BAB 2
TINJAUAN PUSTAKA

2.1 Ruang Linkup Penelitian

Objek penelitian yang sedang diteliti adalah Zoom Pet Care yang berada di Jl. Golf Barat Raya No 22,Arcamanik. Bandung

2.1.1 Logo Zoom Pet Care



Gambar 2-1 Logo Zoom Pet Care

2.1.2 Sejarah Zoom Pet Care

Di latar belakangi hobi terhadap hewan peliharaan terutama kucing dan anjing, pada tanggal 8 Agustus 2004 Zoom Pet Care membuka Toko Pet Shop di Kota Bandung, tepatnya di Jl. Golf Barat Raya No 22. Toko Zoom Pet care awalnya hanya berukuran 250 x 3m2, lokasinya merupakan bekas pantry dan dimanfaatkan sebagai Pet Shop. Tahun 2006 Zoom Pet Care membuka cabang di Cikapayang Dago No. 7

Tahun 2008 Zoom membuat Cattery dengan nama ZoomCattery yang membiakkan kucing ras dengan ras sebagai berikut : Persia,Exotic,Scottish Fold, Sphynx, Abyssinian, Bengal, Dan lain – lainnya.

Tahun 2011 Zoom Membuka Klinik Veterinarian untuk meningkatkan pelayanan terhadap perawatan dan kesehatan kucing pelanggan yang berjalan dengan baik sesuai dengan harapan Zoom Pet Care

Tahun 2015 Zoom Pet Care membuka cabang kedua di Festival CityLink Mall yang berlokasi di Jl. Peta No. 241.

Seiring berkembangnya dunia usaha Pet Shop Zoom mengembangkan diri dalam Pet Boutique dengan brand WOOZ, yang sudah tersebar di seluruh nusantara Indonesia.

Zoom juga dipercaya utuk menjadi distributor utama Brand – brand tertentu, baik untuk Brand Pet Foods, Brands Supplements Vitamin dan beberapa produk lainnya. Dan semoga Zoom Pet care dapat memenuhi kebutuhan konsumen.

2.1.3 Struktur Organisasi Zoom Pet Care

Struktur organisasi pada Zoom Pet Care dapat dilihat pada gambar 2-2 sebagai berikut,



Gambar 2-2 Struktur Organisasi Zoom Pet Care Arcamanik

2.1.4 Deskripsi Tugas

Deskripsi Tugas adalah suatu penjelasan mengenai tugas-tugas dari masing masing jabatan yang ada pada Zoom Pet Care. Berikut adalah deskripsi tugas pada Zoom Pet Care :

Tabel 2-1 Tabel Deskripsi Tugas

Jabatan	Deskripsi Tugas
Pemilik	1. Mengontrol dan mengevaluasi kinerja dari semua toko Zoom Pet Care. 2. Menerima Laporan dari toko.
Penanggung Jawab Toko	1. Bertanggung jawab atas pengembangan bisnis toko. 2. Mengontrol dan mengevaluasi kinerja toko dan karyawan. 3. Membuat laporan penjualan. 4. Bertanggung jawab dalam transaksi penjualan barang dan layanan Grooming.
Staff Grooming	1. Bertanggung Jawab atas hewan dalam layanan Grooming. 2. Melakukan tugas Grooming terhadap
Staff Antar Jemput	Melakukan tugas pengantar jemput hewan.

2.2 Landasan Teori

Landasan teori memberikan gambaran dan teori terkait dengan pembangunan system. Landasan teori akan membahas pengertian Hewan Peliharaan, Grooming, Location Based Service, Global Positioning System, Android, Firebase, Web Service, JSON, Unified Modelling Language, Flow Map, serta *tools* yang digunakan.

2.2.1 Hewan Peliharaan

Hewan peliharaan adalah hewan yang hidup secara dalam lingkungan tertentu, tidak bebas, mereka tumbuh, bergerak, mencari makan dan berkembang biak dengan bantuan manusia secara langsung maupun tidak langsung. Hewan peliharaan dipelihara oleh manusia. Manusia memelihara hewan untuk bebagai macam kepentingan, mulai dari hobi atau kesenangan, mencari keuntungan (sebagai salah satu kegiatan ekonomi), dan melindungi agar tidak punah [1].

Hewan peliharaan sering di anggap sebagai teman atau sahabat oleh pemiliknya, oleh karena itu pemilik hewan selalu memberikan kasih sayang, merawat, dan menjaga hewan peliharaannya. Pemilik hewan di kota Bandung untuk saat ini berjumlah 1.268 anggota sesuai dengan jumlah anggota pada Komunitas Pecinta Kucing Bandung dan 310 anggota pada Komunitas Pecinta Anjing Ras Bandung Selatan.

2.2.2 Layanan Antar Jemput

Layanan berasal dari kata layan yang dalam kamus bahasa Indonesia adalah membantu menyiapkan(mengurus) apa-apa yang diperlukan seseorang, meladeni. Antar dalam kamus bahasa Indonesia adalah membawa sesuatu untuk di berikan kepada seseorang. Jemput dalam kamus bahasa Indonesia adalah pergi mengambil (menjemput). Jadi dapat di simpulkan bahwa layanan antar jemput adalah membantu seseorang untuk membawa sesuatu kepada pengirim dan mengembalikannya kembali.

Layanan antar jemput hewan peliharaan merupakan suatu layanan yang dapat membantu pemilik hewan membawa ke penyedia layanan Grooming dan mengembalikannya ke pemilik hewannya.

2.2.3 Grooming

Perawatan hewan peliharaan sering di sebut *Grooming. Groom* dalam kamus Bahasa Inggris Indonesia artinya adalah mengurus, merawat, rapi atau pelihara. Pemilik hewan peliharaan dapat menggunakan teknik *Grooming* untuk

membantu mempertahankan dan meningkatkan kesehatan hewan peliharaan yang di milikinya [2].

