

BAB 2

LANDASAN TEORI

2.1 Tinjauan Umum Instansi

Pada tinjauan umum instansi ini akan dibahas mengenai sejarah berdirinya perusahaan, struktur organisasi, visi misi serta uraian tugas dari masing-masing bagian yang terlibat.

2.1.1 Sejarah Umum dan Perkembangan Instansi

Puskesmas rancabali merupakan salah satu puskesmas yang ada di Indonesia yang terletak di Kabupaten Bandung, puskesmas rancabali berdiri pada tahun 2006 di kabupaten Bandung tepatnya di kecamatan rancabali desa parengan dengan di kepalai oleh seorang dokter wilayah untuk memberikan pelayanan kesehatan secara menyeluruh kepada masyarakat di wilayah rancabali dan beberapa kecamatan di sekitarnya, pada tahun 2009 puskesmas kecamatan rancabali berpindah ke desa alamendah dengan di kepalai oleh bapak herwan, SKM sampai dengan sekarang.

Perkembangan kesehatan masyarakat di Indonesia dimulai sejak pemerintahan Belanda pada abad ke-16 yaitu adanya upaya pemberantasan penyakit cacar dan cholera yang sangat ditakuti oleh masyarakat. Pada tahun 1968 diterapkan konsep puskesmas yang diluncurkan dalam Rapat Kerja Nasional di Jakarta, yang membicarakan tentang upaya mengorganisasi sistem pelayanan kesehatan di tanah air, karena pelayanan kesehatan pada saat itu dirasakan kurang menguntungkan dan dari kegiatan-kegiatan seperti Balai Kesehatan Ibu dan Anak (BKIA), Balai Pengobatan (BP), Pemberantasan Penyakit Menular (P2M) dan sebagainya masih berjalan sendiri-sendiri dan tidak saling berhubungan.

2.1.2 Visi dan Misi Instansi

2.1.2.1 Visi Instansi

Visi puskesmas ini adalah mewujudkan masyarakat Kecamatan Rancabali dapat melaksanakan perilaku bersih dan sehat.

Mini Endoscope Camera meliputi pengertian teori dasar penyakit THT, aplikasi android, API Clarifai, Camera Endoscope, Pengolahan Citra, FKNN, bahasa pemrograman Java, PHP, JSON, MySQL, dan UML.

2.3 Penyakit THT

Lokasi dan fungsi telinga, hidung dan tenggorokan (untuk selanjutnya disebut THT) berhubungan erat yang dihubungkan oleh saluran yang dinamakan saluran “Eustachian tube”[9]. Dan disebabkan oleh infeksi itu pada hidung bisa menyebabkan menyebarnya virus ke tenggorokan dan dari tenggorokan ke hidung. Kelainan pada organ-organ tersebut didiagnosis dan diobati oleh dokter spesialis THT. Telinga ialah salah satu organ terpenting didalam tubuh yang berguna sebagai pendengaran dan keseimbangan yang terdiri dari telinga luar, telinga tengah dan telinga dalam. Telinga luar menangkap gelombang suara yang dirubah menjadi energi mekanis oleh telinga tengah. Sedangkan hidung merupakan organ penciuman dan jalan utama keluar-masuknya udara dari dan ke paru-paru. Hidung bagian atas terdiri dari tulang dan hidung bagian bawah terdiri dari tulang rawan (kartilago). Rongga hidung dilapisi oleh selaput lendir dan pembuluh darah. Sel-sel pada selaput lendir menghasilkan lendir dan memiliki tonjolan-tonjolan kecil seperti rambut (silia). Dan tenggorokan (faring) terletak di belakang mulut, di bawah rongga hidung dan diatas kerongkongan dan tabung udara (trakea). Tenggorokan terbagi menjadi tiga bagian, atas, tengah dan bawah. Tenggorokan merupakan saluran berotot tempat jalannya makanan ke kerongkongan dan tempat jalannya udara ke paru-paru. Tenggorokan dilapisi oleh selaput lendir yang terdiri dari sel-sel penghasil lendir dan silia. Tonsil (amandel) terletak di mulut bagian belakang, sedangkan adenoid terletak di rongga hidung bagian belakang.

Dan berikut terdapat beberapa jenis penyakit THT yang sering dialami oleh masyarakat, yaitu sebagai berikut [10].

1. Abses Peritonsiler
2. Barotitis Media
3. Faringitis
4. Laryngitis

5. Otitis Media Akut
6. Meniere
7. Tonsilitis
8. Sinusitis Maksilarasis
9. Sinusitis Frontalis
10. Sinusitis Etmoidalis

2.4 Smart Assisted Diagnostic Tool

Proses diagnostik merupakan perpaduan dari aktifitas intelektual dan manipulatif. Diagnosis sendiri didefinisikan sebagai suatu proses penting pemberian nama dan pengklasifikasian penyakit-penyakit pasien, yang menunjukkan kemungkinan nasib pasien dan yang mengarahkan pada pengobatan tertentu. Diagnosis banding, sebagaimana halnya dengan penelitian-penelitian ilmiah, didasarkan atas metode hipotesis. Dengan metode hipotesis ini menjadikan penyakit-penyakit begitu mudah dikenali hanya dengan suatu kesimpulan diagnostik [11]

Diagnosis banding dimulai sejak permulaan wawancara medis dan berlangsung selama melakukan pemeriksaan fisik. Dari diagnosis banding tersebut akan diperoleh pertanyaan-pertanyaan yang terarah, perincian pemeriksaan fisik yang dilakukan untuk menentukan pilihan tes-tes serta pemeriksaan khusus yang akan dikerjakan. Data yang berhasil dihimpun, akan dipertimbangkan dan diklasifikasikan berdasarkan keluhan-keluhan dari pasien serta hubungannya terhadap penyakit tertentu. Berdasarkan gejala-gejala serta tanda-tanda yang dialami oleh penderita, maka penegakkan diagnosis akan lebih terpusat pada bagian-bagian tubuh tertentu. Dengan demikian penyebab dari gejala-gejala dan tanda-tanda tersebut dapat diketahui dengan mudah dan akhirnya diperoleh kesimpulan awal mengenai penyakit tertentu.[11]

Pada kasus ini, digunakan *Smart Assisted Diagnostic Tool* yaitu Mini Kamera Endoskop untuk mendiagnosa penyakit THT, khususnya telinga. Alat ini bekerja dengan cara mengambil gambar dari dalam lubang telinga, dan disesuaikan dengan data latih yang sebelumnya sudah dibuat.

