BAB 2
LANDASAN TEORI

2.1. Profil Umum Rumah Makan Sehat dan Organik Getol

Rumah makan getol berdiri sejak tanggal 2 april 2013 dengan diprakarsai oleh Ibu Merita. Nama rumah makan getol diambil dari lokasi rumah makan yang mana terletak tepat pada ruko didepan gerbang tol Buah batu, oleh karena itu diberikan nama rumah makan getol (Gerbang Tol). Pemilik melihat bahwa pintu tol Buah batu sebagai kawasan potensial untuk membuka usaha kuliner. Selain semakin berkembangnya pembangunan, kawasan tersebut juga banyak dilewati pendatang dari luar kota Bandung yang keluar melalui pintu tol Buah batu. Untuk itu didirikan sebuah rumah makan organik yang bernama rumah makan getol. Rumah makan getol berkomitmen untuk terus memberikan produk terbaik yang sesuai dengan standar mutu kualitas terbaik.

2.1.1. Visi dan Misi Rumah Makas Sehat dan Organik Getol

2.1.1.1. Visi

Menjadi bisnis kuliner organik yang terdepan dalam mutu dan pelayanan di seluruh Indonesia.

2.1.1.2. Misi

1. Menyediakan kualitas unggul, pelayanan prima, dan pilihan terbaik kepada pelanggan kami.
2. Membuat standar operasi prosedur yang baik dalam pengolahan produk untuk mengembangkan kepercayaan.

2.2. Landasan Teori

Landasan Teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan Aplikasi Untuk Rekomendasi Pola Makan dan Tempat Makan Sehat Memanfaatkan API *Clarifai* dan Teknologi GPS Pada *Platform Andorid*, meliputi Aplikasi, Pengertian Aplikasi, Klasifikasi Aplikasi, Pengertian *Mobile*, Pengertian Aplikasi *Mobile*, *Andorid*, Jaringan *Internet*, API *Clarifai*, GPS, LBS, *Google Maps*, React Native, HTML5, PHP7, Javascript,

JSON, XAMPP, *Database* MySQL, *Object Oriented Analysis Desain*, *Structured Analysis and Design*, dan *Web Service*.

2.2.1. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Menurut Kamus Kamus Besar Bahasa Indonesia, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu *program* komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari *user*[6]. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Aplikasi dibagi menjadi tiga katergori, yaitu sebagai berikut :

1. Aplikasi Berbasis *Web*

   Aplikasi berbasis *web* atau disebut juga *web base*, merupakan jenis aplikasi berbasis teknologi *web browser*. Artinya, aplikasi ini berjalan jika ada *browser* dan koneksi *internet*. Selama ada *browser* dan jaringan *internet*, aplikasi berbasis *web* dapat diakses dengan mudah di sistem operasi apapun.

2. Aplikasi Berbasis *desktop*

   Aplikasi *desktop* merupakan aplikasi yang dioperasikan dengan cara menginstalnya terlebih dulu di sistem operasi. Sesuai namanya, aplikasi ini mengacu pada *program* yang diinstal pada perangkat desktop (PC dan laptop) mampu beroperasi secara *offline*, tetapi kita harus menginstalnya sendiri pada laptop atau komputer.

3. Aplikasi *Mobile*

   Pada perkembangan selanjutnya, ketika perangkat *smartphone* populer, maka berkembang pula aplikasi ke arah baru, yakni aplikasi berbasis *mobile*. Aplikasi berbasis *mobile* ini sedikit berbeda dengan aplikasi *web*. Aplikasi berbasis *mobile* umumnya ditujukan untuk perangkat *mobile* seperti *smartphone*. Karena bentuk *smartphone* yang

lebih kecil daripada *desktop,* maka otomatis tampilan pada aplikasi *mobile* disesuaikan dengan kebutuhan dengan *user smartphone*.

Aplikasi memungkinkan kita mempermudah pekerjaan menjadi lebih mudah dan efisien dibandingkan dengan proses yang belum terkompurisasikan seperti pencatatan, pencarian dokumen dll. Aplikasi yang akan dibangun dengan *platform mobile* dan sistem operasinya adalah *andorid,* dikarenakan saat ini *user andorid* di Indonesia sudah mencapai 59.91% dari total penduduk Indonesia[6], dan juga memiliki market share terbesar didunia sebesar 87,7% dari seluruh smart phone yang diaktifkan. Berdasarkan dari uraian masalah yang dipaparkan diatas, maka aplikasi menjadi sebuah media yang akan dibangun dengan tujuan untuk memenuhi kebutuhan *user* dalam menentukan makanan sehat dan tempat makan sehat yang berada di kota Bandung.

2.2.2. Aplikasi *Mobile*

Aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, dan penggunaan. Secara istilah aplikasi adalah *program* siap pakai yang melaksanakan suatu fungsi bagi *user* atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Kata *mobile* mempunyai arti bergerak atau berpindah sehingga aplikasi *mobile* merupakan sebutan untuk aplikasi yang berjalan di *mobile device.* Aplikasi berbasis *mobile* sendiri terbagi menjadi dua, aplikasi berbasis *mobile web* dan *native* apps, penjelasannya adalah sebagai berikut :

1. Aplikasi berbasis *mobile web* (*Hybird*)

   Aplikasi *hybrid* adalah aplikasi *web* yang ditransformasikan menjadi kode *native* pada *platform* seperti IOS atau *Andorid*. Aplikasi *hybrid* biasanya menggunakan *browser* untuk mengijinkan aplikasi *web* mengakses berbagai fitur di *device mobile* seperti *Push Notification*, *Contacts*, atau *Offline Data Storage*. Beberapa *tools* untuk mengembangkan aplikasi *hybrid* antara lain *Phonegap, Rubymotion* dan lain-lain.
2. *Native* Aplikasi

Aplikasi *native* adalah aplikasi yang dibangun dengan Bahasa pemrograman yang spesifik untuk *platform* tertentu. Contoh populernya yakni penggunaan bahasa pemrograman *Objective-C* atau *Swift* untuk *platform* IOS (*Apple*). Adapun *platform Andorid* yang menggunakan bahasa pemrograman *Java*.

