BAB 2
TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

Tahapan tinjauan perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan di PT. Mitra Buana Koorporindo. Adapun tinjauan perusahaan yang dimaksud meliputi profil perusahaan, visi dan misi perusahaan, logo, serta struktur organisasinya.

2.1.1. Profil Perusahaan

PT. Mitra Buana Koorporindo, didirikan pada tanggal 13 Februari 2007 di Jakarta dan telah mengembangkan serta menawarkan komitmen dan solusi berupa Pelayanan Profesional untuk Operasi IT Berbasis standar industri ITIL, berkomitmen untuk mencapai keunggulan operasional di pasar korporasi nasional. Perusahaan ini telah didukung penuh oleh beberapa Merek IT Top Dunia sebagai Mitra Bisnisnya. Memiliki banyak kantor cabang dan gudang yang tersebar diberbagai wilayah terutama daerah Jabodetabek, dan *head office* yang bertempat di Komplek Ketapang Business Center Jl. K.H Zainul Arifin No.20 Blok B.10 & B.11 Jakarta Pusat 10130 Indonesia.

PT. Mitra Buana Koorporindo (MBK) merupakan salah satu perusahaan penyedia jasa Sistem Integrator (SI) skala Nasional yang menyediakan berbagai solusi IT khusus untuk pelanggan bisnis inti, dan juga merupakan "Mitra Bisnis" dari beberapa produk utama IT global dan lokal terkemuka [1]. Ruang lingkup bisnis diantaranya pengadaan, dukungan, pemasangan, pemeliharaan, dan solusi IT yang diperuntukan melayani *desktop management* atau kebutuhan perangkat IT oleh pelanggan, misal perangkat komputer, printer, barcode scanner, dan lain-lain. Selain itu PT Mitra Buana Koorporindo memiliki cangkupan Nasional karena beroperasi di 30 provinsi di seluruh Indonesia, terbukti loyal karena sudah bekerjasaman dengan lebih dari 1000 pelanggan dari Pemerintah, BUMN, Swasta, Jasa Keuangan, Pendidikan, Minyak & Gas, Transportasi, Telekomunikasi, dan lain-lain.

2.1.2. Visi dan Misi Perusahaan

Perusahaan PT. Mitra Buana Koorporindo mempunyai visi jangka panjang serta misi utama yang kuat dalam menentukan arah gerak dan peta jalannya ke depan, visi dan misi tersebut yaitu:

Visi :

"Memimpin pasar korporat nasional Indonesia sebagai Perusahaan IT System Integrator."

Misi :

"Pada setiap tahap siklus hidup TI, layanan perusahaan dapat menyatukan pengalaman, metode yang telah terbukti, dan jalan menuju pikiran batin PT. Mitra Buana Koorporindo untuk mengubah visi Perusahaan menjadi kenyataan praktis dan bisnis. Sehingga dapat memberikan ruang lingkup Layanan dan Solusi yang akan memberi pelanggan keuntungan operasional dan biaya yang optimal (Perangkat Keras, Perangkat Lunak, Jaringan, Pemeliharaan & Layanan) Untuk menciptakan hubungan kemitraan dengan pelanggan".

Gambar 2. 1 Visi Perusahaan

Sumber gambar : http://mbk.co.id/en/about/vision

2.1.3. Logo Perusahaan

Gambar 2. 2 Logo Perusahaan

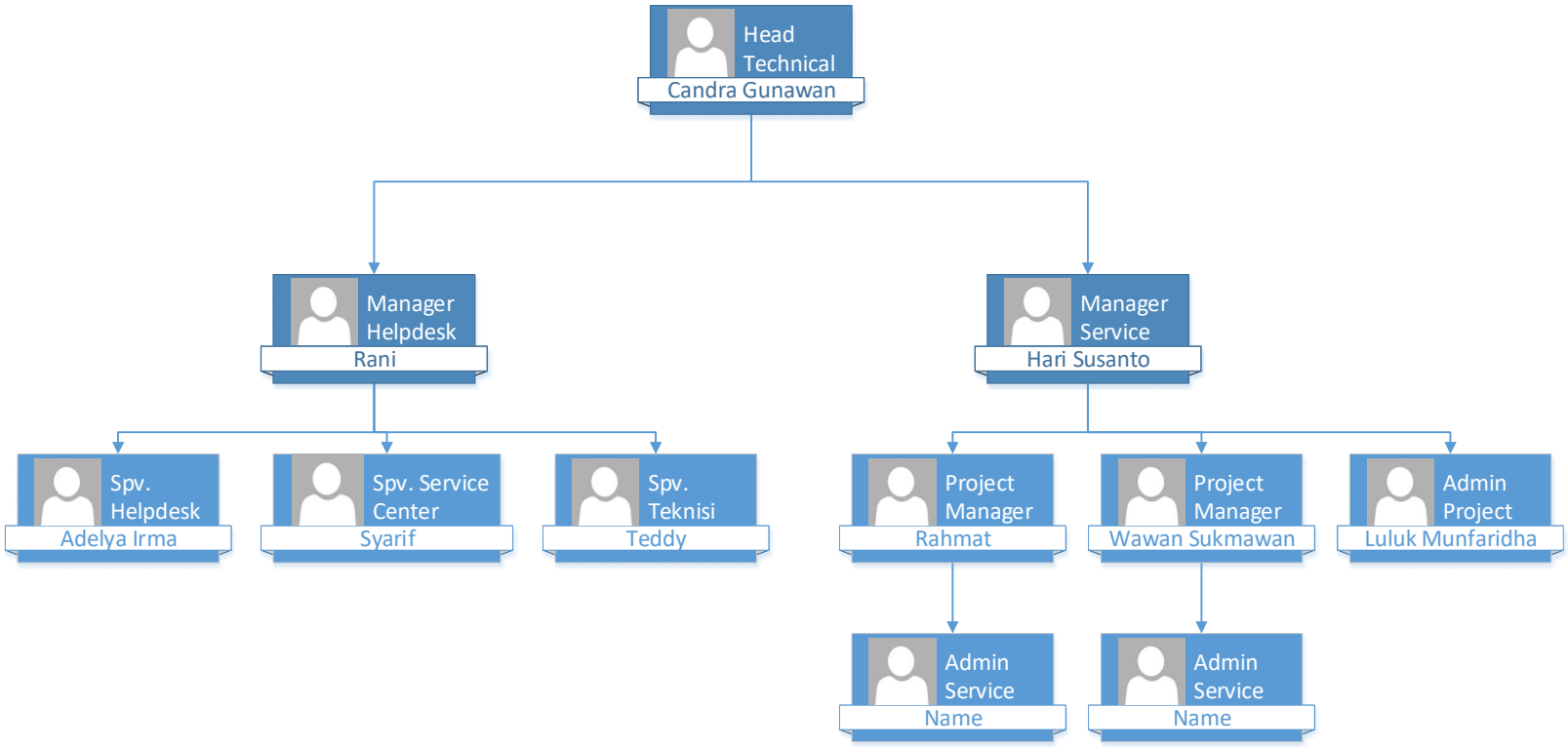
Sumber gambar : http://mbk.co.id/en/about /philosophy

