

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Profil Perusahaan**

Profil perusahaan merupakan uraian mengenai tempat penelitian yang menjadi studi kasus dari tugas akhir ini. Hal yang akan diuraikan di antaranya sejarah, visi dan misi, struktur organisasi dan tugas pokok.

##### **2.1.1 Sejarah Perusahaan**

PT. Tiki Jalur Nugraha Ekakurir atau biasa disebut sebagai JNE merupakan perusahaan jasa di bidang pengiriman ekspres yang berkantor pusat di Jl. Tomang Raya no.11 Jakarta Barat.

Perusahaan ini didirikan oleh Bapak H. Soeprapto Soeparno pada tahun 1990, dengan memulai usaha pada kegiatan kepabeanan atau impor kirim barang atau dokumen serta pengantarannya dari luar negeri ke Indonesia. Pada tahun 1993 JNE mulai membuka jaringan domestik dan pada tahun 1996 mulai membuka sistem keagenan.

##### **2.1.2 Logo**

Berikut ini merupakan logo dari JNE, dapat dilihat pada gambar Gambar 2.1 Logo JNE.



**Gambar 2.1 Logo JNE**

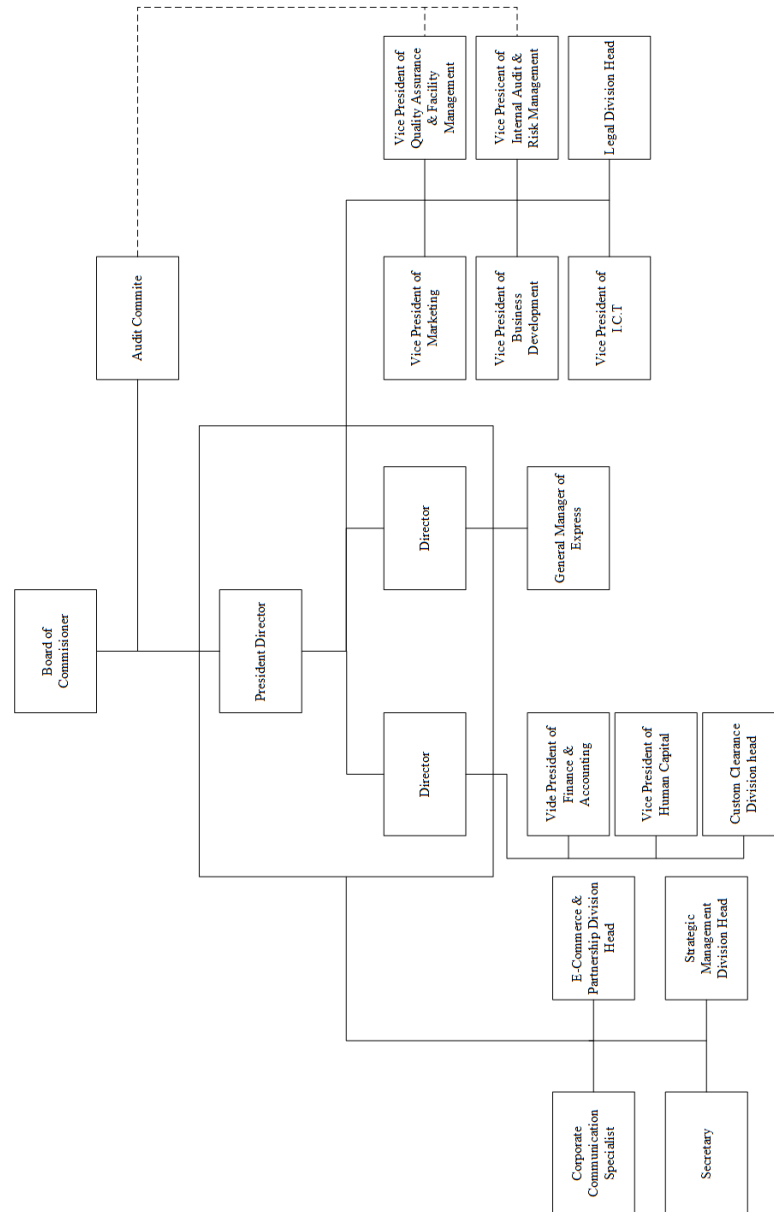
##### **2.1.3 Visi dan Misi**

Visi dan Misi merupakan suatu konsep perencanaan mengenai tujuan serta perencanaan untuk mencapai tujuan tersebut. Visi dari JNE adalah Untuk menjadi

perusahaan rantai pasok global terdepan di dunia. Sedangkan Misi dari JNE adalah Untuk memberi pengalaman terbaik kepada pelanggan secara konsisten.