Para pemilik hewan juga sering menggunakan layanan perawatan untuk hewan peliharaannya seperti *Grooming*. Keberadaan layanan *Grooming* menjadi penting di karenakan para pemilik hewan peliharaan membutuhkan layanan ini untuk merawat hewan peliharaannya, dikarenakan kesibukan yang di miliki oleh para pemilik sehingga tidak memiliki waktu untuk merawat hewan peliharaannya.

2.3 Location Based Service

Layanan berbasis lokasi atau Location Based Service adalah layanan informasi yang dapat diakses melalui mobile device dengan menggunakan mobile network, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari mobile device tersebut. LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Location Based Service merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut [5] .Location Based Service dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau objek tertentu, seperti menemukan lokasi tempat yang diinginkan terdekat atau lokasi lainnya [6]. Teknologi Location Based Service (LBS) merupakan salah satu bagian dari implementasi mobile GIS yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial dibandung fungsionalitas pada teknologi GIS populer untuk Field Based GIS [7].

Location based service dapat di gunakan oleh pengembang aplikasi android untuk mendapatkan informasi lokasi dari pengguna yang butuhkan untuk aplikasi tersebut.

2.3.1 Unsur Unsur Location Based Service

Terdapat dua unsur utama dari Location Based Service adalah :

1) Location Manager (API Maps)

Menyediakan tools/source untuk LBS, Aplication Programming Interface (API) Maps menyediakan fasilitas untuk menampilkan peta beserta feature lainnya seperti tampilan satelit, street (jalan) maupun gabunganya. Paket ini berada pada com.google.android.maps.[6]

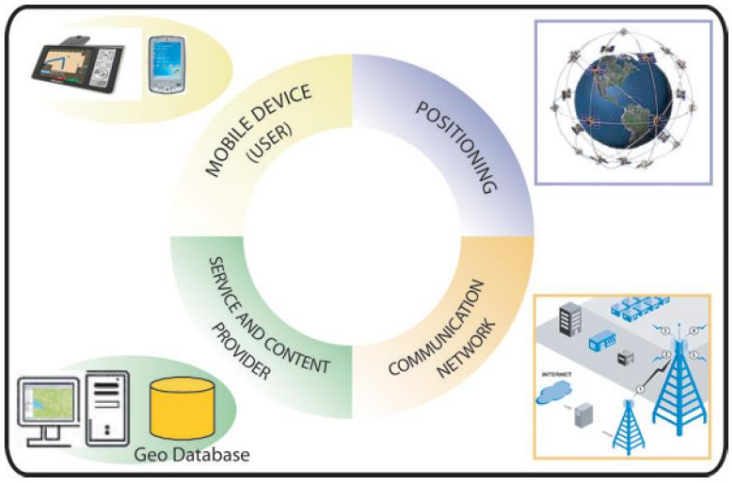
2) Location providers (API Location)

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.[6]

2.3.2 Komponen Location Based Service

Dalam menggunakan Location Based Service terdapat beberapa elemen yang diperlukan antara lain :

3) *Mobile Devices* yaitu sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan. Biasanya perangkat yang memungkinkan yaitu PDA, *Mobile Phone, Laptop,* dan perangkat lainnya yang mempunyai fasilitas navigasi [5[.
4) *Communication Network* adalah jaringan selular yang mengirimkan data pengguna dan permintaan layanan [5].
5) *Positioning Component* untuk pengolahan layanan biasanya posisi pengguna harus ditentukan. Posisi pengguna dapat menggunakan *Global Positioning System* (GPS) [5].
6) *Service and Application Provider* adalah penyedia layanan pengguna selular yang bertanggung jawab untuk memproses layanan [5].
7) *Data and Content Provider* yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna [5].



Gambar 2-3 Komponen Dasar Location Based Service

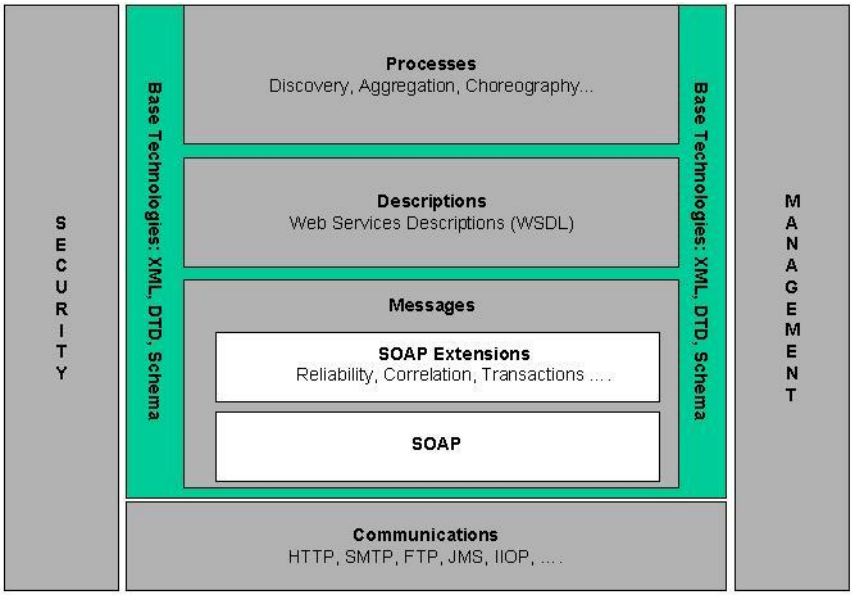
(sumber : Juwita Imaniar, "Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android")[8]

2.4 Web Service

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen. Tujuan web service adalah menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standar protokol yang ditetapkan oleh web service [9]. Web Service digunakan pada pembangunan aplikasi ini untuk mengolah dan menjembatani komunikasi antar program dan juga basis data.

