

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

Penelitian dilakukan di Sekolah Khusus (SKH) PGRI Rangkasbitung yang bertempat di Jl. Siliwangi Pasir Ona Rangkasbitung Lebak, Rangkasbitung Timur, Kec. Rangkasbitung, Kab. Lebak Prov. Banten.



Gambar 2.1 Yayasan SKH PGRI Rangkasbitung

2.1.1 Sejarah SKH PGRI Rangkasbitung

Awal mula berdirinya yayasan SKH PGRI Rangkasbitung ini karena keprihatinan ibu Nurkhayati yang melihat banyak anak berkebutuhan khusus yang masih kurang mendapatkan perhatian khusus seperti pendidikan, jadi ibu Nurkhayati tersebut mendirikan yayasan ini. Bermotivasi niat tulus dan ikhlas ingin membantu anak-anak luar biasa di sekitar kampungnya berdirilah Yayasan SKH PGRI Rangkasbitung ini, yayasan dengan bangunan sederhana dibangun diatas tanah seluas 600 (m²), dan jumlah kelas yang terbatas. Yayasan ini berdiri pada tanggal 18 Februari 1997 dengan tujuan memberikan pelayanan pendidikan pada anak-anak Tuna Netra, Tuna Rungu / Tuna Wicara, Tuna Grahita dan Tuna Daksa.

2.1.2 Logo Sekolah

Logo adalah simbol yang mengandung makna, setiap sekolah, instansi maupun organisasi mempunyai logo sebagai identitas diri. Berikut ini adalah logo dari SKH PGRI Rangkasbitung yang di tunjukan pada Gambar 2.2.



Gambar 2.2 Logo SKH PGRI Rangkasbitung

2.1.3 Visi dan Misi SKH PGRI Rangkasbitung

A. Visi Sekolah

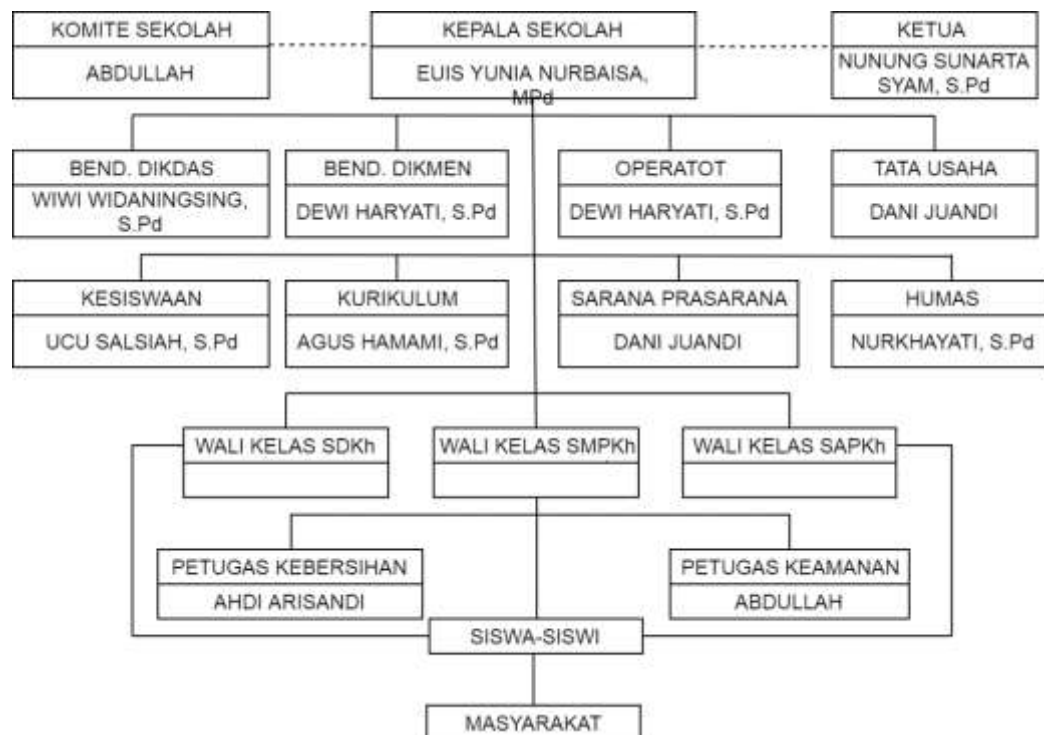
Berbudi Luhur, Berkarakter dan Terampil.

B. Misi Sekolah

1. Melaksanakan pembelajaran dan bimbingan secara efektif sesuai dengan bakat, minat dan potensi yang dimiliki sehingga dapat berkembang secara optimal.
2. Menumbuhkan nilai-nilai karakter sehingga mencerminkan tindakan yang positif.
3. Menerapkan manajemen partisipatif dengan melibatkan seluruh warga sekolah dan komite sekolah.
4. Memberdayakan potensi tenaga pendidik dan kependidikan, siswa dan masyarakat.

