BAB 2
LANDASAN TEORI

2.1 Hadiah

Kamus Besar Bahasa Indonesia mengartikan hadiah sebagai pemberian, persenan, atau kado [1]. Menurut Wolfinbarger, hadiah merupakan sarana untuk menunjukkan ekspresi emosional dan kepribadian atau diri pemberi hadiah [2]. Jr. John menginterpretasikan hadiah sebagai sebuah ajakan pertemanan dan keterlibatan pemberi dalam peristiwa suka dan duka yang dialami penerima. Hadiah juga dapat dijadikan sebagai dasar penciptaan dan pembangunan kembali relasi antara pemberi dan penerima, misalnya ucapan terima kasih, permintaan maaf, simbolisasi cinta, dan berbagai ekspresi emosional yang lain [3].

Pemberian hadiah merupakan aktivitas manusia yang telah mendunia [4]. Praktik kuno pemberian hadiah secara signifikan mempengaruhi budaya di seluruh dunia dan berkembang menjadi sebuah proses menciptakan dan mengelola hubungan sosial [3]. Aktivitas memberi hadiah dalam peristiwa tertentu, seperti perayaan ulang tahun, pernikahan, Natal, Valentine, dan sebagainya, telah biasa dilakukan oleh berbagai orang di seluruh dunia. Mauss mengatakan, pentingnya hadiah biasanya tidak hanya untuk mentransfer objek hadiah itu sendiri, tetapi juga aspek sosial, yaitu keharusan untuk memberi, menerima, dan saling membalas pemberian hadiah [5].

2.2 Android

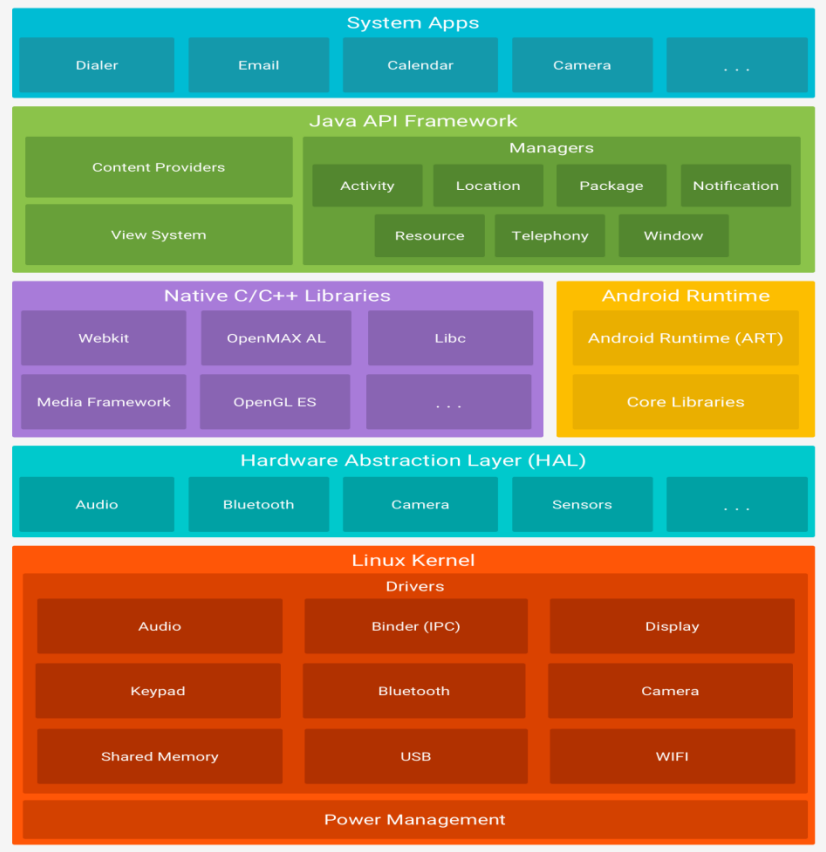
Android merupakan sistem operasi mobile yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainnya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat hardware yang ada. Akan tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang

dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka [6].

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 [6].

2.2.1 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada Gambar 2.1.