2.4.1 Arsitektur Web Service

Arsitektur web service dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Salah satu cara untuk menggambarkan arsitektur web service seperti pada gambar 2-4



Gambar 2-4 Arsitektur Web Service

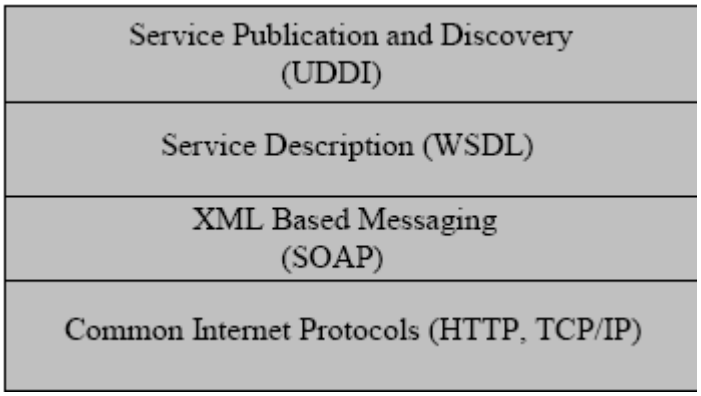
Web service disusun oleh tiga komponen standar, yaitu : [9]

a. Simple Object Access Protocol (SOAP), yaitu protokol yang bertanggungjawab dalam pertukaran informasi dalam lingkungan jaringan terdistribusi.
b. Web Service Definition Langguage (WSDL), dokumen standar dituliskan dalam format XML, dan mendefinisikan kehadiran web service dalam suatu jaringan.
c. Universal Description, Discovery, and Integration (UDDI), yaitu suatu lokasi direktori yang berisikan service (layanan) dan bersifat bebas platform (*Platform Independent*), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam dan luar jaringan.

2.4.2 Komponen Web Service

Web service memiliki empat komponen yaitu :

a. Layer 1 : Protokol Internet standar (HTTP,TCP/IP)
b. Layer 2 : *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis *XML* yang digunakan untuk proses pertukaran data/informasi antar layanan.
c. Layer 3 : *Web Service Definition Language* (WSDL), merupakan suatu bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
d. Layer 4 : *Universal Description, Discovery, and Integration* (UDDI), merupakan lokasi direktori berisikan service dan bersifat bebas platform.



Gambar 2-5 Komponen Web Service

(Sumber : pembangunan perangkat lunak, jardiknas)[10]

2.4.3 Keamanan Pada Web Service

Web Service bersifat open standard (standar pada protokol universal dan diketahui semua pihak) dan belum memiliki sistem penanganan keamanan[11].

Untuk issue keamanan ditangani secara khusus dan berlapis-lapis, antara lain dengan cara [9]:

a. Enkripsi pada data XML yang dipertukarkan saat komunikasi.
b. Pembuatan *Random Password Generator* untuk setiap transaksi yang terjadi.
c. Mengembangkan *Security as a Service* sebagai satu bagian khusus yang akan menangani seluruh aspek *security*.

Keamanan web service menjadi suatu hal yang penting dalam pembangunan aplikasi ini, dikarenakan keamanan data seperti password, identitas pengguna, dan lokasi pengguna harus di lindungi.

2.5 Android

Android merupakan sistem operasi dengan jumlah pengguna terbesar menurut hasil riset dari StatCounter.com[3]. Berikut adalah penjelasan android :

2.5.1 Sejarah Android

Android adalah sistem operasi berbasis linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama. Awalnya, *Android* dikembangkan oleh Android Inc. Perusahaan ini kemudian di beli oleh *Google* pada tahun 2005. Sistem operasi *Android* kemudian diluncurkan bersamaan dengan dibentuknya organisasi *Open Handset Alliance* tahun 2007. Selain *Google,* beberapa nama-nama besar juga ikut serta dalam *Open Handset Alliance*, antara lain *Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel* [12].

2.5.2 Versi Android

Adapun versi android yang pernah dirilis adalah sebagai berikut [14] :

1. Android versi 1.1

Versi 1.1 dirilis pada 9 maret 2009, android ini dilengkapi dengan pembaruan estetis pada aplikasi, alarm, voice search (pencarian suata), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (Cupcake)

Pada pertengahan bulan Mei 2009, Google kembali merilis telepon selules dengan menggunakan Android dan SDK (Software Develoment Kit). Terdapat banyak pembaharuan yakni mampu merekam dan menonton video dengan mous kamera, mengupload video ke youtube dan gambar ke picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung otomatis ke headset Bluetooth.

3. Android v1.6 Donut

Pada September 2009, google kembali merilis android versi 1.6 (Donut). Ini merupakan rilis minor dengan fitur utama berupa dukungan teknologi CDMA, gesture, text-to-speech engine, resolusi VWGA, VPN, dan fitur pencarian cepat.

4. Android v2.0-2.1 Eclair

Pada 3 Desember 2009 Google kembali meluncurkan *Android* dengan versi 2.0/2.1 (*Eclair*). Mengusung fitur baru yaitu live wallpaper, pengoptimalan *hardware,* peningkatan *Google Maps* 3.1.2, Bluetooth 2.1, dan profil bluetooth baru yaitu *Object Push Profile* (OPP) dan *Phone Book Access Profile* (PBAP).

5. Android v2.2 Frozen Yoghurt (Froyo)

Pada 20 Mei 2010 dirilis *Android* 2.2 (*Froyo*). Fitur baru yang dibawa adalah Global Positioning System (GPS), kompas, kamera, dan Accelorometer.

6. Android v2.3 Gingerbread

Pada 6 Desember 2010, *Android* versi 2.3 (*Gingerbread*) diluncurkan. Mengusung beberapa fitur baru, antara lain dukungan banyak kamera, *Near field Communication* (NFC), *download manager service*, dukungan terhadap sensor lain seperti *giroskop*, dan *barometer*.

7. Android v3.0-3.2 Honeycomb

Android *Honeycomb* dirancang khusus untuk tablet diluncurkan pada 22 Februari 2011. *Android* versi ini mendukung ukuran layar yang lebih besar. Pada *Honeycomb* mendukung multi prosesor dan akselerasi perangkat keras (*Hardware*) untuk grafis.