2.1.4 Struktur Organisasi

Gambaran struktur organisasi SKH PGRI Rangkasbitung dapat dilihat pada Gambar 2.3.



Gambar 2.3 Struktur Organisasi

2.2 Landasan Teori

Landasan teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi pembelajaran mengenal warna, huruf dan angka di SKH PGRI Rongkasbitung berbasis android.

2.2.1 Tunagrahita

Anak tunagrahita adalah individu yang secara signifikan memiliki intelegensi dibawah intelegensi normal. Menurut *American Asociation on Mentar Deficiency* mendefinisikan Tunagrahita sebagai suatu kelainan yang fungsi intelektualnya umumnya di bawah rata-rata, yaitu IQ 84 ke bawah. Biasanya anak-anak tunagrahita akan mengalami kesulitan dalam “*Adaptive Behavior*” atau penyesuaian perilaku. Hal ini berarti anak tunagrahita tidak dapat mencapai kemandirian yang sesuai dengan ukuran (standard) kemandirian dan tanggung jawab sosial anak normal yang lainnya dan juga akan mengalami masalah dalam keterampilan akademik dan berkomunikasi dengan kelompok usia sebaya [7].

Masyarakat pada umumnya mengenal tunagrahita sebagai retardasi mental atau terbelakang mental atau idiot. Rachmayana, D. (2016) mengemukakan bahwa

tunagrahita berarti suatu keadaan yang ditandai dengan fungsi kecerdasan umum yang berada dibawah rata-rata disertai dengan berkurangnya kemampuan untuk menyesuaikan diri (berperilaku adaptif), yang mulai timbul sebelum usia 18 tahun. Ia juga mengatakan bahwa orang-orang secara mental mengalami keterbelakangan, memiliki perkembangan kecerdasan (IQ) yang lebih rendah dan mengalami kesulitan dalam proses belajar serta adaptasi sosial [5].

2.2.1.1 Karakteristik Anak Tunagrahita

Untuk mempermudah dalam membuat program dan melaksanakan layanan pendidikan bagi anak tunagrahita sebaiknya para guru atau pendidik mengenal karakteristik dan permasalahan anak tunagrahita, berikut akan diuraikan beberapa karakteristik anak tunagrahita, yaitu [3] :

a. Keterbatasan Intelegansi

Intelegansi merupakan fungsi yang kompleks yang dapat diartikan sebagai kemampuan untuk mempelajari informasi dan keterampilan-keterampilan menyesuaikan diri dengan masalah-masalah dan situasi-situasi kehidupan baru, belajar dari pengalaman masa lalu, berpikir abstrak, kreatif, dan dapat menilai secara kritis, menghindari dari kesalahan-kesalahan, mengatasi kesulitan-kesulitan, dan kemampuan merencanakan masa depan. Anak tunagrahita memiliki kekurangan dalam hal tersebut. Kapasitas belajar anak tunagrahita terutama yang bersifat abstrak seperti belajar dan berhitung, menulis dan membaca juga terbatas. Kemampuan belajarnya cenderung tanpa pengertian atau cenderung belajar dengan membeo.

b. Keterbatasan Sosial

Disamping memiliki keterbatasan intelegansi, anak tunagrahita juga memiliki kesulitan dalam mengurus diri sendiri dalam masyarakat, oleh karena itu mereka memerlukan bantuan. Anak tunagrahita cenderung berteman dengan anak yang lebih muda usianya, ketergantungan terhadap orang tua sangat besar, tidak mampu memikul tanggung jawab sosial dengan bijaksana, sehingga mereka harus selalu dibimbing dan diawasi. Mereka juga mudah dipengaruhi dan cenderung melakukan sesuatu tanpa memikirkan akibatnya.

c. Keterbatasan Fungsi-fungsi Mental Lainnya

Anak tunagrahita memerlukan waktu lebih lama untuk menyelesaikan reaksi pada situasi yang baru dikenalnya. Mereka memperlihatkan reaksi terbaiknya bila mengikuti hal-hal yang rutin dan secara konsisten dialaminya dari hari ke hari. Anak tunagrahita tidak dapat menghadapi sesuatu kegiatan atau tugas dalam jangka waktu yang lama. Anak tunagrahita memiliki keterbatasan dalam penguasaan bahasa. Mereka bukannya mengalami kerusakan artikulasi, akan tetapi pusat pengolahan (perbendaharaan kata atau kosa kata) yang kurang berfungsi sebagaimana mestinya. Karena alasan itu mereka membutuhkan kata-kata konkret yang seritn di dengarnya. Selain itu perbedaan dan persamaan harus ditunjukkan secara berulang-ulang. Latihan sederhana seperti mengajarkan konsep besar dan kecil, keras dan lemah, pertama, kedua, dan terakhir, perlu menggunakan pendekatan yang konkret. Selain itu, anak tunagrahita kurang mampu untuk mempertimbangkan sesuatu, membedakan antara yang baik dan yang buruk, dan membedakan yang benar dan salah. Ini semua karena kemampuannya terbatas sehingga anak tunagrahita tidak dapat membayangkan terlebih dahulu konsekuensi dari suatu perbedaan.