2.5 Aplikasi

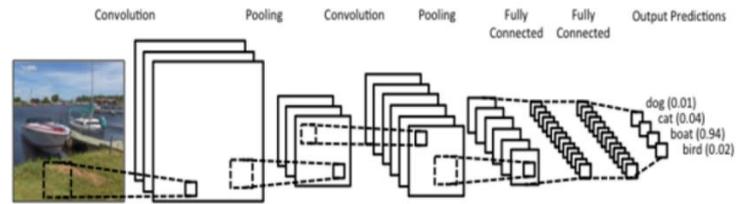
Aplikasi adalah program yang berisikan perintah-perintah untuk mengelolah kumpulan data agar lebih efektif dan efisien[]. Pada dunia teknologi informasi, aplikasi dibuat untuk mempermudah dan membantu perkerjaan manusia dari cara manual menjadi optimal dengan memanfaatkan komputer. Aplikasi dapat dibuat ataupun dikembangkan dengan maksud untuk melakukan tugas yang bersifat umum atau khusus.

Pada dunia teknologi aplikasi merupakan sebuah perangkat lunak yang dibuat untuk membantu atau mempermudah pekerjaan manusia. Aplikasi juga merupakan suatu program perangkat lunak yang berjalan pada suatu sistem tertentu. Sebuah aplikasi dapat dibuat dan dikembangkan dengan tujuan untuk melakukan tugas yang bersifat umum atau juga dapat dikembangkan untuk melakukan tugas yang bersifat spesifik/khusus.

2.6 API Clarifai

Clarifai adalah alat yang dapat mengidentifikasi atau mengenali gambar atau video yang dimasukkan sebagai inputan yang dapat memberikan hasil berupa prediksi tentang apa yang ada di dalam gambar atau video berupa besaran probabilitas kemungkinannya [7].

Clarifai juga mengambil sebuah pixel sebagai input menggunakan library semantic dan visual untuk kecerdasan buatan. Pada sistem ini didasarkan pada jaringan syaraf tiruan, teknik pembelajaran mesin scalable yang dapat menangani skala besar konten visual yang mengalir melalui API [12]. Clarifai juga menggunakan kesamaan semantik dan visual untuk membandingkan gambar yang diunggah dengan gambar lainnya pada library untuk menampilkan tingkat kesamaan pada objek. Clarifai bekerja melalui analisis gambar untuk menghasilkan daftar deskripsi tag dari gambar yang diberikan. Untuk setiap tag dalam daftar ini, sistem juga menyediakan nilai probabilitas. Nilai probabilitas ini nantinya digunakan sebagai tag yang spesifik pada gambar. Clarifai API dapat diakses sebagai layanan web jarak jauh. Skema kerja teknologi Clarifai ditunjukkan pada gambar 2.2 sebagai berikut.



Sumber gambar : Majalah ilmiah unikom (2018) [12].

Gambar 2.2 Skema API Clarifai

2.7 Camera Endoscope

Camera Endoscope adalah kamera dengan antarmuka dan ukuran sangat kecil, kamera ini sering dipergunakan dalam merekam visual video, kamera ini juga dapat melihat kedalam rongga-rongga yang ukurannya sangat kecil, kedalam ruang-ruang sempit dan gelap. Camera endoscope juga dilengkapi dengan koneksi micro USB OTG, lampu LED dan waterproof sehingga dapat digunakan dalam kondisi manapun tanpa khawatir mengalami kerusakan. Berikut beberapa jenis kamera endoscope yang banyak digunakan:

1. Sialendoskop Serat Optik Lentur

Endoskop jenis serat optik lentur akan lebih mudah melewati lekukan pada duktus serta lebih sedikit menimbulkan trauma. Namun penggunaannya relatif lebih sulit daripada endoskop kaku maupun semirigid. Pada penatalaksanaan kasus sialolitiasis, keberhasilannya lebih rendah daripada jika menggunakan semikaku. Endoskopi serat optik lentur lebih rapuh dan lebih mudah rusak daripada endoskop kaku, serta tidak dapat disterilkan dengan autoklav [2].

2. Sialendoskop Kaku

Sialoendoskop jenis kaku menggunakan sistem lensa dengan kualitas superoptikal dan resolusinya lebih baik. Endoskop ini memiliki diameter yang lebih besar sehingga lebih stabil dan dapat disterilkan dengan autoklav. Kameranya terletak pada perlekatan okular dengan endoskop sehingga penggunaannya agak kurang praktis [2].

3. Sialendoskop Semikaku

Merupakan gabungan antara serat optik lentur dan kaku. Bagian yang panjang merupakan fleksibel yang menggunakan serat optik untuk transmisi cahaya. Penggunaan endoskop semikaku akan memudahkan pergerakan dan membutuhkan kekuatan yang minimal untuk mengambil gambar dengan presisi yang tepat [2].

2.8 Pengolahan Citra

Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer [13].

Secara matematis, citra merupakan fungsi kontinyu (continue) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit [14]. Reperesentasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (pixel = picture element) atau elemen terkecil dari sebuah citra.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

2.9 Fuzzy K-Nearest Neighbor

Metode Fuzzy K-Nearest Neighbor (FKNN) diperkenalkan oleh Keller et al (1985) dengan mengembangkan K-NN yang digabungkan dengan teori Fuzzy dalam menyampaikan definisi pemberian label kelas pada data uji yang diprediksi. Seperti halnya pada teori Fuzzy, sebuah data mempunyai nilai keanggotaan pada setiap kelas yang artinya sebuah data bisa dimiliki oleh kelas yang berbeda dengan nilai derajat keanggotaan dalam interval $[0,1]$ Rumus yang digunakan adalah:

$$u(x, ci) = \frac{\sum_{k=1}^K u(xk, ci) * d(x, y)^{\frac{-2}{(m-1)}}}{\sum_{k=1}^K d(x, y)^{\frac{-2}{(m-1)}}} \quad (0.1)$$