8. Android v4.0 Ice Cream Sandwich

Pada 19 Oktober 2011, *Android* versi 4.0 (*Ice Cream Sandwich*) diluncurkan. Pada versi ini membawa perubahan besar pada tampilan *User Interface Android* 3.0 agar cocok pada layar kecil. Fitur baru seperti *Android Beam* dan *Wi-Fi Direct* juga ditambahkan.

9. Android v4.1-4.3 Jelly Bean

Pada 27 Juni 2012, *Android* versi 4.1 (*Jelly Bean*) diluncurkan. Pada versi ini merupakan rilis minor yang membawa perbaikan fungsionalitas dan performa *rendering User Interface* (UI). Versi 4.2 mengusung fitur baru seperti *gesture typing* dan dukungan *multiuser* di piranti tablet. Versi 4.3 membawa perbaikan berupa dukungan *Bluetooth Low Energy* dan *OpenGL ES* 3.0. beberapa fitur baru yaitu dukungan terhadap bahasa internasional dan penulisan teks dua arah.

10. Android v4.4 Kitkat

Android versi 4.4 (*Kitkat*) diluncurkan pada 3 September 2013. Pada versi ini mengusung sejumlah perbaikan dan fitur baru yaitu dukungan *Near Field Communication* melalui *host card emulation*, pencetakan printer nirkabel,

WebView dengan *rendering engine Chromium*, dan dukungan yang lebih baik bagi piranti yang menggunakan *Random Access Memory* (RAM) rendah.

11. Android versi 5.0 (Lollipop)

Versi terbaru saat ini dengan tampilan UI yang lebih menarik, dengan meningkatkan daya hidup baterai 30% lebih tahan lama.

2.5.3 Arsitektur Android

Android terdiri dari lapisan *software*, yaitu lapisan aplikasi, lapisan *framework*, *Android run-time* sebagai lapisan perantara (*middleware*), serta *kernel linux* yang membungkus *hardware* di bawahnya seperti gambar 2-6. berikut penjelasan arsitektur dari android: [12]

a. Aplikasi

Aplikasi yang dibuat berada pada level tertinggi dalam hierarki struktur software di android.

b. Framework Aplikasi

Di bawah aplikasi terdapat sejumlah software pendukung, meliputi layanan pengelola Activity, View, sumber daya, pemberitahuan, dan lain-lain.

c. Pustaka

Android menyediakan sejumlah pustaka dasar seperti pustaka 2D dan 3D, *multimedia playback*, *browser engine*, pencetakan *font*, *database*, dan lain-lain. Aplikasi yang memanfaatkan ini melalui framework aplikasi.

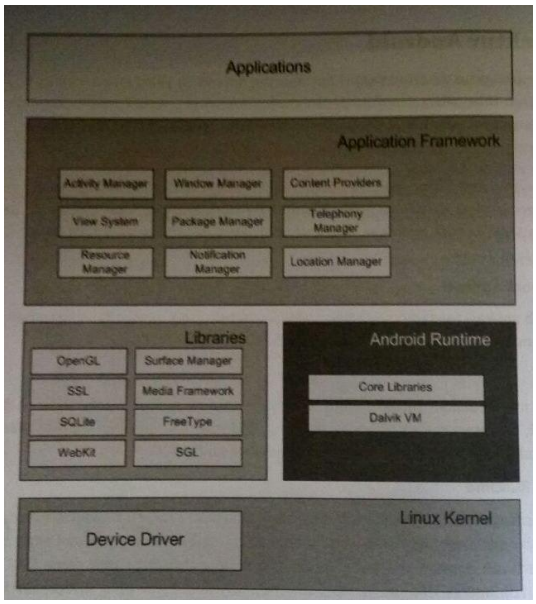
d. Android Runtime

Android menyediakan pustaka inti bagi pemrograman *Java*. Di bawahnya terdapat *Dalvik Virtual Machine* (Dalvik VM) yang akan menjalankan aplikasi. Tiap aplikasi akan dilajankan pada proses terpisah dengan VM yang berbeda sehingga terisolasi satu sama lain. Aplikasi masih dapat berkomunikasi dengan aplikasi lain melalui Framework aplikasi. Dalvik VM

bergantung pada kernel linux untuk *multi-threading* dan pengelolaan memori tingkat rendah.

e. Kernel Linux

Kernel Linux bertanggung jawab menyediakan layanan dasar seperti keamanan, pengelolaan, proses, pengelolaan file, pengelolaan sumber daya memori, dan *hardware*.



Gambar 2-6 Arsitektur Android

2.6 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun 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.[14] .

JSON menjadi bahasa pertukaran data yang ideal untuk di gunakan. JSON terbuat dari dua struktur :

a. Kumpulan pasangan nama/nilai. Pada beberapa bahasa. Hal ini dinyatakan sebagai objek (*object),* rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar kunci (*keyed list*), atau *associative array* [14].

b. Daftar nilai teurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).[14]

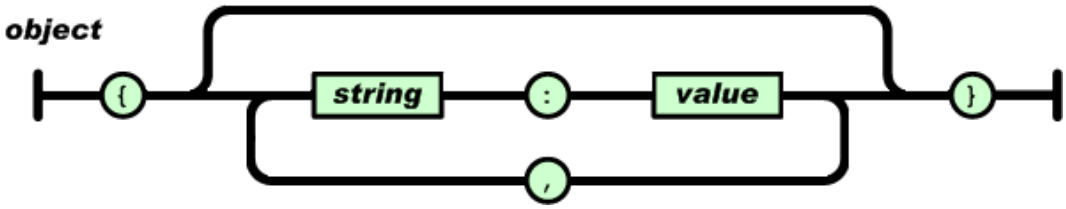
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON di gunakan sebagai format data dalam pertukaran informasi dari aplikasi menuju web service dan juga sebaliknya dalam aplikasi yang akan di bangun.