Gambar 2.1 Arsitektur Android [7]

1. *System Apps*

   System Apps adalah adalah Aplikasi yang Secara Default atau Built-in ,terinstall di berbagai perangkat Android. Aplikasi ini sering kita gunakan sehari-hari ,seperti Dialer untuk Menelpon ,Email untuk mengirim Email ,SMS untuk mengirim pesan text, dan lain-lain. Tentunya berbagai macam perangkat dan OS Android ,akan berbeda pada System Appsnya. Tergantung brand atau vendor dari perangkat tersebut. Aplikasi ini juga umumnya tidak dapat di Uninstall ,harus menggunakan Aplikasi pihak ketiga. Bagi para opreker mungkin sering menggunakan Aplikasinya untuk melegakan memori internal. Bagi para developer Android sering menggunakan System Apps ,untuk melengkapi fitur pada Aplikasi yang sedang dikembangkan. [8]

2. *JAVA API Framework*

   JAVA API Framework adalah adalah kumpulan library Java yang yang digunakan untuk pengembangan Aplikasi Android. Nah disini lah peran Android Developer , dalam membuat dan mengembangkan Aplikasi mereka. Terdapat 3 Komponen utama yaitu View System ,Content Provider dan Managers. Dalam Managers terdiri dari berbagai macam library seperti ,Activity ,Location ,Package ,Notification dan lain-lain. [8]

3. *Native C/C++ Libraries*

   Ini adalah dukungan library yang diperuntukan bagi kalian developer, yang menggunakan bahasa pemrograman C/C++ untuk mengembangkan Aplikasi Android. Terdapat berbagai macam library yang dapat kita kembangkan , seperti WebKit ,OpennGL ES, Media Framework dan lain-lain. [8]

4. *Android Run Time*

   Android pada dasarnya menggunakan Java, sebagai bahasa pemrograman utama. Java mengcompile ke dalam bytecode, kemudian dieksekusi oleh Java Virtual Machine, yang tersedia di berbagai macam perangkat dan sistem operasi . Intinya bytecode ini adalah hasil binary code

(.class) dari instruksi code java (.java) yang telah dicompile ,dan hanya dapat dibaca oleh Java Virtual Machine ,yang kemudian dilanjutkan untuk digunakan oleh Sistem operasi seperti di Windows ,Linux dan Mac.

Sama halnya untuk Android yang telah dikembangkan oleh Google, yang membuat sebuah virtual machine baru yang bernama **Dalvik**, yang dikhususkan untuk perangkat mobile seperti Android. Jadi ketika ketika kita menulis baris code akan dikompile dua kali yaitu dengan menggunakan Java Bytecode dan Dalvik bye code, sebelum diteruskan ke Dalvik Virtual Machine.

Di Android Run Time(ART) ini yang di kembangkan sejak OS Android KitKat, akan mengcompile Dalvik byte code ke dalam system binary, sehingga aplikasi yang dikembangkan untuk Dalvik akan bekerja ,ketika menggunakan ART. Jika di Dalvik virtual Machine ,akan tereksekusi setiap Aplikasi dijalankan(JIT/Just in time) ,berbeda dengan Android Run Time ,yang akan tereksekusi sekali saja, ketika Aplikasi sudah terinstall di perangkat Android. [8]

5. *Hardware Abstraction*

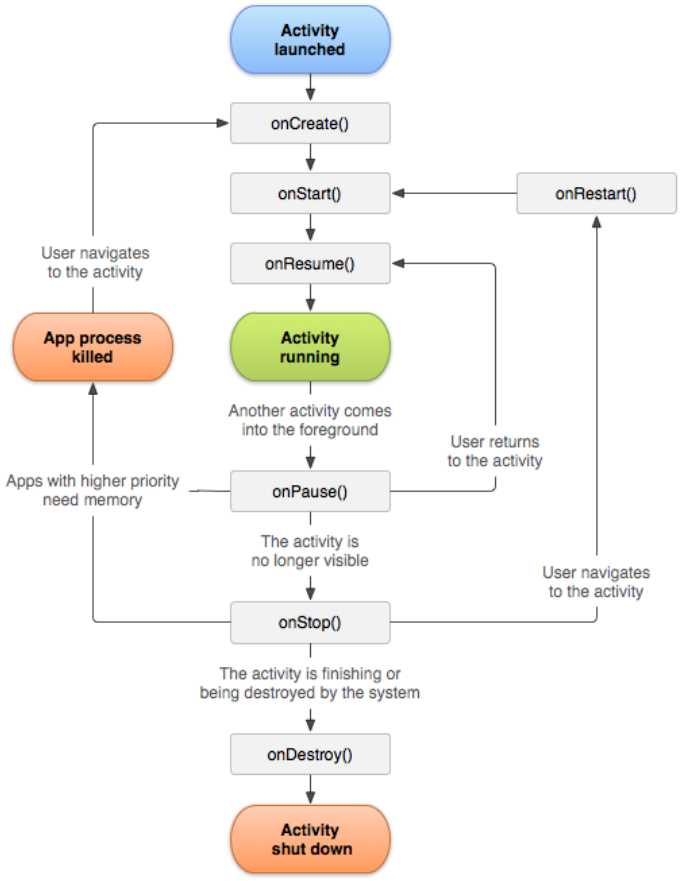
Ini adalah komponen-komponen hardware pada perangkat Smartphone atau tablet pada umumnya , seperti Audio ,Bluetooth ,Camera,Sensors , dan masih banyak lagi. [8]

