BAB 2

LANDASAN TEORI

2.1 Android

Android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Androd menyediakan platform terbuka bagi para pemengembang untuk menciptakan aplikasi mereka sendiri untuk di gunakan oleh bermacam peranti bergerak. Beberapa pengertian lain dari Android, yaitu

1. Merupakan platform terbuka (*open source*) bagi para programmer untuk membuat aplikasi
2. Merupakan sistem operasi yang dibeli Google Inc dari Android Inc.

2.2 Sejarah Sistem Operasi Android

Pada juli 2000, Google bekerjasama dengan Android Inc. Perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. Bekerja pada Google, Diantaranya Andy Rubin,Rich Miner,Nick Sears, Dan Chros White. Saat itu banyak yang menganggap fungsi Android Inc. Hanyalah sebagai perangkat lunak pada telepon seluler. Sejak itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukan indikasi bahwa Google sedang besiap menghadapi persaingan dalam pasar telepon seluler. Versi android tersebut yaitu versi 4.0 (*Ice Cream Sandwitch*). Android juga sudah bergabung dengan beberapa smart mobile seperti Samsung,LG,Sony Ericcson, dan lainnya.

Sekitar september 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus one, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasi nya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia dipasaran pada tahun 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications,diproduksi oleh Asustek komputer Inc, GarminLtd, Softbank,Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.Seiring pembentukan Opne Handset Alliance, OHA mengumumkan produk perdana mereka, Android Perangkat bergerak (Mobile) yang merupakan modifikasi kerel linux 2.6 Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur terbaru.

2.3 Perangkat Lunak Pendukung

2.3.1 PHP

PHP merupakan singkatan dari "*Hypertext Preporcessor*". Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal) dan PHP itu sendiri pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, dan pada saat PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data *form* dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum. PHP adalah sebuah Bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaksnya mirip dengan Bahasa pemograman C, *Java*, ASP dan Perl, ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti.

PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bias menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di *include* atau *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda, yaitu seperti DBM, FilePro (*Personil, Inc*), *Informix, Ingres, Interbase, Microsoft Access*, MSSQL, MySQL, *Oracle*, *Postgre*SQL dan *Sybase*.

2.3.2 Java

Java adalah bahasa pemograman tingkat tinggi yang berorientasi objek, diedarkan oleh sun Microsytems pada awal tahun 1996. Sejarah awal java berwal di tahun 1991 ketika satu group insinyur-insinyur sun, yang dipimpin oleh *Patrick Naughton* dan james Gosling, ingin mendesain sebuah bahasa pemograman komputer yang berukuran kecil yang dapat digunakan untuk peralatan elektronika konsumen seperti *switchboxe* TV kabel.

Dikarenkan peralatan-peralatan ini menggunakan konsumsi daya dan *memory* yang rendah, maka bahasa pemograman tersebut harus berukuran sangat kecil. Juga kerena stiap vendor menggunakan CPUs (Central Processing Unit) yang berbeda, maka bahasa tersebut harus bersifat multiptfrom, tidak terikat hanya pada satu arsitektur (*Architecture Neutral).* Proyek ini diberi nama *Green Project.*

2.3.3 Android Studio

Android Studio adalah IDE resmi untuk membangun aplikasi Android berdasarkan IntelliJ IDEA. IntelliJ IDEA sendiri adalah *Java Integrated Development Environtment* (IDE) yang dikembangkan oleh JetBrains, untuk mengembangkan perangkat lunak komputer. IntelliJ IDE berfungsi dalam membantu anda di dunia percodingan baik dari segi navigasi, penyokong produktivitas, hingga code editor yang cerdas[19]. Berdasarkan website http://developer.android.com, android studio memiliki keunggulan:

1. Android studio didasarkan pada intellij DEA yang memberikan kemungkinan tercepat dalam melakukan coding dan *running workflow* sehingga meng-*coding* dan melakukan iterasi dapat dilakukan dengan cepat.
2. Struktur proyek Android Studio dan pembangunannya yang berbasis *gradle* memberikan fleksibilitas yang dibutuhkan untuk menghasilkan APK untuk semua jenis *device* sehingga konfigurasi pemangunan dapat dlakukan tanpa batas.
3. Android Studio menyediakan fitur-fitur yang membuat pengembang menjadi percaya diri dan dapat membuat kode terbaik.