2.6.1 Bentuk JSON

JSON menggunakan bentuk sebagai berikut:

a. Objek

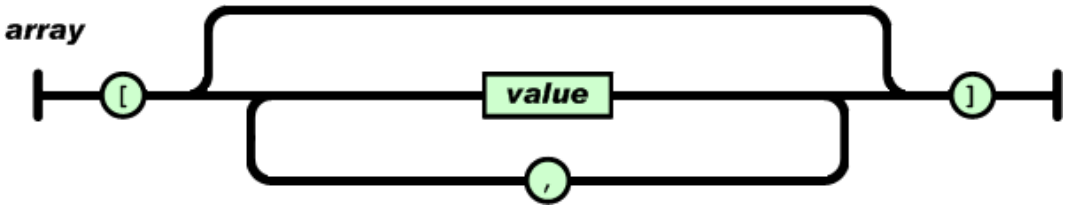
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurang kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). [14]



Gambar 2-7 Objek JSON

b. Larik

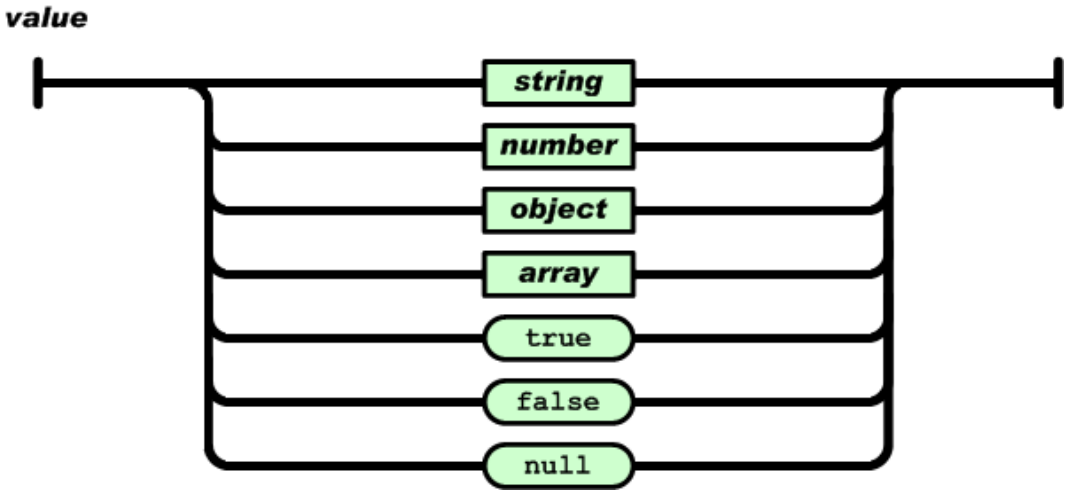
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma). [14]



Gambar 2-8 Larik JSON

c. Value

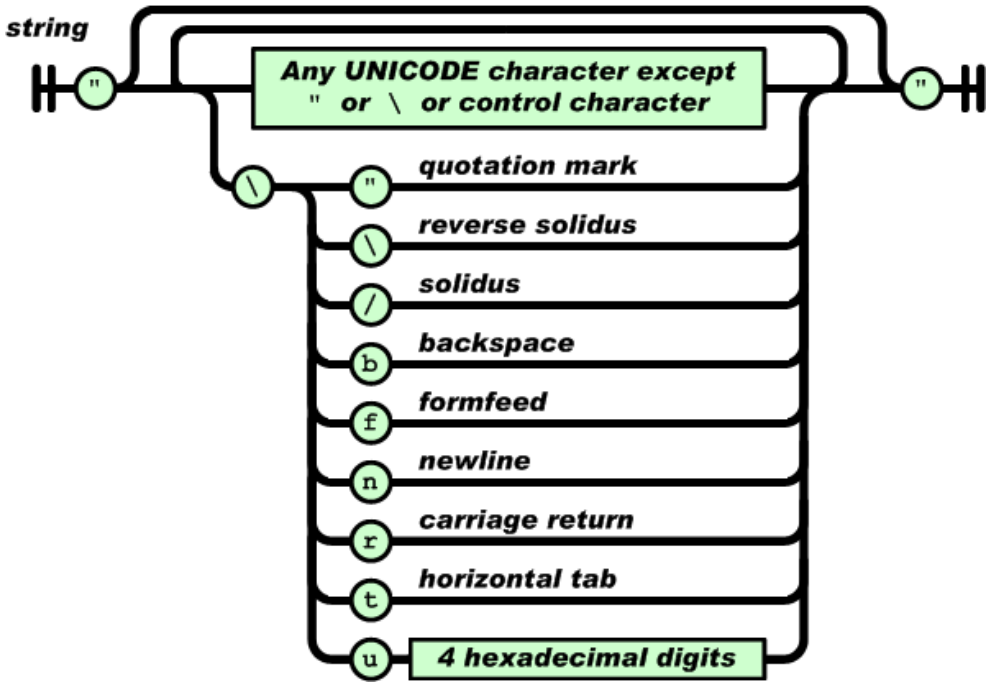
Nilai (*Value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.[14]



Gambar 2-9 Nilai (*Value*) JSON

d. String

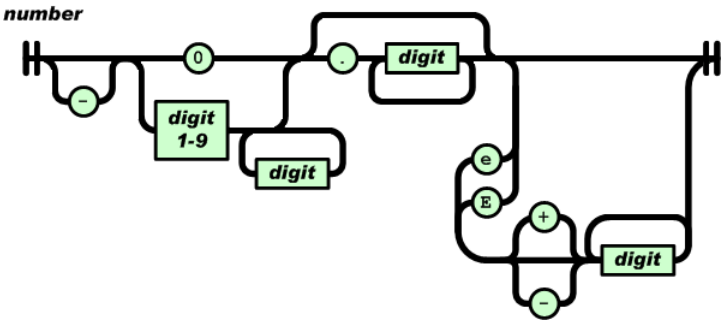
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapers* "/" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.[14]



Gambar 2-10 String JSON

e. Number

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.[14]