### 2.1.4 Struktur Organisasi

Berikut merupakan struktur organisasi yang ada di JNE.



**Gambar 2.2 Struktur Organisasi**

## 2.2 Landasan Teori

Landasan teori digunakan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan.

### 2.2.1 Android

Android adalah sebuah *system* operasi untuk perangkat *mobile* berbasis linux yang mencakup *system* operasi, middleware dan aplikasi Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka Android merupakan generasi baru platform *mobile*, platform yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkan. Pengembangan aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apa pun yang mereka inginkan. Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android di antaranya android studio maupun eclipse.[1]

### 2.2.2 Unified Modelling Language

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.[2]

### 2.2.3 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, yang berdasarkan IntelliJ IDEA. IntelliJ selain editor kode merupakan alat pengembang yang hebat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda saat membangun aplikasi Android[3], seperti:

1. Sistem pembangunan berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Dapat mengembangkan aplikasi semua perangkat Android
4. *Template* kode dan integrasi GitHub untuk membantu Anda membangun fitur aplikasi umum dan mengimpor kode sampel
5. Alat pengujian dan *framework* yang luas
6. Dukungan C ++ dan NDK

7. Dukungan bawaan untuk *Google Cloud Platform*, membuatnya mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*

#### 2.2.4 Metode Haversine

Metode Haversine Formula dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang *latitude* dan posisi garis bujur *longitude* sebagai variabel masukan. Haversine Formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2.[4]

$$d = 2r \sin^{-1} \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos \phi_1 \cos \phi_2 \sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right)} \right)$$

Di mana  $d$  adalah jarak antara dua titik *latitude* dan *longitude* ( $\phi, \varphi$ ) dan  $r$  adalah jari-jari dari bumi.

#### 2.2.5 Global Positioning System

*Global Positioning System* (GPS) adalah teknologi pelacak yang diperkenalkan oleh Amerika Serikat of Defense (DoD), untuk penentuan posisi berbasis navigasi, dan sistem pengaturan waktu. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, di mana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS.[5]

#### 2.2.6 WebSocket

WebSocket merupakan teknologi yang mampu memberikan performa terbaik ketika diimplementasikan dalam sistem dengan rate-request tinggi, dibandingkan dengan teknologi komunikasi lain termasuk AJAX[6]. WebSocket

mampu memberikan reduksi kepadatan lalu lintas jaringan (*network traffic*), *delay rate*, dan latensi, khususnya dalam sistem dengan koneksi simultan dalam jumlah besar. Selain itu, teknologi ini adalah teknologi yang secara umum dianggap sebagai pilihan terbaik untuk membangun sistem dengan komunikasi real-time[7].

### 2.2.7 JSON

JSON atau JavaScript Object Notation adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apa pun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [8].

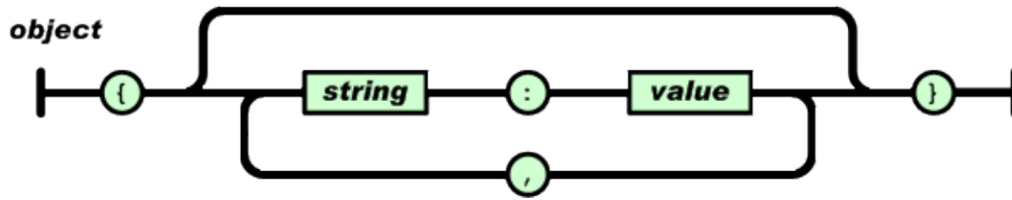
JSON terbuat dari dua struktur, yaitu:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel *hash* (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan ().

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

JSON menggunakan bentuk sebagai berikut:

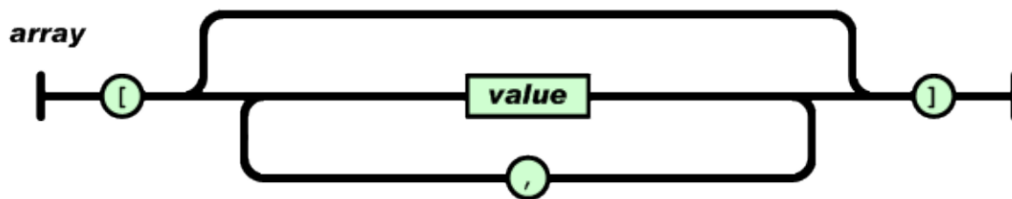
1. Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan “{“ dan diakhiri dengan “}”. Setiap nama diikuti dengan “:” dan setiap pasangan nama/nilai dipisahkan oleh “,”.