4. *Android Studio* mengetahui bahwa tidak seluruh kode ditulis dengan bahasa java dan tidak semua kode dapat berjalan di *device* pengguna. Dengan fitur-fitur yang dimiliki *Android Studio* maka aplikasi menjadi kaya dan terkoneksi satu sama lain.
5. *Android Studio* menyediakan alat GUI untuk mempermudah pengembang dalam merancang tampilan aplikasi sehingga pengembang tidak lelah.

Dengan menggunakan *android studio* pengembang aplikasi dapat mengerjakan aplikasi yang akan dibangunya. Pada pembangunan aplikasi pemandu wisata Kota Bandung *android studio* digunakan untuk menjalankan atau *debugging* program yang dibangun untuk mencari dan mengurangi bug, atau kerusakan didalam sebuah program, sehingga aplikasi dapat bekerja sesuai dengan harapan.

2.4 UML (United Modeling Language)

UML (United Modeling Language) Merupakan bahasa pemodelan grafis yang digunakan untuk mendesain dan membantu pendeskripsikan sistem perangkat lunak, khususnya sistem yang berorientasu objek. UML mencakup berbagai masalah yang meliputi spesifikasi, visualisasi, kontruksi dan documentasi berbagai jenis perangkat lunak sistem, perangkat non-lunak sistem, dan model bisnis, selain itu dapat digunakan dalam pengembangan berbagai tahapan,mulai dari analisis kebutuhan sistem sampai implementasi sistem. UML memenuhi persyaratan objek analisi dan desain karena termasuk diagram alernatif untuk menjelaskan statis properti, penggunaan sistem atau kompones, dan sistem arsitektur. UML merupakan suatu bahasa pemodelan yang terdiri banyak model.

2.4.1 Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah 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

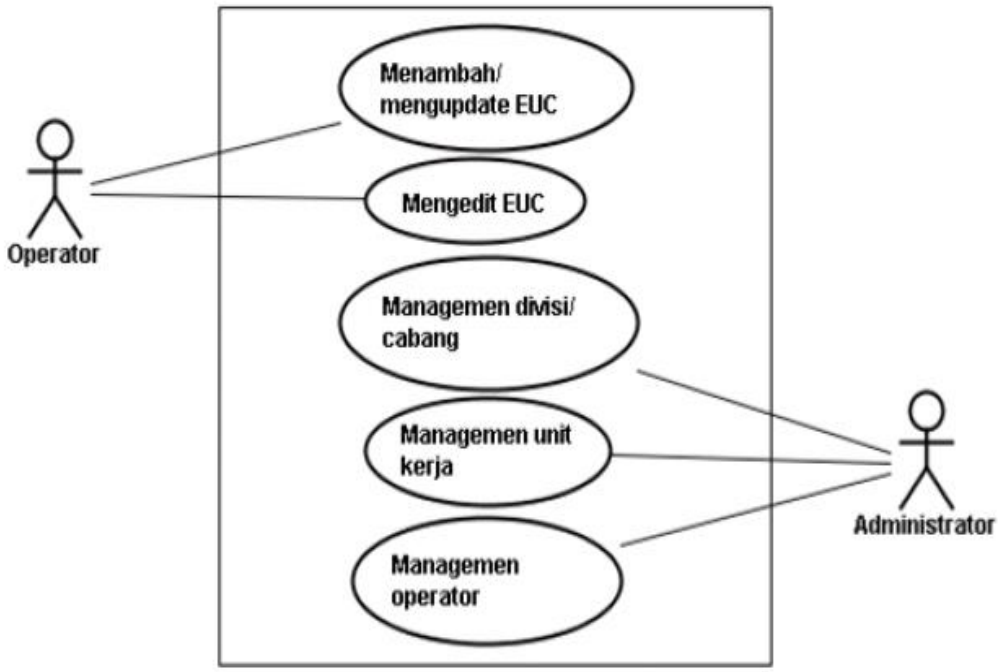
sistem, meng-create sebuah daftar penemuan , dan sebagainya. Seorang aktor adalah sebuah entintas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan pekerjaan tertentu. Adapun komponen-komponen dalam Use Case diagram antaranya.