6. *Linux Kernel*

Linux Kernel ini adalah sebuah inti dari sistem operasi pada umumnya, pada perangkat Android akan menghandle drivers ,agar komponen-komponen hardware yang ada pada perangkat Android bisa berjalan dengan baik. [8]

2.2.2 Android Activity Lifecycle

Terdapat beberapa *state* dalam siklus hidup Android yang terjadi seperti diilustrasikan pada Gambar 2.2.



Gambar 2.2 Android Lifecycle

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() [9].

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 dalan *state* ini [9].

3. Stopped

   Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* di *background* [9]. State lain seperti *Create and Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem onCreate() dipanggil, dengan cepat sistem akan memanggil *method* onStart(), kemudian diikuti oleh onResume().

2.2.3 Fitur

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

1. Multiproses dan App Widget

   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 [7].

   App Widget adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widget dapat menjalankan proses request seperti musik *streaming* atau mendeteksi suhu ruangan secara *background* [7]. Multi-

proses dapat memberikan manfaat berupa user experience yang lebih banyak, namun pengguna fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar [7].

2. Touch Gesture dan Multitouch

   Touchscreen adalah user interface intuitif yang digunakan banyak smartphine di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. Multi-touch adalah kemampuan yang digunakan untuk interaksi memperbesar atau memutar objek. Selain itu pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut. [7]

3. Hard and Soft Keyboard

   Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoprasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang diuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi [7]

2.2.4 Prinsip Desain

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

1. Enchant Me

   Antarmuka yang cantik, animasi yang ditempatkan dengan hati-hati, atau efek suara di saat yang tepat sungguh menyenangkan untuk dinikmati. Efek yang lembut menimbulkan perasaan serba mudah dan kesadaran bahwa kekuatan yang bisa diandalkan ada dalam genggaman. Android juga mengijinkan kita untuk menyentuh dan memanipulasi objek dalam aplikasi. Ini membuat pengguna merasakan pengalaman terbaik dalam sistem. Selain itu Android

memberikan pengalaman untuk menambahkan konfigurasi personal terhadap konfigurasi default yang cantik tetapi tidak menggangu tugas utama. Android dapat mempelajari konfigurasi personal yang dilakukan oleh pengguna secara berulang dan dapat ditampilkan sehingga pengguna dapat melakuan konfigurasi tersebut lebih mudah. [10]

2. Simplify My Life

   Desain Android memudahkan pengguna dengan konsep desain yang simple dan tidak membuat pengguna merasa rumit dalam penggunaannya. Seperti, menggunakan frasa pendek dengan kata-kata sederhana dalam pertanyaan dan kalimat. Karena orang cenderung melewatkan kalimat-kalimat panjang. Mengubah text nama menjadi gambar atau foto yang relevan. Dan menampilkan apa yang hanya kita butuhkan dalam suatu perintah. [10]

3. Make Me Amazing

   Android selalu mengedepankan pengguna. Pengguna merasa senang ketika mereka memahami sendiri sesuatu. Jadikan aplikasi Anda lebih mudah dipelajari dengan memanfaatkan pola visual dari aplikasi Android lainnya. Misalnya, gerakan menggeser dapat menjadi pintasan navigasi yang bagus. Dan Android selalu memberikan tanggapan atau umpan balik terhadap perintah yang kita berikan. [10]

