BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Bagian ini berisi tentang beberapa teori-teori yang mendukung dalam proses analisis dan implementasi berdasarkan masalah yang diangkat dalam pembangunan Aplikasi rekomendasi cerdas untuk persiapan dan panduan hiking pada platform android.

2.2 Pengertian Gunung

Gunung adalah sebuah bentuk tanah yang menonjol di atas wilayah sekitarnya. Sebuah gunung biasanya lebih tinggi dan curam dari sebuah bukit, tetapi ada kesamaaan, dan penggunaan sering tergantung dari adat lokal. Beberapa otoritas mendefinisikan gunung dengan puncak lebih dari besaran tertentu, gunung membutuhkan ketinggian 2000 kaki (610 m) agar bisa didefinisikan sebagai gunung.[1]

2.2.1. Gunung Ceremai

Gunung Ceremai terletak di Kabupaten Kuningan dan Kabupaten Majalengka, Provinsi Jawa Barat. Memiliki tiga jalur resmi yang sering digunakan untuk mendaki, yaitu jalur Apuy, jalur Palutungan dan jalur Linggarjati. Dengan ketinggian 3078 mdpl. Gunung Ciremai termasuk ke dalam kawasan Taman Nasional Gunung Ciremai (TNGC). Gunung ini terletak berjauhan dengan gunung tinggi lainnya dan merupakan gunung tertinggi di Jawa Barat. Gunung Ceremai secara administratif termasuk dalam tiga kabupaten yaitu Kabupaten Cirebon, Kabupaten Kuningan, dan Kabupaten Majalengka, Provinsi Jwa Barat. Gunung ini memiliki kawah ganda, kawah barat yang beradius 400 m terpotong oleh

kawah timur 600 m. pada ketinggian sekitar 2900 mdpl d lereng selatan terdapat bekas titik letusan yang dinamakan Gowa Walet.

Untuk perlengkapan dalam mendaki yang di utamakan yaitu peralatan standar naik gunung seperi tas/ransel gunung, sepatu gunung, senter, jas hujan, jaket tebal, celana training hangat, dan peralatan masak : kompor mini, nesting, korek api, untuk logistik lebih baik bawa makanan mentah bahan masakan missal: beras, tempe,omelet, dan untuk keperluan pribadi bias bawa obat masing-masing.

Dalam perjalanan menuju puncak gunung ciremai di jalur apuy yaitu sangat menanjak (dengkul ketemu dada) karena jalannya tanah ditengah-tengah hutan banyak akar pohon yang mengelilit di tiap jalan. Jalan landai nya hanya pada setiap pos untuk beristirahat. Waktu dalam perjalanan kalau santai 2-3 hari tetapi kalau pengalaman saya cuma 1 hari 1 malam karena berangkat malam lebih cepat dibanding siang, karena tujuannya agar pagi hari sudah sampai puncak, tapi tujuan utama dalam mendaki gunung yaitu mengalahkan diri kita sendiri, menguji sabar dan mempererat tali persaudaraan dan yang terpenting adalah bukan seberapa tinggi gunung yang pernah kita daki tapi seberapa banyak kita berkontribusi untuk memelihara alam.[2]

2.2.2. Gunung Papandayan

Papandayan terkenal dengan bentangan kaldera yang sangat luas, tercatat bentangan kaldera papandayan merupakan yang terluas di asia tenggara. Hal itu juga yang membuat gunung ini sampai bergaung ke mancanegara. Bentangan kaldera ini juga menjadi salah satu jalur yang akan dilalui oleh pendaki.

Secara keseluruhan,kawah papandayan berjumlah empat belas. Keindahan letupan cairan kawah dan asap yang mengepul dari endapan belerang menjadi pemandangan langka yang tidak mungkin terlewatkan. Momen ini,capa kali dimanfaatkan para pendaki untuk rehat sejenak sambil memanjakan pandangan terhadap kecantikan kaldera.kawasan hutan mati disebut hutan mati karena lelehan lahar yang melintasi area ini dari letusan sebelumnya membuat beberapa pohon mati,tak bias tumbuh.meski jika ditilik dari ilmu biologi, kematiannya hanya untuk jangka waktu tertentu. Beberapa pohon-pohon sisa lalapan lahar masih berdiri. Batang-batangnya menghitam,serta daun-daun yang berguguran

membuat kawasan ini bagai kawasan hutan di negeri dongeng dan sangat berbeda dengan kawasan hutan lainnya.

Jika kita berjalan sampai kearah utara hutan,kita dapat melihat bentangan kaldera papandayan dari atas,sebab letak hutan mati tepat berada di atasnya.tak hanya itu titik ini juga sering dijadikan tempat berburu sunset .sunset akan menyembul di atas awan puncak gunung cikurai.

