BAB 2
LANDASAN TEORI

2.1 Multimedia

Multimedia dapat diartikan dalam berbagai perspektif. Multimedia bisa diartikan sebagai materi presentasi yang menggunakan kata-kata serta gambar. Materi yang disajikan dalam bentuk verbal yaitu berupa teks atau berupa lisan. Multimedia dapat diartikan sebagai gambar yang disajikan dalam bentuk grafik statis, yaitu berupa ilustrasi, grafik, foto atau peta, atau menggunakan grafik dinamis yang berupa animasi atau video [6].

2.1.1 Objek Multimedia

Terdapat beberapa objek dalam multimedia yaitu sebagai berikut:

1. Teks

   Teks merupakan objek multimedia yang mendasar dari pengolahan kata serta informasi berbasis multimedia.

2. Gambar

   Gambar merupakan suatu ilustrasi yang dapat menyampaikan sebuah informasi apabila tidak dapat dijelaskan dengan kata-kata.

3. Animasi

   Animasi merupakan sebuah gerakan dari sebuah foto atau video, seperti gerakan sebuah objek yang sedang melakukan kegiatan. Animasi dapat menggambarkan atau menjelaskan suatu informasi yang disajikan dalam beberapa gambar yang sulit dijelaskan.

4. Suara

   Suara merupakan cara untuk memperjelas suatu informasi yang disajikan dalam bentuk video.

5. Video

Video merupakan cara untuk memperjelas informasi yang lebih komunikatif dibandingkan gambar biasa, informasi disajikan dalam kesatuan utuh dari objek yang dimodifikasi sehingga terlihat saling mendukung.

6. Interactive link

Interactuve link merupakan interaksi antara pengguna dengan aplikasi, dimana pengguna dapat menekan mouse atau objek pada screen seperti button atau teks dan menyebabkan program melakukan perintah tertentu.

2.1.2 Jenis-jenis Multimedia

Dari definisi multimedia yang sudah dijelaskan, multimedia dapat dibagi menjadi 2 (dua) jenis, yaitu [7]:

1. Multimedia linier

Multimedia linier merupakan multimedia yang tidak dilengkapi dengan alat pengontrol yang dapat dioperasikan oleh pengguna. Multimedia linier hanya dapat menyajikan media kepada pengguna, seperti televisi dan film atau video.

2. Multimedia interaktif

Multimedia interaktif merupakan multimedia yang dapat dioperasikan oleh pengguna sehingga pengguna dapat melakukan apa yang dikehendakinya, misalnya multimedia pembelajaran, game, dan lain- lain.

2.2 Kebutuhan Energi

Energi dalam tubuh manusia dapat timbul dikarenakan adanya pembakaran karbohidrat, protein dan lemak, dengan demikian agar manusia selalu tercukupi energinya diperlukan zat-zat makanan yang cukup pula ke dalam tubuhnya. Manusia yang kurang makan akan lebih baik kekuatannya, fisiknya maupun daya ingatnya serta daya pemikirannya karena kurangnya zat-zat makanan yang diterima tubuhnya yang dapat menghasilkan energy [8].

Menurut Kartasapoetra dan Marsetyo [8], dalam pengertian makanan sebagai sumber energi ternyata energi makanan dalam proses-proses yang terjadi dalam tubuh hanya sebagian saja yang dapat diubah menjadi panas. Dalam keadaan hanya sedikit melakukan kerja fisik, sebagian besar energi diubah menjadi panas, dan dalam keadaan tidak melakukan pekerjaan fisik maka relatif seluruh energi diubah menjadi panas dan selanjutnya panas akan keluar dari tubuh.

Konsumsi energi berasal dari makanan yang diperlukan untuk menutupi pengeluaran energi sseorang bila ia mempunyai ukuran dan komposisi tubuh dengan tingkat aktivitas yang sesuai dengan kesehatan jangka panjang dan memungkinkan pemeliharaan aktivitas fisik yang dibutuhkan secara sosial dan ekonomi. Kebutuhan energi total orang dewasa diperlukan untuk:

1. Metabolisme Basal
2. Aktivitas Fisik
3. Efek makanan atau pengaruh dinamik khusus (*Specific Dynamic Action)*

Metode Harris-Bennedict (1990) memperhitungkan berat badan, tinggi badan dan umur. Indeks paling adalah berat badan menurut umur dan rumus linier.