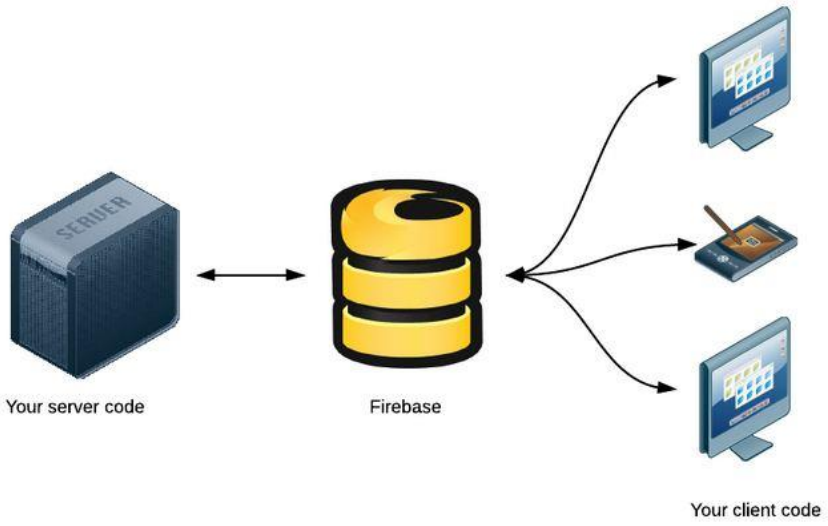


Gambar 2-11 Number JSON

2.7 Firebase

Android firebase adalah sebuah *cloud service* serta bisnis backend yang memungkinkan anda untuk mendapatkan data yang teorganisir untuk aplikasi

mobile.[15]. *Firebase* adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembangan membuat API untuk di selaraskan dengan client yang berbeda-beda dan disimpan pada *firebase.* Pengembangan menggunakan REST APU untuk post data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara real time. [16]



Gambar 2-12 Cara Kerja Firebase

2.8 Database

Database (Basis data) merupakan gabungan file data yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat independen. Adapun basis data adalah Tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut. [17]

2.8.1 SQLite

SQLite adlaah sebuah SQL database engine yang tidak membutuhkan server, tidak membutuhkan konfigurasi yang bersifat transaksional. Sebuah transaksi dalam istilah database berarti memiliki kemampuan untuk menjalankan

sebuah transaksi SQL secara utuh atau dibatalkan sama sekali hal ini misalnya jika terjadi gangguan pada proses akibat crash pada mesin atau program atau yang lainny. SQLite database disimpan dalam sebuah file-file yang bisa di akses dalam berbagai sistem operasi seperti UNIX(Linux, MacOS), Windows, baik mesin 32 bit atau un 64 bit. [18]. Oleh karena itu SQLite sangat cocok untuk pengembangan aplikasi android dikarenakan tidak ada konfigurasi database sehingga relatif mudah digunakan, tidak memiliki server, dan mudah dalam pengembanganya.

2.9 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangakt lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsi pengelolaan komplesitas [19].

Pada saat ini pembangunan perangkat lunak lebih memilih metodologi berorientasi objek di bandingkan metodologi terstruktur dikarenakan banyak menimbulkan masalah dalam mentransformasi hasil analisa dari setiap tahapan, dan pengembangan perangkat lunak yang membutuhkan waktu yang lama.

Keuntungan dalam menggunakan metodologi berorientasi objek adalah sebagai berikut[20]:

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan ojek tersebut.
2. Kecepatan Pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.

3. Kemudahan Pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya Konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan Kualitas Perangkat Lunak

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

2.10 UML (Unified Modeling Language)

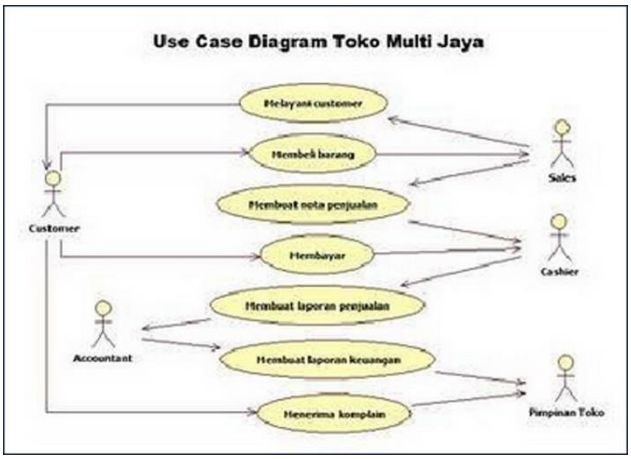
Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [21]. UML adalh bahsa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak lebel abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi[22]. Dalam pembangunan perangkat lunak perlu merancang, menggambarkan sistem perangkat lunak sehingga UML di perlukan dalam pembangunan perangkat lunak.

Diagram UML yang akan di gunakan dalam memvisualisasikan perangkat lunak yang akan di bangun antara lain : use case diagram, class diagram, activity diagram, sequence diagram.

2.10.1 Use Case Diagram

Use case diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing use-case menunjukan sekumpulan use-case, aktor dan hubungannya. Use-case adalah interaksi antara aktor eksternal dan sistem, hasil

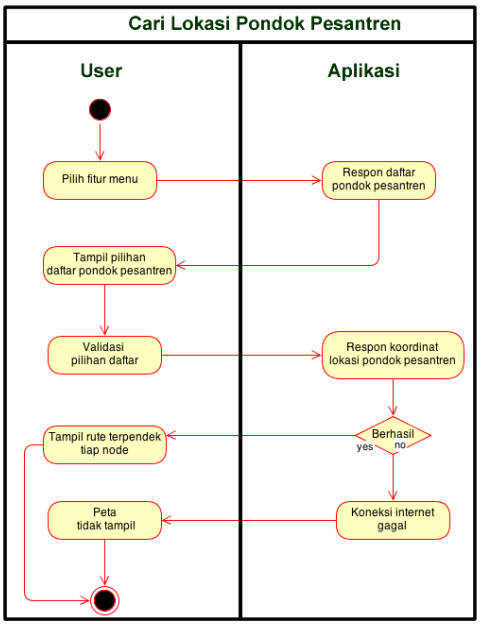
yang dapat diamati oleh aktor, berorientasi pada tujuan, dideskripsikan di diagram use-case dan teks [22]. Use case akan menggambarkan interaksi antara pengguna aplikasi dengan sistem yang akan di bangun, guna mengetahui hubungannya.