Aplikasi *mobile* digunakan pada pembangunan aplikasi untuk rekomendasi pola makan dan tempat makan sehat ini, pada era maju seperti ini banyak orang menggunakan aplikasi *mobile* karena lebih mudah untuk digunakan, tampilan aplikasi *mobile* lebih nyaman dan dapat mendapatkan informasi lebih cepat. Jenis aplikasi yang dipakai adalah *native* aplikasi, menggunakan *framework React Native* berbasis *mobile andorid*.

2.2.3. *Andorid*

Pada Bulan Oktober 2003 *Andorid*, Inc. Didirikan di Palo, California, oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White untuk mengembangkan "perangkat *smartphone* yang lebih sadar akan lokasi dan preferensi *user*nya". Pada awalnya dikembangkan oleh *Andorid* Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google* Inc. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk *Google*, HTC, Intel, Motorola, Qualcomm, T-*Mobile*, dan Nvidia.

Setiap versi yang dirilis dinamakan Secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula berikut versi lengkapnya, Cupcake (1.5), Donut (1.6), Éclair (2.0 - 2.1), Froyo (2.2 - 2.2.3), Gingerbread (2.3 – 2.3.7), Honeycomb(3.0 – 3.2.6), Ice Cream Sandwich (4.0– 4.0.4), jelly Bean (4.1 – 4.3), Kitkat (4.4), Lollipop (5.0), Marshmallow (6.0) dan Nougat (7.0) [7].

Pada pembangun aplikasi untuk rekomendasi pola makan dan tempat makan sehat, sistem operasi yang digunakan adalah adalah *Andorid* versi 5.0 (Lollippop) karena mempunyai material merupakan hasil desain terbaru sabagai spesifikasi minimal cocok untuk pengembangan aplikasi karena sudah

mempunyai hasil desain *andorid* terbaru selain itu. Versi mempunyai kelebihan dalam peningkatan pada pencarian kontekstual sehingga bisa menentukan lokasi dengan tepat ke tempat aplikasi tertentu dan akan mulai secara tepat.

2.2.4. *Andorid Life cycle*

Aplikasi *andorid* terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar *file* musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen *andorid* yang berbeda seperti ditunjukkan pada, klasifikasi tersebut berdasarkan kelas-kelas dasar *java* yang digunakan :

Tabel 2.1 Komponen Aplikasi *Andorid*

Functionality	Java Base Class	Examples
Focused thing a *user* can do	Activity	Edit a note, play a game
Background process	*Service*	Play music, update weather icon
Receive messages	Broadcast Receiver	Trigger alarm upon event
Store and retrive data	Content Provider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service*, *BroadcastReceiver*. *ContentProvider* [8].

Andorid memiliki paradigma pemrograman lain tidak seperti *paradigm* pemrograman biasa di mana aplikasi yang dijalankan pada fungsi *main*(), sistem *andorid* menjalankan kode dalam *method activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi *andorid* memiliki siklus hidup yang berbeda dengan aplikasi *desktop* atau *web*. Hal ini dikarenakan aplikasi *mobile* memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai *user* [8].

2.2.5. Jaringan *Internet*

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut. *Internet* awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat (US *Department of Defense*) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau Advanced Research Project Agency Network. Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan University of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat komputer di lokasi universitas berbeda tersebut[9].

2.2.6. API *Clarifai*

2.2.6.1. API

API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan *program*mer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API atau *Application Programming Interface* juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak.

Dengan adanya API, maka memudahkan seorang *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan *developer* menggunakan sistem *function*. Proses ini dikelola melalui *operating* sistem. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi [10].

Keuntungan dengan menggunakan API adalah sebagai berikut:

1. Portabilitas.

   Developer yang menggunakan API dapat menjalankan *program*nya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut. Lebih Mudah Dimengerti

2. Lebih Mudah Dimengerti

Developer yang menggunakan API dapat menjalankan *program*nya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut. Lebih Mudah Dimengerti API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa sistem call. Hal ini sangat penting dalam hal editing dan pengembangan.

Cara menggunakan API:

1. Dilakukan dengan mengimpor package/kelas
2. Ada beberapa kelas bernama sama dipackage yang berbeda, yaitu:
    a) *Import* salah satu dan gunakan nama lengkap untuk yang lain
    b) Gunakan nama lengkap semua kelas

Kebanyakan Sistem Operasi seperti *Windows*, menyediakan fasilitas API sehingga *program*mer dapat melakukan aktivitas *programming* dengan lebih konsisten. Meskipun API didesain untuk *program*er, namun API juga baik untuk *user* karena setidaknya dapat menjamin bahwa *program* tersebut memiliki *interface* yang sama, sehingga lebih mudah untuk dipelajari.

2.2.6.2. *Clarifai*

Clarifai adalah alat pengenalan gambar dan video yang secara otomatis memberikan tag ke objek dan kategori yang hanya mengambil piksel sebagai *input*. Sistem ini didasarkan pada jaringan saraf, teknik pembelajaran mesin *scalable* yang dapat menangani skala besar konten visual yang mengalir melalui API. *Clarifai* juga menggunakan kesamaan semantik dan visual untuk membandingkan gambar yang diunggah dengan gambar lainnya di *library* mereka untuk menampilkan kesamaan[11]. API *Clarifai* digunakan di aplikasi ini untuk menentukan kandungan kesehatan yang terdapat pada makanan dan minuman, sehingga dapat diketahui kandungan kesehatan dalam makanan dan minuman itu, sehingga dapat membantu *user* untuk mengetahui jenis makanan yang ada pada makanan dan minuman.