Sumber Gambar : <http://www.json.org/json-id.html>

**Gambar 2. 1. Objek JSON**

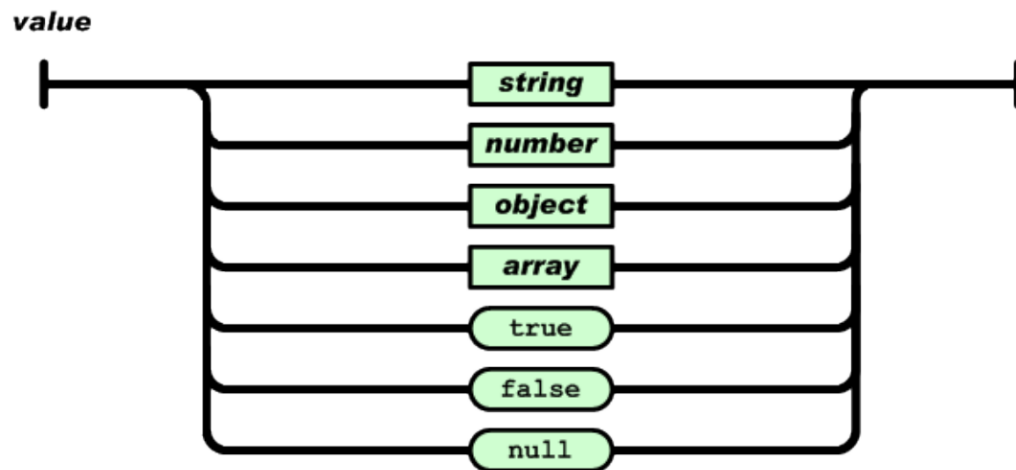
2. Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan “[“ dan diakhiri dengan “]”. Setiap nilai dipisahkan oleh “,”.



Sumber Gambar : <http://www.json.org/json-id.html>

**Gambar 2. 2. Array JSON**

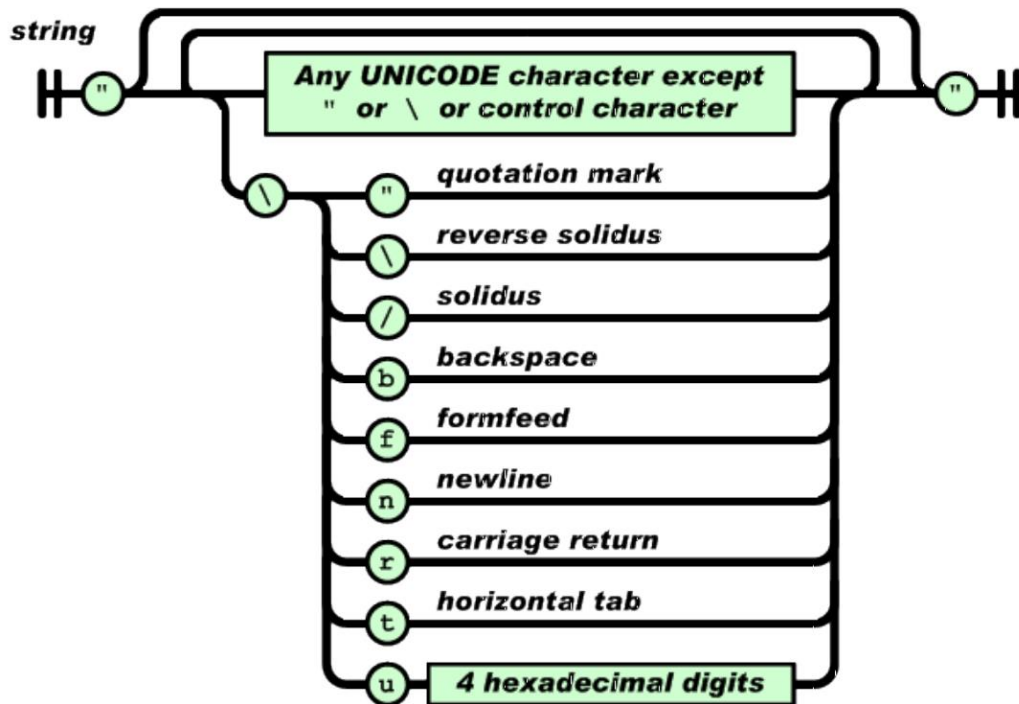
3. Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Sumber Gambar : <http://www.json.org/json-id.html>