2.3 JSON

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

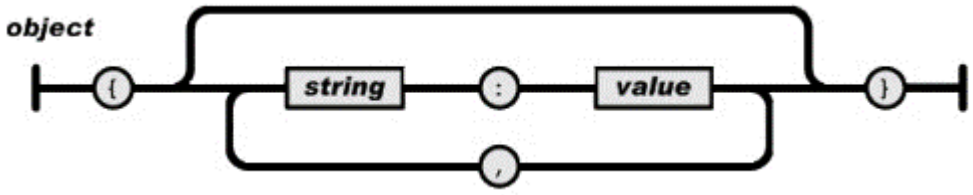
menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), daftar berkunci (*keyed list*), atau *associative* array. [11]
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*). [11]

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

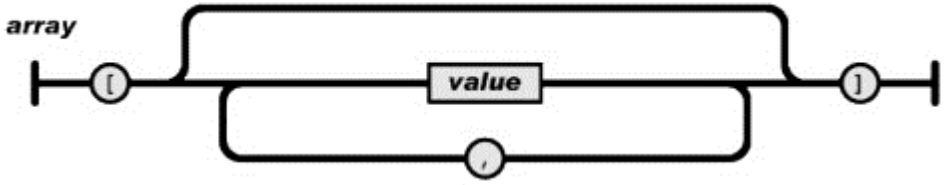
1. Objek

   Objek adalah sepasang nama / nilai yang tidak terurutkan. Seperti yang diilustrasikan pada Gambar 2.5 objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON. [11]



Gambar 2.3 Objek JSON [11]

2. Larik
   Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma) seperti yang diilustrasikan pada Gambar 2.6. Larik dalam JSON dapat digunakan sebagai value dari JSON object hal ini dapat berguna jika JSON menyimpan data bertingkat. [11]



Gambar 2.4 Array JSON [11]

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Bollean dan Nilai NULL. [11]

2.4 API

API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi. API atau *Application Programming Interface* juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak.

Dengan adanya API, maka memudahkan seorang programmer untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan *developer* menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi

lainnya untuk saling berinteraksi. [12]. Keuntungan dengan menggunakan API adalah sebagai berikut:

1. Portabilitas.

   Developer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut.

2. Lebih Mudah Dimengerti

   API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan.

System call interface ini berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan. Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah open().

Perintah open() ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan pengguna dapat dimengerti oleh sistem operasi, maka perintah open() tadi diterjemahkan ke dalam bentuk system call oleh system call interface. Implementasi perintah open() tadi bisa bermacam-macam tergantung dari sistem operasi yang digunakan. [13]

Cara menggunakan API :

1. Dilakukan dengan mengimpor package/kelas
2. Ada beberapa kelas bernama sama dipackage yang berbeda, yaitu :

- Import salah satu dan gunakan nama lengkap untuk yang lain
- Gunakan nama lengkap semua kelas

Kebanyakan Sistem Operasi seperti Windows, menyediakan fasilitas API sehingga programmer dapat melakukan aktivitas programming dengan lebih konsisten. Meskipun API didesain untuk programer, namun API juga baik untuk user karena setidaknya dapat menjamin bahwa program tersebut memiliki interface yang sama, sehingga lebih mudah untuk dipelajari.

2.4.1 API Facebook

API Facebook adalah platform untuk membangun aplikasi yang tersedia untuk anggota jejaring Facebook. API mengijinkan aplikasi yang dibuat terkoneksi dengan profil dan informasi untuk membuat aplikasi lebih mendalam. Dengan API, pengguna dapat menambahkan social konteks pada profil Facebook ke dalam aplikasi dan memanfaatkannya seperti profil , teman , halaman , kelompok , foto , data and event. [14]

Ada 5 fungsi utama pada API Facebook, yaitu :

1. Facebook Login - Mengautentikasi orang dengan kredensial Facebooknya.
2. Account Kit - Masukkan orang-orang hanya dengan nomor telepon atau alamat email mereka.
3. Dialog Bagikan dan Kirim - Aktifkan membagikan konten dari aplikasi Anda ke Facebook.
4. Peristiwa Aplikasi - Mencatat peristiwa di aplikasi Anda.
5. API Graf - Membaca dan menulis ke API Graf.