Dapat disimpulkan dari karakteristik di atas bahwa anak tunagrahita memiliki 3 karakteristik yang sangat berhubungan dengan bahasa, komunikasi, dimana kita tau bahwa anak tunagrahita memiliki permasalahan dalam keterlambatan berbahasa, sedangkan bahasa merupakan alat komunikasi sehingga perlu adanya metode untuk menambah pembendaharaan kata, agar anak tunagrahita dapat berkomunikasi dengan orang lain.

2.2.2 Media Pembelajaran

Menurut Depdiknas (2003) istilah media berasal dari bahasa latin yang merupakan bentuk jamak dari “medium” yang secara harafiah berarti perantara atau pengantar. Makna umumnya adalah segala sesuatu yang dapat menyalurkan informasi dari sumber informasi kepada penerima informasi. Proses belajar mengajar pada dasarnya juga merupakan proses komunikasi, sehingga media yang digunakan dalam pembelajaran disebut media pembelajaran. Media pembelajaran

merupakan bagian dari sumber belajar yang merupakan kombinasi antara perangkat lunak (bahan belajar) dan perangkat keras (alat belajar) [8].

Secara umum, manfaat media dalam proses pembelajaran adalah memperlancar interaksi antara guru dan siswa sehingga kegiatan pembelajaran akan lebih efektif dan efisien. Tetapi secara khusus ada beberapa manfaat media yang lebih rinci. Kemp dan Dayton (dalam Depdiknas, 2003) mengidentifikasi beberapa manfaat media dalam pembelajaran yaitu [8]:

1. Penyampaian materi pelajaran dapat diseragamkan.
2. Proses pembelajaran menjadi lebih jelas dan menarik
3. Proses pembelajaran menjadi lebih interaktif
4. Efisiensi dalam waktu dan tenaga
5. Meningkatkan kualitas hasil belajar siswa
6. Media memungkinkan proses belajar dapat dilakukan di mana saja dan kapan saja
7. Media dapat menumbuhkan sikap positif siswa terhadap materi dan proses belajar
8. Mengubah peran guru ke arah yang lebih positif dan produktif.

2.2.3 Edutainment

Edutainment dari segi bahasa berarti pendidikan yang menyenangkan, sedangkan dari segi makna *edutainment* adalah suatu proses pembelajaran yang didesain sedemikian rupa sehingga muatan pendidikan dan hiburan dapat dikombinasikan secara harmonis sehingga pembelajaran lebih menyenangkan, pembelajaran yang menyenangkan biasanya dilakukan dengan humor atau permainan (*game*), bermain peran (*role play*) dan demonstrasi [4].

Edutainment merupakan kombinasi, fungsi pendidikan dan konten dengan bentuk hiburan, yang bertujuan untuk menciptakan lingkungan belajar yang menarik. Dengan metode *edutainment*, proses pembelajaran di sekolah akan menyenangkan sehingga anak tidak merasa jenuh atau bosan, tapi menjadi lebih bermakna dan mampu mengaitkan materi dengan kehidupan sehari-hari [4].

2.2.4 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran dan penggunaan. Secara istilah aplikasi merupakan suatu program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpaku pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan [6].

Pada saat ini, aplikasi telah berkembang menjadi beberapa *platform*. Terdapat tiga *platform* utama pada pengembangan sebuah aplikasi, yaitu aplikasi berbasis mobile, web dan desktop.

2.2.5 Android Studio

Android Studio adalah IDE (*Integrated Development Environment*) yang resmi untuk pengembangan aplikasi android yang bersifat *open source* atau gratis [6]. Android studio diluncurkan oleh Google pada tanggal 16 Mei 2013. Pengumuman tersebut dilakukan di event Google I/O Conference untuk tahun 2013. Android studio menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi android.

Android Studio dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT (*Android Development Tools*) plugin. Android studio mempunyai beberapa fitur, yaitu sebagai berikut [6]:

1. Projek berbasis pada gradle build.
2. Refactory dan pembenahan bug yang cepat.
3. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung proguard and app-singning untuk keamanan.
5. Mempunyai GUI aplikasi android yang lebih mudah.

Didukung oleh *Google Cloude Platform* untuk setiap aplikasi yang dikembangkan.

2.2.6 Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware atau aplikasi [6]. Android merupakan sistem operasi *smartphone* yang paling banyak digunakan pada saat ini. Android menyediakan *platform* secara gratis atau terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Pada awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru dalam pengembangan peranti lunak untuk ponsel atau *smartphone*. Kemudian dalam pengembangan android ini, Google Inc. membentuk *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia [6].