Logo PT. Mitra Buana Koorporindo terdiri dari tiga buah huruf sebagai inisial perusahaan yang kemudian digabungkan dengan bentuk dua manusia yang saling berhubungan sebagai simbol kebersamaan, menembus keluar dan masuk ke huruf K sehingga membentuk satu kesatuan dan keseimbangan. Yang artinya adalah bersama-sama mencapai kesuksesan, dimaksudkan bahwa Mitra Buana Koorporindo membantu Anda untuk mencapai tujuan dan memiliki kekuatan untuk menuju keberhasilan.

Adapun makna dari warna serta bentuk pada logo :

1. Warna biru tua melambangkan kepercayaan, kejujuran dan kemitraan.
2. Warna hijau dimaksudkan bahwa MBK akan selalu bertumbuh, beruntung dan bersahabat dalam meraih keberhasilan bersama Anda.
3. Warna abu-abu pada logogram berarti teknologi, modern dan kuat.
4. Bentuk kedinamisan dari MBK dapat dilihat dari bentuk full signature yang italic (condong kanan).

2.1.4. Struktur Organisasi



Gambar 2. 3 Struktur Organisasi Divisi Technical MBK

2.2. Kajian Teori

Sebelum pembangunan aplikasi tentu diperlukan pengetahuan-pengetahuan yang terkait dengan penelitian yang diambil, diantaranya yaitu:

1. Konsep Dasar Aplikasi
2. *Maintenance Service*
3. *Standar Level Agreement*
4. Android
5. *Location Based Services*
6. *Global Positioning System*
7. *Application Programming Interface*
8. *Google Maps*
9. *Mapbox*
10. *Geocoding*
11. *Mobile Reporting*
12. *Barcode*
13. *QR-Code*
14. PHP
15. *Laravel*
16. *Flutter*
17. *Unified Modeling Language*

2.2.1. Kosep Dasar Aplikasi

2.2.1.1. Pengertian Aplikasi

Aplikasi berasal dari kata application yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [11].

2.2.1.2. Aplikasi Mobile

Aplikasi mobile berasal dari kata aplikasi dan mobile. Aplikasi yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain

dan dapat digunakan oleh sasaran yang dituju sedangkan mobile dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Aplikasi mobile dapat diartikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, smartphone, dan perangkat sejenisnya [12].

2.2.2. Maintenance Service

Maintenance (pemeliharaan) ialah suatu kegiatan yang dilakukakan secara sengaja (sadar) terhadap suatu perangkat/sistem menggunakan cara atau prosedur tertentu dengan tujuan agar perangkat/sistem tersebut dapat berfungsi, beroperasi dengan lancar, aman, efektif, dan efesien. Layanan pemeliharaan digolongkan menjadi dua kategori utama, yaitu *corrective maintenance* (CM) dan *preventive maintenance* (PM). *Corrective maintenance* dilakukan setelah *system breakdown*. Sedangkan *Preventive maintenance* dilakukan berdasarkan jadwal yang sudah ditetapkan dan dilakukan pada saat sistem masih beroperasi. Tujuannya adalah untuk menjaga sistem dalam fase tersedia dengan melakukan perbaikan pada komponen [2]. Maintenance Service dapat dijelaskan yaitu sebagai suatu layanan pemeliharaan yang dimaksudkan untuk mengelola kendala dan kerusakan-kerusakan alat/perangkat yang dimanage dengan tujuan agar alat/perangkat yang digunakan tersebut oleh pengguna dapat berfungsi kembali baik dengan mengganti perangkat yang rusak ataupun dengan memperbaikinya, sehingga operasional tidak terhambat dan dapat berjalan normal kembali.

Dalam penelitian yang dilakukan di PT. Mitra Buana Koorporindo, aplikasi yang akan dibangun adalah berdasarkan layanan pemeliharaan yang dilakukan setelah *system breakdown* yang disebut *corrective maintenance* (CM). Dimana saat system breakdown terjadi atau terdapat kendala maupun kerusakan pada perangkat yang dalam manage support MBK, maka pelanggan akan melapor untuk kemudian aplikasi yang dibangun mengelola laporan tersebut hingga penyelesaiannya.

2.2.3. *Standar Level Agreement*

Service Level Agreement atau disingkat SLA jika diterjemahkan adalah "Perjanjian Tingkat Layanan". Pengertian SLA adalah bagian dari perjanjian layanan secara keseluruhan antara 2 dua entitas untuk peningkatan kinerja atau waktu pemenuhan tertentu selama masa kontrak. Dua entitas tersebut biasanya dikenal sebagai penyedia layanan dan pelanggan, dan dapat melibatkan perjanjian secara hukum karena melibatkan uang, atau kontrak lebih informal antara unit-unit bisnis internal.

SLA ini biasanya terdiri dari beberapa bagian yang mendefinisikan tanggung jawab berbagai pihak, dimana layanan tersebut bekerja dan memberikan garansi atau jaminan, dimana jaminan tersebut bagian dari SLA yang memiliki tingkat harapan yang disepakati, selain itu dalam SLA dimungkinkan terdapat tingkat ketersediaan, kemudahan layanan, kinerja, operasi atau tingkat spesifikasi untuk layanan itu sendiri. Selain itu, Perjanjian Tingkat Layanan akan menentukan target yang ideal, serta minimum yang dapat diterima.

Adapun pada penelitian ini, SLA yang menjadi acuan adalah SLA yang berkaitan dengan management waktu penyelesaian laporan kendala atau kerusakan-kerusakan perangkat seperti ditunjukan pada tabel 2.1 adalah contoh SLA untuk salah satu project di PT. Mitra Buana Koorporindo.

