BAB 2

LANDASAN TEORI

2.1. Tanaman Hias

Tanaman hias adalah tanaman yang dapat memenuhi kebutuhan psikologis, meningkatkan lingkungan, dan memiliki nilai kepuasan sebagai hobi. Tanaman dan keindahannya menimbulkan efek imajinasi yang luas dan menghidupkan kembali semangat diri. Adanya tanaman dalam lingkungan hidup dapat memberikan efek memanusiakan dan dapat membantu mengembalikan kondisi kesehatan pikiran yang dibebani dengan kesedihan, rasa sakit, dan masalah. Secara fungsional, tanaman hias dapat membantu memperbarui oksigen di udara, membantu mengurangi polutan gas dan kebisingan. Tanaman yang diatur dengan semestinya akan membantu sirkulasi udara secara langsung, menjadi tabir, melembutkan garis arsitektur yang keras, dan memberikan tekstur ke permukaan yang mencolok. [1]

Pada pembangunan aplikasi ini, jenis tanaman hias yang digunakan adalah jenis tanaman hias bunga, tanaman hias daun, tanaman hias hidroponik, dan tanaman hias gantung. Berikut merupakan penjelasan dari setiap jenisnya.

1. Tanaman hias bunga

   Tanaman hias bunga didefinisikan sebagai tumbuhan yang mempunyai bunga berpesona cantik dengan satu atau banyak warna. Tanaman hias bunga dapat menambah keindahan suatu ruangan atau lingkungan di sekitarnya serta membuatnya terlihat lebih asri. Ada banyak jenis tanaman hias bunga yang bisa ditanam di dalam ataupun luar (sekitaran) rumah seperti bunga mawar dan bunga melati.

2. Tanaman hias daun

   Tanaman hias daun merupakan salah satu jenis tanaman hias yang memiliki keindahan dan ciri khas pada bagian daunnya. Ada banyak jenis tanaman hias daun yang bisa ditanam di dalam ataupun luar (sekitaran) rumah seperti bunga aglonema dan bunga keladi *red start*.

3. Tanaman hias hidroponik

   Tanaman hidroponik adalah tanaman yang media tanamnya bukan berupa tanah. Teknik menanam yang satu ini biasanya menggunakan media tanam air. Ada banyak jenis tanaman hias hidroponik yang bisa ditanam di dalam ataupun luar (sekitaran) rumah seperti bunga anggrek.

4. Tanaman hias gantung

   Tanaman hias gantung merupakan jenis tanaman hias yang media tanamnya dapat digantungkan, dengan menggunakan teknik gantung ini, tanaman hias akan terlihat lebih indah dan menarik. Salah satu contoh tanaman hias gantung yaitu seperti tanaman begonia.

2.2. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer,instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. Menurut Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan set instruksi yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini.

Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh aplikasi ialah program pemproses kata dan *Web Browser*. Aplikasi akan menggunakan system operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app.

Secara historis, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. App adalah *sofware* yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang *parallel* dengan yang terjadi antar sistem operasi yang dimunculkan. Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain: [8]

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

Pada penelitian ini, pembangunan aplikasi bertujuan untuk mengkomputerisasikan segala proses yang terlibat dalam aplikasi rekomendasi pencarian tanaman hias.

2.3. Andorid

Menurut Nazaruddin Safaat H dalam bukunya menjelaskan, Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi *Linux*. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak.

Awalnya, *Google Inc*. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Dilain pihak, Google merilis kode–kode Android di bawah lisensi *Apache*, sebuah lisensi

perangkat lunak dan standar terbuka perangkat seluler [9]. Pada penelitian ini pembangunan aplikasi menggunakan perangkat android, karena sistem aplikassi android merupakan *platform* terbuka (*open source*) sehingga banyak di gunakan oleh masyarakat.

2.3.1 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut: [9]

1. *Application and Widgets*

   Application dan widgets ini adalah layer di mana pengguna berhubungan dengan aplikasi saja, di mana biasanya pengguna men-download aplikasi kemudian kita lakukan instalasi dan menjalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk client email, program SMS, kalender, peta, *browse*, kontak dan lain-lain. Semua aplikasi dibuat menggunakan bahasa Java.