2.5 OOAD (Object Oriented Analysis Design)

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah

metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. [15]

2.5.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh model-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan data rancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorium terbuka. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah [15]:

1. Use Case Diagram

   Use Case Diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem secara keseluran yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Yang ditekankan pada diagram ini adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". *Use case* menjelaskan secara sederhana fungsi sistem dari sudut pandang user. Adapun komponen-komponen dalam *Use Case Diagram* anataranya :

   a) Actor

      Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu. Bisa merupakan manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem.

   b) Use Case

      Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini

terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

c) System

Menyatakan batasan sistem dalam relasi dengan actor-actor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Digambarkan dengan segi empat yang membatasi semua use case dalam sistem terhadap pihak mana sistem akan berinteraksi. Sistem disertai label yang menyebutkan nama dari sistem, tapi umumnya tidak digambarkan karena tidak terlalu memberi arti tambahan pada diagram.

d) Association

Mengidentifikasikan interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Digambarkan sebagai garis antara *actor* terhadap *use case* yang bersangkutan. Asosiasi bisa berarah (garis dengan anak panah) jika komunikasi satu arah, namun umumnya terjadi kedua arah (tanpa anak panah) karena selalu diperlukan demikian.

e) Depency

Dependensi <<include>>

1) Mengidentifikasi hubungan antar dua use case dimana yang satu memanggil yang lain.
2) Jika pada beberapa use case terdapat 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.

3) Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis.
4) Arah mata panah sesuai dengan arah pemanggilan

Dependensi <<extend>>

1) Jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>>.
2) Digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan.
3) Note: konsep "extend" ini berbeda dengan "extend" dalam Java.

f) Generalization

Mendefinisikan relasi antara dua *actor* atau dua *use case* yang mana salah satunya meng-*inherit* dan menambahkan atau *override* sifat dari yang lainnya. Penggambaran menggunakan garis bermata panah kosong dari yang meng-*inherit* mengarah ke yang di-*inherit.*

2. Activiy Diagram

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dimana biasanya dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Activity Diagram ini sendiri memiliki struktur diagram yang mirip Flowchart atau data flow diagram pada perancangan terstruktur. Activity Diagram dibuat berdasarkan sebuah atau beberapa use case pada Use Case Diagram.

3. Class Diagram

Class diagram merupakan diagram yang selalu ada di pemodelan system berorientasi objek. Class diagram menunjukkan hubungan antar class dalam system yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai satu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi property dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif,

dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private dan protected.

4. Sequence Diagram

   Sequence Diagram adalah suatu diagram yang digunakan untuk menggambarkan dan menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case. Adapun urutan proses yang dijelaskan yaitu interaksi yang terjadi antar class, operasi yang terlibat, urutan antar operasi, dan informs yang diperlukan oleh masing-masing operasi. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, message yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan progress vertical.

2.6 Sistem Rekomendasi

Sistem rekomendasi adalah suatu sistem yang dirancang untuk memprediksi suatu item yang sesuai dengan minat pengguna, yang mana *item* tersebut akan direkomendasikan pada pengguna [16]. Prediksi informasi minat pengguna dapat diperoleh berdasarkan pola aksi prilaku pengguna atau sering dikatakan sebagai profil pengguna. Pada sistem rekomendasi, terdapat beberapa *item* yang akan disaring untuk direkomendasikan kepada pengguna berdasarkan profil pengguna, skala rating, dan lain-lain sehingga menghasilkan beberapa *item* yang direkomendasikan kepada pengguna. Saat ini banyak sistem yang telah mengadopsi sistem rekomendasi pada penerapannya. Seperti halnya pada sosial media twitter, facebook, youtube.com, amazon.com dan lain-lain.

2.6.1 Metode Sistem Rekomendasi

Dalam sistem rekomendasi terdapat tiga bentuk metode yang digunakan dalam pembangkitan rekomendasi. Metode terebut adalah *Content-based, Collaboratif Filtering*, dan *Hybrid Recomender system*.

a) *Content Based Recommender System*

Metode *content based* membuat rekomendasi dengan menganalisa deskripsi setiap *item* untuk mengidentifikasi *item* mana yang mempunyai hubungan ketertarikan khusus dari seorang *user*. Deskripsi ketertarikan *user* diperoleh dari profil *user* yang didasarkan atas penilaian menarik atau tidaknya suatu *item* yang ada pada *recommender system* ini [17].