Kawasan selanjutnya adalah kawasan tertinggi yang menjadi perhentian langkah terakhir dari pendakian gunung papandayan yaitu tegal alun. Untuk melewatinya dibutuhkan perjalanan 1-2 jam dari kawasan hutan mati.salah satu medannya yang menantang adalah tanjakan mamang. Kemiringannya membuat beberapa pendaki melaluinya dengan setengah memanjat.sebab tak cukup kaki tangan pun ikut berperan

Melalui jalur terakhr ini cukup melelahkan,tapi hal itu akan segera terbayar lunas jika telah menginjakan kaki di kawasan tegal alun.bentangan tanah lapang yang begitu luas ditumbuhi keindahan ribuan bunga edelwis [3].

2.2.3. Gunung Merbabu

Gunung merbabu termasuk gunung yang tidak aktif karena tergolong gunung api tua di pulau jawa.mempunyai lima kawah yaitu: kawah condrodimuko, kawah kombang, kawah kendang, kawah rebab, dan kawah sambernyowo.

Gunung merbabu berbentuk dataran tinggi yang lebar, berbukit-bukit dan terpisah puncaknya akibat erosi bila di bandingkan gunung merapi, gunung merbabu bentuknya besar sekali dibandingan gunung merapi yang sangat ramping. Bagian puncaknya dapat dibagi menjadi tiga satuan yang merupakan sektor graben (daerah sekitar puncak) gunung api, yakni:

1. graben sari dengan arah timur tenggara barat barat laut
2. graben guyangan dengan arah selatan barat daya utar timur
3. graben sipendok dengan arah barat laut timur tenggara

(graben atau slenk adalah hasil dari patahan pada kulit bumi yang mengalami depresi dan terletak di antara dua bagian yang lebih tinggi. Bagian yang lebih

tinggi disebut dengan horst,horst sendiri adalah hasil dari terjadinya patahan pada kulit bumi yang mengalami pengangkatan sehingga menjadi lebih tinggi dibandingkan dengan daerah sekitarnya [4].

2.3. Pengertian Hiking

Hiking adalah aktifitas mendaki gunung. Biasanya *hiking* dilakukan secara berkelompok yang terdiri dari beberapa orang. Jarang sekali ada orang yang hiking sendirian. Aktifitas *hiking* semakin hari semakin digemari baik di dunia maupun di Indonesia. Di Indonesia, khususnya di kampus-kampus, hampir dipastikan ada kelompok pencinta alam yang sering melakukan pendakian [5].

2.4. Pengertian Rekomendasi

Rekomendasi adalah suatu proses komunikasi atas produk atau jasa tertentu yang berguna untuk memberikan informasi secara personal. Contoh rekomendasi yang umum dipakai adalah Word Of Mouth communication (WOM) atau rekomendasi mulut ke mulut.

Menurut Luwis dan Harsini : 2010, Rekomendasi merupakan suatu bentuk komunikasi sekaligus promosi secara tidak langsung yang dilakukan oleh para konsumen yang sudah membeli produk atau jasa yang kemudian diceritakan kembali berbagai pengalamannya yang berkaitan dengan produk atau jasa tersebut kepada orang lain [6].

2.5. Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi Android diluncurkan untuk umum pada tahun 2008. Android sangat berkembang pesat di industri karena dua aspek utama yaitu bersifat opensource dan model arsitekturya. Sebagai sebuah proyek yang bersifat opensource, memungkinkan Android untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada bug program hingga hardware [7].

Aplikasi Android terdiri dari komponen yang bebas digabungkan, diikat menggunakan suatu project manifest yang menggambarkan masing-masing komponen dan bagaimana komponen-komponen itu saling berhubungan. Ada enam komponen dalam Android untuk membangun suatu aplikasi yaitu :

1. *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *activity* menyajikan tampilan visual program yang sedang dijalankan oleh user. Setiap *activity* dideklarasikan dalam sebuah *class* yang bertugas untuk menampilkan antarmuka yang terdiri dari *views* dan *respons* terhadap *event*. Setiap aplikasi mempunyai satu buah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi sedang dijalankan. Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya pada *activity stack*. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ke tumpukan paling atas. Ketika *activity* dipanggil dan disimpan dalam *activity stack* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

2. *Active*

Setiap *activity* yang berada di tumpukan paling atas, maka *activity* tersebut akan terlihat dan menerima masukan dari *user*. Android akan membuat *activity* ini tetap hidup dengan menghentikan *activity* yang berada dibawah tumpukan jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

3. *Paused*

Beberapa *activity* akan terlihat tapi tidak terfokus pada kondisi ini disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat aktif namun tidak bisa diakses oleh *user*.