2. *Applications Frameworks*

   Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk mengembangkan aplikasi yang bagus dan inovatif. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

3. *Libraries*

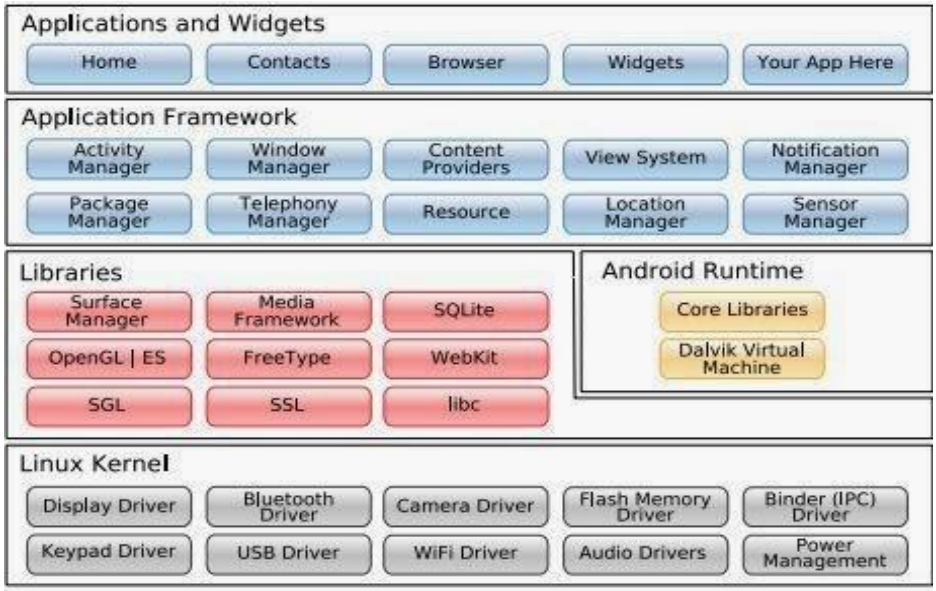
   Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL. Kemampuan ini dapat diakses oleh programmer melewati Android *application framework*. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar.

4. *Android Run Time*

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

5. *Linux Kernel*

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. *Linux kernel* yang digunakan Android adalah *linux kernel release* 2.6. Kernel berperan sebagai *abstraction* layer antara hardware dan keseluruhan software. Namun secara keseluruhan Android bukanlah *linux*, karena dalam Android tidak terdapat paket standar yang dimiliki oleh *linux* lainnya. Pembangunan aplikasi bekerja pada *layer* ke dua yaitu *application framework*, dimana layer ini berfungsi untuk membantu pengembang membangun aplikasi.



Sumber Gambar : http://www.android-indonesia.com/forum/developers/12156-arsitektur-android

Gambar 2.1 Arsitektur Android

2.3.2 Teknologi Android

Android memilik banyak tenologi yang dapat membantu berdasarkan manfaat teknologi itu sendiri, berikut beberapa teknologi android yang penulis gunakan untuk pembangunan aplikasi rekomendasi pencarian tanaman hias.

2.3.2.1 *Global Positioning System* (GPS)

Global Positioning System (GPS) adalah suatu sistem navigasi atau penentu posisi berbasis satelit. Sistem ini didesain untuk memberikan posisi dan informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa tergantung waktu dan cuaca. Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (*longitude* atau *latitude*) [10].

A. Titik Koordinat

Titik koordinat adalah titik yang berpedoman pada garis *latitude* dan *longitude* suatu daerah. Kaitannya dengan *latitude* dan *longitude* adalah, kedua garis lintang dan bujur inilah (*latitude* = garis lintang, *longitude* = garis bujur) yang menentukan di perolehnya suatu nilai derajat dari suatu titik yang diukur. Koordinat pada umumnya dituliskan dengan format DD (*Degree Decimal*), terkadang juga dituliskan dengan format DMS (*Degree-Minute-Second*).