Tabel 2. 1 Contoh *Standar Level Agreement*

Severity	**Response Time**	**Resolve Time (non perangkat)**	**Pergantian Perangkat**		
			Jabodetabek	**Non Jabodetabek**	
Low	<= 15 Menit	36 jam	1 x 24 Jam	2 x 24 Jam	Regular GA*
Medium	< =15 Menit	16 Jam	1 x 24 Jam	2 x 24 Jam	Regular GA*
High	< =15 Menit	8 jam	1 x 24 Jam	2 x 24 Jam	Regular GA*
Critical	< =15 Menit	2 jam	1 x 24 Jam	2 x 24 Jam	Regular GA*

Berikut penjelasan klasifikasi prioritas yang dimaksud :

1. Priority Code 1 / Critical

   Insiden yang termasuk prioritas critical adalah insiden yang menyebabkan layanan IT terhenti / stop dan berdampak pada layanan bisnis dan menghentikan operasional bisnis. Contoh: Operating System (OS) / VM Down, Disk corrupt sehingga layanan tidak dapat di akses sama sekali.

2. Priority Code 2 / High

   Insiden yang masuk prioritas High adalah insiden yang berdampak pada beberapa bagian yang berhubungan operasional bisnis. Pegawai masih bisa menjalankan proses bisnis namun dengan kapasitas terbatas. Contoh: Service down yang menyebabkan banyak user yang tidak bisa mengakses layanan.

3. Priority Code 3 / Medium

   Insiden yang masuk prioritas Medium adalah insiden yang berdampak pada beberapa layanan bisnis namun tidak mengganggu operasional bisnis. Permasalahan ini tergolong menengah ke bawah, dan operasional bisnis tetap berjalan seperti biasa. Contoh: Over utilize (Processor, Memory), Services Down yang menyebanbkan ada sebagian layanan yang tidak bisa diakses.

4. Priority Code 4 / Low

   Insiden yang masuk prioritas Low adalah insiden yang tidak berdampak pada layanan IT yang sedang berjalan, seperti permintaan penyelesaian untuk masalah teknis rutin; seperti aplikasi, navigasi, instalasi atau konfigurasi; bug yang mempengaruhi sejumlah kecil pengguna. Contoh: Layanan Request, Backup Full yang tidak berdampak pada layanan yang sedang berjalan.

2.2.4. Android

Android merupakan sebuah sistem operasi berbasis linux yang dirancang untuk perangkat telepon seluler layar sentuh seperti smartphone dan tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi

mereka sendiri [13]. Android awalnya dikembangkan oleh Android, Inc. Dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007.

Sistem Android menggunakan database untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun device dimatikan. Sistem Android menggunakan SQLite untuk melakukan penyimpanan pada database, dimana SQLite tersebut merupakan suatu open source database yang banyak digunakan di berbagai perangkat keras berukuran kecil.

Android secara konsisten memberikan pembaruan terhadap perangkatnya hampir setiap tahun. Versi android dapat dilihat pada Tabel 2.2 Versi Android [14].

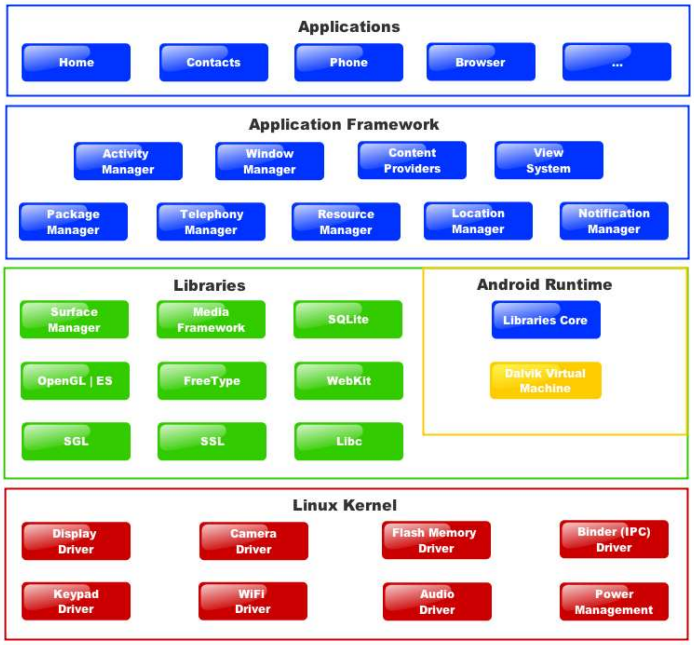
Tabel 2. 2 Versi Android

No	Versi	Tanggal Rilis
1	Apple Pie / Alpha (1.0)	September 2008
2	Banana Bread (1.1)	Febrari 2009
3	Cupcake (1.5)	April 2009
4	Donut (1.6)	September 2009
5	Eclair (2.0 – 2.1)	Desember 2009
6	Froyo : Frozen Yoghurt (2.2)	Mei 2010
7	Gingerbread (2.3)	Desember 2010
8	Honeycomb (3.0 – 3.2)	Februari 2011
9	Ice Cream Sandwich (4.0)	Oktober 2011
10	Jelly Bean (4.1 – 4.3)	Juli 2012
11	Kitkat (4.4)	Oktober 2013
12	Lolipop (5.0)	Juni 2014
13	Marshmallow (6.0)	Agustus 2015
14	Nougat (7.0)	Agustus 2016
15	Oreo (8.0)	Agustus 2017
16	Pie (9.0)	Maret 2018

Setiap aplikasi Android memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. API yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem itu sendiri.

2.2.4.1. Arsitektur Android

Android dibangun dengan berbagai macam arsitektur sebagai komponen yang terdapat pada perangkat tersebut adapun arsitektur dari android tersebut dapat ditunjukan seperti Gambar 2.4.



Gambar 2. 4 Arsitektur Android

Sumber gambar : http://www.eazytutz.com/android/android-architecture

Berikut ini adalah penjelasan dari arsitekur android :

1. *Application* ini adalah *layer* dimana kita terhubung dengan aplikasi saja, dimana biasanya kita mengunduh aplikasi kemudian kita melakukan instalasi dan menjalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk pelanggan, *email*, program SS, kalender, peta, *browser*, dan

kontak. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java*.