4. *Stopped*

Ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi

yang ada. Namum akan di eksekusi oleh sistem jika sistem membutuhkan sumberdaya lebih.

5. *Inactive*

Kondisi ini ketika *activity* telah dihentikan atau sebelum dijalankan. *Inactive activity* telah ditiadakan dari *activity stack* sehinga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

6. *Services*

Suatu *services* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi.

7. *Content Provoders*

Content providers digunakan untuk mengelola dan berbagi *database*. Data dapat disimpan dalam *file system*, dalam *database* SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *content provider* memungkinkan antar aplikasi untuk saling berbagi data.

8. *Intens*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intens* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi sesuatu, semisal pesan masuk.

9. *Broadcast Receivers*

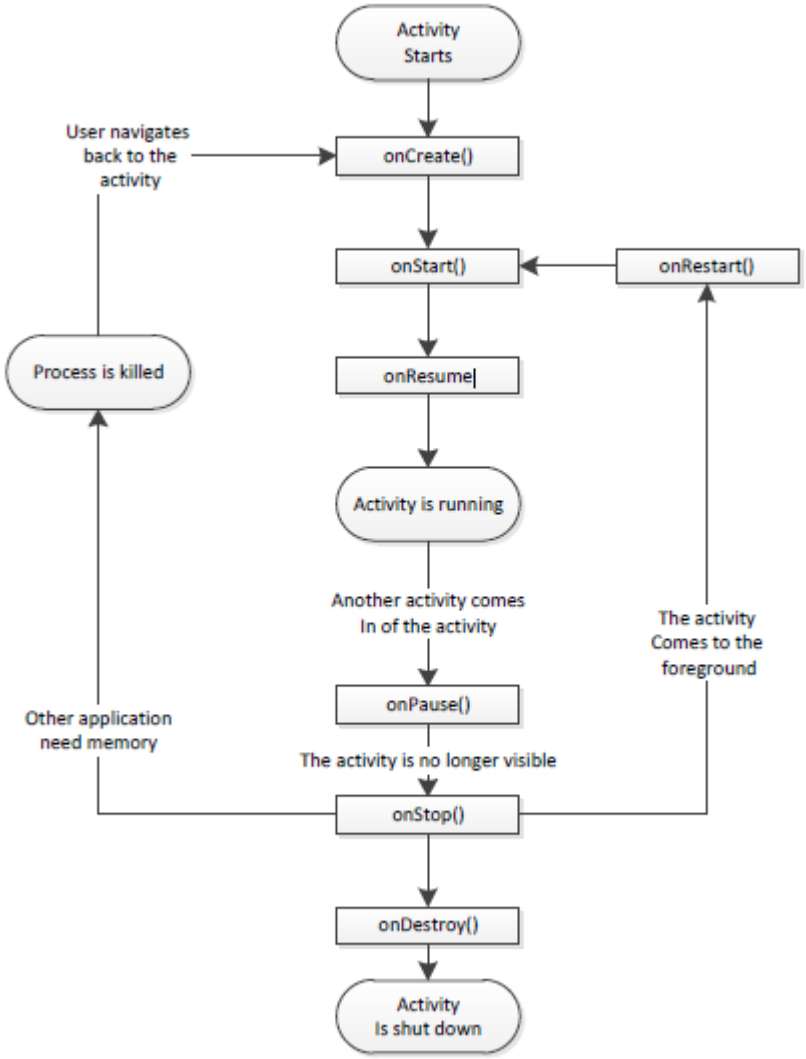
Broadcast receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *broadcast* berasal dari sistem contohnya baterai sudah hampir habis, informasi zona waktu sudah berubah, atau user telah mengganti bahasa *default* pada perangkat Android. Sama halnya dengan *service*, *broadcast receivers* tidak menampilakan antarmuka *user*. Namun, *broadcast receivers* dapat menggunakan *notification* manager untuk memberitahukan sesuatu kepada *user*.

10. *Notifocations*

Notifications merupakan kerangka pemberitahuan untuk *user*. *Notifications* memberikan isyarat atau peringatan tanpa mengganggu aktivitas yang sedang dilakukan oleh user.

2.5.1. Siklus Hidup Aplikasi Android

Siklus hidup aplikasi Android dikelola oleh sistem, berdasarkan kebutuhan user dan sumber daya yang tersedia. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika user sedang menjalankan sebuah activity, maka sistem akan memprioritaskan aplikasi tersebut. Sebaliknya, jika suatu activity tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka activity yang prioritas rendah akan ditutup[5]. Berikut adalah siklus hidup activity Android pada gambar 2.1:



Gambar 2.1 Siklus hidup aplikasi Android

Pengguna mengakses method init pada activity class, dan Android akan memanggilnya menurut waktu yang tepat.