**Gambar 2. 3. Value JSON**

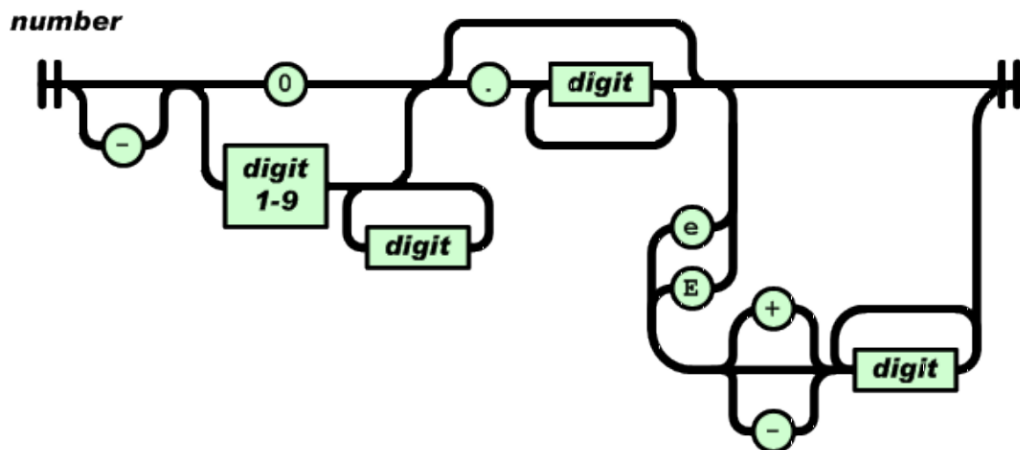
4. *String* adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.



Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2. 4. String JSON

5. Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2. 5. Number JSON



### 2.2.8 JSON WEB TOKEN (JWT)

JSON Web Token (JWT) adalah standar terbuka (RFC 7519) yang mendefinisikan cara yang ringkas dan mandiri untuk mentransmisikan informasi antar pihak secara aman sebagai objek JSON. Informasi ini dapat diverifikasi dan dipercaya karena ditandatangani secara digital. JWT dapat ditandatangani menggunakan rahasia (dengan algoritma HMAC) atau pasangan kunci publik / pribadi menggunakan RSA atau ECDSA [9].

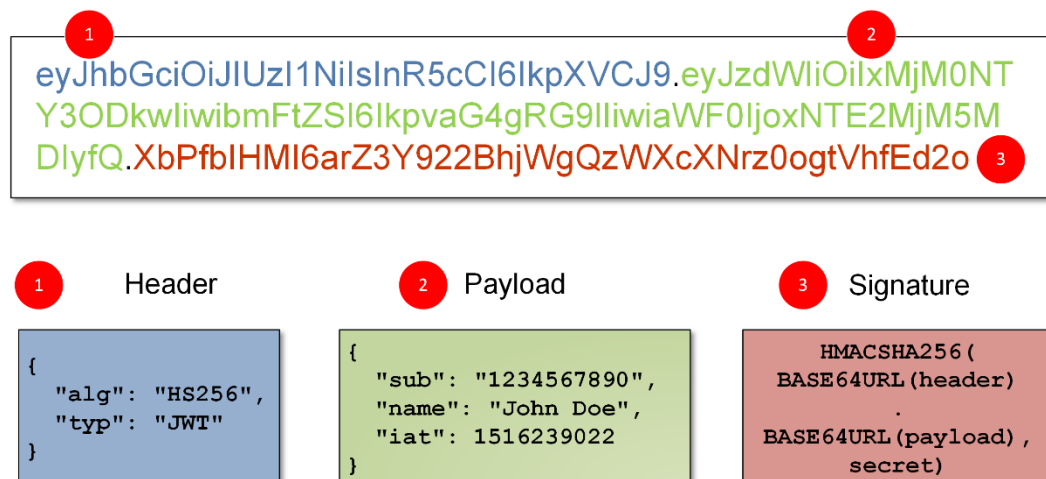
Meskipun JWT dapat dienkrpsi untuk juga memberikan kerahasiaan antar pihak, JWT akan fokus pada token yang ditandatangani. Token yang ditandatangani dapat memverifikasi integritas yang terkandung di dalamnya, sementara token terenkrpsi menyembunyikan klaim tersebut dari pihak lain. Ketika token ditandatangani menggunakan pasangan kunci publik / pribadi, tanda tangan juga menyatakan bahwa hanya pihak yang memegang kunci pribadi yang menandatangani [9].