Yang mana:

$u(x, ci)$: nilai keanggotaan data x ke kelas ci

K : jumlah tetangga terdekat yang digunakan

$u(xk, ci)$: nilai keanggotaan data tetangga dalam K tetangga pada kelas ci ,
nilainya 1 jika data latih xk milik kelas ci atau 0 jika bukan milik kelas ci

$d(x, y)$: jarak dari data x ke data y dalam K tetangga terdekat

m : bobot pangkat (weight exponent), yang besarnya $m > 1$

Dalam algoritma Fuzzy KNN menggunakan nilai keanggotaan data pada setiap kelas tetapi untuk mendapatkan keluaran akhir Fuzzy KNN tetap harus memberikan kelas akhir akan hasil prediksi, Fuzzy KNN memilih kelas dengan nilai keanggotaan terbesar pada data tersebut. [4]

1. Komponen Masukan

Mewakili sejumlah data yang masuk ke dalam sistem informasi. Input tersebut termasuk metode-metode dan media untuk memperoleh data yang akan dimasukkan yang dimana dapat berupa dokumen-dokumen dasar komponen proses terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di *database* dengan cara tertentu untuk menghasilkan keluaran (*output*) yang diinginkan.

2. Komponen Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi dan dokumentasi yang dapat digunakan untuk semua tingkatan manajemen dan semua pemakai sistem.

3. Komponen Teknologi

Teknologi merupakan *toolbox* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

4. Komponen *Hardware*

Hardware berperan penting sebagai suatu peralatan vital bagi sistem informasi yang digunakan untuk menyimpan dan mengolah informasi.

5. Komponen *Software*

Software berfungsi sebagai kumpulan perintah komputerisasi yang berguna untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

6. Komponen Basis data

Database merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan pada perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Pengelolaan *database* umumnya dikenal dengan nama DBMS (*Database Management System*).

7. Komponen Kendali

Bagian pengendalian dirancang dan diterapkan untuk memelihara sistem dari hal-hal yang dapat merusaknya, seperti faktor-faktor alamiah temperatur, air, api, debu, virus, sabotase / *hijacking*, dan sebagainya.

2.10 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. 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, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008[16].

Android memiliki komponen utama yang dapat membuat berjalan dengan baik dan disebut dengan Arsitektur *Platform* Android. Berikut diagram komponen dari Android pada gambar 2.3 :



Sumber: <https://developer.android.com/guide/platform/index.html?hl=id>

Gambar 2.3 Arsitektur Platform Android

Penjelasan dari setiap komponen yang ada pada Arsitektur Platform Android antara lain:

2.10.1 *Linux Kernel*

Fondasi platform pada sistem Android adalah Linux kernel. Sebagai contoh, Android Runtime (ART) bergantung pada Linux kernel untuk menjalankan fungsionalitas dasar seperti *threading* dan manajemen memori tingkat rendah.

Menggunakan Linux kernel memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup terkenal.

2.10.2 *Hardware Abstraction Layer*

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau *bluetooth*. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

2.10.3 *Android Runtime*

Perangkat Android yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap *Android Runtime* (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format *bytecode* yang didesain khusus untuk Android yang dioptimalkan untuk *footprint* memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke *bytecode* DEX, yang dapat berjalan pada platform Android. Beberapa fitur utama ART mencakup :

- a. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT)
- b. Pengumpulan sampah (GC) yang dioptimalkan
- c. Dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostik mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

2.10.4 *Native C/C++ Libraries*

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda. Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka platform asli langsung dari kode asli.

2.10.5 *Java API Framework*

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

- a. Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan
- b. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout
- c. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status
- d. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum

Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri. Developer memiliki akses penuh ke API kerangka kerja yang sama dengan yang digunakan oleh aplikasi sistem Android.

2.10.6 Mysql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*), database ini multithread, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus yang bersifat khusus.

Kekuatan MySQL tidak ditopang oleh sebuah komunitas, seperti Apache, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing-masing, tetapi MySQL didukung penuh oleh sebuah perusahaan profesional dan komersial, yakni MySQL AB dari Swedia[17].

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan mudah secara otomatis.

2.10.7 System Apps

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama platform tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri—sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang Anda tetapkan

2.11 Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Cupcake (1.5), Donut (1.6), Eclair (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+), hingga yang terbaru adalah Nougat (7.0+) dan selanjutnya versi android terbaru yang ditunggu-tunggu adalah Android O (8.0+).

Pada tanggal 3 September 2013, Google mengumumkan bahwa sekitar 1 miliar perangkat seluler aktif di seluruh dunia menggunakan OS Android. Berikut adalah versi android dan perubahan fitur berdasarkan tanggal rilis:

1. Android 1.0 (API level 1)

Android 1.0 merupakan versi komersial pertama Android yang dirilis pada 23 September 2008. Awalnya versi 1.0 ini akan dinamai “Astro”, tapi dikarenakan ada permasalahan hak cipta dan trademark nama “Astro” maka penamaan pada versi pertama tersebut dibatalkan oleh Google. Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream

2. Android 1.2 (API level 2)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Bender", meskipun nama ini tidak digunakan secara resmi dikarenakan masalah yang sama dengan versi pertamanya. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur.

3. Android 1.5 Cupcake (API level 3)

Versi ini dirilis pada tanggal 27 April 2009, menggunakan kernel Linux 2.6.27. Mulai dari versi ini, android melakukan penamaan berdasarkan nama pencuci mulut. Karena ini merupakan versi ketiga yang dirilis, sehingga penamaan diawali dari huruf "C", yaitu "Cupcake". Pembaruan pada versi ini mulai terlihat dari fitur baru dan perubahan UI. Pada versi ini juga sudah mulai terdapat "widget" yang dapat di rubah ukurannya.