Pada saat perilisan perdana android pada 5 November 2007, android bersama *Open Handset Alliance* menyatakan mendukung pengembangan aplikasi open source pada perangkat mobile. Google juga merilis kode-kode android dibawah linesi apache, sebuah lisensi perangkat lunak dan open *platform* perangkat lunak [6].

Dalam perkembangannya, Android saat ini sudah sampai pada generasi ke 9, yaitu Android 9.0 Pie. Perkembangan versi android dari 2008 hingga sekarang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Versi Android

No	Versi Android	Nama
1	1.0	Astro
2	1.1	Bender
3	1.5	Cupcake
4	1.6	Donut
5	2.0 – 2.1	Éclair
6	2.2 – 2.23	Froyo
7	2.3 – 2.37	Gingerbread
8	3.0 – 3.2.6	Honeycomb

9	4.0 – 4.0.4	Ice Cream Sandwich
10	4.1 – 4.3.1	Jelly Bean
11	4.4 – 4.4.4	Kitkat
12	5.1 – 5.1.1	Lolipop
13	6.0 – 6.0.1	Marshmallow
14	7.1 – 7.1.2	Nougat
15	8.0 – 8.1	Oreo
16	9	Pie

Berikut adalah penjelasan versi android berdasarkan waktu rilisnya:

1. Android 1.0 Astro (Alpha)

Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008. Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android 1.1 Bender (Beta)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Petit Four", meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur.

3. Android 1.5 Cupcake

Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27. Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut ("Cupcake"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya. Pembaruan pada versi ini termasuk beberapa fitur baru dan perubahan UI.

4. Android 1.6 Donut

Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29. versi ini hadir untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

5. Android 2.0 – 2.1 Éclair

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29. Versi 2.0.1 rilis tanggal 3 desember 2009 dan versi 2.1 rilis tanggal 12 januari 2010. Beberapa penambahan fitur pada versi ini seperti Bluetooth, flash pada kamera, digital zoom pada kamera, multi touch, live wallpaper dan lainnya.

6. Android 2.2 – 2.2.3 Froyo

Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, yang berbasis kernel Linux 2.6.32.

7. Android 2.3 – 2.3.7 Gingerbread

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.

8. Android 3.0 – 3.2 Honeycomb

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

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.

10. Android 4.1 – 4.3 Jelly Bean

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). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. 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

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'.

12. Android 5.0 – 5.1 Lollipop

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.

1. Desain antarmuka (tampilan) yang dinamakan "Material Design".
2. 64-bit ART compiler
3. Project volta, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.
4. 'factory reset protection'. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).

13. Android 6.0 Marshmallow

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang,

menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, *Bluetooth* dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan autentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan autentikasi 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.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API "MNC".

14. Android 7.0 – 7.1 Nougat

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 – 8.1 Oreo

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.

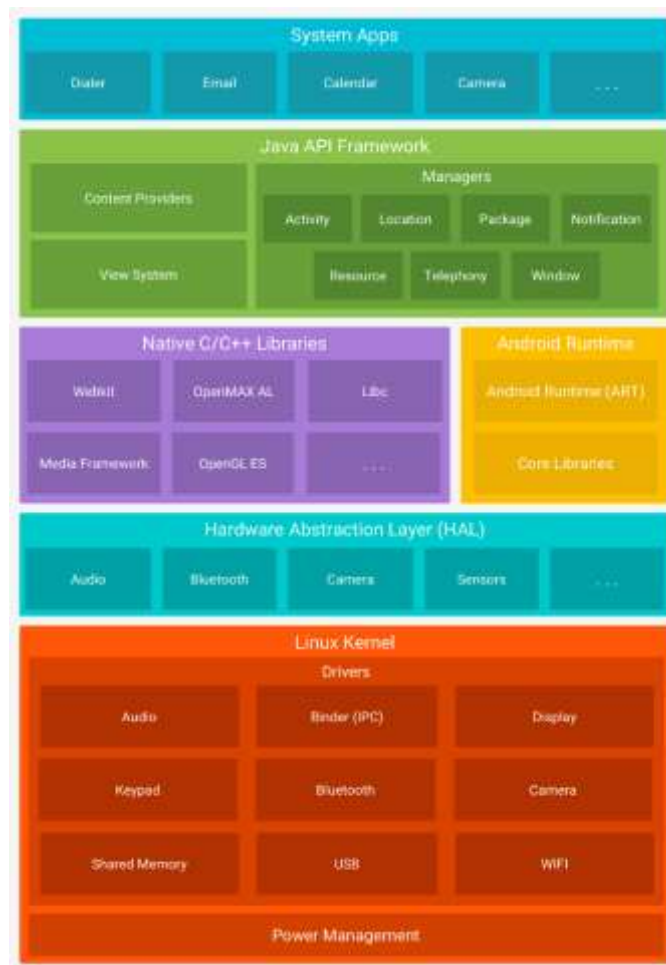
16. Android 9.0 Pie

Terakhir ada Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak ubahan, terutama untuk HP dengan desain baru.