Model adalah blok pemrosesan yang mengubah masukan (misalnya gambar) ke prediksi (misalnya konsep, kotak pembatas, dll.). Ketika Anda menjalankan

sebuah *input* melalui sebuah model, akan mengembalikan prediksi yang dilihat model pada *input*.

Setiap model dilatih untuk mengenali rangkaian *output* yang unik, seperti konsep, wajah, dan warna. Misalnya, jika Anda menjalankan masukan Anda melalui model 'makanan', ia akan mengembalikan konsep yang diketahui oleh 'makanan', seperti nama hidangan tertentu. Jika Anda menjalankan *input* yang sama melalui model 'warna', itu akan mengembalikan prediksi tentang warna dominan pada gambar Anda.

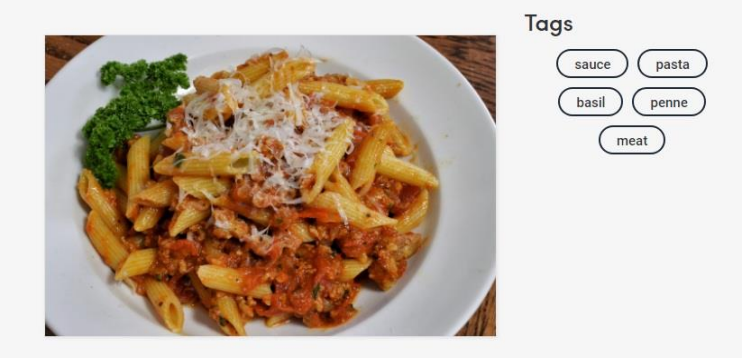
Clarifai menyediakan beragam model pra-terlatih yang bisa Anda gunakan untuk membuat prediksi. Model ini dibangun di rumah dan telah diuji secara menyeluruh. Sebelum melatih model Anda sendiri, sebaiknya coba ini untuk memastikan apakah sesuai dengan kebutuhan anda[11].

Berikut ini ada contoh model *clarifai*:

1. The ‘Food’ Model

   Model 'Makanan' menganalisa gambar dan video dan mengembalikan nilai probabilitas pada kemungkinan gambar atau video tersebut mengandung bahan makanan dan hidangan yang dikenal.

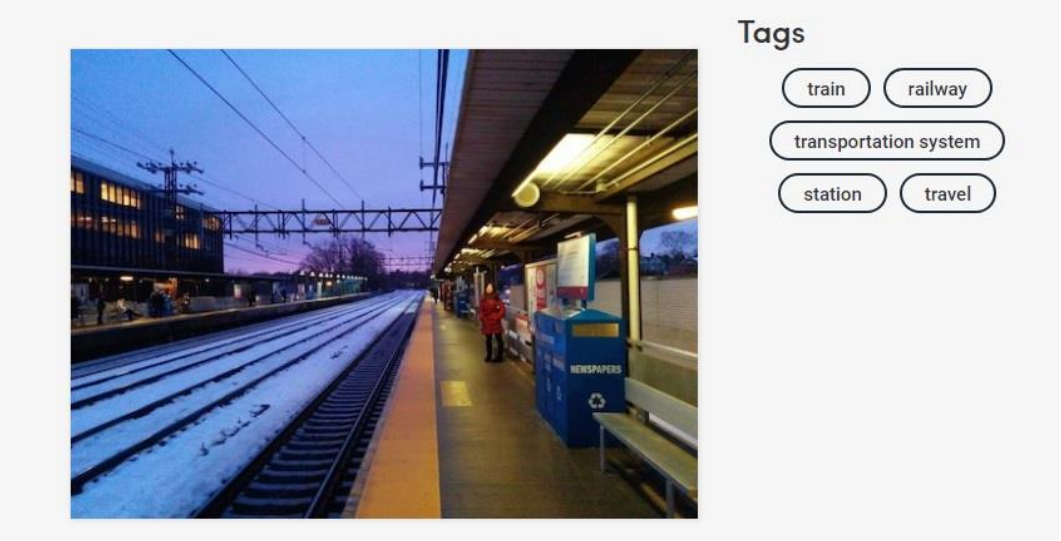
   Model saat ini dirancang untuk mengidentifikasi bahan makanan tertentu dan bahan yang terlihat[11].



Gambar 2.1 Model *Food Items*

2. General Model

   Model 'General' berisi berbagai tag di berbagai topik yang berbeda. Dalam kebanyakan kasus, tag yang dikembalikan dari model umum akan cukup mengenali apa yang ada di dalam gambar atau video anda[11].

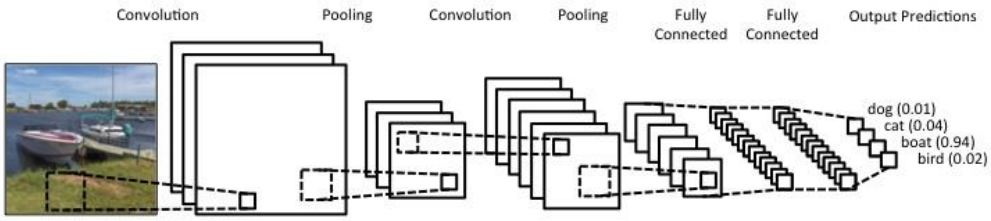


Gambar 2.2 Model *General*

2.2.6.4. Cara Kerja *Clarifai*

Model komputer kami dilatih pada daftar kemungkinan keluaran (tag) untuk diterapkan pada masukan (konten Anda). Menggunakan pembelajaran mesin, sebuah proses yang memungkinkan komputer belajar dari data dan menarik kesimpulannya sendiri, model kami dapat secara otomatis mengidentifikasi tag yang benar untuk gambar atau video tertentu. Model ini kemudian dibuat mudah diakses melalui API sederhana kami.