Menurut Cakrawati dan Mustika [9], aktivitas fisik memerlukan energi di luar kebutuhan untuk metabolisme basal. Pengertian aktivitas fisik adalah gerakan yang dilakukan otot tubuh dan sistem penunjangnya. Selama aktivitas fisik, otot memerlukan energi di luar metabolisme untuk bergerak. Jantung dan paru-paru memerlukan tambahan energi untuk mengantarkan zat-zat gizi dan oksigen ke seluruh tubuh dan mengeluarkan sisa-sisa dari tubuh. Kebutuhan energi untuk aktivitas fisik tergantung dari banyaknya otot yang bergerak, waktu dan beban pekerjaan yang dilakukan sehingga seseorang yang gemuk memerlukan energi lebih besar daripada seseorang yang kurus.

Kebutuhan energi untuk pengaruh termis makanan atau kegiatan dinamik khusus adalah energi tambahan yang diperlukan tubuh untuk pencernaan makanan, absorpsi dan metabolisme zat-zat gizi yang menghasilkan energi. SDA

bergantung pada jumlah energi yang dikonsumsi yaitu ± 10% kebutuhan energi untuk metabolisme basal dan untuk aktivitas fisik .

2.3 Android

Android adalah sebuah sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan platfotm yang terbuka bagi para *developer* untuk menciptakan aplikasi ciptaan mereka sendiri. Pada mulanya,, Google.Inc. membeli Android Inc. Pendatang baru yang membuat piranti lunak untuk handphone. Kemudian untuk mengembangkan Android, dibentuklah Handset Alliance, konsursium dari 34 perusahaan piranti keras, piranti lukan dan telekomunikasi [10].

Pada saat pertama kali Android dirilis pada 5 November 2007, Android bersama dengan Open Handset Alliance menyatakan untuk mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat software dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau dikenal sebagai Open Handset Distribution (OHD).

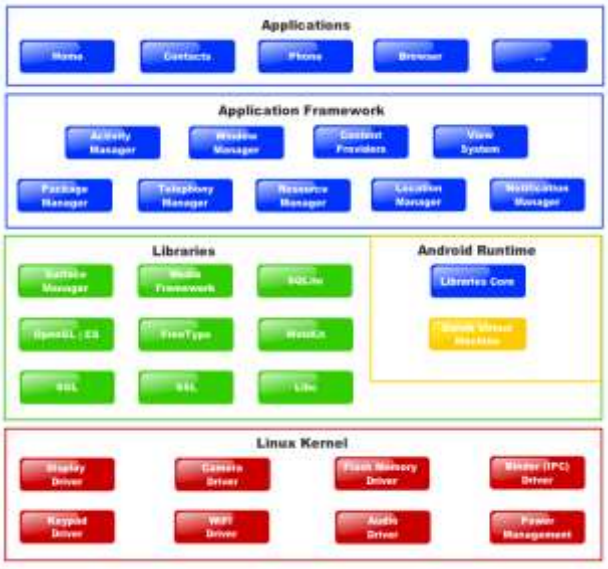
Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

Android merupakan sistem operasi yang berkembang dengan pesat, namun tidak menjadikannya sistem operasi yang sempurna ada beberapa kekurangan dari sistem operasi Android diantaranya Android terkesan rumit, karena mempunyai banyak sekali *widget* maupun aplikasi dengan banyak pengaturan sehingga pengguna harus banyak belajar mengenai Android, selain itu Android yang

merupakan sistem operasi terbuka sehingga pengguna dapat memasang aplikasi di luar toko aplikasi yang ditawarkan oleh perangkat Android tersebut sehingga sangat rentan terkena ancaman *malware* atau *virus*. Tidak semua perangkat Android dapat langsung memperbarui sistem operasi terbaru, karena produsen *Smartphone* lebih mementingkan produk baru untuk diberi sistem operasi yang terbaru, dibanding dengan memberi pemberitahuan tentang update sistem operasi terbaru sehingga membutuhkan waktu lama untuk memperbarui sistem operasi bagi beberapa perangkat.