1. *onCreate*( )

   Dipanggil ketika *activity* pertama kali dibuat, disini user melakukan pengaturan statis secara normal, misal membuat *view*, memasukan data dan sebagainya. selalu diikuti dengan *onStart*( ).

2. *onStart*( )

   Dipanggil sebelum *activity* terlihat oleh pengguna. Selalu diikuti dengan *onResume*( ).

3. *onResume*( )

   Dipanggil sebelum *activity* mulai berinteraksi dengan *user*. Pada titik ini, *activity* berada pada *stack activity* yang paling atas. Selalu diikuti dengan *onPause*( ).

4. *onPause*( )

   Dipanggil ketika sistem akan memulai *activity* lain. *Method* ini biasa digunakan untuk melakukan perubahan yang tidak disimpan kedalam data yang tetap, menghentikan animasi, dan hal lain yang memakai *CPU*. Diikuti *onResume*( ) jika *activity* tetap berada dilayar atau *onStop*( ) jika *activity* menjadi tidak terlihat oleh user.

5. *onStop*( )

   Dipanggil ketika *activity* tidak lagi terlihat oleh *user*. Ini mungkin terjadi karena *activity* kemudian dihancurkan atau jika *activity* lain yang sudah ada atau yang baru telah dilanjutkan. Diikuti dengan *onRestart*( ) jika *activity* kembali berinteraksi dengan user atau dengan *onDestroy*( ) jika *activity* akan ditutup.

6. *onRestart*( )

   Dipanggil setelah *activity* dihentikan, dan merupakan prioritas untuk dijalankan kembali dan selalu diikuti *onStart*( ).

7. *onDestroy*( )

   Dipanggil sebelum *activity* dihancurkan, ini merupakan panggilan terakhir yang diterima *activity*. Bisa dikatakan *activity* diakhiri (*finish*( )), atau karena sistem secara *temporary* menghancurkan *instant activity* untuk menghemat ruang.

8. *onSaveInstanceState*( )

   Android akan memanggil *method* ini untuk mengizinkan *activity* menyimpan setiap contoh keadaan, seperti posisi kursor dalam sebuah *text field*.

9. *onRestoreInstanceState*( )

   Dipanggil ketika *activity* di inisialisasi ulang sejak keadaan sebelum disimpan dari *onSaveInstanceState*( ).

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin *virtual* pengolah sendiri, dengan melindungi penggunaan memori pada aplikasi. Selain itu juga Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karena itu Android menjadi sangat sensitif dengan siklus hidup aplikasi dan komponen-komponennya.

2.5.2. Karakteristik Android

Andorid merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, middleware, dan aplikasi inti yang di-release oleh Google. SDK adalah suatu tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform atau linkungan Android. Pengembangan aplikasi Android menggunakan bahasa pemrograman Java. Seperti kita ketahui, SDK ini dikembangkan oleh OHA (Open Handsate Alliance). Organisasi OHA ini terdiri atas Google, Intel, Motorola, Qualcomm, T-Mobile, dan NVIDIA [8].

2.5.3. Perkembangan Android

Dikutip dari http://id.wikipedia.org/wiki/Android_(sistem_operasi) Android telah banyak melakukan perkembangan, berikut adalah perkembangan dari android :

1. Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android Versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara

otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 (Donuts)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android Versi 2.1 (Éclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5. Android Versi 2.2 (Froyo)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android Versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan

WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android Versi 3.0 (HoneyComb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. Android Versi 4.0 (Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

9. Android Versi 4.1 (Jelly Bean)

Diumumkan ada tanggal 27 Juni 2012 meruupakan pembaruan yang 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.

10. Android Versi 4.4 (Kitkat)

Google mengumumkan Android 4.4 KitKat dengan pembaruan antarmuka bar status dan navigasi transparan pada layar depan, dengan optimasi kinerja perangkat dengan spesifikasi yang lebih rendah, perluasan fungsionalitas bagi

layanan pendengar notifikasi dengan menambahkan dukungan Bluetooth Message Access Profil (MAP).

11. Android Versi 5.0 (Lollipop)

Salah satu perubahan yang paling menonjol dalam rilis Lollipop adalah user interface yang didesain ulang dan dibangun dengan yang dalam bahasa desain disebut sebagai "material design". Perubahan lain termasuk perbaikan pemberitahuan, yang dapat diakses dari lockscreen dan ditampilkan pada banner di bagian atas screen. Google juga membuat perubahan internal untuk platform, dengan Android Runtime (ART) secara resmi menggantikan Dalvik untuk meningkatkan kinerja aplikasi, dan dengan perubahan yang ditujukan untuk meningkatkan dan mengoptimalkan penggunaan baterai, yang dikenal secara internal sebagai Project Volta.