2. *Appllication frameworks*, pengembang memiliki akses penuh ke APIs *frameworks* yang sama digunakan oleh aplikasi inti sehingga dapat memberikan kesempatan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *services backgrounds*, mengatur alarm, dan menambahkan suatu *notifications*, dan lain sebagainya.
3. *Libraries*, merupakan *layer* dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernal, layer ini meliputi berbagai C/C++ inti seperti Libc dan SSL.
4. Android *Run Time*, *layer* yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Core Libraries* dan *Dalvik Virtual Machine*.
5. *Linux Kernel*, merupakan *layer* dimana inti *operating system* dari android itu berada. Berisi *file-file system* yang mengatur *system processing*, *memory, resource, drives*, dan sistem-sistem operasi android. *Kernel* juga bertindak sebagai lapisan abstraksi antara *hardware* dan *software stack.*

2.2.4.2. Komponen Aplikasi Android

Aplikasi android ditulis dalam bahasa pemrograman *Java*. Kode *java* di-*compile* bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" kedalam paket android sehingga menghasilkan file dengan ektensi .apk (dot apk). *File* apk ini yang disebut dengan aplikasi, dan kemudian dapat di instal di perangkat android. Komponen aplikasi android diantaranya:

1. ***Activities***

Suatu *activity* menyediakan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android mungkin hanya

terdapat satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi itu sendiri, satu *activity* biasanya dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan oleh pengguna. Untuk pindah dari satu *activity* ke *activity* lainnya, dapat dilakukan dengan *even*, misalnya klik tombol, memilih pilihan atau menggunakan *trigger* tertentu.

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam menjalankan musik, *service* mungkin menjalankan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya media *player* sedang memutar lagu dari *list* yang ada, aplikasi ini memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu, atau menulis pesan/*chat* dengan media *player* tetap berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

3. *Broadcast Reciever*

Broadcast reciever berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Sebagai contoh *broadcast* seperti notifikasi zona waktu berubah, baterai lemah, gambar telah selesai diambil oleh kamera, atau pengubahan referensi bahasa yang digunakan.

Broadcast receiver tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau getaran perangkat.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system* seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (*map*), atau aplikasi yang membutuhkan untuk mengakses data

kontak dan navigasi, maka inilah fungsi *Content Provider* untuk menyediakan data yang diperlukan oleh aplikasi.

2.2.4.3. Android Develovment Tools

Dalam proses merancang dan membangun sebuah aplikasi berbasis android tentu dibutuhkan beberapa *tools* pendukung, diantaranya :

1. **Android Software Development Kit (SDK)**

   Android SDK adalah *tools Application Programming Interface* (API) yang dibutuhkan untuk mengembangkan aplikasi android dengan menggunakan bahasa pemrograman *java*. Android SDK menyediakan *library* API dan *developer tools* yang digunakan untuk *build, test*, dan *debug* sebuah aplikasi android.

2. **Android Development Tools (ADT)**

   Android Development Tools (ADT) adalah *plugin* untuk *eclipse* yang menyediakan seperangkat alat yang terintegrasi dengan *eclipse* IDE (*Integrated Development Environment*). ADT memberikan akses ke banyak fitur sehingga mudah untuk membangun aplikasi android. Berikut ini adalah penjelasan fitur antara *eclipse* dan IDE.

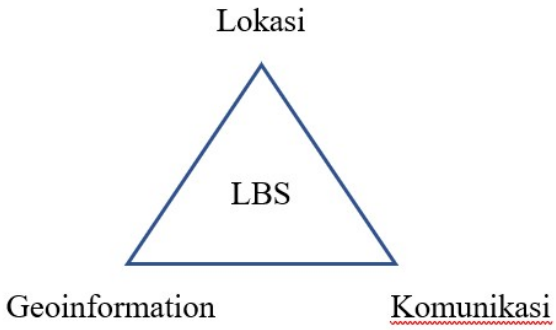
3. **Eclipse Integrated Development Environment (IDE)**

   Eclipse merupakan program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. *Eclipse* tersedia secara bebas untuk merancang dan mengembangkan aplikasi android. *Eclipse* merupakan IDE terpopuler dikalangan *developer* android, karena *eclipse* memiliki android *plugin* lengkap yang tersedia untuk mengembangkan aplikasi android, selain itu *eclips*e juga mendapat dukungan langsung dari *Google* untuk menjadi IDE pengembangan android, membuat *project* android dimana *source software* langsung dari situs resminya *Google*. Sampai saat ini *eclipse* memiliki 4 versi *package*, yaitu: *Indigo Package, Helios Package, Galileo Package, Ganymade Package*, dan *Europa Package*.

2.2.5. *Location Based Services*

Location Based Services adalah salah satu teknologi mobile yang dapat menyediakan informasi berdasarkan tempat yang berlandaskan pada Georaphic Information System (GIS) atau electronic map yang ditunjukkan oleh garis lintang dan bujur sehingga mendapatkan titik lokasi yang akurat [4].

Location Based Services didasarkan pada kebutuhan orang untuk mengorientasikan diri di lingkungan yang tidak diketahui. Layanan berbasis lokasi merupakan layanan komputer yang menyediakan fungsionalitas spesifik berdasarkan lokasi saat ini dari entitas seluler. Dalam beberapa kasus, entitas seluler adalah perangkat mobile yang lokasinya diidentifikasi dengan lokasi orang yang menggunakan perangkat, Dalam kasus lain, lokasi beberapa objek akan ditentukan dan dilacak, misalnya, dalam aplikasi manajemen armada atau logistik. Ada beberapa definisi layanan basis lokasi dalam literatur. Intinya, seperti yang akan kita amati dalam contoh berikut, layanan berbasis lokasi mengandung tiga aspek: lokasi, komunikasi, dan geoinformasi [15].



Gambar 2. 5 Aspek LBS

Sumber gambar : Buku Indoor Location-Based Services [15]

2.2.5.1. Unsur-Unsur Location Based Services

Adapaun unsur-unsur utama LBS adalah sebagai berikut :

1. *Location Manager* (API *Maps*): Menyediakan *tools/source* untuk LBS, *Aplication Programming Interface* (API) *maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta *feature* lainnya seperti tampilan

satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada com.google.android.maps.

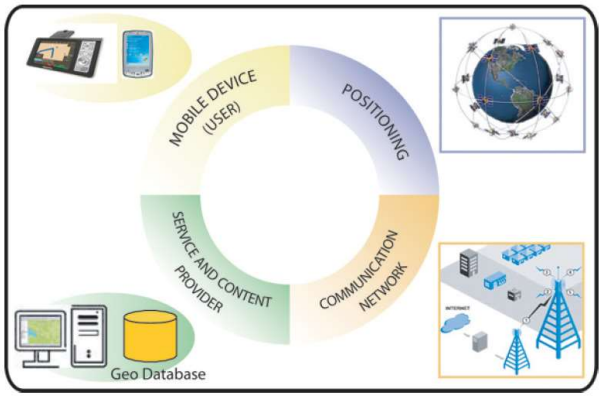
2. *Location Providers* (API *location*): Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API *location* berada pada paket android yaitu *android.location*. Dengan *location manager*, kita dapat menentukan lokasi kita saat ini dan rute menuju tempat tertentu.