#### Struktur JWT

Dalam bentuknya yang ringkas, JSON Web Token terdiri dari tiga bagian yang dipisahkan oleh titik (.), Yaitu:

1. Header
2. Payload
3. Signature

Oleh karena itu, JWT biasanya terlihat seperti pada Gambar 2.3 Struktur JWT



Gambar 2.3 Struktur JWT

### 1. Header

Header biasanya terdiri dari dua bagian: jenis token, yaitu JWT, dan algoritma penandatanganan yang digunakan, seperti HMAC SHA256 atau RSA.

### 2. Payload

Payload biasanya berisi data yang dienkripsi menggunakan base64url

### 3. Signature

Signature terdiri dari hasil enkripsi header, payload dan secret key.

## 2.2.9 Structured Query Language (SQL)

SQL adalah pendekatan dari *Structural Query Language*, yang merupakan Bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan *database*. Perintah dalam SQL diklasifikasikan menjadi lima bagian besar [10], yaitu:

### 1. *Data Definition Language* (DDL)

*Data Definition Language* (DDL) adalah sekumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan dan penghapusan *database* prosedur/fungsi, *trigger*, dan sebagainya.

Perintah SQL yang termasuk kategori DDL adalah:

1. CREATE DATABASE, digunakan untuk membuat *database* dan objek-objek di dalam *database*.
  2. CREATE TABLE, digunakan untuk membuat tabel di dalam *database*.
  3. CREATE INDEX, digunakan untuk membuat *index*.
  4. CREATE VIEW, digunakan untuk membuat *view*.
  5. DROP DATABASE, digunakan untuk menghapus basis data.
  6. DROP TABLE, digunakan untuk menghapus tabel.
  7. DROP INDEX, digunakan untuk menghapus *index*.
  8. DROP VIEW, digunakan untuk menghapus *view*.
  9. ALTER TABLE, digunakan untuk mengubah struktur suatu tabel.
- ### 2. *Data Manipulation Language* (DML)

*Data Manipulation Language* (DML) adalah kumpulan perintah SQL yang berkaitan dengan data atau isi dari suatu tabel. Dengan perintah-perintah di dalam

DML, kita dapat memanipulasi (menambah, mengubah, dan menghapus) data yang terdapat pada suatu tabel secara mudah. Perintah-perintah yang termasuk dalam DML adalah:

- a. *INSERT*, berfungsi untuk menambah atau memasukkan data baru ke dalam tabel.
- b. *UPDATE*, berfungsi untuk mengubah data dalam tabel dengan nilai baru.
- c. *DELETE*, berfungsi untuk menghapus data dari suatu tabel.

### 3. *Data Control Language (DCL)*

*Data Control Language (DCL)* adalah salah satu komponen SQL yang berfungsi untuk mengontrol hak akses *user*. Perintah yang termasuk ke dalam DCL adalah GRANT dan REVOKE. Berikut penjelasannya.

- a. GRANT, Perintah GRANT digunakan untuk memberikan hak akses (privilege) kepada *user* tertentu. Melalui perintah hak akses semacam ini, seseorang *user* memiliki keterbatasan dalam menggunakan database sehingga data akan aman dari pihak-pihak yang tidak berkepentingan. Untuk melihat daftar hak akses yang dimiliki oleh seorang *user*, gunakan perintah SHOW GRANTS. Untuk mengeksekusi perintah GRANT, anda perlu memiliki hak akses GRANT OPTION yang sebelumnya diberikan oleh administrator kepada anda.
- b. REVOKE, Perintah REVOKE merupakan kebalikan dari perintah GRANT, yang berfungsi untuk mencabut salah satu atau beberapa hak akses dari *user* tertentu di dalam database. Sama seperti perintah GRANT, untuk menjalankan perintah REVOKE anda perlu memiliki hak akses GRANT OPTION.

### 4. *Transaction control Language (TCL)*

Selain pengolahan hak akses *user*, dalam SQL kita juga dapat mengontrol transaksi data yang telah dilakukan. Perintah-perintah yang berkaitan dengan pengontrolan transaksi data digolongkan ke dalam *Transactional Control Language (TCL)*. Perintah yang termasuk ke dalam TCL adalah COMMIT dan ROLLBACK. Berikut penjelasannya.