2.4.2 Actor

Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi computer. Jadi aktor ini bias berupa orang, perangkat keras atau objek yang lain dalam sistem yang sama. Biasanya yang dilakukan oleh aktor adalah memberi informasi pada sistem atau memerintahkan sistem melakukan sesuatu

1. Use Case

   Use Case mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permitaan user/aktor. Setiap use case mengekpresikan tujuan dari sistem yang harus dicapai. Diberi nama sesuai dengan tujuannya dan digambarkan dengan elips dengan nama di dalamnya.



Gambar 2. 1 Contoh UseCase

a. System

Menyatakan batasan sistem dalam relasi dengan aktor-aktor yang menggunakan (diluar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Di gambarkan dengan segi empat yang membatasi semua use case dalam sistem terhadap pihak mana sistem akan berinteraksi.

b. Assosication

Menginditifikasi interaksi antara setiap aktor tertentu dengan setiap use case tertentu. Digambarkan sebagai garis antara aktor terhadap use case yang bersangkutan

c. Dependency

Dependency (include)

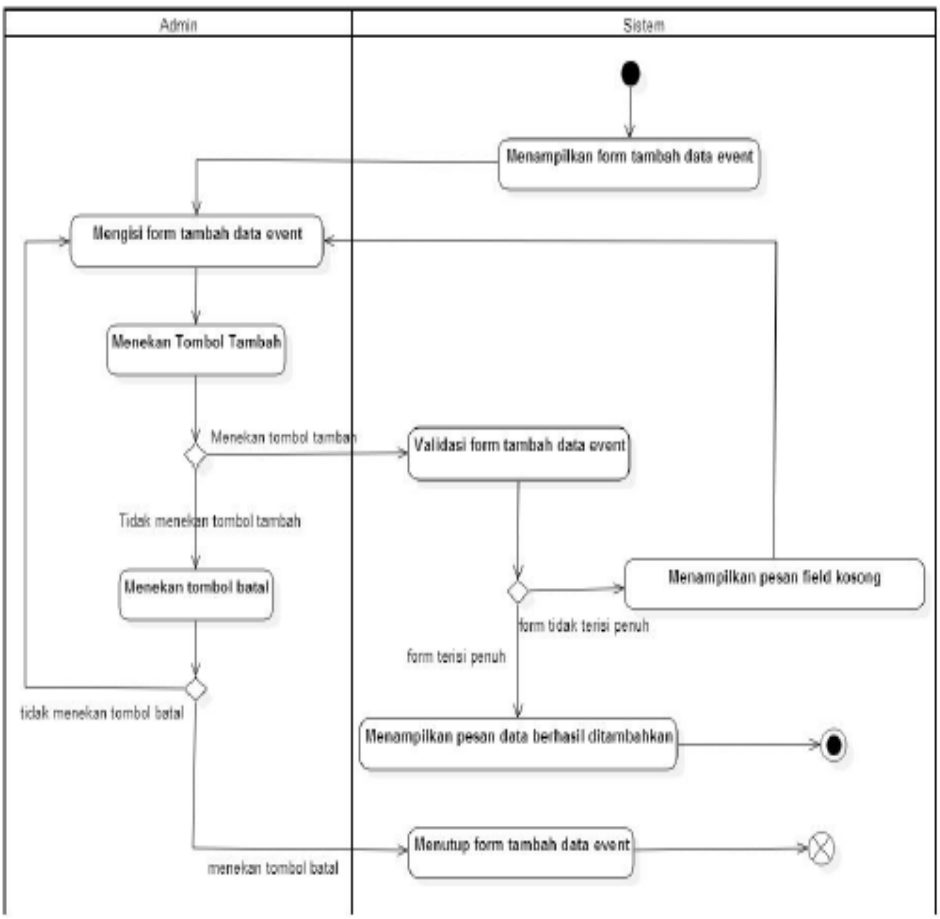
1. Mengidentifikasi hubungan antara 2 use case dimana yang satu memanggil yang lain.
2. Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis.
3. Jika beberapa use case terhadap bagian yang memiliki aktivitas yang sama maka bagian aktivitas tersebut biasanya dijadikan use case tersendiri dengan relasi dependensi setiap use case semula ke use case yang baru ini sehingga memudahkan pemeliharaan.