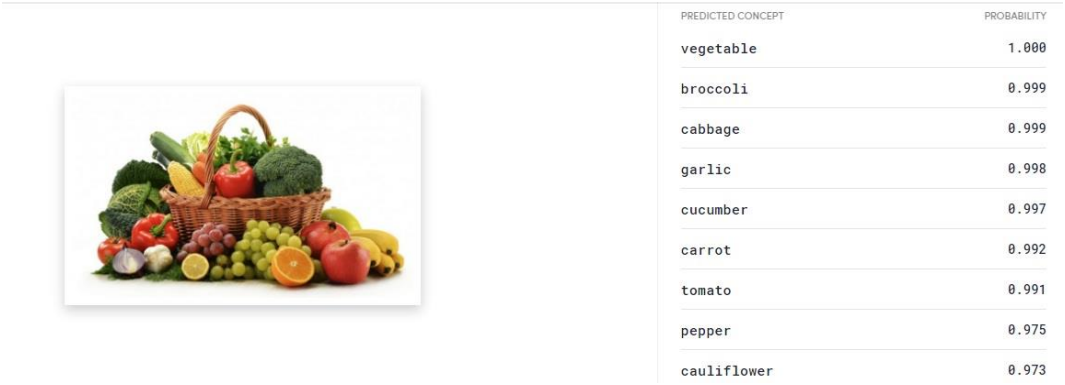
Model inti kami mengidentifikasi 11.000+ konsep umum seperti objek, gagasan, dan emosi. Bergantung pada tingkat spesifisitas yang Anda butuhkan untuk data Anda, Anda mungkin memerlukan model tambahan. Misalnya, model inti kami dapat memberi tahu Anda apakah ada sesuatu "makanan" tapi Anda memerlukan model Makanan untuk memberi tahu Anda jenis makanan apa, seperti "pizza".



Gambar 2.3 Skema Proses *Clarifai* API

2.2.6.5. Gambar Deskriptor Anotasi *Clarifai*

Proses untuk menghasilkan *Clarifai* deskriptor dimulai dengan langkah awal di mana kita menemukan semua label termasuk dalam do*main* masalah kita. Hal ini dilakukan dengan memproses semua gambar yang tersedia dengan API *Clarifai*, dan kemudian menghapus nilai-nilai label digandakan. Langkah ini dapat dilihat sebagai codebook atau kamus Proses generasi. Setiap gambar *input* kemudian dikodekan sebagai daftar probabilitas nilai-nilai yang panjangnya ditentukan oleh ukuran kamus. Nilai-I di deskripsi ini akan sesuai dengan probabilitas dan dikembalikan oleh *Clarifai* untuk menandakan label.



Gambar 2.4 Label dan probabilitas API *Clarifai*

2.2.7. *Global Positioning* Sistem (GPS)

GPS adalah singkatan dari Global Positioning Sistem, 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[12]. Sinyal ini diterima oleh alat penerima (receiver) di

permukaan, dimana GPS receiver ini akan mengumpulkan informasi dari satelit GPS,seperti:

1. GPS *Receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. GPS memberikan informasi lokasi dalam tiga dimensi yaitu, Latitude, Longitude dan Elevasi
3. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. GPS dapat menunjukkan arah tujuan atau arah perjalanan.
5. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS receiver.
6. Komulasi 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.

GPS dapat melakukan perhitungan dan menentukan posisi *user* dan menampilkan dalam *maps*. Jika sudah dapat menyimpan posisi *user* selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah tuju, rute, tujuan perjalanan serta jarak tujuan. Dalam pembangun aplikasi untuk rekomendasi pola makan dan tempat makan sehat teknologi GPS digunakan untuk perhitungan waktu yang akurat sangat menentukan informasi lokasi *user*.

2.2.8. Location Based *Service* (LBS)

Location Based Service adalah *Service* yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* (GPS) dan *Google*'s cell-based location. Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis, namun sebagai *user* kita membutuhkan alamat atau posisi realtime kita bukan nilai lintang dan bujur [12], Dua unsur utama LBS adalah:

1. Location Manager (API *Maps*)

   Menyediakan tools/resource untuk LBS, Application *Program*ming *Interface* (API) *Maps* menyediakan fasilitas untuk menampilkan,

memanipulasi *maps*/peta beserta feature-feature lainnya seperti tampilan satelit, street (jalan), maupun gabungannya. Paket ini berada pada com.*google*.*andorid*.*maps*.