2.3.1 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada Gambar 2.1 Arsitektur Android. Secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:



Gambar 2.1 Arsitektur Android

1. *Apllication and Widget*

Application and Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

Applications Frameworks Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan *service background*, mengatur alarm, dan menambah status notifikasi, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*). Sehingga bisa kita simpulkan *Application Frameworks* ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content *providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

1) Views
2) Content Provider
3) Resource Manager
4) Notification Manager
5) Activity Manager

3. *Libraries*

Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas Kernel, *layer* ini meliputi berbagai *library* C/C++ inti seperti Libc SSL, serta:

1) *Libraries* media untuk pemutaran media audio dan video.
2) *Libraries* untuk manajemen tampilan.
3) *Libraries* Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4) *Libraries* SQLite untuk dukungan *database*.

5) *Libraries* SSL dan WebKit terintegrasi dengan *web browser* dan *security*.
6) *Libraries* LiveWebcore mencakup modern *web browser* dengan *engine embedded web view*.
7) Libraries 3D yang mencakup implementasi OpenGL ES1.0 API's.

4. *Android Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu:

5. *Core Libraries*

Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*,sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *Core Libraries.*

6. *Dalvik Virtual Machine*

Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsifungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan *threading* dan manajemen tingkat rendah.

7. *Linux Kernel*

Linux Kernel adalah layer dimana inti dari sistem operasi Android itu berada.

Berisi *file* sistem yang mengatur sistem *processing*, *memory*, *resource*, drivers, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6. Keunikan dari nama sistem operasi (OS) Android adalah dengan menggunakan nama makanan hidangan penutup (*Dessert*). Selain itu juga nama-nama OS Android memiliki huruf awal berurutan sesuai abjad. Adapun beberapa nama dan versi Android yang sudah diluncurkan adalah sebagai berikut:

1) Android v 1.0 *Astro* (*Alpha*)

Sebenarnya sebelum mereka memberikan nama-nama kudapan sebagai nama untuk versi OS nya, Android sempat memiliki 2 versi awal dengan nama Android *Alpha* dan *Beta*. Nama untuk versi pertama ini sendiri sebenarnya adalah Android Astro, namun karena alasan hak cipta (*trademark*), nama ini tidak jadi digunakan. Di versi awal ini belum ada perangkat dengan sistem operasi Android yang dijual secara komersil.

2) Android v 1.1 *Bender* (*Beta*)

Versi ini dirilis pada tanggal 5 November 2007 yang merupakan versi lanjutan dari Android Astro (Alpha). Sama seperti versi awalnya, nama Bender juga juga tak jadi digunakan karena alasan hak cipta (*trademark*). Kemudian lahirlah telepon seluler pertama dengan sistem operasi Android yang dijual secara komersil yakni HTC Dream.

3) Android v 1.5 *Cupcake*

Ini merupakan versi pertama yang menggunakan nama makanan manis sebagai kode nama untuk tiap versi Android yang kemudian tradisi untuk menamai versi Android dengan nama makanan manis masih diteruskan hingga saat ini. Android *Cupcake* dirilis pada tanggal 30 April 2009.

4) Android v 1.6 *Donut*

Dirilis tidak sampai setahun setelah perilisan Android *Cupcake*, yakni pada tanggal 15 September 2009. Versi ini dihadirkan untuk menutupi *bug* pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

5) Android v 2.0 – 2.1 *Éclair*

Sistem operasi ini juga dirilis tidak sampai setahun setelah perilisan dua versi

sebelumnya yakni pada tanggal 26 Oktober 2009. Mereka masih berfokus untuk menutupi *bug* yang ada dan juga menambahkan beberapa fitur seperti *Bluetooth*, *flash* pada kamera, fitur digital zoom pada kamera, *multi-touch*, *live wallpaper*, dan lainnya. Hadirnya perangkat seri Nexus dari Google yang pertama kali muncul yakni HTC Nexus One juga menggunakan versi OS Android *Eclair*.

6) Android v 2.2 *Frozen Yoghurt* (Froyo)

Dirilis pada tanggal 20 Mei 2010. Perangkat dengan OS Android semakin banyak dan kehadirannya mulai dilirik oleh pasar meski masih jauh dibawah kepopuleran OS lain seperti Symbian dan Windows Mobile.

7) Android v 2.3 *Gingerbread*

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 v 3.0 – 3.2 *Honeycomb*

Versi ini dirilis pada tanggal 10 Mei 2011 dan dirancang khusus untuk perangkat Tabelt, yang kala itu mulai populer di pasaran salah satunya berkat promosi Samsung dan juga kepopuleran Apple iPad.