4. Android 1.6 Donut (API level 4)

Pada 15 September 2009, Android 1.6 dirilis. Versi ini dinamai "Donut", dan menggunakan kernel Linux 2.6.29. Versi ini hadir untuk menutupi bug pada versi sebelumnya sekaligus untuk penambahan beberapa fitur, misalnya dukungan untuk perangkat ukuran layar yang lebih besar

5. Android 2.0 – 2.1 Eclair (API level 5-7)

Pada 26 Oktober 2009, SDK Android 2.0 dirilis. Versi ini diberi nama Eclair, dan menggunakan kernel Linux 2.26.29. Pada tanggal 12 Januari 2010, versi 2.1 dirilis. Terdapat perubahan UI dan beberapa penambahan fitur ini, seperti *Bluetooth*, *flash* pada kamera, *digital zoom* pada kamera, *multi touch*, *live wallpaper*, dan lainnya

6. Android 2.2 – 2.2.3 Froyo (API level 8)

Pada 20 Mei 2010, SDK Android 2.2 dirilis dan diberi nama Froyo (singkatan dari *Froze Yoghurt*) menggunakan kernel Linux 2.6.32. Perangkat pertama yang menggunakan versi Android ini adalah Google Nexus One. Pada versi ini sudah mengukung kecepatan dalam membuka dan menutup aplikasi, serta penambahan penyimpanan aplikasi dengan memanfaatkan *SD Card*.

7. Android 2.3 – 2.3.7 Gingerbread (API level 9-10)

Pada tanggal 6 Desember 2010, SDK Android 2.3 (Gingerbread) dirilis, berbasis kernel Linux 2.6.35. Versi 2.3.1 yang rilis desember 2010 dan versi 2.3.2 yang rilis januari 2011 berisi perbaikan bug pada Google Nexus S. Versi 2.3.3 sampai versi 2.3.7 rilis antara bulan februari 2011 sampai bulan september 2011.

Dirilis pada tanggal 6 Desember 2010 bersamaan dengan dihadirkannya Nexus S yang merupakan perangkat smartphone seri Nexus yang diproduksi oleh Samsung. Versi OS ini juga mengawali kesuksesan Android di jagad smartphone meski masih kalah populer dengan BlackBerry OS.

8. Android 3.0 – 3.2 Honeycomb (API level 11-13)

Pada 22 Februari 2011, SDK Android 3.0 (Honeycomb) – pembaruan pertama Android yang ditujukan hanya untuk komputer tablet – dirilis, berdasarkan kernel Linux 2.6.36. Perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011. Versi 3.1 rilis tanggal 10 mei 2011. Versi 3.2 rilis tanggal 15 juli 2011. Google TV generasi pertama dan kedua menggunakan Honeycomb.

9. Android 4.0 – 4.0.4 Ice Cream Sandwich (API level 14 – 15)

SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1, dirilis pada 19 Oktober 2011. Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoretis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu. Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011. Pembaruan pada versi ini antara lain:

Ice Cream Sandwich adalah versi terakhir yang mendukung Flash player Adobe System. Versi 4.0.3 rilis tanggal 16 desember 2011. Versi 4.0.4 rilis tanggal 29 maret 2012.

10. Android 4.1 – 4.3 Jelly Bean (API level 16 – 18)

Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Penambahan fitur yang diberikan diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencariia melalui *Voice Search* yang lebih cepat. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.

11. Android 4.4 Kitkat (API level 19)

Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'. Fitur andalan yang diberikan pada Android versi 4.4 ini adalah pengoptimalan perangkat dengan minimum RAM yaitu 512 MB. Dikarenakan untuk menjalankan versi ini dibutuhkan RAM setidaknya 340 MB.

12. Android 5.0 Lollipop (API level 21)

Android 5.0 pertama kali diperkenalkan di bawah codename "Android L" pada 25 Juni 2014 selama presentasi keynote pada konferensi pengembang Google I/O. Di samping Lollipop, presentasi difokuskan pada sejumlah platform Android yang berorientasi dan teknologi baru, termasuk Android TV, pada platform Android Auto, dapat dipakai pada platform komputasi Android Wear, dan platform pelacakan kesehatan Google Fit.

Bagian dari presentasi didedikasikan untuk bahasa desain cross-platform baru yang disebut sebagai "material design". Memperluas pada "kartu" motif pertama kali terlihat di Google Now, adalah desain dengan peningkatan penggunaan layout berbasis grid, animasi dan transisi responsif, padding, dan efek kedalaman seperti pencahayaan dan bayangan.

13. Android 6.0 Marshmallow (API level 23)

Android 6.0 ini dirilis pada tanggal 25 Juni 2014 dan diberi kode nama Marshmallow pada saat acara Google I/O. Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

14. Android 7.0 – 7.1 Nougat (API level 24 -25)

Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui

over-the-air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

Pada tanggal 19 Oktober 2016, Google merilis Android 7.1.1 sebagai pratinjau pengembang untuk Nexus 5X, Nexus 6P dan Pixel C. Pratinjau kedua mulai tersedia pada 22 November 2016, sebelum versi final diluncurkan ke publik pada bulan Desember. 5, 2016.

15. Android 8.0 Oreo (API level 26)

Android Oreo adalah rilis utama ke 8 dari sistem operasi Android. Ini pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017, dengan gambar pabrik untuk perangkat Nexus dan Pixel saat ini. Pratinjau pengembang terakhir dirilis pada tanggal 24 Juli 2017, dengan rilis stabil yang diharapkan pada bulan Agustus atau September 2017.

16. Android 9.0 Pie (API level 27)

Android 9.0 resmi dirilis pada Agustus 2018 dengan menggunakan nama kode “Pie”. Pada versi ini Google mengoptimalkan penggunaan baterai, sehingga menjaga ketahanan perangkat untuk bisa digunakan dalam jangka waktu lama. Fitur ini diperkenalkan Google dengan sebutan *adaptive battery*, yang menggunakan pembelajaran mesin untuk meramalkan kapan suatu aplikasi tidak digunakan. Android akan “membekukan” aplikasi pada saat-saat tertentu.

Pada Android Pie, Google secara resmi mengintegrasikan dukungan untuk fitur kamera ganda, yang sebelumnya sudah diperkenalkan pada beberapa model ponsel Android unggulan. Pembuat ponsel akan lebih mudah memberikan dukungan terhadap fitur tersebut dari perangkat lunak.