2.2.5.2. Komponen-Komponen Location Based Services

Penggunaan *Location Based Services* (LBS) juga memerlukan beberapa komponen yang terbagi menjadi 5 elemen, diantaranya [16]:

1. *Mobile Device* yaitu sebuah alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Perangkat yang memungkinkan yaitu perangkat yang memiliki fasilitas navigasi PDA, *mobile phone*, laptop, dan lainnya.
2. *Communication Network* adalah jaringan selular yang mengirimkan data pengguna dan permintaan layanan.
3. *Positioning Component*, biasanya posisi pengguna harus ditentukan untuk pengolahan layanan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS).
4. *Services and Application Provider*, adalah penyedia layanan pengguna selular yang bertanggung jawab untuk memproses layanan.
5. *Data and Content Provider*, yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna.

   Kelima komponen tersebut dapat ditunjukan pada Gambar 2. 6 :



Gambar 2. 6 Komponen LBS

Sumber gambar : Jurnal Aplikasi Pencarian Rute Perguruan Tinggi Berbasis Android Menggunakan Location Based Services di Kota Semarang [16]

2.2.5.3. Kategori Location Based Services

Secara garis besar jenis *Location Based Services* dapat dibagi menjadi dua bagian yaitu:

1. *Pull Services:* Layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi.
2. *Push Service*: Layanan ini diberikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan.

2.2.6. Global Positioning System

Global Posotioning System atau yang sering disingkat dan dikenal dengan GPS adalah system navigasi yang menggunaan satelit yang didesain agar dapat menyediakan data posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat dipermukaan bumi, setiap saat dan dalam kondisi cuaca apapun. Penerimaan sinyal satelit sering terbatas, oleh karena itu alternatif untuk mengatasinya yaitu berdasarkan pancaran transmisi dari stasiun pemancar atau Menara BTS terdekat [17].

Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (longitude atau latitude) [18].

1. Titik Kordinat

   Titik koordinat adalah titik yang berpedoman pada garis latitude dan longitude suatu daerah. Kaitannya dengan latitude dan longitude adalah, kedua garis lintang dan bujur inilah (latitude = garis lintang, longitude = garis bujur) yang menentukan di perolehnya suatu nilai derajat dari suatu titik yang diukur. Koordinat pada umumnya dituliskan dengan format DD (Degree Decimal), terkadang juga dituliskan dengan format DMS (Degree-Minute-Second).

2. Konversi Derajat Bujur dan Lintang

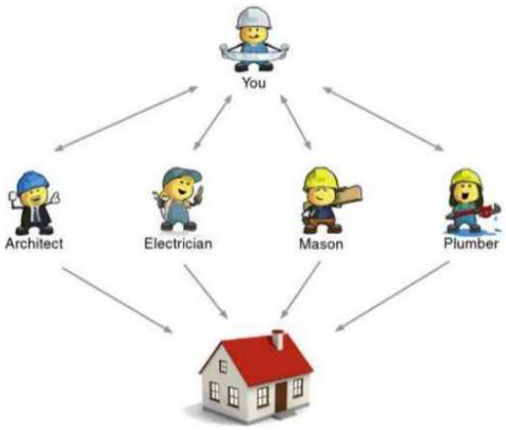
   Perlu diketahui pula konversi derajat menit („) dan detik (“) pada letak astronomis suatu tempat dalam satuan jarak (km dan m). Dalam pemakaian GPS sering didapatkan titik koordinat suatu posisi lokasi dalam bentuk Latitude (Lintang), Longitude (Bujur) atau Degree (Derajat). Dalam hal ini ada suatu formula yang digunakan untuk mengkonversi bilangan tersebut dari bentuk Degree Minute Second (DMS) menjadi Degree Decimal (DD), atau sebaliknya konversi Degree Decimal (DD) menjadi Degree Minute Second (DMS)

Penggunaan GPS untuk penentuan posisi saat ini diantaranya untuk kegiatan-kegiatan pribadi maupun kelompok seperti pergi hiking, pelayaran, berburu, atau pun dijadikan sebagai penunjuk jalan saat mengemudi, dan lain sebagainya. Contoh lain adapalah penggunaan untuk navigasi kapal, fleet tracking, pengendalian mesin, Teknik sipil, survey daratan, GIS dan pemetaan, analisis informasi dan masih banyak lagi yang lainnya termasuk seperti kegunaan-kegunaan pada penelitian lain yang sebelumnya sudah dipaparkan pada pendahuluan.

Program GPS dan operasionalnya saat ini sebagai besar bersumber pada Department of Defense (DoD) Amerika Serikat yang dapat dikatakan merupakan pembuat dan tempat lahirnya system ini. Manajemennya sendiri dilaksanakan oleh US Aid Force dengan panduan dari komite eksekutif DoD Positioning/Navigasi Komite ini menerima eksekutif oleh Positioning/Navigastion. Komite ini menerima masukan dari komite yang sama dari Department of Transportation (DoT) yang bertindak sebagai suara sipil untuk urusan aturan GPS (NAPA 1995).

2.2.7. Application Programming Interface

Application Programming Interface (API) merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan *software* yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari *software* tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut [19].



Gambar 2. 7 Analogi API

Sumber gambar : Jurnal Teknologi Terpadu [19]

Dalam API terdapat fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*. Keuntungan menggunakan API diantaranya:

1. Probabilitas, API dapat digunakan untuk bahasa pemrograman ataupun untuk sistem operasi mana saja asalkan paket-paket API sudah terpasang.

2. Lebih mudah dimengerti, API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal *editing* dan pengembangan.
3. Mudah dikembangkan, dengan adanya API memudahkan *programmer* untuk mengembangkan suatu sistem.

2.2.8. Google Maps

Seperti yang tercatat oleh Svennerberg (Beginning Google Maps API 3), Google Maps adalah yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei tahun 2010 ini menyatakan bahwa 43% mashup (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps. Beberapa tujuan dari penggunaan Google Maps adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan Google Maps [20].

Dalam pembangunan aplikasi dengan *Google Maps API* dapat menggunakan urutan sebagai berikut:

1. Memasukkan *Maps API JavaScrip*t ke dalam HTML kita.
2. Membuat *element div* dengan nama *map_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Meng-inisiasi peta dalam *tag body* HTML dengan *event onload*.

Gambar 2. 8 Google Maps

Sumber gambar : https://www.google.com/maps

2.2.9. Mapbox

Mapbox adalah sebuah *platform* pemetaan *open source* yang bekerja dan merilis sebagai kode sebanyak mungkin. Sebagian besar data *Mapbox* menggunakan bantuan serta berinvestasi pada berbagai macam sumber data misalnya *OpenStreetMap*, USGS, *Landsat*, dan *OpenAddresses* [21]. *Mapbox* mendukung berbagai macam aplikasi yang akan digunakan oleh penggunanya, baik *mobile* maupun *online*. Produk yang tersedia di *Mapbox* terdiri dari peta, satelit, server atlas, *geocoding*.