9) Android v 4.0 Ice Cream Sandwich

Dirilis pada 16 Desember 2011. Bisa dibilang merupakan Android *Honeycomb* yang disempurnakan, dan dioptimalkan untuk penggunaan baik

Smartphone maupun Tabelt. Perubahan yang paling terlihat dari versi ini dibanding denganversi sebelumnya adalah dari segi *User interface* yang nampak lebih bersih danelegan. Versi ini juga lebih dioptimalkan untuk urusan multitasking. Bersamaan dengan diperkenalkannya Android ICS, Google juga memperkenalkan perangkat Galaxy Nexus yang merupakan seri *Smartphone* Nexus yang diproduksi oleh Samsung. Setelah versi ini, Google kemudian secara rutin memperkenalkan perangkat seri Nexus pada tiap kali mereka memperkenalkan versi Android terbaru.

10) Android v 4.1 – 4.3 *Jelly Bean*

Dirilis pada 9 Juli 2012. Bersamaan dengan diperkenalkannya versi OS 4.1 pada 27 Juni 2012, Google juga memperkenalkan Nexus 7 yang diproduksi oleh ASUS. Nexus 7 (generasi 1) merupakan seri Nexus pertama yang merupakan perangkat Tabelt. Jelly Bean mengalami 3x update versi yakni 4.1, 4.2 hingga 4.3. Selanjutnya mereka memperkenalkan Android v4.2 bersamaan dengan dihadirkannya Nexus 4, *Smartphone* yang diproduksi oleh LG plus Nexus 10, perangkat Tabelt yang diproduksi oleh Samsung.Pada saat versi 4.3 dirilis, Google juga merilis Nexus 7 generasi 2 yang masih diproduksi oleh ASUS yang mana ia memiliki beberapa peningkatan seperti misalnya penambahan kamera belakang serta dukungan untuk konektivitas internet.

11) Android v 4.4 *Kitkat*

Nama *Kitkat* diambil dari sebuah produk cemilan wafer berlapis coklat yang dimiliki oleh Nestle. Sebelumnya Android versi "K" ini disebut-sebut sebagai *Key Lime Pie*, namun atas beberapa pertimbangan akhirnya Google lebih memilih untuk memberi nama *Kitkat*. Ceritanya, *Kitkat* adalah salah satu cemilan yang tersedia di dapur kantor yang biasanya juga menemani para programer Google. Hingga seseorang berkata *"Hey, kenapa kita tidak menamainya sebagai Kitkat?"*.Sesaat setelah ide itu muncul, Google segera menghubungi pihak Nestle sebagai pemilik merk dagang Kitkat dan mereka menyetujui pemberian nama Kitkat untuk versi Android K. Karyawan Google sendiri tidak mengetahui bahwa Android 4.4 akan diberi nama Kitkat karena

yang mereka tau versi Android K adalah *Key Lime Pie*. Mereka baru mengetahuinya setelah patung maskot Android Kitkat diletakkan di kantor pusat Google.Versi ini diklaim lebih ramah terhadap perangkat dengan spesifikasi seadanya. Bahkan perangkat dengan RAM 512 MB masih bisa menjalankan OS versi ini dengan mulus. Berbeda dengan *Jelly Bean* yang minimal harus memiliki RAM diatas 756 MB agar dapat berjalan dengan mulus. Bersamaan dengan dirilisnya Android Kitkat pada tanggal 31 Oktober 2013, Google juga merilis *Smartphone* Nexus 5 yang diproduksi oleh LG.

12) Android v 5.0 *Lollipop*

Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih *flat* dengan konsep *material design*. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk penggunaan RAM diatas 3 GB pada *hardware* perangkat. Penggunaan prosesor 64-bit pun makin banyak diadopsi oleh para *vendor*, mulai dari penerapan pada perangkat flagship hingga perangkat kelas menengah kebawah.