2. Location Providers (API Location)

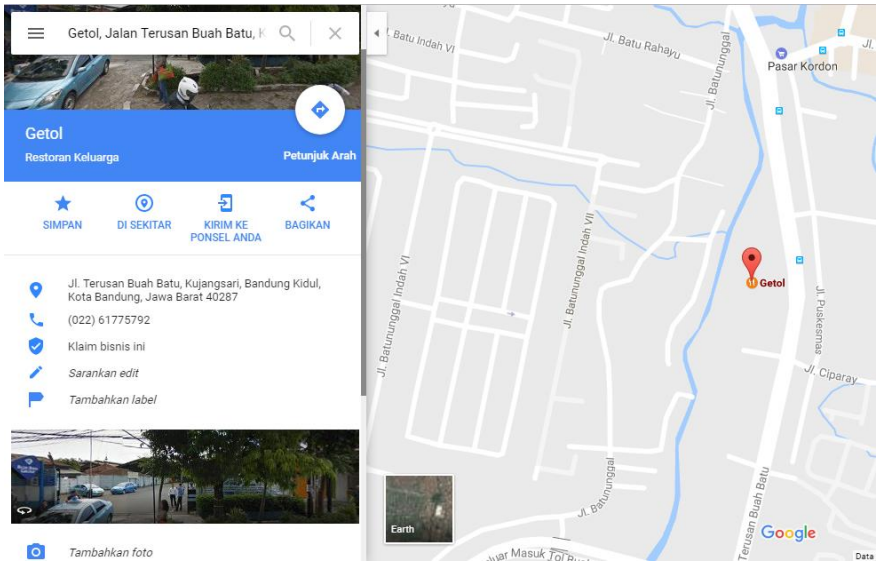
   Menyediakan teknologi pencarian lokasi yang digunakan oleh device/perangkat. API Location berhubungan dengan data GPS dan data lokasi real-time. API Location berada pada paket *Andorid* yaitu dalam paket *andorid*.location. Dengan Location Manager, kita dapat menentukan lokasi kita saat ini, Track gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

Teknologi ini dipakai dalam pembangun aplikasi untuk rekomendasi pola makan dan tempat makan sehat, untuk menentukan lokasi geografis tempat makan sehat, Selain dapat mengetahui posisi *user*, aplikasi LBS juga dapat menentukan posisi tempat-tempat tertentu. Dan dengan kombinasi ini, aplikasi LBS akan mencari rute untuk menghubungkan posisi *user* dengan suatu tempat.

2.2.9. *Google Maps* API

GoogleMaps adalah layanan gratis yang diberikan oleh *Google* dan sangat popular. *GoogleMaps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah seperti pada gambar 7. Cara membuat *GoogleMaps* untuk ditampilkan pada suatu *web* atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi *Internet* yang sangat stabil. Dengan menggunakan *GoogleMaps* API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan [13].

GoogleMaps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur *GoogleMaps* dalam *web* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *GoogleMaps* API. *GoogleMaps* API adalah suatu library yang berbentuk JavaScript. Layananan *Google Maps* digunakan untuk menampilkan tempat Makas sehat yang direkomendasikan dalam pembangun aplikasi untuk rekomendasi pola makan dan tempat makan sehat.



Gambar 2.5 Tampilan Peta Pada Google *Maps*

2.2.10. *Object Oriented Analysis Desain*

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaiut analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem[14].

2.2.10.1 *Unified Modeling Language* (UML)

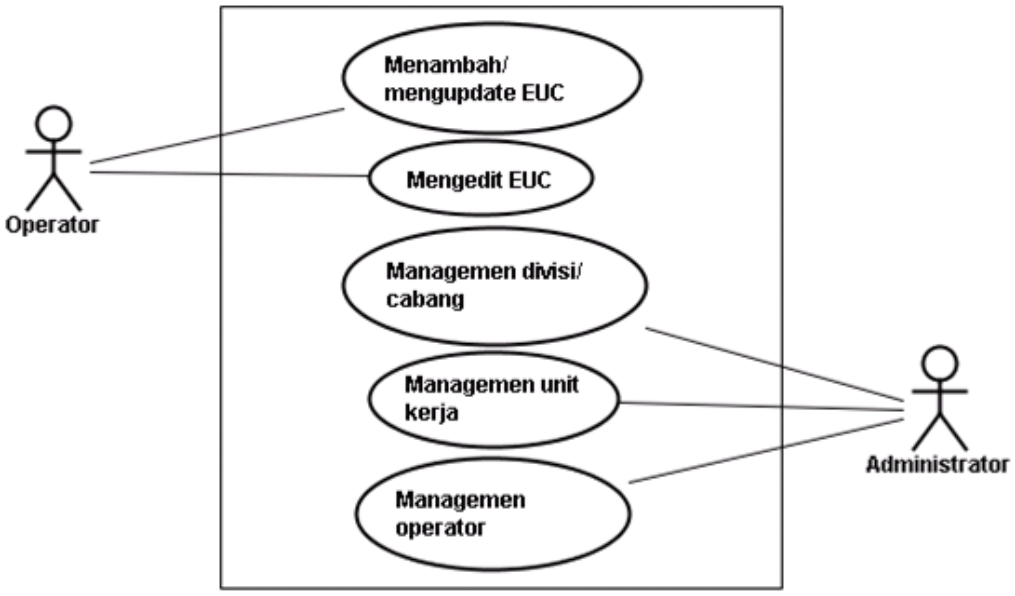
Metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Unified Modeling Language (UML) adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasi artefacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan blueprint *software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak[15].

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa

pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut :

1. *Use Case Diagram*

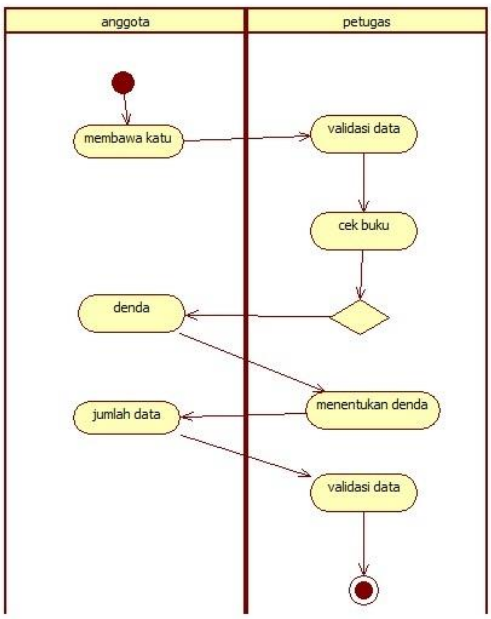
   Use Case diagram menggambarkan sejumlah eksternal aktor dan hubungannya ke *Use Case* yang diberikan oleh sistem.