2.12 Siklus Hidup Android

Aplikasi Android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-

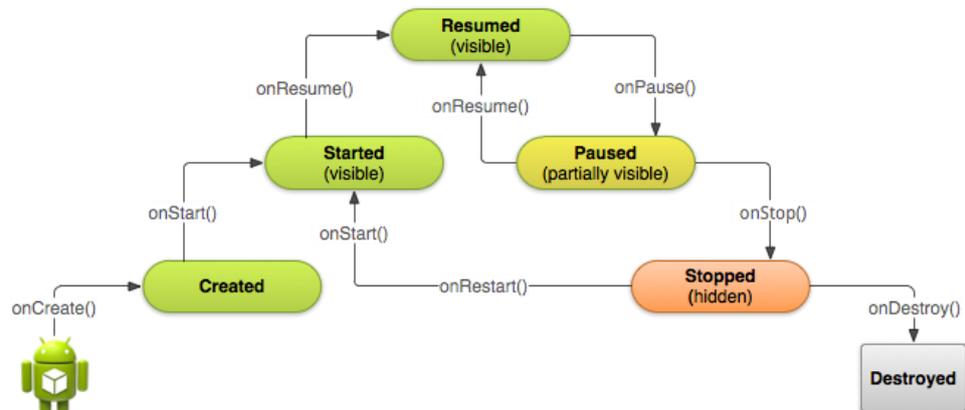
fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada tabel 2.1, dan klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan[18].

Tabel 0.1 Komponen Aplikasi Android

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	BroadcastReceiver	Trigger alarm upon event
Store and retrieve data	ContentProvider	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*[18].

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi `main()`, sistem android 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 android 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 pengguna[16]. Siklus hidup Android dapat dilihat pada gambar 2.4 di bawah ini.



Gambar 2.4 Siklus Hidup Android

Sumber: <https://developer.android.com/training/basics/activity-lifecycle/starting?hl=id>

Terdapat beberapa *state* dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.4, akan tetapi hanya beberapa dari *state* tersebut yang menjadi statis diantaranya:

1. Resumed

Resumed terjadi ketika aplikasi berjalan setelah *state paused*. State ini akan menjalankan perintah program yang ditulis pada method `onResume()`[16].

2. Paused

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam *state* ini[16].

3. Stopped

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* di *background*[16]. *State* lain seperti Created dan Started bersifat sementara dan sistem dengan

cepat menjalankan *state* berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem `OnCreate()` dipanggil, dengan cepat sistem akan memanggil method `OnStart()`, kemudian diikuti oleh `onResume()`[18].

2.12.1 Fitur

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah:

1. *Multi-Process* dan *App Widgets*

Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara *multitasking*. Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan game, proses lain dapat berjalan di belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan[16].

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti *homescreen*. *App widgets* dapat menjalankan proses request seperti musik streaming atau mendeteksi suhu ruangan secara *background*[16]. *Multi-process* dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar[18].

2. *Touch Gestures* dan *Multi-touch*

Touchscreen adalah *user interface* intuitif yang digunakan banyak *smartphone* di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. *Multi-touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu, Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut[18].

3. *Hard dan Soft Keyboard*

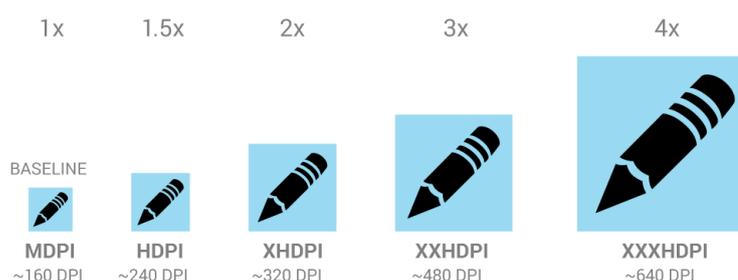
Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi[18].

2.12.2 Prinsip Desain

Android memiliki beberapa prinsip desain yang dapat menjadi acuan dalam membuat desain aplikasi android diantaranya adalah

A. *Multiple Assets*

Android mendukung jutaan smartphone, tablet dan perangkat lain dalam berbagai ukuran layar dan ukuran, untuk itu *Multiple Assets* sangat disarankan digunakan untuk mengatasi fragmentasi pada android. Seperti ilustrasi pada gambar 2.5 tentang Klasifikasi Ukuran Ikon, android menciptakan beberapa klasifikasi ukuran icon yaitu MDPI, HDPI, XHDPI, XXHDPI dan XXXHDPI. MDPI dan HDPI dikhususkan untuk icon yang akan digunakan pada device berukuran *smartphone* sedangkan untuk XHDPI, XXHDPI dan XXXHDPI digunakan pada *device* berukuran tablet[19].



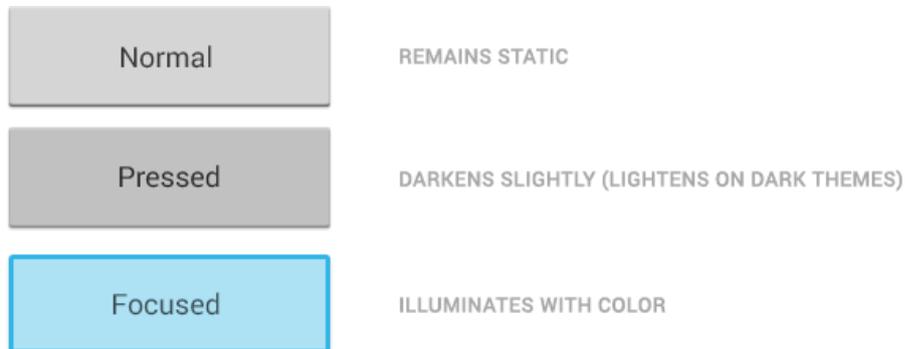
Gambar 2.5 Klasifikasi Ukuran Ikon

Sumber : <https://developer.android.com/google-play/resources/icon-design-specifications?hl=id>

B. *Touch Feedback*

Touch Feedback dalam android digunakan sebagai respon setiap objek yang ditekan pengguna. Hal ini bertujuan untuk memberi tahu pengguna objek mana

yang berinteraksi dengan pengguna[19]. Ilustrasi tentang *Touch Feedback* dapat dilihat pada gambar 2.6.