Sistem rekomendasi *content based* merupakan suatu sistem yang bertujuan memperkirakan informasi yang menarik bagi penggunanya dan membantu calon konsumen dalam memutuskan barang apa yang akan dibelinya. Perkiraan informasi ini bersifat personil yang didasarkan atas profil dari pengguna sistem. Profil pengguna umumnya didasarkan atas penilaian menarik-tidaknya suatu infomasi yang pernah dibaca oleh pengguna. Sistem rekomendasi sudah banyak diterapkan pada berbagai model bursa elektronis.

Secara umum, metode content-based filtering membentuk profil penggunanya berdasarkan atribut pembentuk suatu item. Sebagai contoh untuk suatu item dokumen, atribut pembentuknya adalah kata-kata/*term* yang terdapat pada dokumen tersebut. Parameter pembentuk profil pengguna ini juga diberi nilai bobot berdasarkan kriteria tertentu, misalnya untuk kasus barang adalah frekuensi suatu kategori/term dalam dokumen tersebut.

Content-based filtering memberikan suatu rekomendasi berdasarkan hasil analisa kemiripan item yang telah dinilai oleh para penggunanya. Metode ini pada awalnya diterapkan dalam sistem rekomendasi dokumen. Variasi metode ditawarkan berdasarkan suatu analisis isi dokumen dan pencarian keteraturan dalam berbagai dokumen. Pendekatan metode lain dilakukan menggunakan metode klasifikasi dengan tujuan memperkirakan golongan suatu item atau teks,

misalnya golongan disukai atau tidak disukai. Metode lain juga melakukan klasifikasi dengan melakukan pembobotan suatu item atau teks dalam suatu nilai numerik.

Secara umum, metode *content-based filtering* mempunyai 2 teknik umum dalam membuat rekomendasi yaitu *heuristic-based* dan *model-based*. *Cosine similarity*, *Boolean query*, teknik TF-IDF (*term frequency-invers document frequency*) dan *Clustering* termasuk dalam golongan *heuristic-based* sedangkan yang masuk dalam golongan *model-based* adalah teknik *Bayesian classifier & Clustering, Decision Tree* dan *Artificial Neural Network*. Penelitian ini akan menggunakan algoritma TF-IDF (*term frequency-invers document frequency*) sebagai pembentuk profil pengguna.

Algoritma metode *content-based filtering* dapat dijelaskan dalam langkah-langkah :

1. Suatu item barang dibagi-bagi berdasarkan suatu vektor komponen pembentuknya. Misalnya untuk sebuah film dibagi atas komponen aktor, sutradara, jenis film, dll.
2. Pengguna memberi penilaian suka atau tidak suka atas item tersebut.
3. Sistem akan membuat profil pengguna berdasarkan bobot vektor komponen pembentuk suatu item. Pembuatan profil pengguna dapat menggunakan algoritma TF-IDF (*term frequency-invers document frequency*). TF adalah jumlah term dalam suatu dokumen. Sedangkan nilai IDF dapat dihitung menggunakan rumus:

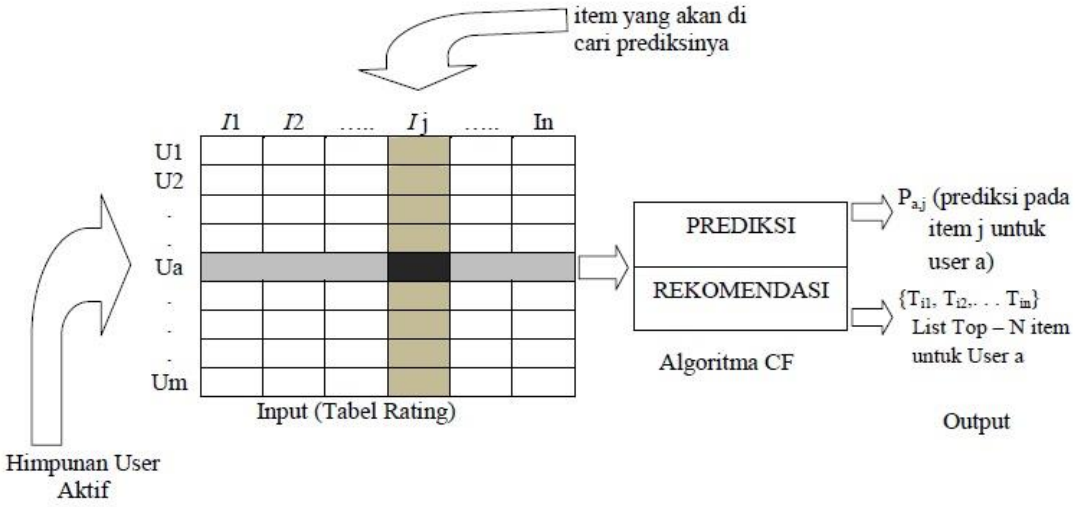
$$idf_i = \log\left(\frac{n}{df_i}\right)$$