Gambar 2.6 Contoh *Use Case Diagram*

2. *Activity Diagram*

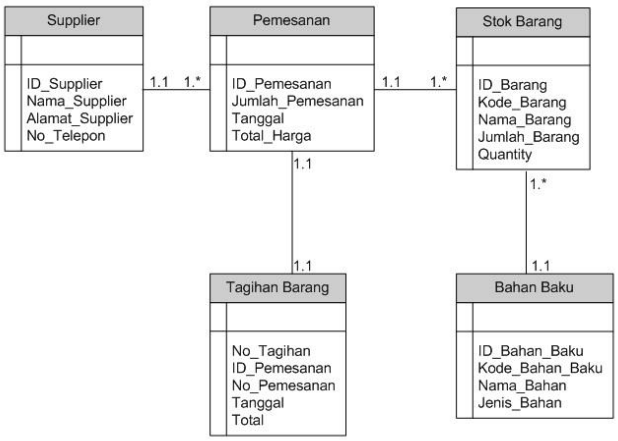
   Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *Use Case*.



Gambar 2.7 Contoh *Activity Diagram*

3. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi class, package (paket) dan objek beserta hubungan satu sama lain seperti containment (penahanan), pewarisan, asosiasi dan lain-lain.

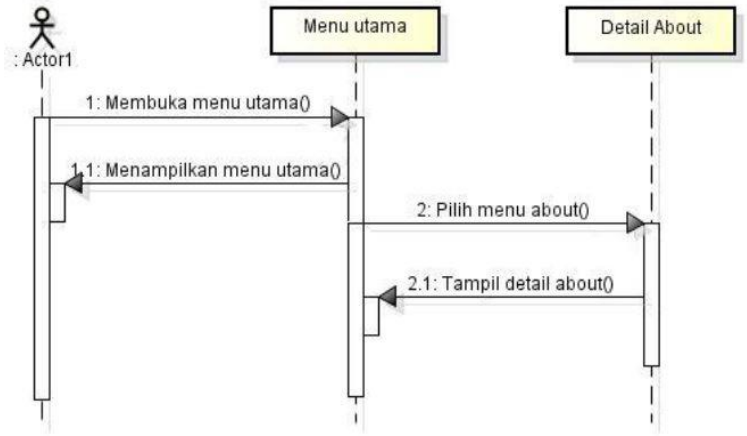


Gambar 2.8 Contoh *Class Diagram*

4. *Sequence Diagram*

Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim

antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



Gambar 2.9 Contoh *Sequence Diagram*

2.2.11. *Structured Analysis and Design*

Structured Analysis and Design adalah pendekatan sistem analisis dan desain sistem informasi secara hierarki. SADM diproduksi untuk CCTA, kantor pemerintah Inggris yang berkaitan dengan penggunaan teknologi di pemerintahan dari tahun 1980 dan seterusnya[16].

2.2.11.1. *Diagram Konteks*

Diagram Con*text*, adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem)[16].

2.2.11.2. *Data Flow Diagram*

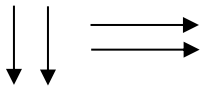
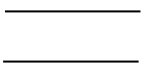
Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. DFD terdiri dari empat komponen utama diantaranya adalah

1. Entitas
2. Proses

3. Data Store
4. Data Flow

DFD dapat dikatakan merupakan breakdown dari diagram konteks. Analisis mengenai sistem yang lebih detail dapat dilakukan dengan melakukan breakdown pada DFD level 1, 2 dan seterusnya[16]. Berikut adalah Tabel 2 Simbol Diagram Konteks dan DFD :

Tabel 2.2 Simbol *Diagram* Konteks dan DFD

Simbol	Nama Simbol	Fungsi
	Entitas Luar	Menggambarkan entitas eksternal yang berhubungan dengan sistem
	Sistem (Konteks) / Proses (DFD)	Menggambarkan proses yang ada dalam suatu sistem
	Aliran Data/Informasi	Menggambarkan aliran data antar proses, data store dan entitas luar
	Data Storage	Mengambarkan tempat penyimpanan data di dalam sistem

2.2.11.3. Spesifikasi Proses

Spesifikasi Proses adalah tabel yang berisi keterangan atau deskripsi dari semua proses yang terdapat pada DFD. Logika proses yang ditulis pada spesifikasi proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau *pseudecode* tetapi tidak boleh mengkombinasikan kedua bahasa tersebut[16]. Berikut adalah Tabel 3 Contoh Spesifikasi Proses :

Tabel 2.3 Contoh Spesifikasi Proses

No	Proses	Keterangan
1.	No Proses	Menyatakn no proses
	Nama Proses	Menyatakan nama proses
	Source (Sumber)	(Menyatakan sumber data *input* menuju proses)
	Input	(menyatakan isi data yang masuk ke proses)
	Output	(menyatakan informasi yang keluar dari proses)
	Destination	(menatakan tujuan informasi *output* dari

No	Proses	Keterangan
		proses)
	Logika Proses	(menyatakan algoritma dari proses)

2.2.11.4. Kamus Data

Kamus data adalah tabel yang berisi deskripsi dari data yang mengalir pada DFD, Penjelasan struktur data dapat berupa field tiap data harus sama dengan yang sudah dimodelkan di ERD[16]. Berikut adalah Tabel 4 Contoh Kamus Data

Tabel 2.4 Contoh Kamus Data

Nama aliran data	Data Siswa Nama dari data yang digunakan
Where used/How used	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data.