Gambar 2-13 Contoh Use Case Diagram

2.10.2 Activity Diagram

Pada dasarnya, diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain. Kita menggunakan diagram ini untuk memodelkan aspek dinamis sistem. Aktivitas adalah eksekusi nonatomik yang berlangsung di state machine. Diagram aktivitas mendeksipsikan aksi-aksi dan hasilnya. Diagram aktivitas berupa operasi-operasi dan aktivitas-aktivitas di use-case [22]. Activity digram dapat memodelkan aliran kendali aktivitas pada perangkat lunak yang di bangun sehingga dapat di ketahui aksi dan hasilnya.



Gambar 2-14 Contoh Activity Diagram

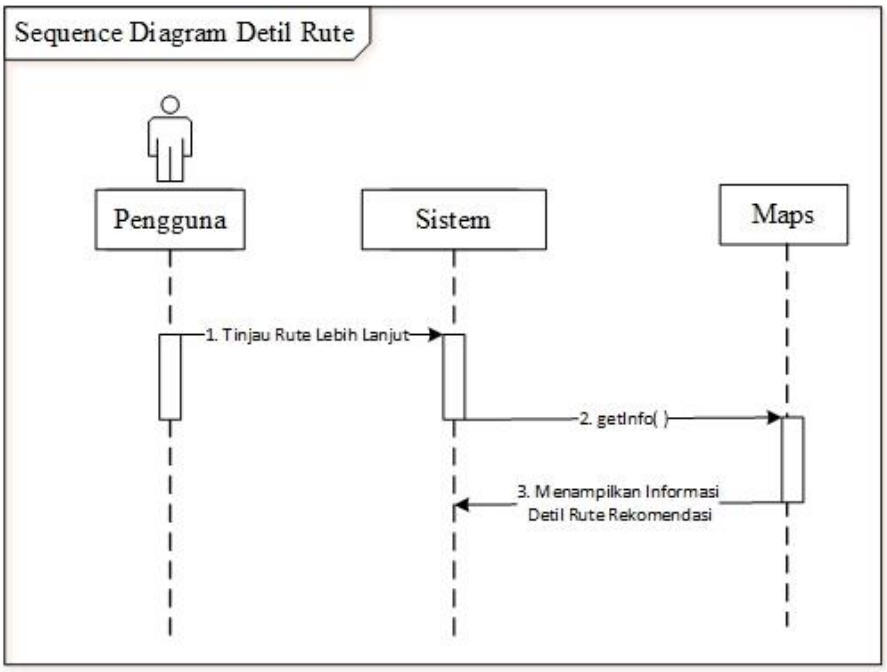
2.10.3 Sequence Diagram

Digram sekuen (Sequence Diagram) mendeskripsikan komunikasi di antara objek-objek, meliputi pesan-pesan yang ada dan urutan pesan tersebut muncul. Diagram sekuan menggambarkan urutan kejadian suatu kegiatan [22].

Diagram sekuen digunakan untuk [22]:

1. Overview perilaku sistem
2. Menunjukan objek-objek yang diperlukan.
3. Mendokumentasikan skenario-skenario dari suatu digram use case.
4. Memeriksa jalur-jalur pengaksesan.

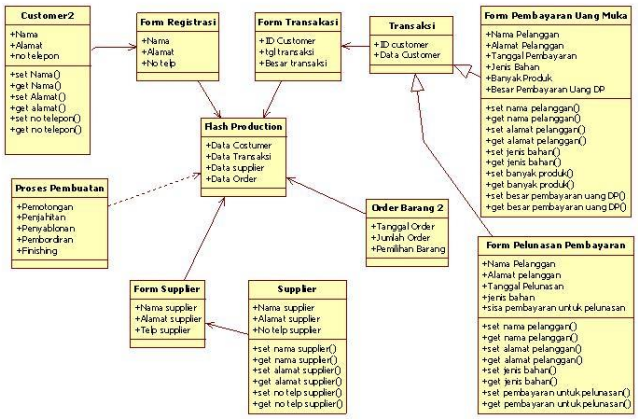
Pada pembangungan aplikasi, diagram sekuan di perlukan untuk menggambarkan urutan kejadian pada setiap kegiatan pada aplikasi .



Gambar 2-15 Contoh Sequence Diagram

2.10.4 Class Diagram

Digram kelas merupakan digram paling umum dipakai di semua pemodelan berorientasi objek. Pemodelan kelas menunjukan kelas-kelas yang ada di sistem dan hubungan antar kelas-kelas itu, atribut-atribut dan operasi-operasi di kelas-kelas. Diagram kelas menunjukan aspek statik sistem terutama untuk mendukung kebutuhan fungsional sistem. Kebutuhan fungsional berarti layanan-layanan yang harus disediakan sistem ke pemakai [22]. Diagram kelas di gunakan untuk menggambarkan kelas-kelas pada sistem, sehingga dapat menunjukan hubungan antar kelas dan mendukung fungsionalitas perangkat lunak.



Gambar 2-16 Contoh Class Diagram

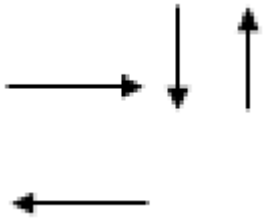
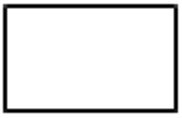
2.11 Flowmap

Flowmap merupakan bagian yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagian ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

Flowmap menggambarkan pergerakan proses diantara unit kerja yang berbeda-beda, sekaligus menggambarkan arus dari dokumen, aliran data fisik, entitasntitas sistem informasi dan kegiatan operasi yang berhubungan dengan sistem informasi. Simbol-simbol pada flowmap diantaranya terdapat pada Tabel 2-2 [23]

Tabel 2-2 Simbol-simbol pada flowmap

Simbol	Nama Simbol	Fungsi
	Dokumen	Menunjukan dokumen *input* dan *output* pada proses manual dan proses berbasis komputer.