B. Konversi Derajat Bujur dan Lintang

Perlu diketahui pula konversi derajat menit (‘) dan detik (“) pada letak astronomis suatu tempat dalam satuan jarak (km dan m). Dalam pemakaian GPS sering didapatkan titik koordinat suatu posisi lokasi dalam bentuk *Latitude* (Lintang), *Longitude* (Bujur) atau *Degree* (Derajat). Dalam hal ini ada suatu formula yang digunakan untuk mengkonversi bilangan tersebut dari bentuk *Degree Minute Second* (DMS) menjadi *Degree Decimal* (DD), atau sebaliknya konversi *Degree Decimal* (DD) menjadi *Degree Minute Second* (DMS).

a. Contoh perhitungan Konversi *Degree Decimal* (DD) menjadi *Degree Minute Second* (DMS)

Bila diperoleh angka 109.036889 maka diubah menjadi 109˚. Angka dibelakang koma dikalikan dengan bilangan desimal 60, maka 0,036389 x 60 = 2,18334, urutan kedua yaitu 2 menit.

Sisa angka dibelakang koma diatas (18334) dikalikan dengan bilangan desimal 60, maka 0,18334 x 60 = 11,0004, urutan ketiga yaitu 11 detik. Hasil dari konversi, secara lengkap ditulis 109˚ 2' 11", hal ini juga berlaku pada *Latitude*.

b. Contoh perhitungan Konversi *Degree Minute Second* (DMS) menjadi *Degree Decimal* (DD)

Bila diperoleh angka 7˚13' 13", maka dapat diubah menjadi,

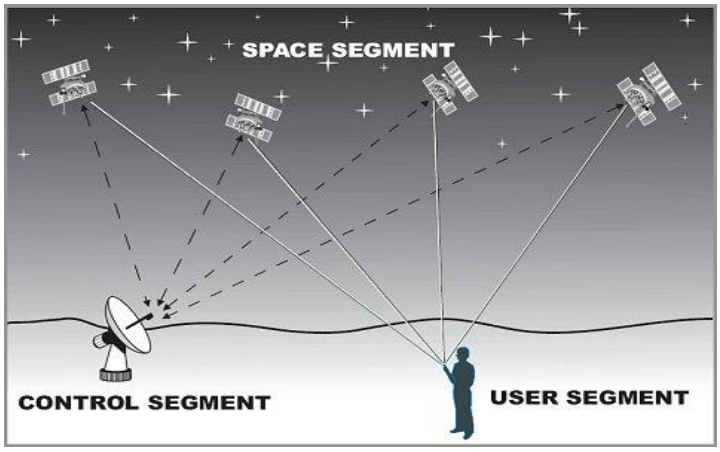
= (Nilai Degree) + (Nilai Minute/60) + (Nilai Second/3600)

= 7 + (13/60) + (13/3600) = 7.220278

Maka hasil konversi DMS ke DD adalah 7.220278.

Secara umum, GPS terdiri dari tiga bagian yaitu satelit, stasiun bumi (*ground station*), dan penerima (*receiver*). Adapun cara kerja GPS dapat dilihat dari gambar 3.2 sebagai berikut.

1. Bagian terpenting dari navigasi GPS adalah beberapa satelit yang berada di orbit bumi, satelit ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke stasiun bumi.
2. Stasiun bumi menangkap sinyal tersebut dengan radarnya kemudian mengrimkan sinyal tersebut dan ditangkap oleh alat penerima sinyal tersebut atau GPS Tracker.
3. Penerima sinyal dari satelit tersebut, akan memperhitungkan seberapa jauh mereka dari beberapa satelit. Setelah penerima menghitung jarak dari empat atau lebih satelit, maka akan diketahui persis di mana seseorang yang menggunakan GPS itu berada.



Sumber gambar : http://pengetahuan-asikmenarik.blogspot.co.id/2011/01/cara-kerja-gps.html

Gambar 3.2 Cara Kerja GPS

Penggunaan gps pada aplikasi ini digunakan untuk beberapa fungsi yaitu untuk *geotagging* dan untuk LBS. Gps pada *geotagging* digunakan untuk mendeteksi lokasi pengambilan data yang akan disimpat melalui format penyimpanan berupa EXIF foto. Selain itu, gps digunakan juga untuk teknolog LBS, dimana teknologi pada aplikasi ini untuk mencari lokasi keberadaan tanaman terdekat.