Gambar 2.6 Touch Feedback

Sumber : <http://www.androiddocs.com/design/style/touch-feedback.html>

C. *Pattern Gesture*

Touch gesture berguna untuk meningkatkan experience pengguna dalam menggunakan aplikasi. Terdapat beberapa *gesture* yang didukung oleh android diantaranya adalah[19]:

1. *Touch*

Touch adalah *gesture default* sebagai *event* untuk item tertentu yang paling sering digunakan dalam perangkat android, hampir seluruh kontrol utama aplikasi ini menggunakan *touch*.

2. *Long Press*

Biasanya digunakan untuk seleksi data, dengan *gesture* ini dimungkinkan untuk memilih satu atau lebih item dalam sebuah tampilan dan menjalankan suatu fungsionalitas tertentu.

3. *Swipe or Drag*

Swipe adalah menyentuh sebuah titik pada layar dan menggerakkan jari yang tetap tersentuh pada layar ke titik lain pada layar. *Swipe* biasanya dilakukan untuk kebutuhan berpindah layar atau tergantung dari aplikasi. *Swipe* dapat dilakukan dari dan ke arah mana saja.

4. *Double Touch*

Pada *smartphone* dan tablet android, melakukan dua kali *tapping* secara berturut-turut pada satu obyek, fungsinya berbeda dengan double klik mouse komputer. Pada Android, teknik ini biasanya dipakai untuk melakukan zoom in dan zoom out atau memperbesar / memperkecil sebuah obyek gambar.

5. *Pinch Open*

Teknik lain yang biasa digunakan adalah dengan menggunakan dua jari, di mana kedua jari tersebut menyentuh dua titik pada layar yang terpisah di mana ujung dari dua jari tersebut tidak bersentuhan, kemudian kedua jari tersebut sambil tetap menyentuh layar bergerak saling mendekati. Teknik ini digunakan untuk membuka aplikasi tertentu.

6. *Pinch Close*

Teknik *Pinch Close* adalah kebalikan dari *Pinch Open* dimana *spread* dilakukan dengan berawal dari dua jari yang bersentuhan pada ujungnya dan ditempelkan sebuah titik yang sama pada layar, kemudian kedua jari tersebut bergerak memisahkan atau menjauhi satu sama lain. Gerakan ini untuk menutup layar.

2.13 *User-Generated Content*

User-generated Content atau disingkat UGC, dikenal juga dengan istilah *consumer-generated media* (CGM) atau *user-created content* (UCC) adalah merujuk pada berbagai tipe konten materi atau media yang tersedia untuk umum dan diproduksi oleh *end-user* atau para non-media profesional. Kemajuan teknologi web memungkinkan konten website tidak lagi dimonopoli oleh profesional media, melainkan dapat dibuat oleh para penggunanya. UGC merupakan salah satu ciri dominan Web 2.0. Contoh penerapan pada UGC adalah YouTube hampir semua konten yang dimiliki situs tersebut dibuat dan diupload oleh penggunanya[20].

Konsep UGC pada dasarnya telah banyak merubah cara berinteraksi pengguna dengan internet begitu juga dalam media periklanan. Bagi media

periklanan jejaring sosial dengan konsep UGC memiliki potensi besar menyediakan market yang lebih terarah dan terpusat bagi mereka[20].

2.14 *Web Services*

W3C mendefinisikan *web service* sebagai sebuah *software* aplikasi yang dapat teridentifikasi oleh URI dan memiliki *interface* yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan *software* aplikasi yang lain dengan menggunakan *message* berbasis XML atau JSON melalui protokol internet. *Web service* adalah sebuah *software* aplikasi yang tidak terpengaruh oleh platform, menyediakan *method-method* yang dapat diakses oleh network[21]. *Web Service* juga akan menggunakan XML untuk pertukaran data, khususnya pada dua *entities* bisnis yang berbeda. Beberapa karakteristik dari *web service* adalah:

1. *Message-based*
2. *Standards-based*
3. *Programming language independent*
4. *Platform-neutral*

Beberapa *key standard* di dalam *web service* adalah: JSON, XML, SOAP, WSDL and UDDI.

2.15 **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 Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena

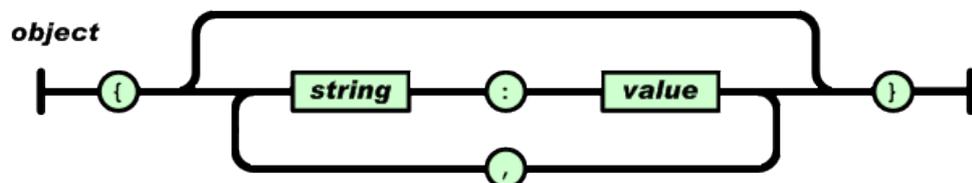
sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur[22]:

- 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 berkunci (*keyed list*), atau *associative array*.
- B. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*)[22].

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

A. Objek

Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON[22]. Ilustrasi objek JSON dapat dilihat pada gambar 2.7 di bawah ini.

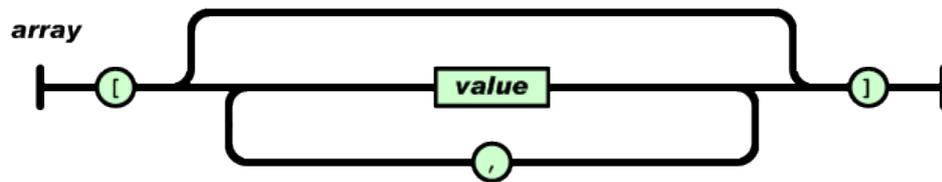


Gambar 2.7 Objek JSON

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

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). Larik dalam JSON dapat digunakan sebagai *value* dari JSON *object* hal ini dapat berguna jika JSON menyimpan data bertingkat[22]. Ilustrasi tentang larik / *array* JSON dapat dilihat pada gambar 2.8 di bawah ini.