	Proses manual	Menunjukan proses yang dilakukan secara manual.
	Penyimpanan magnetik	Menunjukan media penyimpanan data/informasi pada proses berbasis komputer. File dapat disimpan pada harddisk, disket, CD dan lain-lain.
	Arah alir dokumen	Menunjukan arah aliran dokumen antar bagian yang terkait pada suatu sistem. Bisa dari sistem keluar ataupun dari luar ke sistem dan antar bagian diluar sistem.
	Penghubung	Menunjukan alir dokumen yang terputus atau terpisah pada halaman alir dokumen yang sama
	Proses komputer	Menunjukan proses yang dilakukan secara komputerisasi
	Pengarsipan	Menunjukan simpanan data non-komputer/informasi file pada proses manual. Dokumen dapat

		disimpan pada lemari arsip, map file dan lain-lain.
	Input keyboard	Menunjukan *input* yang dimasukan melalui *keyboard.*
	Penyimpanan manual	Menunjukan media penyimpanan data/informasi secara manual.

Dalam membuat *flowmap* perlu beberapa hal yang harus diperhatikan diantaranya:

a. Memodelkan aliran dokumen pada suatu sistem yang sedang berjalan.
b. Bentuk dokumen yang digunakan dapat berupa manual atau berupa file komputer.
c. Dalam satu aliran dokumen terdiri dari input-proses-output.
d. Tidak boleh ada dokumen yang hilang dalam runtunan prosesnya.

2.12 Perangkat Lunak Pendukung

Berikut adalah perangkat lunak pendukung dalam pembangunan aplikasi yang di bangun :

2.12.1 Android Studio

Android Studio adalah Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersigat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android [24]. Android

Studio digunakan pada pembangunan aplikasi ini di karenakan IDE resmi untuk pembangunan aplikasi android .



Gambar 2-17 Android Studio

2.12.2 Google Maps API

Google maps merupakan layanan dari google yang mempermudah penggunanya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan google maps API memungkinkan pengembangan untuk mengintegrasikan Google Maps ke dalam situs web. Dengan menggunakan Google Maps API memungkinkan untuk menanamkan situs Google Mas ke dalam situs eksternal, dimana situs data tertentu dapat dilakukan *overlay* [25].



Gambar 2-18 Contoh Google Maps

Meskipun pada awalnya hanya *Javascript* Api, API Maps sejak diperluas untuk menyertakan sebuah API untuk Adobe Flash aplikasi, layanan untuk mengambil gambar peta status, dan layanan web untuk melakukan geocoding, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil elevasi.

Kelas kunci dalam perpustakaan Maps adalah *MapView,* sebuah *subclass* dari *ViewGrou* dalam standar perpustakaan Android. Sebuah MapView menampilkan peta dengan data yang diperoleh dari layanan Google Maps. Bila yang ditekan dan gerakan sentuh untuk pan dan zoom peta secara otomatis, termasuk penanganan permintaan jaringan untuk ubin peta tambah. Ini juga menyediakan semua elemen UI yang diperlukan bagi pengguna untuk mengendalikan peta. Aplikasi tersebut juga dapat menggunakan metode MapView kelas untuk mengontrol MapView secara terprogram dan menarik sejumlah jenis tampilan di atas peta,

Secara umum, kelas MapView menyediakan pembungkus di Google Mas API yang memungkinkan aplikasi tersebut memanipulasi data Google Maps melalui metode kelas, dan itu memungkinkan dikerjakan dengan data Maps seperti jenis lain Views [20].

2.13 Metode Pengujian

Metode pengujian yang digunakan dalam pembangunan aplikasi ini yaitu Blackbox, kuesioner, dan skala likest. Pengujian di lakukan untuk mengidentifikasi kesalahan-kesalahan pada perangakt lunak yang di buat.

2.13.1 Blackbox

Pengujian Blackbox berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program [26].

Tujuan metode ini mencari kesalahan pada[26]:

1. Fungsi yang salah atau hilang.
2. Kesalahan pada interface.
3. Kesalahan pada struktur data atau akses database.
4. Kesalahan performasi
5. Kesalahan inisialisasi dan tujuan akhir

Pengujian blackbox di lakukan pada pembangunan perangkat lunak ini untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak, mengevaluasi User Interface dan fungsionalitasnya.

2.13.2 Kuesioner

Angket atau kuesioner merupakan suatu teknik atau cara pengempulan data secara tidak langsung (peneliti tidak langsung bertanya-jawab dengan responden. Angket adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respons sesuai dengan permintaan pengguna. Tujuan penyebaran angkat ialah mencari informasi lengkap mengenai suatu masalah dari responden dari responden tanpa merasa khawatir bila responden memberikan jawaban yang tidak sesuai dengan kenyataan dalam pengisian daftar pertanyaan [27]. Pada pengujian perangkat lunak kuesioner digunakan untuk mencari permasalahan yang ada pada layanan antar jemut Grooming hewan peliharaan sehingga di dapatkan kebutuhan para pemilik hewan peliharaan.

2.13.3 Skala Likert

Skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi sesorang atau sekelompok orang tentang kejadian atau gejala sosial. Dalam

penelitian gejala sosial, peneliti telah menetapkan secara spesifik skalanya dan selanjutnya disebut variabel penelitian.

Dengan menggunakan skala likest, variabel yang akan diukur dijabarkan menjadi dimensi, lalu dimensi dijabarkan menjadi subvariabel dan subvariabel dijabarkan lagi menjadi indikator yang dapat diukur. Akhirnya indikator-indikator yang terukur dapat menjadi titik tolak untuk membuat item instrumen berupa pertanyaan atau pernyataan yang perlu dijawab oleh responden. Setiap jawaban dihubungkan dengan bentuk pernyataan atau dukungan sikap yang diungkapkan dengan kata kata [27].