13) Android v 6.0 *Marshmallow*

Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya Google juga memperkenalkan 2 perangkat *Smartphone* Nexus sekaligus yang diproduksi oleh 2 *vendor* yang berbeda. Nexus 5X adalah versi *Smartphone* Nexus kelas menengah dengan ukuran layar 5.2 inch yang diproduksi oleh LG. Sedangkan yang satunya lagi memiliki bentang layar yang lebih lebar yakni 5.7 *inch* yang diberi nama Nexus 6P yang merupakan *Smartphone* flagship hasil kerjasama Google dengan Huawei. Sejak Android 6.0 (Marshmallow), Google memberi perhatian khusus terhadap usaha penghematan konsumsi baterai. Pada Marshmallow, Google memperkenalkan dua fitur baru untuk keperluan ini: *Doze* dan *App Standby*. Bila Android mendeteksi bahwa peranti tidak digunakan dalam waktu lama (layar mati dan peranti tidak bergerak), sistem bisa masuk dalam mode *Doze*. Dalam mode ini, Android akan membatasi akses aplikasi terhadap berbagai perangkat keras seperti

prosesor dan jaringan. *App Standby* pada dasarnya serupa, hanya saja di sini pembatasan dilakukan terhadap aplikasi tertentu, bukan terhadap sistem secara keseluruhan. Pada versi ini, Google juga memperkenalkan dukungan terhadap porta USB Type C, dan untuk pertama kalinya memberikan dukungan terhadap pembaca sidik jari.

14) Android v 7.0 *Nougat*

Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi
bahwa kemungkinan besar, pemberian nama untuk Android versi "N" ini adalah Nutella. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan *Nougat* (yang sepintas lebih mirip dengan tempe itu). Sebelumnya, Google telah mengundang para penggunanya untuk memberikan ide penamaan pada versi ini. Beberapa nama termasuk Nutella dan Nastar pun muncul, hingga akhirnya Google lebih memilih nama Nutella. Google memperkenalkan peningkatan terhadap kompilator aplikasi yang memungkinkan pemasangan yang lebih cepat (sampai 75%) dan ukuran aplikasi yang lebih kompak (sampai 50%), selain eksekusi yang lebih cepat. Pada *Nougat* juga, Google menyertakan platform realitas virtual (*Daydream*), penghematan data (*Data Saver*), serta dukungan yang lebih baik untuk tampilan multijendela (*Multiwindow*). Yang terakhir ini diperlukan untuk multitugas lebih baik pada peranti dengan layar lebar, seperti phablet dan komputer Tabelt. Untuk pertama kalinya juga, Google memperkenalkan dukungan terhadap teknologi grafis baru Vulkan, yang diharapkan akan menggantikan OpenGL pada Android.

15) Android v 8.0 Oreo

Android Oreo pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017. Android Oreo mengusung fitur baru yang membuat *Smartphone* lebih cepat, pintar dan *powerful* dibandingkan dengan versi android

yang sebelumnya. Beberapa fitur unggulan dari versi ini adalah *background limit*, *Google Play Protect*, Emoji baru dan *picture in picture*.

16) Android v 9.0 Pie

Android 9.0 (Pie) resmi dirilis pada Agustus 2018. Seperti sebelumnya, Google terus mencari cara untuk menghemat baterai. Fitur baru yang diperkenalkan pada Android 9.0 (Pie) untuk keperluan ini adalah *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 sisi perangkat lunak.

2.3.2 Android Life Cycle

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, klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan. Tabel 2.1 adalah Komponen Aplikasi Mobile.

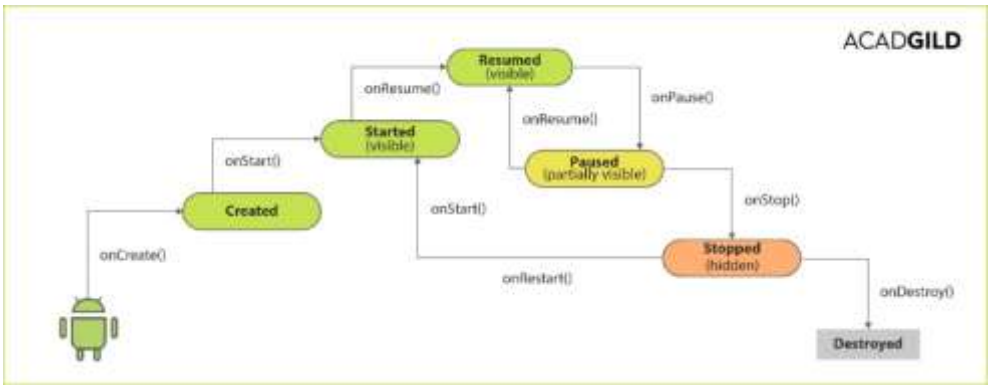
Tabel 2.1 Komponen Aplikasi Mobile

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

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan

mencantumkan specified permission sebelum digunakan seperti komponen *Service*, *Broadcast Receiver*, *Content Provider*.