Mapbox mendukung beberapa aplikasi pengembang, diantarannya *Javascript*, iOS, Android dan API. *Mapbox* telah digunakan untuk aplikasi *Foursquare*, *Pinterest* dan *Evernote* yang memudahkan penggunanya untuk menandai lokasi mereka kapanpun dan dimanapun. Pengguna *Mapbox* yang akan mendaftar disediakan berbagai pilihan akses data dengan berbagai pilihan biaya yang tentu saja mempengaruhi keberagaman fasilitas yang didapat oleh pengguna.

Gambar 2. 9 Mapbox

Sumber gambar: https://blog.mapbox.com

2.2.10. Geocoding

Geocoding adalah salah satu teknologi yang merupakan proses untuk mencari koordiat geografis (*latitude* dan *longitude* atau garis lintang dan bujur) dari alamat tertentu (nama jalan, nama kota, kode pos, nama negara). Kebalikan dari *geocoding* adalah *reverse geocoding*, yaitu mencari data geografis berdasarkan koordinat geografis [16]. Jika ada suatu koordinat yang dimasukan lalu akan muncul data geografis yang ada pada koordinat itu seperti nama jalan, kode pos, negara, kota, provinsi.

Geocoding diartikan sebagai proses penyimpanan identifikasi lokasi menjadi bagian dari *record*. Jika data telah di *geocoding* berarti pengidentifikasian

telah ditambahkan pada *record* ke lokasi pada peta [19]. *Geocoding* sebenarnya adalah proses penggabungan dua tabel dimana kedua tabel tersebut dapat dilakukan penggabungan karena memiliki *field* yang sama.

2.2.11. Mobile Reporting

Perkembangan teknologi informasi memberikan banyak terobosan-terobosan baru bagi dunia terlebih dalam dunia bisnis. Gambaran yang nyata diantaranya terobosan baru dalam pelayanan pemeliharaan perangkat yang semula konvensional kini dapat dilakukan dengan mobile reporting.

Dalam penelitian ini mobile reporting yang dimaksud adalah proses dimana pelanggan dapat melakukan pelaporan kendala atau kerusakan pada perangkat yang digunakan, melakukan penyesuaian jadwal bertemu, hingga memperoleh informasi yang akurat terkait progress penanganan laporannya, dan kemudian melakukan validasi laporan dapat terselesaikan dengan baik dari manapun dan kapanpun menggunakan perangkat mobile. Aplikasi mobile reporting terkait *maintenance service* ini berguna untuk mempermudah pelaporan kendala atau kerusakan yang terjadi serta mempercepat penanganannya.

Pada saat sekarang ini mulai digunakannya aplikasi content pada dunia bisnis, yang menuntut akan penggunaan teknologi penyedia layanan content yang aman dan reliable. Teknologi yang aman akan menciptakan peluang-peluang baru dalam dunia bisnis. Dengan mobile reporting ini dapat melakukan pelaporan kendala atau kerusakan secara online. Hal ini sangat membantu pengguna dalam melakukan pelaporan kerusakan dan melakukan pengelolaan *maintenance service* yang dimaksud, karena layanan ini sangat praktis dan dapat menghemat waktu.

2.2.12. Barcode

Barcode adalah sekumpulan kode yang berbentuk garis yang masing-masing ketebalan garisnya berbeda sesuai dengan karakter yang diwakilkannya. Barcode ini biasanya akan ditempelkan di label dan kemudian ditempelkan di produk yang akan diwakilinya atau dicetak langsung pada produknya untuk mengidentifikasikan produk tersebut. Informasi yang terdapat barcode tersebut

dapat berupa nomor seri, nomor model, kode produksi, nomor identitas dan lain-lainnya sehingga dapat dengan mudah dan cepat diidentifikasikan oleh sistem komputer.

Untuk dapat membaca dan menterjemahkan kode-kode batang ini ke karakter yang dikenal oleh manusia atau sistem komputer, kita memerlukan alat yang biasanya kita sebut dengan Barcode Scanner. Saat ini juga terdapat banyak aplikasi smartphone (ponsel pintar) yang dapat membaca atau menterjemahkan barcode ini melalui kameranya. Seiring dengan perkembangannya, banyak jenis barcode atau kode batang yang tersedia di pasaran. Bahkan barcode yang kita kenal saat ini tidak hanya berbentuk batang, namun juga berbentuk kotak-kotak kecil yang umumnya berbentuk bujursangkar.

Barcode dapat diklasifikasikan menjadi dua kategori utama yaitu barcode 1 dimensi (1D) dan barcode 2 dimensi (2D). Dalam laporan ini yang akan digunakan dan dibahas adalah barcode 1D dan barcode 2D *(QR-Code)*. Berikut ini adalah jenis-jenis barcode 1D yang sering ditemukan di berbagai produk di sekitar kita :

1. **Code 39**

   Barcode jenis Code 39 adalah jenis barcode alfanumerik tertua yang hingga saat ini masih digunakan oleh industri manufaktur, kesehatan, industry otomotif, pertahanan dan pemerintahan. Code 39 ini dapat menyandikan angka 0 hingga 9 dan huruf A hingga Z serta beberapa simbol khusus seperti spasi, -, $, /, +, %.

2. **Code 128**

   Kode 128 adalah barcode linier yang digunakan oleh industri pengiriman dan pengemasan di seluruh dunia ini. Kode ini dirancang agar sangat ringkas dan menggunakan jumlah ruang yang paling sedikit dari semua jenis barcode satu dimensi (1D barcode). Kode 128 ini mencakup angka 0-9, huruf A-Z (huruf besar dan kecil) dan semua simbol ASCII standar serta beberapa karakter khusus.