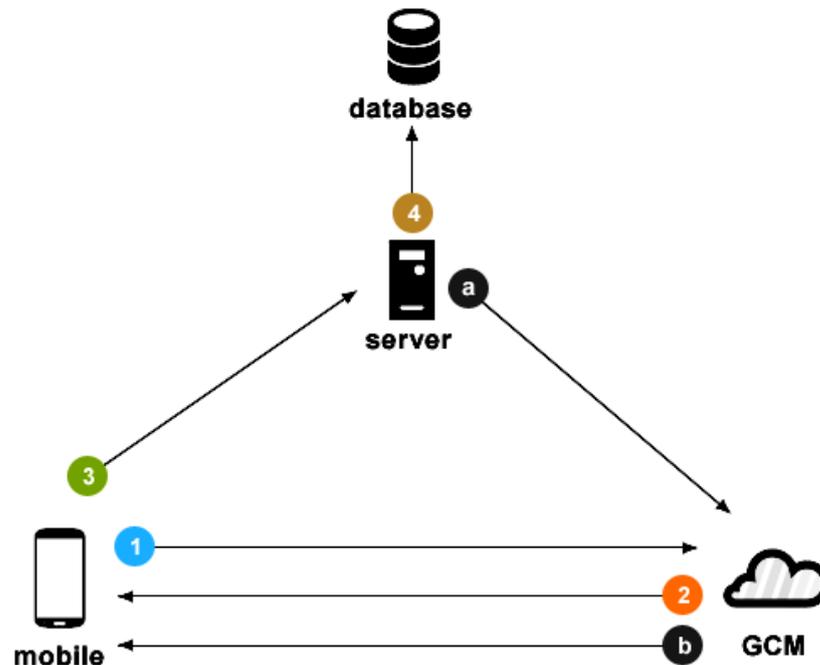
Gambar 2.8 Array JSON

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

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Boolean dan nilai NULL[22].

2.16 Google Cloud Messaging

Google Cloud Messaging adalah teknologi untuk membantu pengembang mengirim data dari server menuju *device* android secara langsung melalui server google, pemanfaatan teknologi ini biasanya digunakan untuk *push messaging*. Pengiriman pesan menggunakan teknologi ini mengharuskan adanya API key, key ini terkait langsung dengan akun google yang digunakan untuk mempublikasikan aplikasi melalui google play[18]. Ilustrasi tentang *Google Cloud Message* dapat dilihat pada Gambar 2.9 tentang GCM Model.



Gambar 2.9 GCM Model

Sumber : <https://jmsliu.com/3009/add-push-notification-in-android-rss-reader-using-google-cloud-messaging.html>

Berdasarkan ilustrasi dari Gambar 2.9, terdapat dua server yang terlibat dalam proses *push messaging*, pada setiap server yang terlibat terdapat proses pengiriman data yang berbeda-beda diantaranya[18]:

1. Device android mengirimkan sender id dan application id menuju GCM Server untuk registrasi.
2. Setelah berhasil registrasi, GCM server akan mengirimkan registration id menuju device android.
3. Setelah device android mendapatkan registration id, device tersebut akan mengirimkan registration id menuju server aplikasi.
4. Server aplikasi akan menyimpan registrasi id yang dikirimkan untuk digunakan sebagai id saat menggunakan *push messaging*.

2.17 Java

Java adalah bahasa pemrograman yang dapat dijalankan dimanapun dan di sembarang platform apapun di beragam lingkungan: *internet, intranet, consumer electronic products, dan computer applications* [23]. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi berbasis java pada umumnya di compile menjadi p-code (bytecode) dan dapat berjalan pada beragam *Java Virtual Machine* (JVM) [23].

Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi. Berikut contoh dari bahasa pemrograman java:

```
Public class HelloWorld () {  
    Public static void main (String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

2.18 PHP

PHP (HyperText Preprocessor) adalah sebuah bahasa utama *script server-side* yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi dekstop. PHP bersifat *open-source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, dalam artian bisa digunakan pada sistem operasi Windows ataupun Linux[17].

Tentunya bahasa pemrograman PHP berbeda dengan HTML, pada PHP Script/kode yang di buat tidak dapat di tampilkan pada halaman/muka *website* begitu saja, tapi harus diproses terlebih dahulu oleh web server lalu di tampilkan dalam bentuk halaman website di *web browser*, Script PHP juga dapat di sisipkan pada HTML dan script PHP selalu diawali dengan `<?php` dan di akhiri dengan `?>`. Berikut contoh penggunaan PHP :