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* ataupun web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang cukup 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 seperti pada Gambar 2.2.



Gambar 2.2 Siklus Hidup Android

(Sumber gambar: https://acadgild.com)

diilustrasikan pada Gambar 2.2 siklus hidup android 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().

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 .

3) *Stopped*

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground. State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cyclecallback* berikutnya. Artinya, setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), kemudian diikuti oleh onResume().

2.3.3 Fitur Android

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah [11]:

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.

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widgets dapat menjalankan proses *request* seperti musik *streaming* atau mendeteksi suhu ruangan secara background.

Multi-proses dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar.

2. *Touch Gesture* dan *Multi-Touch*

Touchscreen adalah *user interface* intuitif yang digunakan banyak *Smartphone*
didunia. 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.

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.

2.3.4 Android SDK

Android SDK adalah tools *API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone atau *Smartphone*. Beberapa fitur-fitur android yang paling penting adalah [11]:

1. *Framework:* Aplikasi yang mendukung pengganti komponen dan reusable.
2. *Dalvik Virtual Machine dioptimalkan untuk perangkat mobile.*
3. *Integrated Browser berdasarkan engine open source WebKit.*
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi hardware).
5. SQLite untuk penyimpanan data.
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung hardware).
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
8. Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware).
9. Lingkungan *Development* yang lengkap dan termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin.

2.4 Java

Inovasi bahasa komputer dimotivasi oleh dua faktor: perbaikan dalam seni pemrograman dan perubahan dalam lingkungan komputasi, tidak terkecuali Java. Dibangun di atas warisan yang kaya dari C dan C++, Java menambahkan perbaikan dan fitur yang mencerminkan keadaan seni dalam pemrograman saat ini. Menanggapi munculnya lingkungan online, Java menawarkan fitur yang merampingkan pemrograman untuk arsitektur yang sangat terdistribusi [12].

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsistems 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-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM) [12].

Versi awal Java ditahun 1996 sudah merupakan versi rilis sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. java.lang: Peruntukan kelas elemen-elemen dasar.
2. java.io: Peruntukan kelas input dan output, termasuk penggunaan berkas.
3. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Seperti telah dibahas sebelumnya, banyak jenis komputer dan sistem operasi yang terhubung ke Internet. Untuk program-program untuk secara dinamis didownload ke semua berbagai jenis platform, beberapa sarana untuk menghasilkan kode dieksekusi porTabel diperlukan. Mekanisme yang sama yang membantu menjamin keamanan juga membantu menciptakan portabilitas karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Java memliki beberapa kelebihan dibandingkan dengan bahasa pemograman
lain,diantaranya:

1. Multiplatform
2. OOP (*Object Oriented Programming)*
3. *Library Class* yang lengkap
4. Mewarisi Kekayaan C/C++

5. Pengumpulan Sampah Otomatis
6. Mudah didekompilasi

2.5 Augmented Reality

Augmented Reality (AR) adalah sebuah teknologi yang menggabungkan sebuah benda dua dimensi maupun tiga dimensi dalam bentuk dunia digital ke dalam lingkungan dunia nyata. Fungsi *augmented reality* ini yaitu memberikan informasi yang tidak dapat diterima oleh manusia dalam berinteraksi antara manusia dengan komputer sehingga manusia menggunakan produk yang telah disediakan oleh sistem secara *virtual* dalam keadan *real- time* pad Gambar 2.3.

Gambar 2.3 Contoh Augmented Reality

(sumber: https://www.forbes.com)

Augmented reality memiliki teknik dalam penggunaannya, yaitu sebagai berikut [13]:

1. *Marker Augmented Reality (Marker Based Tracking)*

   Marker Augmented Reality (Marker Based Tracking) merupakan metode *augmented reality* yang memerlukan *marker* khusus yang berupa ilutrasi hitam dan putih berbentuk persegi dengan bata hitam tebal dan latar belakang berwarna putih.