3. **Interleaved 2 of 5**

   Kode Batang jenis Interleaved 2 of 5 atau ITF barcode ini umumnya digunakan untuk memberikan label pada bahan kemasan, karena barcode jenis

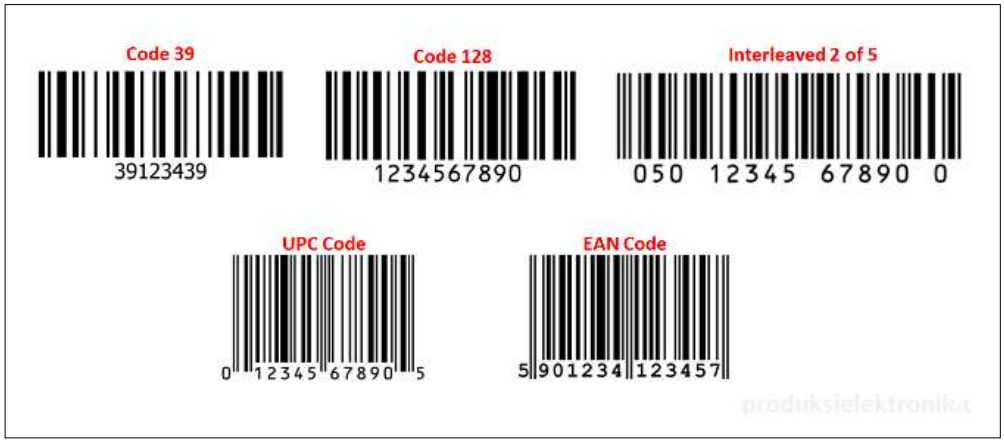
ini memiliki toleransi yang tinggi dan cocok digunakan pada karton yagn bergelombang. Kode Interleaved 2 of 5 adalah barcode yang hanya terdiri dari numerik yang digunakan untuk menyandikan pasangan angka. Setiap dua digit dipasangkan untuk membuat satu simbol. Jumlah digit yang digunakan harus genap agar format ini berfungsi, jadi nol biasanya ditambahkan pada akhir set angka ganjil.

4. Universal Product Codes (UPC)

Barcode UPC dapat kita temukan di hampir setiap produk ritel. Barcode ini awalnya dibuat untuk toko grosir untuk menyediakan pencetakan tanda terima dan pelacakan inventaris yang cepat. Setelah mendapatkan nomor UPC, pabrikan akan menerima nomor perusahaan unik untuk digabungkan dengan nomor produk masing-masing.

5. International Article Number (EAN)

Barcode EAN pada dasarnya adalah superset dari UPC, barcode ini digunakan secara khusus oleh penjual buku, perpustakaan, universitas dan grosir untuk penelusuran buku. 13 kode digit ini dibuat dari Nomor Buku Standar Internasional (ISBN) untuk setiap buku yang dilacak. Seperti UPC, ini distandarisasi untuk identifikasi unik penerbit.

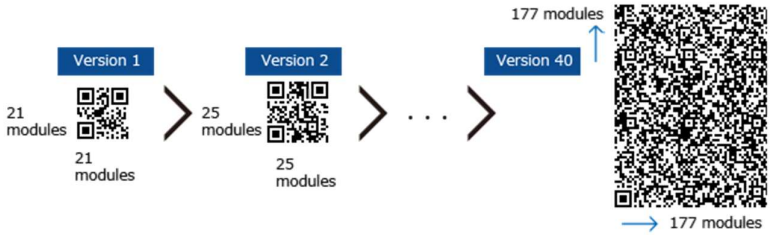


Gambar 2. 10 Jenis-Jenis Barcode 1D

2.2.13. QR-Code

QR code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak kedalam suatu media yang ringkas. QR adalah singkatan dari *Quick Response* karena ditujukan untuk diterjemahkan isinya dengan cepat. *QR code* merupakan pengembangan dari barcode satu dimensi, *QR code* salah satu tipe dari *barcode* yang dapat dibaca menggunakan kamera *handphone.* QR Code adalah symbol matriks dua dimensi yang terdiri dari sebuat untaian kotak persegi yang disusun dalam sebuah pola persegi yang lebih besar. Kotak persegi ini kemudian disebut modul, luasnya pola persegi menentukan versi dari QR-Code [22]. *QR code* mampu menyimpan semua jenis data, seperti data angka/numerik, *alphanumeric*, biner, kanji/kana. Selain itu *QR code* memiliki tampilan yang lebih kecil dari pada *barcode*. Hal ini dikarenakan *QR code* mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari tampilannya, gambar *QR code* bisa hanya sepersepuluh dari ukuran sebuah barcode [23].

Versi simbol *QR code* berkisar dari versi 1 ke versi 40. Setiap versi memiliki konfigurasi modul yang berbeda atau jumlah modul (Modul ini mengacu pada titik-titik hitam dan putih yang membentuk *QR code*). Konfigurasi modul mengacu pada jumlah modul yang terkandung dalam simbol, dimulai dengan versi 1 (21 x 21 modul) sampai ke versi 40 (177 x 177 modul). Setiap nomor versi lebih tinggi terdiri dari 4 modul tambahan per samping. Setiap versi simbol *QR code* memiliki kapasitas data yang sesuai dengan jumlah data, jenis karakter dan tingkat kesalahan koreksi. Untuk itu pemeriksaan data dengan kapasitas maksimum ditentukan pada setiap versinya. Untuk versi dan kapasitas data maksimum, maka jumlah data dan modul akan meningkat sehingga simbol *QR code* semakin besar. Penggunaan *QR code* dalam penelitian ini dimanfaatkan untuk proses validasi status laporan, dimana tiket laporan yang dihandle oleh teknisi dianggap selesai jika teknisi sudah ketemu dengan pelanggan dan menscan barcode tertentu pada halaman aplikasi pelanggan.



Gambar 2. 11 Versi Simbol QR Code

Sumber gambar: https://www.qrcode.com

2.2.14. PHP

PHP merupakan suatu bahasa pemrograman sisi server yang dapat kita gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page* (ASP) dan *Java Server Page* (JSP). Dalam suatu halaman HTML kita dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi [24].

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (Form Interpreted). Pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah Personal Home Page/Form Interpreter. Dengan pelepasan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C.

PHP merupakan produk *open source* sehingga kita dapat mengakses *source code*, menggunakan, dan mengubahnya. Karena kekayaan akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki tingkat popularitas yang cukup tinggi dikalangan *developer*.

```
<?php
 $a = 1;
 $b = “Ini Adalah Syntax Php”;
 $c = 27,0895;
      Echo $b;
?>
```

2.2.15. Laravel

Laravel yaitu framework PHP open source yang dibuat dengan model view controller dan dibekali dengan berbagai macam sintaks. Diluncurkan pada tahun 2011, sejak itu dalam membuat website dengan Laravel banyak digemari oleh berbagai komunitas programmer di Github, sebelum kemudian menyebar ke seluruh dunia. Framework ini menyediakan beberapa jenis PHP library dan beberapa fungsi lain yang bisa mempermudah dalam menulis baris kode. Framework Laravel dibuat dengan tujuan memperindah cara untuk membuat website, terkenal sederhana dan elegan karena pembuatannya memang ditujukan untuk end-user. Selain itu framework ini terkenal dengan dokumentasinya yang sangat lengkap dan selalu diperbarui [25].