12. Android Versi 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.

2.5.4. Kelebihan Android

Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul

sebelumnya pada platform lain [8]. Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap system karena berbasiskan Linux dan open source. Pembuat perangkat menyukai hal ini karena dapat membangun platform yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang software menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
3. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja system menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
4. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan OpenGL memungkinkan membuat aplikasi maupun game yang berbeda.
5. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan Keyboard, layar sentuh, trackball dan resolusi layar semua dapat disesuaikan dengan program.

2.5.5. Kekurangan Android

Tak bisa dipungkiri disamping kelebihan tentu saja ada kekurangan dari sistem operasi ini beberapa kekuranganya yaitu seperti :

1. Koneksi Internet yang terus menerus, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif.

Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.

2. Iklan – Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

2.5.6. 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 memberikan kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone/Smartphone [9]. Beberapa fitur-fitur android yang paling penting adalah :

1. Framework : aplikasi yang mendukung pengganti komponen dan reusable.
2. Dalvik Virtual Machine dioptimalkan untuk perangkat mobile
3. Integrated Browser verdasarkan 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 pernagkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Android Studio.

2.5.7.Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangkan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu Intel IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment [10]. Fitur-fitur lainnya adalah sebagai berikut :

1. Menggunakan Gradle-based build system yang fleksibel.
2. Bisa mem-build multiple APK.
3. Template support untuk Google Services dan berbagai macam tipe perangkat.
4. Layout editor yang lebih bagus.
5. Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Mesbdvcessaging dan App Engine.
6. Import library langsung dari Maven repository

Satu hal tambahan lagi yang membuat Android Studio unggul adalah dukungan layout xml editor secara visual yang jauh lebih baik daripada Eclipse.

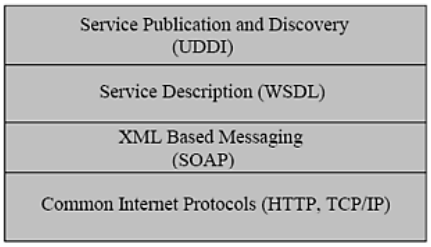
2.6. *Web Services*

Web service merupakan suatu komponen software yang merupakan selfcontaining, aplikasi modular self-describing yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada web. Web service adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola program-to-program (P2P). Fokus web selama ini didominasi oleh komunikasi program-to-

user dengan interaksi business-to-consumer (B2C), sedangkan transactional web akan didominasi oleh program-to-program dengan interaksi business-to-business [11].

Gambar 2.2 merupakan blok bangunan web service yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur web service.

1. Layer 1 : Protokol internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.
2. Layer 2 : Simple Object Access Protocol (SOAP) berbasiskan XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.
3. Layer 3 : Web service Definition Language (WSDL) digunakan untuk mendiskripsikan attribute layanan.
4. Layer 4 : Universal Description, Discovery and Integration, yang mana merupakan direktori pusat untuk deskripsi layanan.



Gambar 2.2 Blok Bangunan Web Service

2.6.1.*JavaScript Object Notation (JSON)*

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python

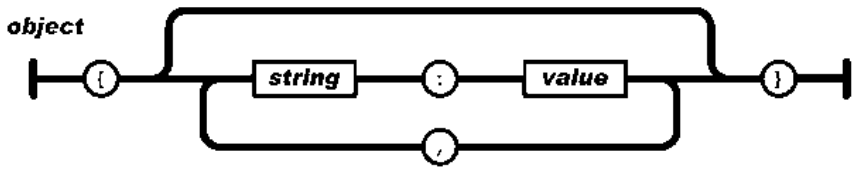
dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [12].

JSON terbuat dari dua struktur :

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.
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*).

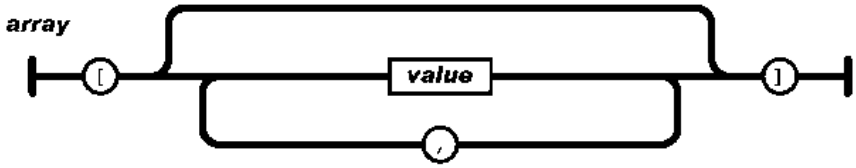
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga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

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).