2. *Markerless Augmented Reality*

Markerless Augmented Reality merupakan metode *augmented reality* yang dimana tidak memerlukan sebuah marker untuk menampilkan elemen-elemen

digital. Berikut adalah teknik-teknik yang dapat digunakan dengan menggunakan *markerless tracking*, yaitu:

a) Face tracking

Merupakan teknik yang dapat mengenali bagian-bagian wajah manusia seperti: mata, hidung mulut dan dapat mengabaikna objek-objek lainnya

b) 3D object tracking

Merupakan teknik yang dapat mendeteksi semua bentuk benda seperti:mobil meja, rumah dan lainnya.

c) Motion tracking

Merupakan teknik yang dapat menangkap setiap gerakan, dan teknik ini biasanya dipakai untuk pembuatan film animasi atau mencoba memanipulasikan suatu gerakan.

d) GPS based tracking

Merupakan sebuah teknik yang memanfaatkan fitur GPS dan kompas yang terdapat pada perangkat serta dapat mengambil data berupa posisi koordinat dari perangkat tersebut dan kemudian secara *realtime* akan memberikan tampilan dalam bentuk arah yang diinginkan.

e) Text Recognition Teknik pendeteksian teks pada dasarnya menggunakan

Teknologi OCR (Optical Character Recognition). OCR yaitu sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. Penggunaan OCR pada augmented reality telah dikembangkan sebagai media translator pada *Smartphone* untuk kemudahan proses terjemahan secara realtime.

2.6 Unified Modeling Language (UML)

Unified Modeling Languange (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan

pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif [14].

Dalam kerangka spesifikasi, UML menyediakan model-model yang tepat,tidak mendua-arti (ambigu), serta lengkap. Secara khusus, UML menspesifikasi langka-langka penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive sistem*). Dalam hal itu, UML bukanlah merupakan Bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam Bahasa pemrograman perorientasi objek, katakanlah Java, Borland Delphi, Visual BASIC, C++, dan lain-lain [14].

UML lahir dari penggabungan banyak Bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997, UML menghancurkan Menara Babel tersebut menjadi sejarah. Pada intinya peran UML dalam pengembangan perangkat lunak, orang-orang memiliki cara-cara yang berbeda dalam penggunaanya, perbedaan-perbedaan yang masih dibawa dari Bahasa-bahasa pemodelan grafis lain. Perbedaan-perbedaan ini mengakibatkan perselisihan yang panjang dan keras tentang bagaimana UML seharusnya digunakan.

Berdasarkan pemaparan mengenai *Unified Modeling Languange* (UML) dari beberapa sumber referensi, maka dapat disimpulkan UML merupakan alat bantu dalam melakukan pemodelan yang saling berhubungan secara langsung dalam pembangunan sebuah sistem agar lebih efektif.

2.6.1 *Use Case* Diagram

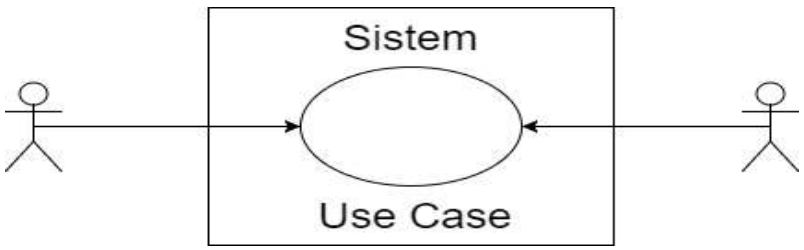
Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara

pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai [14].

Use Case diagram digunakan untuk menggambarkan analisis kebutuhan dari sistem dari level atas melalui fungsionalitas dari sistem dan interaksi diantara para aktor. Aktor adalah sesuatu yang berinteraksi dengan sistem.

Secara umum, tujuan dari *Use Case* diagram adalah sebagai berikut:

1. Digunakan untuk mengumpulkan kebutuhan dari sebuah sistem
2. Untuk mendapatkan pandangan dari luar sistem
3. Untuk mengidentifikasi factor yang mempengaruhi sistem baik internal maupun eksternal
4. Untuk menunjukkan interaksi dari para actor dari sistem Ilustrasi dari actor, *Use Case* dan boundary dapat dilihat pada Gambar 2.4.