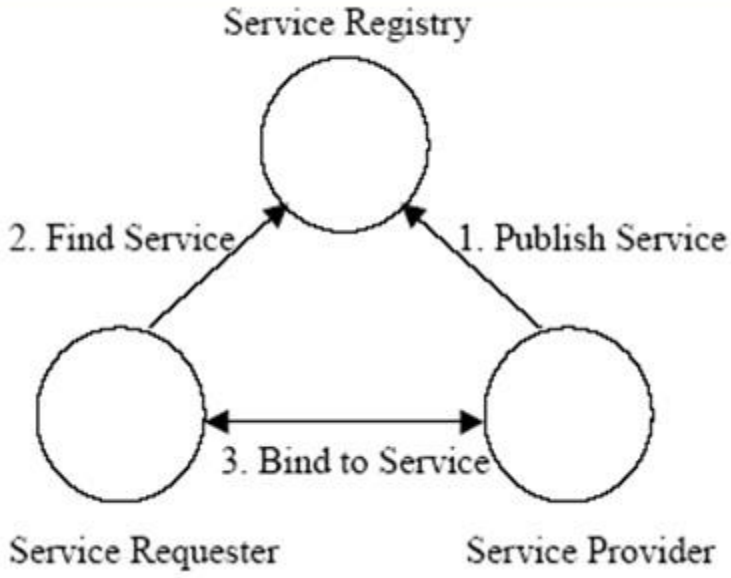
2.2.12. *WEB Service*

Web Service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web Service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web* site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*Service*) yang disediakan oleh suatu sistem yang menyediakan *web Service*. *Web Service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa compiler [17].

Web Service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

1. *Web Service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web Service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web Service* cukup di-upload ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web Service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web Service* tidak memerlukan konfigurasi khusus di sisi firewall.

Web Service memliliki tiga entitas dalam arsitekturnya yaitu, *Service request*er (peminta layanan), *Service* provider (penyedia layanan) dan *Service* registry (daftar layanan) berikut gambaran dan penjelasan dari arsitektur *web Service*.



Gambar 2.10 Arsitektur *WEB Service*

1. *Service* provider, berfungsi untuk menyediakan layanan dan mengolah sebuah registry agar layanan tersebut dapat tersedia.
2. *Service* resgistry, berfungsi sebagai lokasi tengah yang mendeskripsikan semua layanan yang telah di register.

*Service requesto*r, berfungsi sebagai peminta layanan yang menacri dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.2.13. *React Native*

React Native adalah sebuah framework javascript yang di kembang kan oleh *facebook* dan memungkin pengembang untuk membuat aplikasi *mobile andorid* atau ios menggunakan teknologi *web*. Seperti yang di ketahui sekarang ini juga sudah banyak *framework javascript* yang bisa kita gunakan untuk membuat aplikasi *andorid* atau ios, akan tetapi untuk *React Native* ini berbeda dengan f*ramework - framework javascript* yang lainya. Dengan *React Native* ini kita tidak membuat aplikasi *Hybrid* yang dimana aplikasi ini berjalan di *javascript runtime* akan tetapi kita akan membuat *real* Aplikasi dimana untuk *andorid* akan di *compile* di *java* dan untuk IOS akan di *compile* di *Objective-C*[18].

Didalam *React Native* ini terdapat beberapa kompenen yang membuat nya bisa gunakan untuk membuat aplikasi *mobile cross-platform*. Berikut ini adalah beberapa fitur dari *React Native* ini :

1. *ReactJS*

   Sebelum *react native* di keluarkan oleh *facebook* yang pertama di kenalkan adalah *ReactJS* yaitu *framework javascript* yang bisa digunakan untuk membuat aplikasi *web* dengan *java script*.

2. *Native*

   Di *Andorid* dan IOS memiliki arsitektur yang berbeda, pada pembangunan aplikasi, *Native* ini yang berfungsi untuk menghandle komponen dari masing - masing *platform*.

3. *Platform*

   Untuk saat ini *platform* yang sudah terintegrasi adalah *Andorid* dan IOS. Mungkin kedepanya bisa lebih dari dua *platform* ini yang bisa terintegrasi dengan *react native*.

Pada pembangun aplikasi untuk rekomendasi pola makan dan tempat makan sehat, menggunakan *framework React Native* untuk pembangunan aplikasi berbasis *mobile* dengan *javascript* dan bersifat *opensource* bagi pengembang aplikasi.

2.2.14. *Hypertext Propocessor* (PHP)

PHP singkatan dari *Personal Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua script PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP pertama kali di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, namun sekarang di ambil oleh oleh *The PHP Group*. Pada awalnya PHP adalah singkatan dari *Personal Home Page*, namun dalam perkembangannya, di ubah menjadi PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya *web server*. PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *Windows* maupun Linux [19].

PHP juga dibangun sebagai modul pada *web apache* dan sebagai *binary* yang dapat berjalan sebagai CGI. Pada pembangunan aplikasi untuk rekomendasi pola makan dan tempat makan sehat khususnya halaman *backend* menggunakan *script* PHP untuk membuat halaman pelaporan.

2.2.15. HTML 5

HTML5 (*Hypertext Markup Language version* 5) adalah sebuah bahasa markah yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada *internet*. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan elemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi *web* yang lebih kompleks [20].

HTML5 merupakan salah satu karya Konsortium *Waring Wera Wanua* (*World Wide Web Consortium, W3C*) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5

merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat *web*.

Pada pembangunan aplikasi untuk rekomendasi pola makan dan tempat makan sehat khususnya halaman backend menggunakan script HTML5 untuk membuat halaman *admin*istrator. Karena sudah *responsive* atau bisa menyesuaikan bentuk halaman.