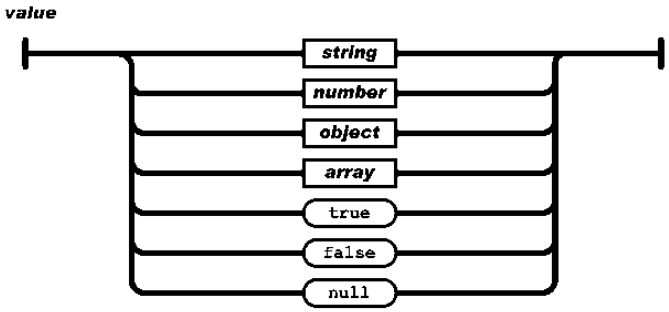
Gambar 2.3 Objek JSON

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



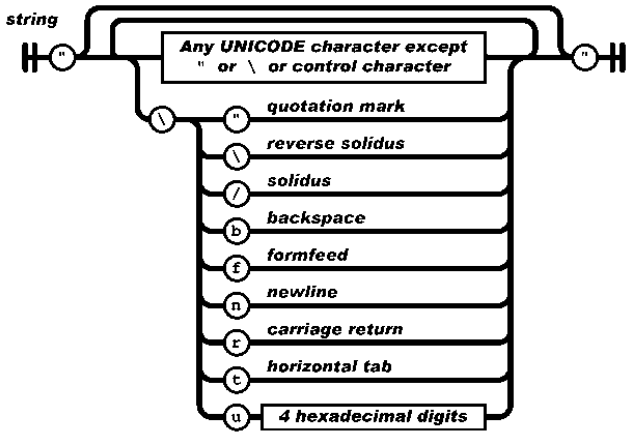
Gambar 2.4 Array JSON

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



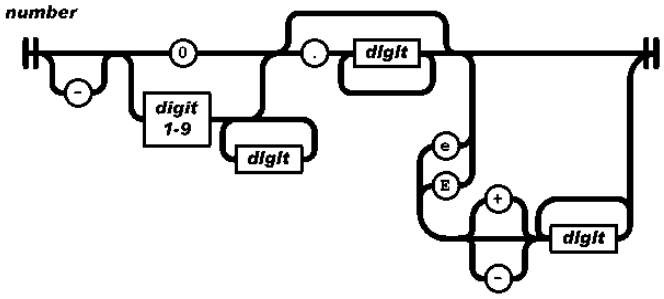
Gambar 2.5 Value JSON

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



Gambar 2.6 String JSON

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2.7 Number JSON

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

2.6.2.Application Programming Interface (API)

Application Programming Interface (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun

bahasa pemograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak.

Dalam suatu pemograman dibutuhkan setidaknya ribuan *system call* per detik oleh karena itu banyak *programmer* yang menggunakan API. Didalam API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih mudah dimengerti oleh *programmer*. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system call* sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari *system call* sama dengan nama API.

Kelebihan pemograman menggunakan API adalah:

1. Portabilitas
   Programmer yang menggunakan API dapat menjalankan programnya dengan sistem operasi mana saja asalkan sudah terinstal API tersebut.
2. Mudah Dimengerti
   API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*.

2.7. Global Positioning System (GPS)

Global positioning System adalah sistem navigasi berbasis satelit terdiri dari jaringan 24 satelit ditempatkan ke orbit oleh Departemen Pertahanan Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978. Layanan GPS dahulu hanya dipergunakan untuk keperluan militer namun mulai terbuka untuk publik. 24 satelit GPS tersebut berada sekitar 12.000 mil di atas bumi bergerak mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, memiliki baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari dan memiliki roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS receiver, sehingga harus selalu

dipelihara agar posisinya tepat. Posisi satelit-satelit tersebut selalu dipantau oleh stasiun pengendali. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS receiver yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS receiver mengambil informasi tersebut dan melakukan perhitungan triangulation untuk menentukan lokasi pengguna dengan tepat. GPS receiver membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit. Dengan mengetahui jarak tersebut, GPS receiver dapat melakukan perhitungan dan menentukan posisi pengguna dan menampilkan dalam peta elektronik. Setelah menentukan posisi pengguna, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS sangat presisi dalam mengirim informasi karena satelit tersebut menggunakan jam atom. Jam atom yang ada pada satelit merupakan partikel atom yang diisolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Keistimewaan GPS adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat GPS bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi Wide Area Augmentation System (WAAS) keakuratannya sampai 3 meter. Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan di tempat terbuka. Penggunaan di dalam ruangan, atau di tempat yang menghalangi arah satelit (di angkasa), maka GPS tidak akan bekerja secara akurat dan maksimal. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi. Semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data posisi yang dihasilkan. Selain itu, ketinggian juga mempengaruhi proses kerja GPS, karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit.

2.7.1.Location Based Services (LBS)

Location based service adalah *Service* yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* **(**GPS) dan *Google's Cell-Based Location*. Map dan layanan berbasis lokasi menggunakan garis lintang dan garis bujur untuk menentukan letak geogragis, namun sebagai user kita membutuhkan alamat atau posisi *real time* kita bukan nilai garis lintang dan garis bujur. Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding.* Menggunakan *geocoder,* anda dapat mengkonversi nilai garis lintang dan garis bujur menjadi alamat dunia nyata atau sebaliknya.