Misal Android 9.0 Pie memberikan navigasi berupa gesture yang menggantikan tombol fisik Home, Back dan Recent Apps. Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan hingga sistem screenshoot terbaru yang lebih memudahkan.

2.2.6.1 Arsitektur Android

Android adalah tumpukan perangkat lunak berbasis *linux* sumber terbuka yang dibuat untuk berbagai perangkat dan faktor bentuk [7]. Android memiliki beberapa komponen utama yang disebut dengan Arsitektur *Platform*. Komponen utama pada android dapat dilihat pada Gambar 2.4.



Sumber Gambar : https://developer.android.com/guide/platform/images/android-stack_2x.png?hl=ID

Gambar 2.4 Arsitektur Android

a. Linux Kernel

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

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

b. Hardware Abstraction Layer (HAL)

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.

c. Android Runtime

Untuk perangkat 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:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT).
2. Pengumpulan sampah (GC) yang dioptimalkan.
3. Dukungan debug yang lebih baik, mencakup *profiler* sampling terpisah, pengecualian diagnostik *detail* 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.

d. Pustaka C/C++ Asli

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.

e. Kerangka Kerja Java API

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:

1. 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.
2. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout.
3. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status.
4. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum.
5. 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.

f. Aplikasi Sistem

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.

2.2.6.2 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 Astro (1.0), Bender (1.1), 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 – 5.1), Marshmallow (6.0), hingga yang terbaru adalah Nougat (7.0 – 7.1), Android O (8.0 – 8.1) dan Android Pie (9.0).

Berikut adalah versi-versi android berdasarkan waktu rilisnya:

1. Android 1.0 Astro (Alpha)

Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008. Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android 1.1 Bender (Beta)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Petit Four", meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur.

3. Android 1.5 Cupcake

Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27. Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut ("Cupcake"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya. Pembaruan pada versi ini termasuk beberapa fitur baru dan perubahan UI.

4. Android 1.6 Donut

Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29. versi ini hadir untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

5. Android 2.0 – 2.1 Éclair

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29. Versi 2.0.1 rilis tanggal 3 desember 2009 dan versi 2.1 rilis tanggal 12 januari 2010. Beberapa penambahan fitur pada versi ini seperti Bluetooth, flash pada kamera, digital zoom pada kamera, multi touch, live wallpaper dan lainnya.

6. Android 2.2 – 2.2.3 Froyo

Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, yang berbasis kernel Linux 2.6.32.

7. Android 2.3 – 2.3.7 Gingerbread

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. Beberapa vendor mulai serius untuk menggarap perangkat dengan OS Android. Pada saat itu, Samsung dengan Galaxy series nya berperan besar dalam kesuksesan Android. Promosi yang luar biasa gencarnya membuat orang awam mulai mengenal sistem operasi Android. Bahkan saat itu sebagian besar orang beranggapan bahwa OS Android adalah milik Samsung karena kuatnya branding yang dilakukan oleh Samsung. Ini juga menjadi awal mula kedigdayaan Samsung di jagad smartphone.

8. Android 3.0 – 3.2 Honeycomb

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

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.

10. Android 4.1 – 4.3 Jelly Bean

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). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. 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

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'.

12. Android 5.0 – 5.1 Lollipop

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

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang,

menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, *Bluetooth* dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan autentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan autentikasi 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.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API "MNC".

14. Android 7.0 – 7.1 Nougat

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 – 8.1 Oreo

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

Terakhir ada Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak ubahan, terutama untuk HP dengan desain baru.

Misal Android 9.0 Pie memberikan navigasi berupa gesture yang menggantikan tombol fisik *Home*, *Back* dan *Recent Apps*.

Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan hingga sistem screenshoot terbaru yang lebih memudahkan kamu loh.

2.2.7 Web Service

Web Service adalah sekumpulan aplikasi logik beserta objek-objek dan metode-metode yang dimilikinya, yang terletak disuatu server yang terhubung ke internet sehingga dapat diakses menggunakan protokol HTTP (*Hypertext Transfer Protocol*) maupun SOAP (*Simple Object Access Protocol*) [6].

Dalam penggunaannya, *web service* dapat digunakan untuk memeriksa data user yang *login* ke sebuah website ataupun untuk digunakan pada transaksi perbankan online yang rumit [6].

Tujuan dari *web service* adalah untuk memudahkan beberapa aplikasi komponennya untuk saling terhubung dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat dengan *platform* dan tidak terikat dengan bahasa pemrograman yang digunakan [6].

2.2.8 Pengerian *Eclipse*

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (platform-independent). Berikut adalah sifat dari Eclipse [11]:

1. *Multi-platform*: Target sistem operasi Eclipse adalah *Microsoft Windows*, *Linux*, *Solaris*, *AIX*, *HP-UX* dan *Mac OS X*.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi test perangkat lunak, pengembangan *web*, dan lain sebagainya.