   n merupakan jumlah semua dokumen sedangkan df adalah jumlah dokumen yang memiliki term *i*.
4. Berdasarkan profil pengguna tersebut, sistem akan memperkirakan penilaian suka atau tidak suka suatu item berdasarkan analisis kemiripan profil

pengguna dengan vektor komponen pembentuk item. Jika sistem memperkirakan bahwa item tersebut akan disukai oleh pengguna maka item tersebut akan direkomendasikan ke pengguna. Metode ini memiliki kekurangan yang utama yaitu ketidakmampuan merekomendasikan jenis item yang baru atau belum pernah dilihat kepada seorang pengguna. Hal ini disebabkan oleh karena metode ini dibuat berdasarkan item-item yang pernah dinilai oleh pengguna tersebut.

b) *Collaborative Filtering Recommender System*

Pendekatan *collaborative filtering* merekomendasikan *item* kepada *user* berdasarkan korelasi antara *rating user* yang berbeda terhadap *item* atau transaksi *user* di dalam sistem. Sementara *user merating item*, sistem akan menggunakan data *rating user* tertentu untuk melakukan perhitungan prediksi dan memberikan rekomendasi *item* yang belum pernah dirating oleh *user* tersebut [18]. *Collaborative Filtering* merupakan proses penyaringan atau pengevaluasian item menggunakan opini orang lain. Pada prosesnya metode *Collaborative filtering* melakukan penyaringan data berdasarkan tingkah laku karakteristik pengguna sehingga dapat memberikan informasi yang baru kepada pengguna lainnya karena sistem memberikan informasi berdasarkan pola satu kelompok pengguna yang hampir sama. Berikut merupakan skema dari *Colaboratif Filtering* dalam pemberian rekomendasi kepada user aktif.



Gambar 2.5 Skema Collaborative Filtering

Dalam skenario CF terdapat daftar pengguna m *user* U = {u1, u2,…, um} dan daftar *item* I = {p1, p2,…, pn}. Setiap ui *user* mengekspresikan pendapatnya tentang daftar *item* miliknya. Kumpulan *set* dari pendapat itu disebut dengan rating dari *user* ui dan dilambangkan dengan Iui. Setelah sistem ini menentukan ketetanggaan terdekat, maka sistem akan merepresentasikan *item* yang mungkin disukai *user* dalam dua bentuk. [19]

1. Prediksi, merupakan nilai numerik dimana Pa,j adalah nilai prediksi rating item j yang mungkin disukai oleh active user (Ua). Nilai prediksi ini digunakan dengan skala yang sama dengan nilai yang disediakan (misalnya, dari skala 1 sampai 5).
2. Rekomendasi adalah daftar N item yang mungkin akan disukai oleh user Ua. Daftar yang direkomendasikan biasanya terdiri dari item yang belum pernah dibeli atau dirating oleh active user. Output dari algoritma CF ini juga dikenal sebagai Top-N Recommendation.

Gambar 2.5 menunjukkan diagram skema dari proses collaborative filtering. Algoritma CF merepresentasikan seluruh m x n user-item sebagai matriks rating

dimana setiap entri merupakan nilai rating dari user untuk setiap item. Active user (Ua) pada skema ini merupakan user yang akan dicari item yang mungkin disukainya dengan menggunakan algoritma CF. [19]

Menurut teori dan penggunaannya Schafer membagi algoritma *collaborative filetering* ke dalam dua kelas yang berbeda yaitu dengan menggunakan algoritma probabilistik dan non probabilistik. Suatu algoritma dianggap probabilistik bila algoritma tersebut berdasarkan model probabilistik. Algoritma tersebut menggunakan opini orang lain. [20]

