test*

Availability didefinisikan sebagai waktu reaksi yaitu waktu yang dibutuhkan untuk mendapatkan informasi yang diminta atau waktu yang dibutuhkan oleh *firmware* yang diinstal pada perlengkapan komputer untuk bereaksi. *Availability* adalah yang paling penting dalam aplikasi online sistem informasi yang sering digunakan. Kegagalan *firmware software* untuk memenuhi persyaratan ketersediaan dapat membuat perlengkapan tersebut tidak berguna.

2. *Realiability test*

Realiability berkaitan dengan fitur yang dapat diterjemahkan sebagai kegiatan yang terjadi sepanjang waktu seperti waktu rata-rata antara kegagalan (misalnya 500 jam), waktu rata-rata untuk *recovery* setelah kegagalan sistem (misalnya 15 menit) atau *average downtime* per bulan (misalnya 30 menit per bulan). Persyaratan realibilitas memiliki efek selama *regular full-capacity* operasi sistem. Harus diperhatikan bahwa penambahan faktor *software realiability* juga berkaitan dengan perangkat, sistem operasi, dan efek dari sistem komunikasi data.

3. *Stress test*

Stress test terdiri dari 2 tipe pengujian yaitu *load test* dan *durability test*. Suatu hal yang mungkin untuk melakuakn pengujian-pengujian tersebut setelah penyelesaian sistem *software*. *Durability test* dapat dilakukan hanya

setelah firmware atau sistem informasi *software* diinstall dan siap untuk diuji. Pada *load test* berkaitan dengan *functional performance system* dibawah beban maksimal operasional, yaitu maksimal transaksi per menit, hits per menit ke tempat internet dan sebagainya. *Load test*, yang biasanya dilakukan untuk beban yang lebih tinggi dari yang diindikasikan spesifikasi persyaratan merupakan hal yang penting untuk sistem *software* yang rencananya akan dilayani secara simultan oleh sejumlah pengguna. Pada sebagian besar kerja sistem software, beban maksimal menggambarkan gabungan beberapa tipe transaksi. Selanjutnya *durability test* dilakukan pada kondisi operasi fisik yang ekstrem seperti temperatur yang tinggi, kelembaban, mengendara dengan kecepatan tinggi, sebagai detail persyaratan spesifikasi durabilitas. Jadi, dibutuhkan untuk *real-time firmware* yang diintegrasikan ke dalam sistem seperti sistem senjata, kendaraan transport jarak jauh, *Internet Of Things* (IOT) dan keperluan meterologi. Isu ketahanan pada *firmware* terdiri dari respon *firmware* terhadap efek cuaca seperti temperatur panas atau dingin yang ekstrem, debu, kegagalan operasi ekstrem karena kegagalan listrik secara tiba-tiba, loncatan arus listrik dan putusnya komunikasi secara tiba-tiba.

4. *Training usability test*

Ketika sejumlah besar pengguna terlibat dalam sistem operasi, *training usability requirement* ditambahkan dalam agenda pengujian. Lingkup dari *training usability test* ditentukan oleh sumber yang dibutuhkan untuk melatih pekerja baru untuk memperoleh level pengenalan dengan sistem yang ditentukan atau untuk mencapai tingkat produksi tertentu. Detail dari pengujian ini, sama halnya dengan yang lain, didasarkan pada karakteristik pekerja. Hasil dari pengujian ini harus menginspirasi rencana dari kursus pelatihan dan follow-up serta memperbaiki sistem operasi *software*.

5. *Operational usability test*

Fokus dari pengujian ini adalah produktifitas operator, yang aspeknya terhadap sistem yang mempengaruhi performance dicapai oleh operator sistem. *Operational usability test* dapat dijalankan secara manual.

Revision class partitioning adalah sebuah metode *black box* lainnya yang merupakan faktor dasar yang menentukan keberhasilan paket suatu *software*, pelayanan jangka panjang, dan keberhasilan penjualan ke sejumlah besar populasi pengguna [21]. Berkaitan dengan hal tersebut terdapat tiga kelas pengujian revisi sebagai berikut:

1. *Reusability test*

Reusability menentukan bagian mana dari suatu program (modul, integrasi, dbs) yang akan dikembangkan untuk digunakan kembali pada proyek pengembangan *software* lainnya, baik yang telah direncanakan maupun yang belum. Bagian ini harus dikembangkan, disusun, dan didokumentasikan menurut prosedur perpustakaan *software* yang digunakan ulang.

2. *software interoperability test*

software interoperability berkaitan dengan kemampuan *software* dalam memenuhi perlengkapan dan paket *software* lainnya agar memungkinkan untuk mengoperasikannya bersama dalam satu sistem komputer kompleks.

3. *Equipment interoperability test*

Equipment interoperability berkaitan dengan perlengkapan *firmware* dalam menghadapi untuk perlengkapan lain dan atau paket *software*, dimana persyaratan mencantumkan *specified interfaces*, termasuk dengan *interfacing standard*. Pengujian yang relevan harus menguji implementasi dari *interoperability requirements* dalam sistem.

Keuntungan dan kekurangan dari *black box testing*, beberapa keuntungan dari *black box testing*, diantaranya sebagai berikut:

1. *Black box testing* memungkinkan kita untuk memiliki sebagian besar tingkat pengujian, yang sebagian besarnya dapat diimplementasikan.
2. Untuk tingkat pengujian yang dapat dilakukan baik dengan *white box testing* maupun *black box testing*, *black box testing* memerlukan lebih sedikit sumber dibandingkan dengan yang dibutuhkan oleh *white box testing* pada pake *software* yang sama.

Sedangkan kekurangan dari *black box testing* adalah sebagai berikut:

1. Adanya kemungkinan untuk terjadinya beberapa kesalahan yang tidak disengaja secara bersama-sama akan menimbulkan respon pada pengujian ini dan mencegah deteksi kesalahan (*error*). Dengan kata lain, *black box test* tidak siap untuk mengidentifikasi kesalahan-kesalahan yang berlawanan satu sama lain sehingga menghasilkan output yang benar.
2. Tidak adanya kontrol terhadap *line coverage*. Pada kasus dimana *black box test* diharapkan dapat meningkatkan *line coverage*, tidak ada cara yang mudah untuk menspesifikasikan parameter-parameter pengujian yang dibutuhkan untuk meningkatkan *coverage*. Akibatnya, *black box test* dapat melakukan bagian penting dari baris kode, yang tidak ditangani oleh set pengujian.

Ketidakmungkinan untuk menguji kualitas pembuatan kode dan pendekatannya dengan standar pembuatan kode.