Dependency (Extend)

1. Digambarkan serupa dengan dependensi <<include>> kecuali panah beralawanan.
2. Jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>>

2. Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *use case*.

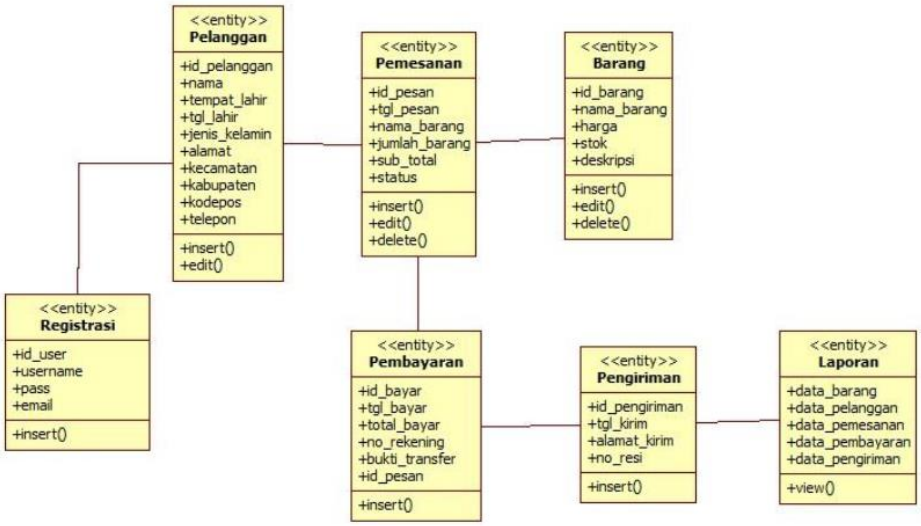


Gambar 2. 2 Contoh Activity Diagram

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 (attribut/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi class, packpage dan objek beserta hubungan satu sama lain seperti containment,pewaris,asosiasi dan lain-lain. Class dapat merupakan implementasi dari sebuah interface, yaitu Class Abstrak yang hanya memiliki metode. Interface tidak dapat langsung diinstansiasikan, tetapi harus

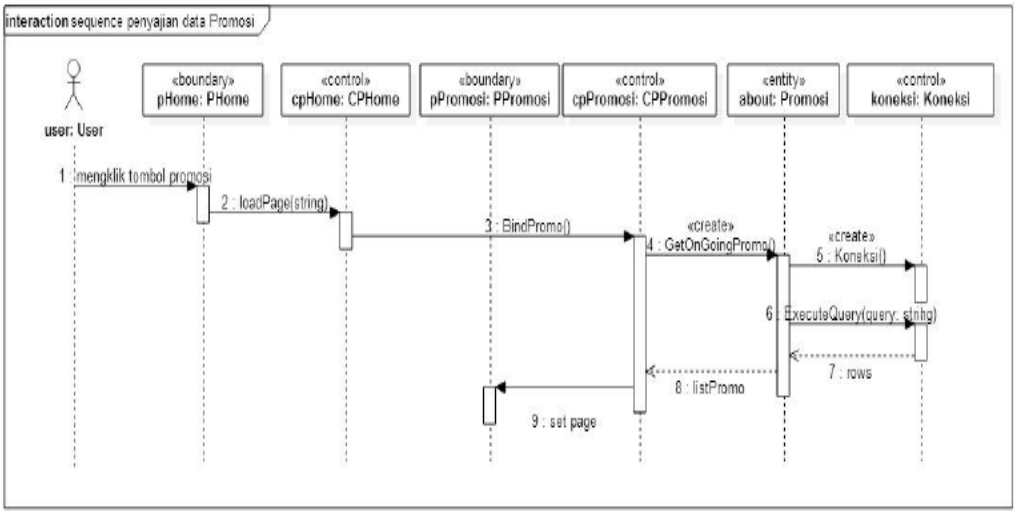
diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metode pada saat run-time.