Gambar 2.4 Use Case Diagram

Sumber gambar: https://www.uml-diagrams.org/use-case-diagrams-examples.html

2.6.2 *Activity Diagram*

Activity Diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika procedural, proses bisnis, dan aliran kerja suatu bisnis bias dengan mudah dideskripsikan dalam *Activity Diagram*. *Activity Diagram* memiliki peran seperti halnya flowchart, akan tetapi perbedaannya adalah *Activity Diagram* bisa mendukung perilaku pararel [14].

Tujuan dari *Activity Diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya. Secara umum tujuan *Activity Diagram* adalah sebagai berikut:

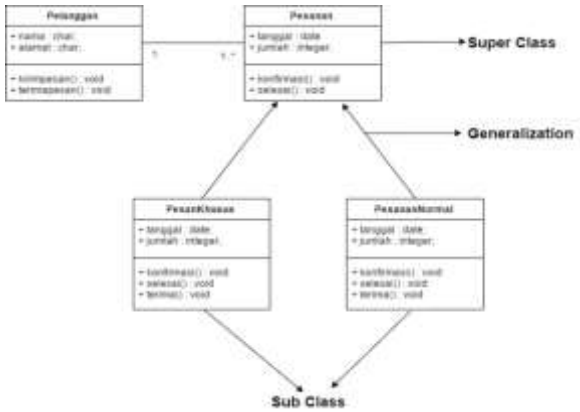
1. Menggambarkan aliran aktivitas dari sistem
2. Menggambarkan urutan aktifitas dari satu aktifitas ke aktifitas lainnya
3. Menggambarkan pararelisme, percabangan dan aliran konkuren dari sistem

2.6.3 Class Diagram

Class Diagram adalah diagram statis yang mewakili pandangan statis dari suatu aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executeTabel code*) dari aplikasi perangkat lunak. *Class diagram* menunjukan koleksi kelas, antarmuka, asosiasi, kolaborasi, dan constraint. *Class diagram* juga dikenal sebagai diagram struktural [14] contohmya seperti pada Gambar 2.5.

Tujuan dari *class diagram* adalah untuk memodelkan pandangan statis suatu aplikasi. Secara lebih rinci tujuannya adalah sebagai berikut:

1. Analisis dan desain pandangan statis aplikasi.
2. Menjelaskan tanggung jawab suatu sistem.
3. Basis untuk diagram komponen dan penyebaran (*deployment*)
4. *Forward and reverse engineering*



Gambar 2.5 Contoh Class Diagram Sistem Pemesanan

2.6.4 Sequance diagram

Sequance diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *Use Case*.

Komponen utama *sequance diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Simbol – symbol yang ada pada *sequance diagram* dapat dilihat pada gambar 14.

2.6.5 Bangun Dasar UML

Metodologi UML menggunakan 3 bangunan dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan, yaitu:

1. Sesuatu (*Things*).

Ada 4 macam '*things*' dalam UML, yaitu:

1) Structural Things.
2) Behavioral Things.
3) Grouping Things.
4) Annotational Things.

2. Relasi (*Relationship*).

Yang dimaksud *relationship* adalah hubungan-hubungan yang terjadi antarelemen dalam UML. Hubungan-hubungan ini penting sekali dalam UML. Dapat dikatakan, tidak mungkin membuat model-model UML tanpa *relationship* ini.

3. *Diagrams*.

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbeda-beda sehingga kita bisa mendapatkan pemahaman secara menyeluruh.

Secara umum UML diterapkan dalam pengembangan sistem/perangkat lunak berorientasi objek sebab metodologi UML ini umumnya memiliki keunggulan-keunggulan, yaitu:

1) ***Uniformity.*** Dengan metodologi UML (atau metodologi berorientasi objek pada umumnya), para pengembang cukup menggunakan 1metodologi dari tahap analisis hingga perancangan. UML juga memungkinkan kita merancang komponen antarmuka penguna (*User Interface*) secara terintegrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data.

2) ***Understandability.*** Dengan metodologi ini kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah dipahami siapa pun juga.

3) ***Stability.*** Kode program yang dihasilkan relative stabil sepanjang waktu sebab sangat mendekati permasalahan sesungguhnya dilapangan.

4) ***Reusability.*** Dengan metodologi perorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannya akan sangat mempercepat waktu untuk pengembangan perangkat lunak (atau sistem informasi).

2.7 Pengujian *Black Box*

Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program [15].

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program [15].

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing.*

Black Box Testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.