2.3.2.1 Geotagging

Geotagging adalah proses penambahan identifikasi ke berbagai media seperti foto, video, *website*, pesan SMS, atau RSS feed berupa metadata g*eospasial* . Data ini biasanya terdiri dari koordinat lintang dan bujur (*latitude* dan *longitude*). Data lainnya juga dapat mencakup ketinggian, jarak, akurasi data, dan nama tempat. Hal ini biasanya digunakan untuk memberikan *geotag* pada foto [10].



Sumber Gambar : http://kiosbukugema.blogspot.co.id/2011/03/gps-photo-tagging.html

Gambar 2.2 Contoh *Geotagging*

Dasar untuk *geotagging* adalah posisi. Posisi ini akan,dalam hampir setiap kasus, berasal dari global *positioning system* (GPS). GPS secara formal diketahui sebagai *Navigation Satellite Timing and Ranging* (NAVSTAR) GPS yang sebenarnya dikembangkan untuk keperluan militer. Karena kemampuan navigasi populernya dan karena teknologi GPS bisa diakses dengan perlatankecil yang tidak terlalu mahal maka pemerintah membuat sistem ini bisa dinikmati oleh rakyat sipil. USA memiliki teknologi GPS dan Departemen Pertahanan yang memeliharanya. [10] Pada pembangunan aplikasi ini, *geotagging* digunakan untuk mendapatkan koordinat lokasi (*latitude* dan *longitude*) tanaman hias yang akan dipromosikan.

2.3.2.2 *Location Based Service* (LBS)

Location Base Services (LBS) suatu layanan yang memberikan informasi berdasarkan posisi pengguna (*device*) sebagai acuan utama. Saat ini LBS digunakan dalam berbagai konteks seperti kesehatan, pencarian objek, hiburan, pekerjaan dan lain-lain. Dua unsur utama dalam LBS yaitu: [9]

1. *Location Manager (API Maps)*

   Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Aplication Programming Interface* (API) menyediakan fasilitas untuk menampilkan atau memanipulasi peta.

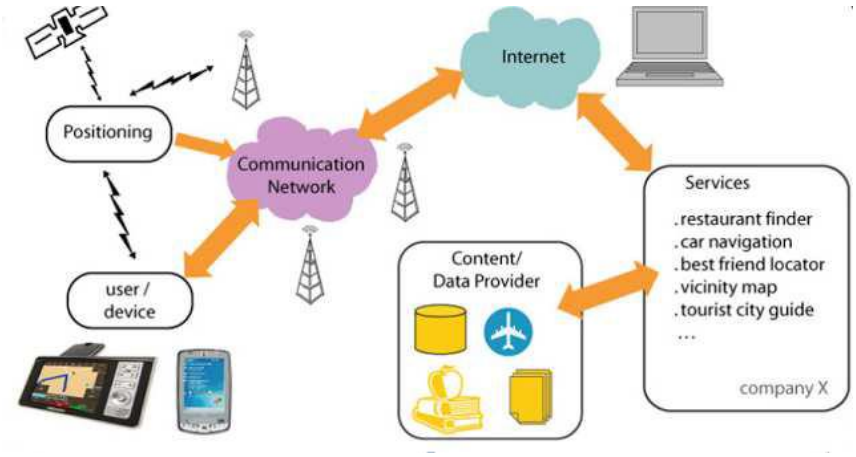
2. *Location Prividers (API Location)*

Menyediakan teknologi pencarian lokasi yang di gunakan oleh perangkat. API Location berhubungan dengan data GPS dan data lokasi *real time.*

Data lokasi pengguna biasanya di dapatkan melalui jaringan telepon seluler ataupun menggunakan GPS. LBS memiliki komponen-komponen yang menunjang dalam prosesnya yaitu :