2.8. Unified Modeling Language (UML)

Saat ini piranti lunak semakin luas dan besar lingkupnya, sehingga tidak bisa lagi dibuat asal-asalan. Piranti lunak saat ini seharusnya dirancang dengan memperhatikan hal-hal seperti scalability, security, dan eksekusi yang robust walaupun dalam kondisi yang sulit. Selain itu arsitekturnya harus didefinisikan dengan jelas, agar bug mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya.

Keuntungan lain dari perencanaan arsitektur yang matang adalah dimungkinkannya penggunaan kembali modul atau komponen untuk aplikasi piranti lunak lain yang membutuhkan fungsionalitas yang sama. Pemodelan (modeling) adalah proses merancang piranti lunak sebelum melakukan pengkodean (coding). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung.

Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Dengan menggunakan model, diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti scalability, robustness, security, dan sebagainya. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur,

yang kemudian terkenal dengan sebuan segitiga sukses (the triangle for success). Ketiga unsur tersebut adalah metode pemodelan (notation), proses (process) dan tool yang digunakan.

1. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

2. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan

perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

3. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok : 1. Nama (dan stereotype) 2. Atribut 3. Metoda Atribut dan metoda dapat memiliki salah satu sifat berikut : • Private, tidak dapat dipanggil dari luar class yang bersangkutan • Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya • Public, dapat dipanggil oleh siapa saja Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package. Hubungan Antar Class 1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class. 2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”). 3. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi. 4. Hubungan dinamis,

yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

2.9. Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java. Pada Desember 1998, Sun memperkenalkan nama "Java 2" (J2) sebagai generasi kedua dari java platform. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME) [12].

Ada tiga platform Java yang masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda:

1. Standard Edition (J2SE) : J2SE merupakan inti dari bahasa pemrograman Java. J2SE didesain untuk jalan pada komputer desktop dan komputer workstations.
2. Enterprise Edition (J2EE) : Dengan built-in mendukung untuk servlets. JSP, dan XML, edisi ini ditujukan untuk aplikasi berbasis server.
3. Micro Edition (J2ME) : Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

2.10. Database Management System (DBMS)

DBMS (*Database Management System*) adalah suatu system aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. suatu system aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data secara bersamaan
4. Mampu menangani backup data

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi Relational DBMS yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan open source saat ini berbasis Relastional DBMS. Pada tahun 1980an mulai berkembang Object Oriented DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Saat ini OODBMS juga cukup berkembang namun belum menggeser kepopuleran RDBMS.MySQL adalah perangkat lunak DBMS atau server basis data yang dapat melayani banyak pengguna, dan dapat melakukan proses dalam waktu yang bersamaan (multi threaded). MySQL juga sangat cepat dan fleksibel dan dapat menyimpan data yang berkapasitas besar. Dalam perancangannya, MySQL mengutamakan kecepatan proses, ketangguhan dan kemudahan bagi penggunanya. Pada saat ini MySQL adalah server basis data yang paling banyak diminati oleh banyak perusahaan di dunia. MySQL menyediakan lingkungan yang mudah dan efisien untuk menyimpan data dan mengakses basis data berkapasitas besar lebih cepat dari pada server basis data lainnya. Keuntungan lainnya adalah bahwa MySQL server tidak membutuhkan spesifikasi hardware yang mahal. MySQL server memungkinkan membuat aplikasi client untuk mengakses dan memanipulasi data di server basis data melalui basis data provider. Keuntungan utama menggunakan MySQL server adalah mudah digunakan, berkecepatan tinggi dan membutuhkan sedikit memori serta tidak membutuhkan spesifikasi hardware yang mahal [13].

2.11. Case-Based Reasoning (CBR)

Metode Case Based Reasoning adalah salah satu metode untuk membangun sistem pakar dengan pengambilan keputusan dari kasus yang baru dengan berdasarkan solusi dari kasus-kasus sebelumnya. Konsep dari metode case based reasoning ditemukan dari ide untuk menggunakan pengalaman - pengalaman yang terdokumenatsi untuk mengyelesaikan masalah yang baru. Para *decisionmaker*

kebanyakan menggunakan penglaman - pengalaman dari problem solving terdahulu untuk menyelesaikan masalah yang dihadapi sekarang [13].