2.2.16. *Andorid* Studio

Andorid Studio adalah IDE resmi untuk membangun aplikasi *Andorid* berdasarkan IntelliJ IDEA. IntelliJ IDEA sendiri adalah *Java Integrated Development Environtment* (IDE) yang dikembangkan oleh JetBrains, untuk mengembangkan perangkat lunak komputer. IntelliJ IDE berfungsi dalam membantu anda di dunia percodingan baik dari segi navigasi, penyokong produktivitas, hingga code editor yang cerdas[22]. Berdasarkan *web*site http://developer.*andorid*.com, *andorid* studio memiliki keunggulan:

1. *Andorid* studio didasarkan pada intellij DEA yang memberikan kemungkinan tercepat dalam melakukan coding dan *running workflow* sehingga meng-*coding* dan melakukan iterasi dapat dilakukan dengan cepat.
2. Struktur proyek *Andorid* Studio dan pembangunannya yang berbasis *gradle* memberikan fleksibilitas yang dibutuhkan untuk menghasilkan APK untuk semua jenis *device* sehingga konfigurasi pemangunan dapat dlakukan tanpa batas.
3. *Andorid* Studio menyediakan fitur-fitur yang membuat pengembang menjadi percaya diri dan dapat membuat kode terbaik.
4. *Andorid* Studio mengetahui bahwa tidak seluruh kode ditulis dengan bahasa java dan tidak semua kode dapat berjalan di *device user*. Dengan fitur-fitur yang dimiliki *Andorid* Studio maka aplikasi menjadi kaya dan terkoneksi satu sama lain.
5. *Andorid* Studio menyediakan alat GUI untuk mempermudah pengembang dalam merancang tampilan aplikasi sehingga pengembang tidak lelah.

Dengan menggunakan *andorid* studio pengembang aplikasi dapat mengerjakan aplikasi yang akan dibangunya. Pada pembangunan aplikasi untuk rekomendasi makan sehat dan tempat makan sehat *andorid* studio digunakan untuk menjalankan atau *debugging program* yang dibangun untuk mencari dan mengurangi *bug*, atau kerusakan didalam sebuah *program*, sehingga aplikasi dapat bekerja sesuai dengan harapan.

2.2.17. *Java Script*

Java Script pada awalnya diciptakan untuk menangani validasi dari sebuah formulir dalam *website* yang sebelumnya ditugaskan pada bahasa pemograman *server* seperti Perl. Namun dengan berkembangnya zaman *Java Script* menjadi pe*main* penting dalam perkembangan teknologi *website*. *Java Script* adalah bahasa *scripting* yang dibuat untuk berinteraksi dengan halaman *website* dan dibuat dengan 3 bagian :

1. *ECMAScript*, yang didefinisikan dalam ECMA-262 dan menjadi bagian utama dalam menjalankan fungsionalitas.
2. *Document Object Model* (DOM), yang menyediakan *method* dan *interface* untuk bekerja dengan isi dari halaman *web*site.
3. *Browser Object Model* (BOM), yang menyediakan *method* dan *interface* untuk berinteraksi dengan browser.

2.2.18. *Java Script Object Natation* (JSON)

Java Script Object Notation (JSON) pertama kali dikenalkan oleh Douglas Crockford adalah bagian dari *Java Script*, dibuat dengan menggunakan pola dari JavaScript untuk merepresentasikan data terstruktur. Crockford membuat JSON sebagai alternatif dari *(eXtended Markup Language*) XML untuk mengakses data terstruktur dari *Java Script* dengan menggunakan *method eval()* dan tidak membutuhkan pembuatan dari DOM. Bagian terpenting dari JSON adalah format data bukan bahasa pemrogramannya. JSON bukan bagian dari *JavaScript* walaupun mereka saling berbagi *syntax*, JSON juga tidak semata-mata digunakan oleh *JavaScript*. JSON merupakan data format, jadi JSON memiliki *parsers* dan *serializers* yang ada dalam banyak bahasa pemograman. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*Object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash Tabel*), 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 (*sequence*).

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

2.2.19. *Database* MYSQL

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu *program* komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang multithread, dan multi-*user*. MySQL adalah implementasi dari sistem manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. Dalam MYSQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu:

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu *program* komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang multithread, dan multi-*user*. MySQL adalah implementasi dari sistem manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. Dalam MYSQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu:

1. *Create*

   Create adalah operasi penambahan data baru kedalam Tabel. Querynya adalah sebagai berikut :

   INSERT INTO Tabel_name(column1,column2,column3,...)

   VALUES (value1,value2,value3,...);

2. *Read*

   Read adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam *database*. Querynya adalah sebagai berikut:

   SELECT column_name,column_name
   FROM Tabel_name;

3. *Update*

   Update adalah operasi untuk mengubah data yang ada di dalam *database*. Querynya adalah sebagai berikut:

   UPDATE Tabel_name SET column1=value1,column2=value2,...
   WHERE some_column=some_value;

4. Delete

   Delete adalah operasi untuk menghapus data yang ada di dalam Tabel. Querynya adalah sebagai berikut:

   DELETE FROM Tabel_name
   WHERE some_column=some_value;

2.2.20. XAMPP

XAMPP adalah sebuah *software web server* apache yang didalamnya sudah tersedia *database server* mysql dan support php *program*ming. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache *Web Server*, MySQL *Database Server*,

PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk *file* terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah *server*

secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan [23].

Dalam XAMPP banyak diaplikasikan dan digunakan oleh kalangan *user* komputer di bidang pemrograman *web* XAMPP berfungsi sebagai *server offline* yang berdiri sendiri (seringkali disebut sebagai localhost). XAMPP terdiri dari beberapa *program* yaitu Apache HTTP *Server*, MYSQL *Database*, PHP, dan Pearl XAMPP juga dilengkapi fitur manajemen *database* PHPMy*Admin* seperti pada *server* hosting sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis *database* secara mudah.