- a. COMMIT, perintah COMMIT berfungsi untuk menyimpan perubahan-perubahan yang dilakukan terhadap *database* (melalui perintah INSERT, UPDATE, atau DELETE) secara permanent.
- b. ROLLBACK, perintah ROLLBACK merupakan kebalikan dari perintah COMMIT, yang berfungsi untuk membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam *database* (melalui perintah INSERT, UPDATE, atau DELETE). Dengan melakukan pembatalan transaksi, data di dalam *database* akan kembali ke keadaan awal (keadaan sebelum dilakukan perubahan). Sehingga dengan kata lain, perubahan yang dilakukan tidak akan pernah disimpan ke dalam *database*.

#### **2.2.10 Firebase Cloud Messaging (FCM)**

*Firestore Cloud Messaging* adalah layanan untuk mengirimkan *push* notifikasi kepada perangkat yang dituju oleh pengirim dengan mengisikan *device id* dari perangkat penerima.

Setiap perangkat yang memiliki aplikasi menggunakan FCM, akan mempunyai *registration\_id* dari setiap perangkat yang bersifat unik. *Registration ID* atau biasa disebut dengan *token*. Pengembang aplikasi dapat menggunakan API dengan protokol HTTP untuk mengirim notifikasi (dalam format JSON) ke perangkat pengguna dengan *token* mereka[11].

#### **2.2.11 Personal Home Page (PHP)**

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor*, dan menampilkan jumlah pengunjung dari suatu *website*. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia *web* saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page (PHP)* [12].

Kebutuhan PHP sebagai *tool* yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengonversikan data yang di masukkan melalui *Form* HTML menjadi suatu variabel, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasikannya, akhirnya Lerdorf mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan Perl. Tahun 1997, PHP versi 2.0 di rilis, dengan nama *Personal Home Page Form Interpreter* (PHP-FI). PHP Semakin populer, dan semakin diminati oleh *programmer* web dunia.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia *web development*. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat di luar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutsman selaku *core developer* (*programmer* inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan Zend *scripting engine*, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4.

13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan di segala sisi, dan wajar jika *netcraft* mengumumkan PHP sebagai bahasa web populer di dunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai *server side* scriptingnya. PHP saat ini telah Mendukung XML dan Web Services, Mendukung SQLite. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai *server* scriptingnya. Benar-benar PHP sangat mengejutkan.

Yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka filenya terlebih dahulu, kemudian menambahkan isi ke dalam *file* tersebut. Beda halnya dengan PHP. Bagi anda yang pernah menggunakan CMS seperti *wordpress* atau *joomla* yang dibangun dengan PHP tentunya, ketika akan menambahkan konten kedalam website, anda tinggal masuk ke dalam halaman admin, kemudian

pilih *new artikel* untuk membuat halaman/*content* baru. Artinya hal ini, seorang *user* tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu. Keunggulan PHP antara lain:

a. Gratis

Apa yang membuat PHP begitu berkembang sangat pesat hingga jutaan domain menggunakan PHP, begitu populernya PHP? Jawabannya adalah karena PHP itu gratis.

b. *Cross Platform*

Artinya dapat di gunakan di berbagai sistem operasi, mulai dari linux, windows, mac os dan os yang lain.

c. Mendukung banyak *database*

PHP telah mendukung banyak *database*, ini mengapa banyak *developer web* menggunakan PHP Adabas D Adabas D, dBase dBase, Empress Empress, FilePro (*read-only*) FilePro (*read-only*) Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase mSQL, Direct MS-SQL, MySQL MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL SQLite, Solid, Sybase, Velocis, Unix dbm.

d. On The Fly

PHP sudah mendukung *on the fly*, artinya dengan php anda dapat membuat *document text*, Word, Excel, PDF, menciptakan *image* dan *flash*, juga menciptakan file-file seperti zip, XML, dan banyak lagi.

### 2.2.12 CodeIgniter

CodeIgniter merupakan aplikasi open *source* berupa *framework* PHP dengan model MVC (Model, View, Controller) untuk membangun aplikasi web dinamis dengan cepat dan mudah. CodeIgniter memiliki desain dan struktur *file* yang sederhana, didukung dengan dokumentasi yang lengkap sehingga *framework* ini lebih mudah dipelajari. CodeIgniter ini memungkinkan para pengembang untuk menggunakan framework secara parsial atau secara keseluruhan.

Artinya bahwa CodeIgniter masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi

menggunakan cara konvensional atau dengan *syntax* umum di dalam PHP, tidak harus menggunakan aturan penulisan kode di CodeIgniter [13].