```
<?php
    echo "Hello World...!!!";
    echo "<br />";
    print "Hello Again World...!!!";
    print "<br /><br />";
?>
```

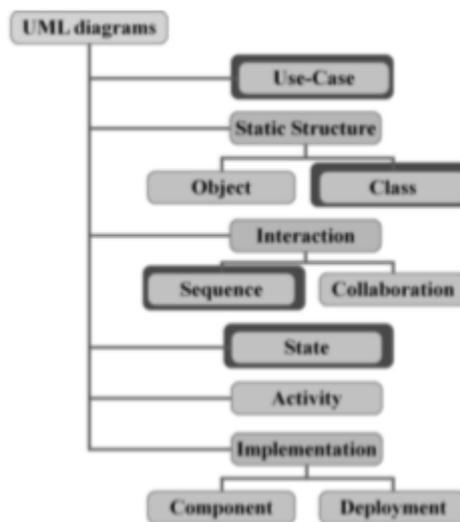
2.19 Object Oriented Analysis Desain

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa *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[24].

2.19.1 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak [25]

Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem, dan berbagai macam domain aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan masing-masing diagram. UML bertujuan menyatukan teknik-teknik pemodelan berorientasi objek-objek menjadi terstandarisasi[25]. Diagram UML pada gambar 2.10 dapat digambarkan sebagai berikut :



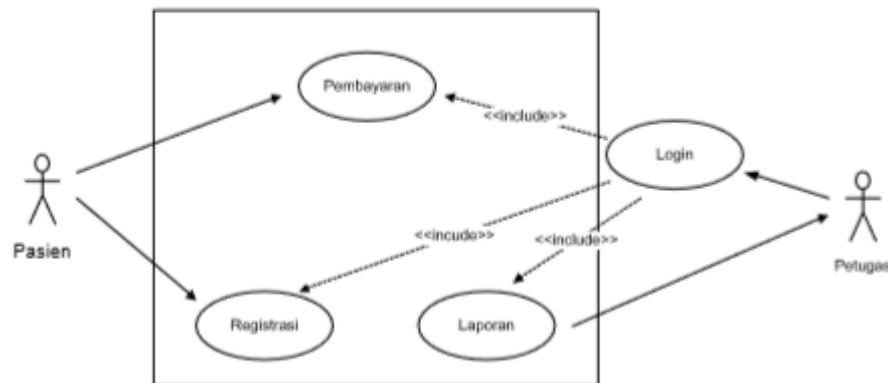
Sumber Gambar : JT-IBSI [12]

Gambar 2.10 Diagram UML

Untuk mendapatkan banyak pandangan terhadap sistem yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain[26]:

- a. *Use case Diagram*

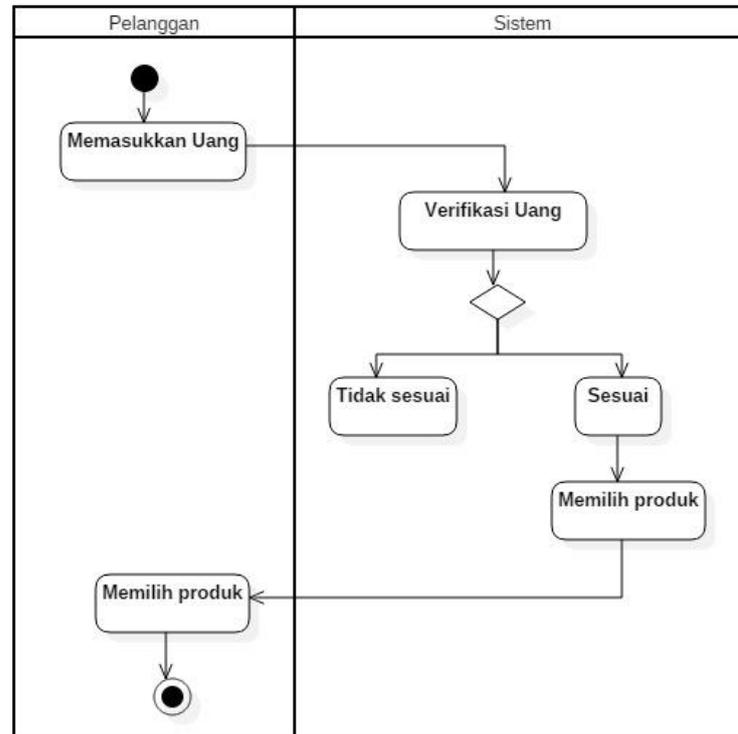
Use case diagram menyajikan interaksi antara use case dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai seperti pada gambar 2.11



Gambar 2.11Contoh Use Case Diagram

b. *Activity* Diagram

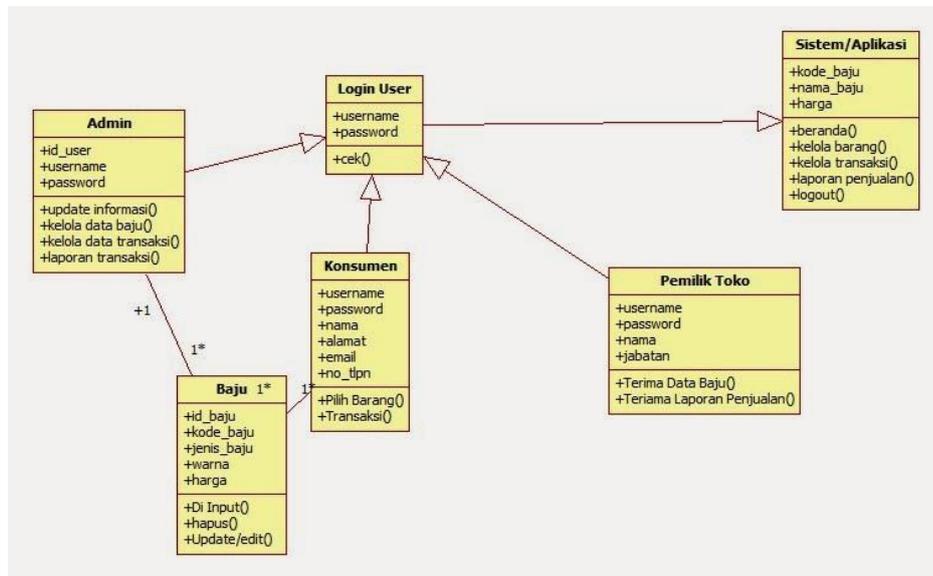
Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. *Activity* diagram juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan serta rancang menu yang ditampilkan pada perangkat lunak seperti pada gambar 2.12 :



Gambar 2.12 Contoh Activity Diagram

c. *Class Diagram*

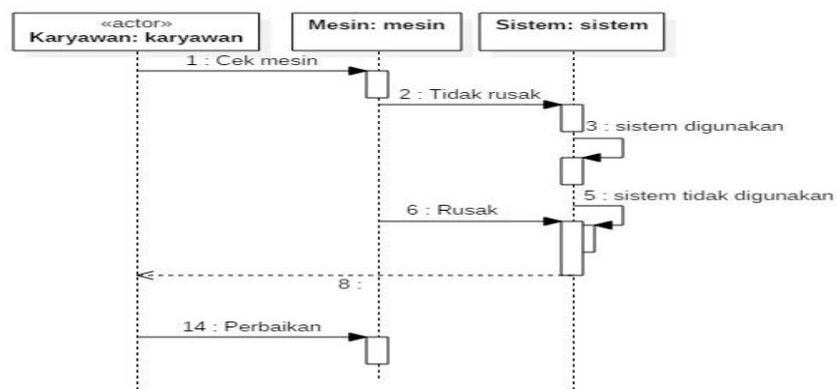
Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class* diagram memiliki apa yang disebut atribut dan metode atau operasi. *Class* diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. *Class Diagram* juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut seperti yang terlihat pada gambar 2.13 :



Gambar 2.13 Contoh Class Diagram

d. *Sequence Diagram*

Sequence diagram merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu dan objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Pada gambar 2.14 terlihat contoh *sequence diagram*



Gambar 2.14 Contoh Sequence Diagram

2.20 Object Relational Mapping

Object Relational Mapping (ORM) adalah salah satu teknik untuk memetakan basis data relasional ke model objek. Pemetaan dapat dikategorikan menjadi 4, yaitu[24]:

1. *Basic mapping*
2. *Relationship mapping*
3. *Composition mapping*
4. *Inheritance mapping*