Gambar 2. 3 Contoh Class Diagram

4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara objek didalam dan di sekitar sistem (termasuk pengguna,display,dan sebagainya) berupa massage Yang di gambarkan terhadap waktu. Sequence diagram terdiri antara dimensi vertical (waktu) dan dimensi horizontal (objek-objek tang terkait). Sequence diagram biasa digunakan untuk menggambarkan scenario atau rangkain langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk mengahasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

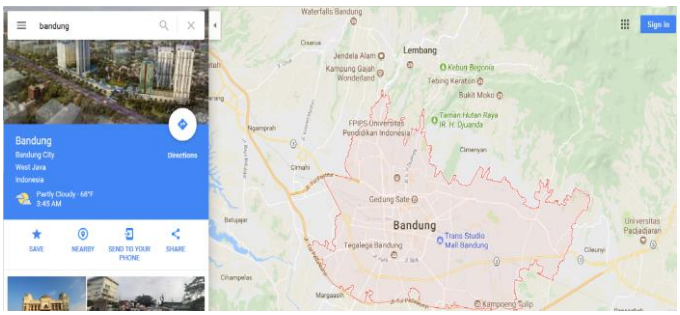


Gambar 2. 4 Contoh Sequence Diagram

2.5 Google Map Service

Google Map Service adalah sebuah jasa peta global virtual gratis dan online yang disediakan oleh perusahaan Google. *Google Maps yang dapat ditemukan di alamat* http://maps.google.com Google Maps menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia Google Maps juga menawarkan pencarian suatu tempat dan rute perjalanan.

Google Maps API adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk mempaatkan *Google Maps* dalam mengmbangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulaso peta, dan menambahkan konten melalui berbagai jenis *service* yang dimiliki, serta menginjinkan kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya.



Gambar 2. 5 Contoh Google Maps

2.6 Global Position System (GPS)

GPS atau Global positioning system, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana lokasinya berada(secara Global) di permukaan bumi yang berbasis stalite. Data dikrim dari satelit berupa sinyal radio dengan data digital. Dimana pun pengguna tersebut berada, maka GPS bisa membatu menunjukan arah.

2.7 Geolocation

Geolocation adalah sebuah cara untuk mengetahui suatu lokasi didunia. Ada beberapa metode untuk menemukan lokasi, yaitu dengan IP *address*, sambungan wireless atau BTS, dan dedicated GPS atau embeded GPS pada telepon seluler. Geolocation menggunakan data koordinat latitude dan longitude yang dimiliki oleh komputer atau seluler.

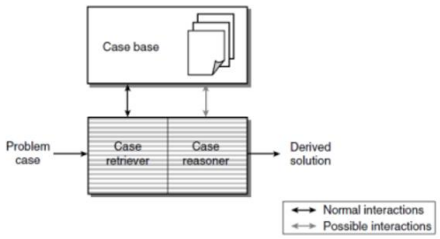
2.8 Google Cloud Vision

API Google Cloud Vision merupakan pengembang untuk memahami konten gambar dengan mengenkapsulasi model pembelajaran mesin yang canggih dengan API REST yang mudah digunakan. Dengan cepat mengklasifikasikan gambar ke dalam ribuan kategori misal Perahu layar, singa, Menara Eiffel, mendeteksi objek dan wajah individual dalam gambar, dan menemukan dan membaca kata-kata tercetak yang terdapat di dalam gambar. Anda dapat membangun metadata pada katalog gambar Anda, konten ofensif yang moderat, atau mengaktifkan skenario pemasaran baru melalui analisis sentimen gambar. Analisis gambar yang diunggah dalam permintaan atau integrasikan dengan penyimpanan gambar Anda di Google Cloud Storage.