Adapun tools andalan yang dimiliki Laravel, yaitu sebagai berikut :

1. Composer

   Composer merupakan tool yang di dalamnya terdapat dependencies dan kumpulan library. Seluruh dependencies disimpan menggunakan format file composer.json sehingga dapat ditempatkan di dalam folder utama website.

2. Artisan

   Artisan merupakan command line interface yang dimiliki oleh Laravel. Artisan mencakup sekumpulan perintah yang membantu untuk membangun sebuah website atau aplikasi web.

2.2.16. Flutter

Flutter hadir menjadi bagian dari salah satu framework open-source yang menyemarakkan dunia pemrograman, terkhusus dibidang mobile development. Flutter dikembangkan oleh Google sebagai framework untuk membuat UI (User

Interface) android dan iOS sekaligus dalam satu frame yang sama dengan menggunakan bahasa pemrograman Dart.

Ada 3 hal yang menjadi poin penting yang ditawarkan oleh Flutter kepada para penggunanya, yakni:

1. Fast Development

Pada web programming proses penerjemahan setelah code diketikkan, kemudian di-reload maka hasilnya dapat dilihat karena menggunakan sistem interpreter. Berbeda halnya, dalam desktop / mobile development, karena menggunakan sistem compiler maka perubahan dari code program yang kita punya, harus di-compile terlebih dahulu sebelum dapat ditampilkan. Flutter menawarkan fitur Hot Reload yang di-claim dalam hitungan detik kamu sudah dapat melihat terjemahan code program yang telah kamu buat.

2. Expressive and Flexible UI

Secara default, Flutter menawarkan komponen UI (User Interface) yang menarik dengan proses rendering yang cepat.

3. Native Performance

Flutter juga menyediakan widgets yang dapat digunakan guna menunjang proses development agar menjadi lebih efisien, seperti scrolling, navigation, icon, dan font. Namun tetap memberikan performance serasa native, baik untuk versi Android maupun iOS.

2.2.17. Unified Modeling Language

Desain sistem merupakan salah satu hal penting dan sangat berguna untuk membuat suatu program. Tidak heran jika banyak perusahaan-perusahaan yang sangat membutuhkannya untuk suatu aplikasi atau project yang sedang mereka kerjakan. Untuk memenuhi kebutuhan tersebut, bisa menggunakan Unified Modeling Language atau yang disingkat dengan UML.

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk system atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang rumit/kompleks sedemikian rupa agar menjadi lebih mudah

dipahami dan dipelajari [26]. Sebagai contoh, alih-alih membawa kapal selam ke meja kita, kita mungkin akan membawa replika miniature kapal beserta buku manualnya ke meja kita untuk dipromosikan kepada calon pembeli. Adapun tujuan pemodelan (dalam kerangka pembangunan system/perangkat lunak aplikasi) adalah sebagai sarana analisi, pemahaman, visalisasi, dan komunikasi antar yang membangun perangkat lunak aplikasi (jika pembangunan dikerjakan secara tim/banyak anggota) maupun dengan penggunanya, serta sebagai sarana dokumentasi agar bisa menelaah secara seksama hingga melakukan pengujian terhadap aplikasi yang dibangun.

Berikut bagian utama Unified Modeling Language :

2.2.17.1. View

Bagian yang pertama yaitu view. Bagian ini berguna untuk melihat sebuah sistem yang telah dimodelkan dari sejumlah aspek berbeda. Dengan kata lan, view merupakan suatu abstraksi dan berisi beberapa diagram. Jenis view yang ada di dalam UML yaitu logical view, use case view, concurrency view, serta deployment view.

1. Logical View

   Logical view digunakan untuk mendeskripsikan fungsionalitas dari struktur statis (class, relationship, dan juga object), sistem, dan juga kolaborasi dinamis ketika object mengirim sebuah pesan ke object yang lain di dalam fungsi tertentu. Logical view ini digambarkan di dalam class diagrams. Selan itu, logical view cocok digunakan bagi para designer (perancang) dna developer (pengembang).

2. Use Case View

   Use case view digunakan untuk mendeskripsikan fungsionalitas sebuah sistem yang harus dilakukan sesuai apa yang diinginkan oleh eternal actors. Actor yang melaukan interaksi degan sebuah sistem bisa berupa user maupun sistem lainnya. Use case view digunakan oleh designer, pelanggan, developer, dan juga penguji sistem.

3. Concurrency View

Concurrency view digambarkan di dalam 2 diagram yiatu diagram dinamis (squence, state, activity diagrams, dan juga collaboration) dan juga diagram implementasi (deployment dan component diagram). Selain itu, Concurrency view membagi sistem ke proses dan prosesor. View ini digunakan untuk para developer,integrator (pengintegrasi), serta penguji.

4. Deployment View

Deployment view merupakan view yang mendeskripsikan fisik suatu sistem seperti komputer serta perangkat (nodes) dan juga bagaimana hubungannya dengan yang lain. Deployment view digunakan bagi pengembang(developer), penguji, dan juga pengintegrasi serta dideskripsikan dalam deployment diagram.

2.2.17.2. Diagram

Diagram dengan bentuk grafik menunjukkan simbol suatu elemen model dan disusun untuk menggambarkan aspek atau bagian tertentu. Sebuah diagram adalah bagian dari sebuah view tertentu, saat digambarkan umumnya dialokasikan bagi view tertentu. Untuk jenisnya, ada beberapa jenis diagram yaitu:

1. Class Diagram

Class diagram merupakan sebuah diagram yang mengilustrasikan struktur class yang statis (statis class) dalam sebuah sistem. Class sendiri merepresentasikan suatu hal yang ditangani sistem dan bisa berhubungan satu sama lain dengan berbagai cara seperti associated, specialed, dependent, dan juga package. Sebuah sistem pada umumnya memiliki sejumlah class diagram.

2. Use Case Diagram

Use case diagram menggambarkan beberapa external actors serta hubungannya dengan use case (deskripsi fungsi, disediakan sistem dengan bentuk teks) Use case dideskripsikan hanya yang terlihat oleh actor dari luar.

3. Squence Diagram

Squence diagram adalah diagram yang menggambarkan kolaborasi yang dinamis antara beberapa object. Kegunaannya yaitu menunjukkan serangkaian pesan antara interaksi object yang dikirim.

4. Activity Diagram

Activity diagram menggambarkan serangkaian aliran aktivitas serta digunakan untuk menggambarkan aktivitas yang dibentuk ke dalam suatu operasi.