1. Perangkat *Mobile*, Pengguna membutuhkan perangkat *mobile* untuk menggunakan layanan LBS ini. Sepertihalnya *smartphone*, tablet dan lain-lain.
2. Jaringan Komunikasi, Jaringan kominkasi digunakan untuk menghubungkan perangkat *mobile* dengan perangkat lainya.
3. Komponen pengambil posisi *latitude* dan *longitude* (satelit), satelit merupakan alat yang menentukan posisi pengguna. Seperti jarak, lokasi dan lain-lain.
4. Data dan *provider content*, data yang di dapatkan akan di proses di *server* dan dikirim kembali ke pengguna berupa data yang telah akurat.
5. *Web Map Server* (WMS), merupakan *server* dimana tempat pengumpulan dan pemrosesan data.

Berikut ini adalah cara kerja dari LBS :



Sumber Gambar : http://eprints.umm.ac.id/34206/1/jiptummpp-gdl-anggaefend-45496-1-pendahul-x.pdf

Gambar 2.3 Cara Kerja Location Based *Service*

1. Pertama *Smartphone* membuka aplikasi LBS yang sudah terinstall / jika menggunakan aplikasi yang berbasis *browser*, maka buka *browser* dan ketik alamat tujuan situsnya.
2. Aplikasi LBS akan melakukan sambungan dengan jaringan provider (seperti telkomsel, xl, axis, dan lain-lain) yang dipakai oleh si *User* (pengguna).
3. Jaringan mengirimkan *request* ke satelit untuk menentukan *longitude* (garis bujur) dan latitude (garis lintang) dari si pengguna aplikasi tersebut.
4. Provider menghubungkan aplikasi (di *smartphone*) dengan *server* LBS dan meminta data yang diinginkan *User*.
5. User mendapatkan data dan ditampilkan di *Smartphone*.

Pada dasarnya, LBS terbagi menjadi 2 yaitu :

1. *Pull Service*

Pengguna secara aktif mengirimkan informasi yang dibutuhkan. Sama seperti mengakses sebuah halaman web di browser, kita memasukan halaman web yang di tuju dan kita memperolah informasi dari halaman web yang tampil di browser. *Pull service* terbagi menjadi dua bagian yaitu berdasarkan

fungsional seperti memesan taksi dengan menekan tombol pada *device* atau layanan *service* mencari lokasi restoran yang terdekan dengan kita.

2. *Push Service*

Memberikan informasi kepada pengguna yang mana tidak secara langsung diminta oleh pengguna. Aplikasi *Push Service* cocok digunakan untuk memata - matai seseorang melalui *SmartPhone* yang digunakannnya, misalnya orang tua yang ingin tau anaknya sedang berada dimana, maka dengan adanya aplikasi *Push Service* ini orang tua bisa mendapatkan Data lokasi anaknya yang dikirim ke *smartphone* orang tua yang berasal dari *smartphone* anaknya secara *Push Service*. sehingga si anak tidak mengetahui kalau orang tua nya sedang memata-matai pergerakan dari si Anak dan menghindari hal-hal yang tidak dinginkan seperti penculikan.

Penggunaan teknologi LBS pada pembangunan aplikasi ini bertujuan untuk mendapatkan lokasi keberadaan tanaman hias yang paling dekat dengan pengguna.

2.3.2.3 Sensor

Sensor adalah alat yang dapat digunakan untuk mengukur, menganalisa, memantau suatu kondisi dan kemudian merespon terhadap perubahan di sekitarnya. Alat ini dapat ditemukan pada perangkat modern seperti *smartphone* dengan sistem operasi android, sebagai ponsel pintar yang memiliki sepaket teknologi canggih [11] [12]**.**

Beberapa macam kategori tipe sensor yang umum:

1. Sensor Gerak atau *Motion Sensors*.

   Motion sensors adalah alat yang berfungsi untuk mengukur kekuatan akselerasi dan rotasi sepanjang tiga sumbu (X,Y,Z). Kategori ini meliputi kecepatan atau akselerasi, sensor gravitasi, *gyroscope*, dan rotasi vektor.