### 2.2.13 Model View Presenter (MVP)

Model-View-Presenter merupakan turunan dari pola MVC. Menurut Mike Potel (1996) melalui Lappalainen Sami dan Takashi Kobayashi (2017) tujuan dibuatnya MVP adalah untuk memperbaiki pemisahan tugas yang ada pada MVC dan memfasilitasi automated unit tests untuk *User* interface. Ada dua variasi utama MVP, yaitu Passive View dan Supervising Controller[13].

Pada MVP Supervising Controller, presenter adalah controller yang sedikit dimodifikasi yang bertanggung jawab untuk memperbarui model dan memberitahukan perubahan dalam model. Tampilan (view) masih mengakses langsung ke model, membuat pengujian *User* interface menjadi sulit. Masalah ini telah dibahas dalam versi *passive view* di mana presenter bertindak sebagai koneksi antara tampilan dan model.

Presenter berinteraksi dengan model ketika mendeteksi masukan pengguna dan memperbarui tampilan melalui antarmuka ketika model berubah. Presenter memiliki akses langsung ke tampilan (view) dan model. Tampilan juga memiliki sinkronisasi *one way* yang mengikat ke model. Presenter bertanggung jawab untuk menangani *view state logic* dan memanggil metode logika bisnis pada model.

### 2.2.14 Model View Controller (MVC)

MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data atau query (Model) dari tampilan atau *user interface* (View) dan cara pemrosesannya (Controller). Pembuatan program yang menggunakan MVC ini biasanya dikemas dalam kerangka kerja (framework), sehingga pengembang aplikasi tinggal menggunakan kerangka kerja yang sudah disediakan [14].

1. Model, merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. Model adalah suatu objek yang tidak mengandung informasi tentang *user interface*.

2. View, merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam *user interface* (semua hal *user* dapat melihat dan merespon pada layar, seperti tombol, tampilan kotak, dan sebagainya).
3. *Controller*, merupakan kelas yang menghubungkan model dan *view*, dan digunakan untuk berkomunikasi antara kelas dalam model dan *view*.

### 2.2.15 Web Service (API)

*Web service* adalah *standar* yang digunakan untuk pertukaran data antar aplikasi atau sistem. Standar diperlukan karena masing-masing aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda. Contoh implementasi dari *web service* antara lain adalah SOAP dan REST [15].

Peneliti menggunakan *web service* yang ditempatkan di sebuah server untuk dapat menjalankan sistem sesuai dengan yang diharapkan sehingga aplikasi dapat mengambil serta menyimpan data yang diperlukan seperti data pengguna, data *monitoring* dan lain sebagainya. Adapun metode yang digunakan untuk format pertukaran data adalah *Javascript Object Notation* (JSON).

### 2.2.16 Representational State Transfer (REST)

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protocol untuk komunikasi *data*. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000 [15].

Pada arsitektur REST, REST *server* menyediakan *resources* (sumber daya/data) dan REST *client* mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

Keuntungan REST antara lain:



1. Bahasa dan *platform agnostic*.
2. Lebih sederhana/simpel untuk dikembangkan daripada SOAP.
3. Mudah dipelajari, tidak bergantung pada *tools*.
4. Ringkas, tidak membutuhkan *layer* pertukaran pesan (*messaging*) tambahan.
5. Secara desain dan filosofi lebih dekat dengan web.

Kelemahan REST antara lain:

1. Mengasumsi model *point-to-point* komunikasi tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara.
2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri").
3. Berkaitan dengan model *transport* HTTP (*Hypertext Transfer Protocol*).

Berikut metode HTTP (*Hypertext Transfer Protocol*) yang umum digunakan dalam arsitektur berbasis REST:

1. *GET*, menyediakan hanya akses baca pada *resource*.
2. *PUT*, digunakan untuk menciptakan *resource* baru.
3. *DELETE*, digunakan untuk menghapus *resource*.
4. *POST*, digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru.
5. *OPTIONS*, digunakan untuk mendapatkan operasi yang disupport pada *resource*.

*Web service* yang berbasis arsitektur REST kemudian dikenal sebagai RESTful *web services*. Layanan web ini menggunakan metode HTTP (*Hypertext Transfer Protocol*) untuk menerapkan konsep arsitektur REST.