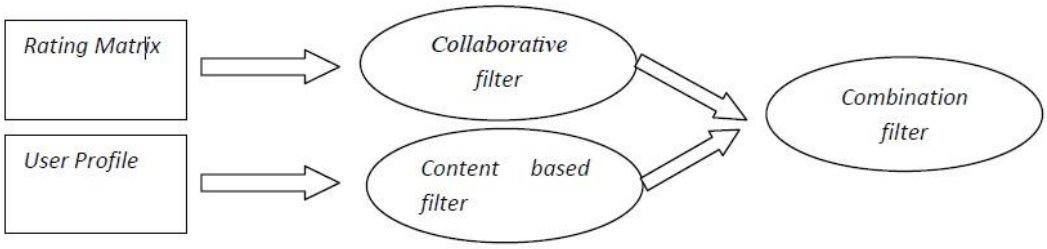
c) *Hybrid Recommender System*

Secara umum pendekatan *hybrid recommendations* adalah dengan menggabungkan lebih dari satu metode yang ada pada *recommender system*, kombinasi yang ada pada teknik ini misalnya dengan menggabungkan metode *content based* dengan *collaborative filtering* [21].

2.7 Hybrid Recommender System

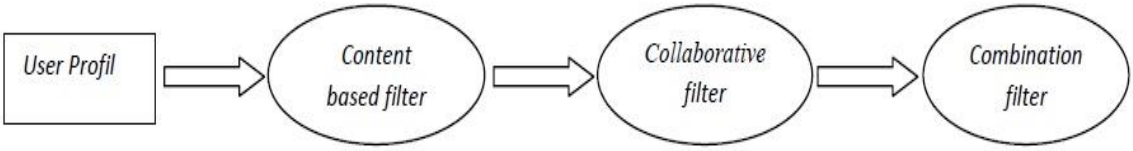
Hybrid Recommender System menggabungkan dua atau lebih teknik rekomendasi untuk mendapatkan hasil yang lebih akurat. *Hybrid Recommender System* menggabungkan metode content-based dan collaborative untuk membantu keterbatasan yang terdapat pada kedua metode tersebut. Terdapat beberapa cara penggabungan yang dapat dilakukan dalam metode hybrid yaitu :

1. Penggabungan secara Liner (*Linier Combination*) Penggabungan ini menggabungkan hasil rating dari metode *content-based* dan *collaborative*. Penggabungan ini dilakukan dengan cara pemberian ranking atau rating. Penggabungan ini digambarkan pada gambar berikut.



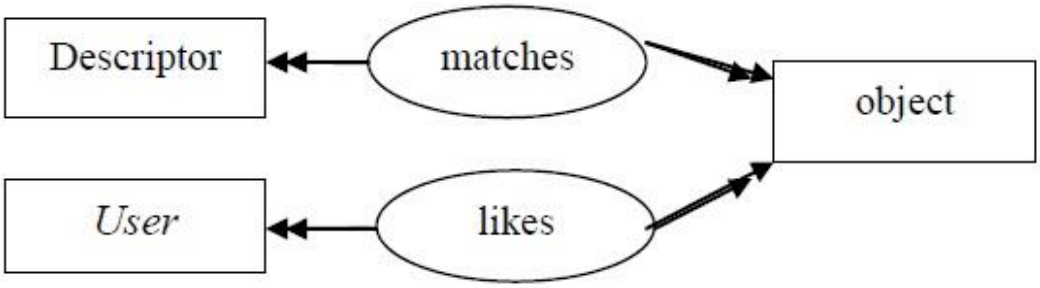
Gambar 2.6 Kombinasi Linier

2. Penggabungan secara Sekuensial (*Sequential Combination*) Penggabungan ini melakukan perhitungan pada salah satu metode (misalnya *content-based*) kemudian hasilnya digabungkan dengan perhitungan metode lainnya (*collaborative*). Penggabungan ini digambarkan pada gambar berikut :



Gambar 2.7 Kombinasi Sekuensial

Berikut ini merupakan gambar model hubungan entitas dari metode *hybrid*



Gambar 2.8 Model Hubungan Entitas Hybrid System

Hubungan entitas diatas menyediakan fungsionalitas dari sebuah sistem *content-based* dan juga fungsionalitas dari *collaborative system*.