2.9 Case-Based Reasoning (CBR)

Case-Based Reasoning (CBR) adalah metode penyelesaian masalah dengan menggunakan solusi masalah-masalah sebelumnya yang serupa. CBR sendiri

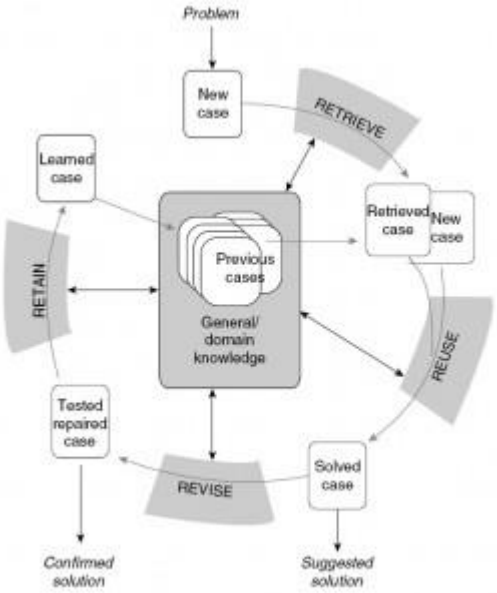
adalah metode yang umum digunakan manusia dalam menyelesaiakan masalah sehari-hari.



Gambar 2. 6 Komponen Case-Based Reasoning

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*.

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



Gambar 2. 7 Siklus Case-Based Reasoning

1. *Retrieve* : yaitu mengambil kembali permaslaahan yang sama. Pada langkah ini dilakukan proses pencarian atau kalkulasi dari kasus-kasus yang memiliki kesamaan.
2. *Reuse* : yaitu menggunakan kembali informasi dan pengetahuan dalam kasus tersebut untuk mengatasi masalah baru. Pada langkah ini dicari solusi dari kasus serupa pada kondisi sebelumnya untuk permasalahan baru.
3. *Revise* : yaitu meninjau kembali solusi yang diberikan. Pada langkah ini dicari solusi dari kasus serupa pada kondisi sebelumnya untuk permasalahan yang terjadi kemudian.
4. *Retain* : yaitu mendalami bagian dari pengalaman sebelumnya untuk digunakan dalam pemecahan masalah berikutnya.

2.10 Web Service

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-toprogram (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.

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.

2.11 User Generated Content

User Generated Content adalah merunjuk pada berbagai tipe konten materi atau media yang tersedia untuk umum dan di produksi oleh end-user atau para non-media professional. Kemajuan teknologi web memungkinkan konten. Website tidak lagi dimonopoli oleh professional media, melainkan dapat dibuat oleh para penggunanya. Menggunakan *User Generated Content* (UGC) juga merupakan cara yang bagus untuk membangun hubungan dan *loyalitas follower* Anda. Jika mereka melihat anda memperhatikan apa yang mereka katakan, mereka akan lebih bersemangat dengan merk.

2.12 Xampp

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia database *server* mysql dan *support* php *programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache* Web Server, MySQL Database Server,

PHP *Support* (PHP 4 dan PHP 5) dan beberapa *module* lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah *server* secara *grafis,* sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan [20].

Dalam XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web XAMPP berfungsi sebagai *server offline* yang berdiri sendiri (seringkali disebut sebagai *localhost*). XAMPP terdiri dari beberapa program yaitu *Apache* HTTP Server, MYSQL *Database*, PHP, dan Pearl XAMPP juga dilengkapi fitur manajemen *database PHPMyAdmin* seperti pada

server hosting sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis database secara mudah.

2.13 Metode Pengujian Sistem

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.13.1 Pengujian *Black Box*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut[21]:

1. Fungsi tidak benar atau hilang.
2. Kesalahan *interface* atau antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.13.2 *Skala Likert*

Skala likert adalah suatu skala *psikometrik* yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama *skala Likert*. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena [22].

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan *negative* yang berfungsi untuk mengukur sikap *negative* objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (TS), 2 untuk tidak setuju (S), 3 untuk raguragu

(CS), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. *Skala Likert* digunakan untuk mengukur kesetujuan dan ketidak setujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan *Skala Likert*:

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan *Skala Likert*:

1. Mudah dibuat dan di terapkan.
2. Skala Likert lebih mudah membuatnya dibanding lain sepertti skala *Thurstone*.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa *alternative*, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.