2.2.9 Corel Draw

Corel Draw adalah sebuah program komputer yang melakukan editing pada garis vektor. Program ini dibuat oleh Corel, sebuah perusahaan software yang berkantor pusat di Ottawa, Kanada. Corel draw memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan dalam bidang pulikasi atau percetakan ataupun pekerjaan di bidang lain yang membutuhkan proses visualisasi [12].

2.2.10 Java

Java merupakan bahasa pemrograman yang berorientasi objek yang diciptakan oleh Sun Microsystem pada tahun 1995 [13]. Java merupakan salah satu bahasa pemrograman yang banyak digunakan saat ini. Java dapat membuat seluruh bentuk aplikasi, baik dalam bentuk web, desktop, maupun mobile.

Java merupakan bahasa pemrograman *open source* yang artinya terbuka untuk umum sehingga para developer dapat menggunakan bahasa pemrograman secara gratis dalam membuat sebuah produk berupa aplikasi, baik dalam bentuk mobile, desktop, maupun web.

Java mempunyai beberapa karakteristik, yaitu sebagai berikut [13]:

1. Sederhana
2. Berorientasi objek
3. Dapat didistribusikan dengan mudah
4. Diinterpretasi oleh interpreter
5. Aman
6. Portable
7. Multithreading
8. Dinamis
9. Netral
10. Robust

2.2.11 MySQL

MySQL adalah suatu perangkat lunak *database* relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis di bawah lisensi GPL (*General Public License*) [11]. MySQL merupakan *database* yang bersifat gratis digunakan oleh siapa saja, akan tetapi tidak diperbolehkan untuk dijadikan produk turunan yang dijadikan closed source atau komersial. MySQL merupakan jenis basis data yang paling populer saat ini.

Berikut adalah contoh sintak pada MySQL:

1. Membuat *Database*

```
CREATE DATABASE peminjaman_buku;
```

2. Melihat *Database*

```
SHOW DATABASES;
```

3. Membuat Tabel

```
CREATE TABLE operator(
    id VARCHAR (20) NOT NULL,
    nama VARCHAR (50) NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at DATETIME NOT NULL,
    updated_at TIMESTAMP,
    PRIMARY KEY (id)
);
```

4. Melihat Tabel

```
SHOW TABLE;
```

5. Menghapus Tabel

```
DROP TABLE operator;
```

6. Menghapus *Database*

```
DROP DATABASE peminjaman_buku;
```

2.2.12 JSON

JSON (*Javascript Object Notation*) adalah format pertukaran data 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, menjadi JSON ideal sebagai bahasa pertukaran data [15]. JSON terbuat dari dua struktur [15]:

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

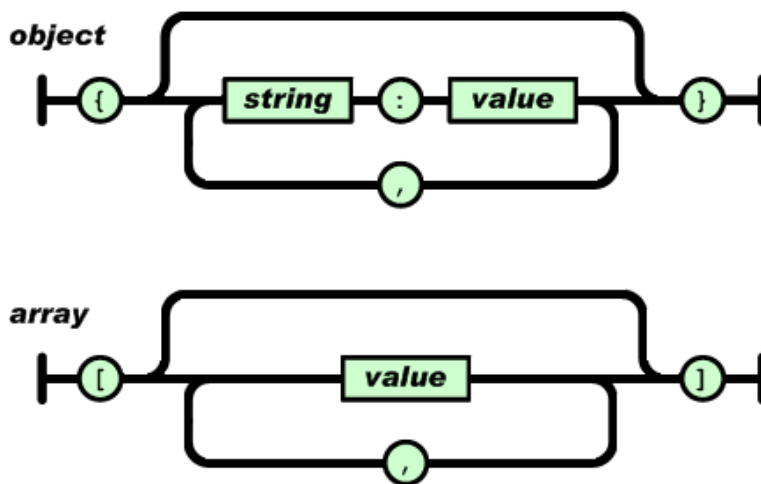
Stuktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [15].

JSON menggunakan bentuk sebagai berikut:

1. 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).

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

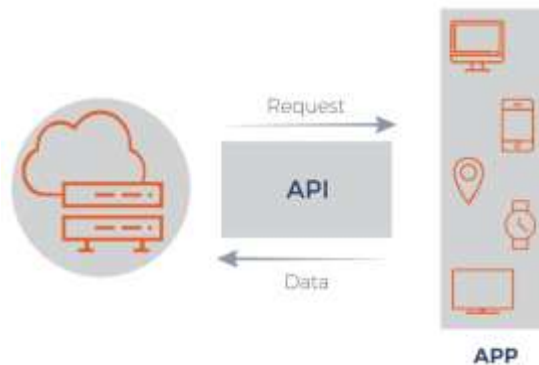


Sumber gambar : Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) [15]

Gambar 2.5 JSON Object dan Array

2.2.13 Application Programming Interface (API)

Application Programming Interface (API) adalah seperangkat definisi subrutin, protokol, dan alat untuk membangun perangkat lunak aplikasi. API yang bagus memudahkan pengembangan program komputer dengan menyediakan semua blok bangunan, yang kemudian disatukan oleh pemrogram. API mungkin untuk sistem berbasis *web*, sistem operasi, sistem basis data, perangkat keras komputer atau perpustakaan perangkat lunak.