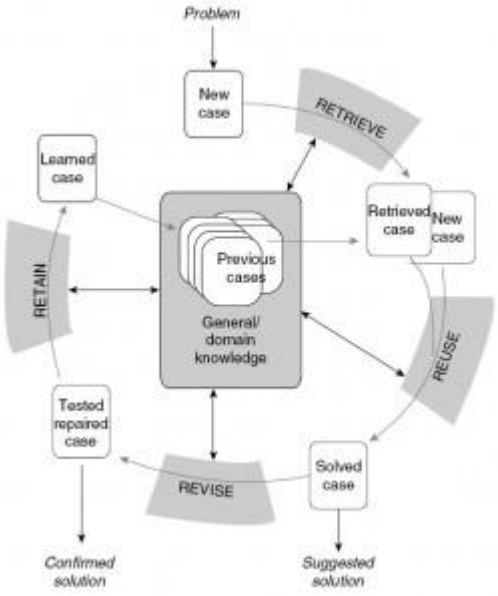
Metode Case Babe Reasoning menggunakan pendekatan kecerdasan buatan yang menitikberatkan pemecahan masalah dengan didasarkan pada *knowledge* dari kasus-kasus sebelumnya. Apabila ada kasus baru maka akan disimpan pada basis pengetahuan sehingga sistem akan melakukan *learning* dan *knowledge* yang dimiliki oleh sistem akan bertambah. Secara umum, metode ini terdiri dari empat langkah, yaitu:

1. Retrieve (memperoleh kembali) kasus-kasus yang paling mirip. Task ini dimulai dengan pendeskripsian satu/sebagian masalah dan berakhir apabila telah ditemukan kasus sebelumnya yang paling cocok. Sub tasknya mengacu pada identifikasi fitur, pencocokan awal, pencarian dan pemilihan.
2. Reuse (menggunakan) informasi dan pengetahuan dari kasus tersebut untuk memecahkan permasalahan. Proses reuse dari solusi kasus yang telah diperoleh dalam konteks kasus baru difokuskan pada dua aspek yaitu:
    1. Perbedaan antara kasus yang sebelumnya dan yang sekarang
    2. Bagian apa dari kasus yang telah diperoleh yang dapat ditransfer menjadi kasus baru
3. Revise (meninjau kembali/memperbaiki) usulan solusi. Fase ini terdiri dari dua tugas, yaitu:
    1. Mengevaluasi solusi kasus yang dihasilkan oleh proses reuse. Jika berhasil, maka dilanjutkan dengan proses retain.
    2. Jika tidak maka memperbaiki solusi kasus menggunakan domain spesifik pengetahuan.
4. Retain (menyimpan) bagian-bagian dari pengalaman tersebut yang mungkin berguna untuk memecahkan masalah di masa yang akan datang. Proses ini terdiri dari memilih informasi apa dari kasus yang akan disimpan.

Komponen penyelesaian masalah CBR memiliki dua bagian utama : *case retriever* dan *case reasoner*. *Case retriever* bertugas untuk mencari masalah-masalah yang mirip dengan masalah yang dimasukkan pengguna di dalam basis pengetahuan (*case base*), sementara *case reasoner* bertugas untuk mencari solusi masalah pengguna dengan melihat atau menyesuaikan solusi masalah yang ditemukan *case retriever*.

Jadi Metode CBR sendiri terdapat tahap-tahap yang harus diselesaikan secara berurutan. dalam mengembangkan aplikasi,pengguna harus membuka tahap-tahap dalam aplikasi. Setiap proses tersebut harus dikerjakan sampai paham.jadi jika ada kebutuhan tambahan maka harus menunggu sampai tahap akhir.

Tahapan pada metode CBR (*Case-Based Reasoning*) ada 4 yaitu, *retrieve, reuse, revise* dan *retain.* Hubungan antara tahapan pada metode CBR (*Case-Based Reasoning*) [9] dapat dilihat pada gambar berikut.



Gambar 2.8 Siklus Case-Based Reasoning

2.12. Skala Likert

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuisioner dan meruoakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Skala likert ini merupakan skala yang

dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena. Pada evaluasi, skala likert digunakan untuk menilai keberhasilan suatu program, menilai manfaat pelaksanaan suatu program, dan mengetahui kepuasan *stakeholder* terhadap pelaksanaan suatu program.

Rensis Likert telah mengembangkan suatu skala untuk mengukur sikap masyarakat di tahun 1932 yangsekarang terkenal dengan nama skala likert. Skala likert hanya menggunakan item yang secara pasti baik dan pasti buruk. Skala likert menggunakan ukuran ordinal, karena hanya dapat membuat rangking, tetapi tidak dapat diketahui berapa kali satu responden lebih baik atau kebih buruk dari responden lainnya dalam skala [15].

2.13. Pengujian Black-Box

Menurut Pressman (2010, p495) *Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemusatakan kesalahan