Cara Kerja RESTful web services yaitu:

Sebuah *client* mengirimkan sebuah *data* atau *request* melalui HTTP *Request* dan kemudian *server* merespon melalui HTTP *Response*. Komponen dari HTTP *request*:

- a. *Verb*, HTTP *method* yang digunakan misalnya *GET*, *POST*, *DELETE*, *PUT* dll.
- b. *Uniform Resource Identifier* (URI) untuk mengidentifikasi lokasi *resource* pada *server*.
- c. *HTTP Version*, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
- d. *Request Header*, berisi metadata untuk HTTP *Request*. Contoh, *type client/browser*, format yang didukung oleh *client*, format dari *body* pesan, seting *cache* dll.
- e. *Request Body*, konten dari *data*.

Sedangkan komponen dari *http response* yaitu:

1. *Status/Response Code*, mengindikasikan status *server* terhadap *resource* yang *direquest*. misal: 404, artinya *resource* tidak ditemukan dan 200 *response OK*.
  2. *HTTP Version*, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
  3. *Response Header*, berisi *metadata* untuk HTTP *Response*. Contoh, *type server*, panjang *content*, tipe *content*, waktu *response*, dll
- Response Body*, konten dari data yang diberikan.

### 2.2.17 Global Positioning System (GPS)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, di mana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
  - a) Latitude
  - b) Longitude
  - c) Elevasi

3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
5. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.

Kumulasi data. GPS *receiver* dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya [16].

### 2.2.18 Bahasa Pemrograman Java

Java merupakan sebuah Bahasa pemrograman berorientasi objek yang dapat berjalan pada *platform* yang berbeda, baik di Windows, Linux, serta system operasi lainnya. Dengan menggunakan Java kita dapat mengembangkan banyak aplikasi yang dapat digunakan pada lingkungan yang berbeda, seperti pada *Desktop*, *Mobile*, *Internet* dan lain-lain [17].

Untuk menginstalasi dan menggunakan Java, Sun Micro *System* selaku pengembang Java menyediakan paket instalasi sesuai dengan kebutuhan kita dalam membangun suatu aplikasi. Berikut ini uraian singkat mengenai paket aplikasi Java yang tersedia.

#### 1. J2ME (Java 2 *Micro Edition*)

Paket instalasi ini dapat digunakan untuk membangun *software* yang berjalan pada perangkat yang memiliki memori dan sumber daya yang kecil, seperti pada *handphone*, PDA, dan *Smartcard*.

#### 2. J2SE (Java 2 *Standard Edition*)

Paket instalasi ini digunakan untuk mengembangkan aplikasi yang berjalan pada lingkungan *workstation*, seperti aplikasi desktop.

#### 3. J2EE (Java 2 *Enterprise Edition*)

Paket instalasi ini dapat digunakan untuk mengembangkan aplikasi pada lingkungan internet maupun aplikasi skala *enterprise*.

Java juga merupakan bahasa pemrograman resmi yang digunakan untuk pembangunan aplikasi android yang didukung penuh oleh Google. Namun meskipun demikian saat ini java bukanlah satu-satunya Bahasa yang dapat digunakan untuk membangun aplikasi Android seperti Xamarin dengan menggunakan Bahasa pemrograman C#, Cordova dengan menggunakan bahasa pemrograman web seperti HTML, CSS, dan Javascript dan lain-lain.

### **2.2.19 Skala Likert**

Skala *Likert* adalah suatu skala *psikometrik* yang umum digunakan dalam kuesioner. Skala *Likert* merupakan skala yang paling banyak digunakan dalam riset berupa survey seperti mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan Skala *Likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam Skala *Likert*, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti: [18]

- a. Sangat tidak setuju
- b. Tidak setuju
- c. Netral
- d. Setuju
- e. Sangat setuju

Penulis kuesioner harus memutuskan apakah memasukkan titik tengah atau tidak sesuai dengan pernyataan yang diberikan kepada responden. Meskipun penggunaan respon kategori tengah tidak mempengaruhi reliabilitas dan validitas dalam penelitian ini, namun direkomendasikan bahwa penilaian. Ahli lain bahwa menyediakan kategori tengah memungkinkan responden untuk menunjukkan

respon yang netral dan lebih diskriminatif dalam respon mereka, membuat nilai skala yang lebih handal dan skala yang lebih disukai oleh responden.

Berikut ini rumus skala *likert* yang digunakan untuk mencari nilai persentase dari jawaban kuesioner :

$$\text{Rumus : } P = \frac{S}{\text{Skor Ideal}} \times 100\%$$

Keterangan :

P : Nilai persentase yang dicari

S : Jumlah frekuensi dikalikan dengan bobot yang ditetapkan

Skor Ideal : Nilai tertinggi dikalikan dengan jumlah sampel