Sumber gambar : <https://qatestlab.com/resources/knowledge-center/application-programming-interface/> [16]

Gambar 2.6 Application Programming Interface (API)

2.2.14 Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan merekomendasikan sistem perangkat lunak. Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET [6].

2.2.14.1 Diagram UML

Setiap diagram yang kompleks dapat dipandang dari sudut yang berbeda-beda, hal ini dapat membuat kita mendapatkan pemahaman secara menyeluruh. Dalam upaya untuk mendapatkan pemahaman secara menyeluruh, UML menyediakan sembilan jenis diagram yang dapat dikelompokkan sifat statis atau dinamis. Berikut adalah sembilan jenis diagram tersebut [6].

1. Diagram Kelas

Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi.

2. Diagram Objek

Diagram objek bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.

3. Use Case Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Sequence Diagram

Diagram ini bersifat dinamis. Diagram *sequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu.

5. Collaboration Diagram

Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan (message).

6. Statechart Diagram

Diagram ini bersifat dinamis. Diagram ini memperlihatkan state – state pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif.

7. Activity Diagram

Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas, antarmuka – antarmuka serta kolaborasi – kolaborasi.

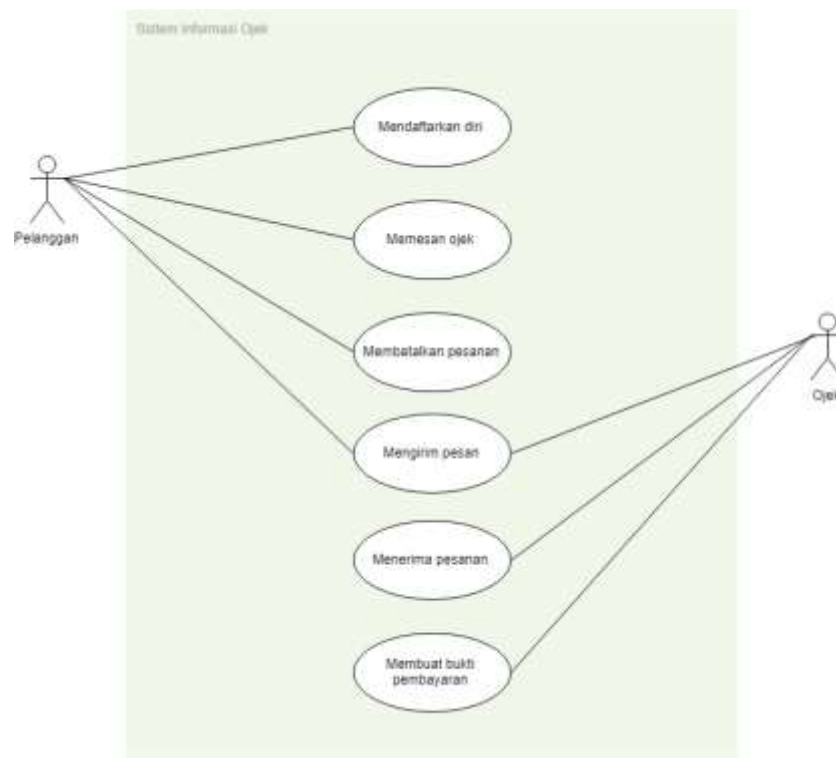
9. *Deployment Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat run time). Dengan ini memuat simpul – simpul (node) beserta komponen – komponen yang ada di dalamnya. Deployment diagram berhubungan erat dengan diagram kompoen dimana deployment diagram memuat satu atau lebih komponen – komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing).

Kesembilan diagram yang sudah disebutkan di atas tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semua dibuat sesuai dengan kebutuhan.

2.2.15 *Use Case Diagram*

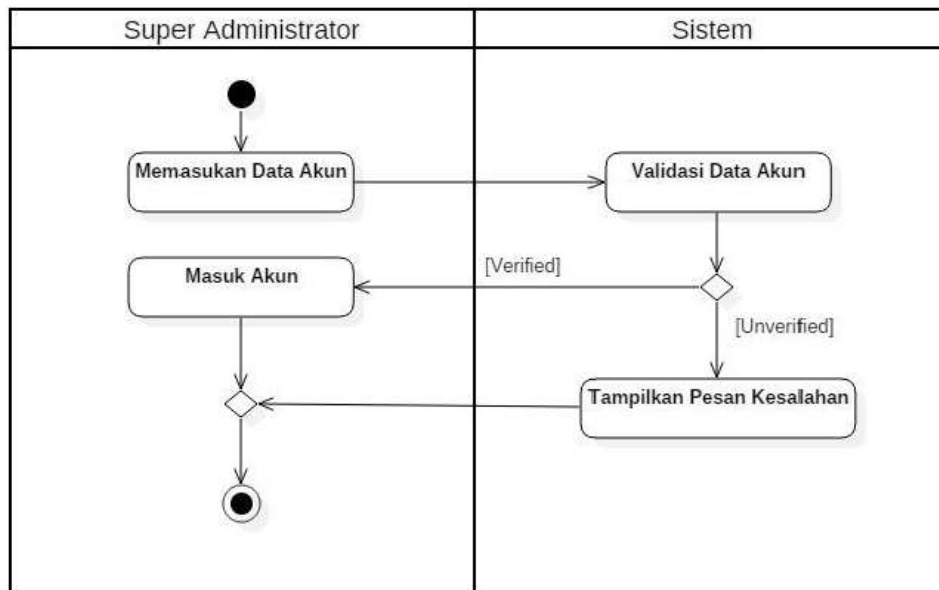
Use case diagram merupakan pemodelan untuk kelakuan atau tingkat laku suatu sistem informasi yang nantinya akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sistem informasi dan siapa saja yang berhak menggunakan aplikasi tersebut [6]. Contoh *use case diagram* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Use Case Diagram

2.2.16 Activity Diagram

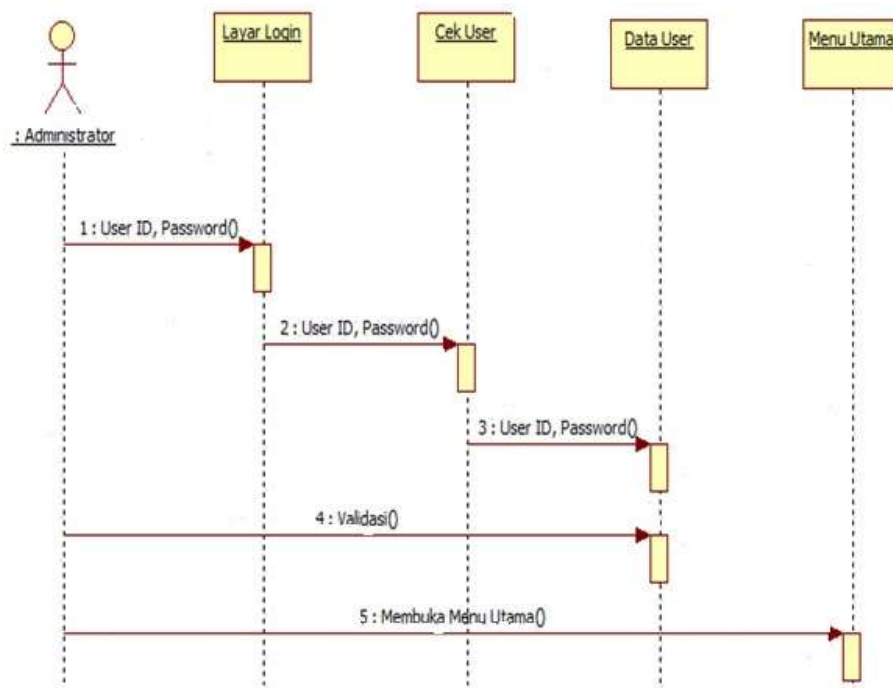
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis [6]. Contoh *activity diagram* dapat dilihat pada Gambar Gambar 2.8 .



Gambar 2.8 Activity Diagram

2.2.17 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek [6]. Contoh *sequence diagram* dapat dilihat pada Gambar 2.9 .



Gambar 2.9 Sequence Diagram

2.2.18 Black Box Testing

Black Box Testing yaitu menguji perangkat lunak dari segi fungsional untuk mengetahui fungsi-fungsi masukan dan keluaran dari perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan (Maulana et al. 2017:21) [17].

a. Keunggulan *Black Box Testing*

Meskipun dalam pelaksanaan *testing* kita dapat menguji keseluruhan fungsional perangkat namun formal *black box testing* yang sebenarnya kita dapat memilih *subset test* yang secara efektif dan efisien dapat menemukan cacat. Dengan cara ini *black box testing* dapat membantu memaksimalkan program [17].

b. Kelemahan *Black Box Testing*

Pengujian *black box testing* memiliki kelemahan ketika *tester* melakukan *black box testing*, *tester* tidak akan pernah yakin apakah perangkat lunak yang diuji telah benar-benar lolos pengujian. Hal ini terjadi karena kemungkinan masih ada beberapa jalur eksekusi yang belum pernah diuji. Untuk menemukan cacat perangkat lunak menggunakan *black box testing*, *tester* seharusnya membuat setiap kemungkinan [17].