2. Sensor Lingkungan atau *Environmental Sensors*.

   Environmental sensors adalah alat yang berfungsi untuk mengukur berbagai parameter keadaan sekitar, seperti suhu udara dan tekanan,

pencahayaan, dan kelembaban. Kategori ini termasuk *barometers*, *photometers*, dan *thermometers*.

3. Sensor Posisi atau *Position sensors*.

Position sensors adalah alat yang berfungsi untuk mengukur posisi fisik perangkat. Kategori ini meliputi sensor orientasi dan *magnetometer*.

Dalam pembangunan aplikasi rekomendasi ini, penulis menggunakan tiga jenis sensor yang ada pada kategori sensor lingkungan atau *Environmental Sensors*, diantaranya [11] [12].

1. *Ambient Light Sensor*

Ambient Light Sensor adalah alat yang digunakan untuk mengatur kecerahan layar secara otomatis berdasarkan cahaya sekitar. Selain itu, sensor ini juga berfungsi sebagai pengukur atau penghitung intensitas cahaya yang ada pada era kita, sehingga sensor akan mendeteksi dan menampilkan intensitas cahaya yang masuk.

Intensitas cahaya matahari adalah kualitas cahaya matahari untuk membantu tanaman untuk tumbuh dan berfotosintesis. Cahaya matahari adalah sumber energi utama bagi kehidupan seluruh makhluk hidup didunia. Bagi manusia , hewan dan tumbuhan cahaya matahari adalah penerang dunia ini. Tanaman memerlukan cahaya matahari tumbuh hijau. Dengan air tanpa cahaya matahari, tanaman akan tumbuh tinggi dengan cepat, namun akan terlihat kuning dan kekurangan air, meskipun saat disentuh, daunnya teraba amat basah. Selain itu, bagi tumbuhan khususnya yang berklorofil cahaya matahari sangat menentukan proses fotosintesis.

Intensitas cahaya matahari menunjukkan pengaruh primer pada fotosintesis, dan pengaruh sekundernya pada morfogenetik. Pengaruh terhadap morofogenetik hanya terjadi pada intensitas rendah, pengaruh tanaman dalam kaitannya dengan intensitas cahaya salah satunya adalah penempatan daun dalam posisi di mana akan diterima intersepsi cahaya maksimum. Daun yang menerima intensitas maksimal adalah daun yang

berada pada tajuk utama yang terkena sinar matahari. Masing-masing tanaman memiliki reaksi yang berbeda terhadap intensitas cahaya. Berdasarkan perbedaan reaksi tersebut, tanaman dibedakan menjadi tanaman C3, C4, CAM. Tanaman C3 adalah tanaman yang hidup baik pada intensitas cahaya rendah, dan tanaman C4 adalah tanaman yang hidup baik pada intensitas cahaya tinggi, sedangkan tanaman CAM adalah tanaman yang hidup didaerah kering. Pada pembangunan aplikasi rekomendasi tanaman hias ini, sensor tersebut digunakan untuk mengukur seberapa terang intensitas cahaya yang ada dilingkungan tersebut agar bisa merekomendasikan tanaman yang sesuai dengan lingkungan tersebut dengan satuan yang digunakan adalah lx.

2. *Temperature Sensor*

Temperature Sensor adalah alat yang digunakan untuk mengukur suhu yang ada pada lingkungan sekitarnya. Suhu udara adalah suatu keadaan panas atau dinginnya udara disuatu tempat pada waktu tertentu,yang dipengaruhi oleh banyaknya atau sedikitnya panas matahari yang diterima bumi. Faktor iklim sangat menentukan pertumbuhan dan produksi tanaman, apabila tanaman ditanam di luar daerah iklimnya, maka produktivitasnya sering kali tidak sesuai dengan yang diharapkan. Menurut Sutarno at all, studi tentang perilaku kejadian tiap organisme atau tumbuhan dalam hubungannya dengan perubahan-perubahan iklim disebut dengan fenologi.

Untuk faktor iklim yang dipergunakan untuk penelitian fenologi pada umumnya adalah curah hujan hal ini adalah karena curah hujan secara langsung atau tidak langsung penting untuk pengaturan waktu dan ruang dalam pembentukan bunga dan buah pada tumbuhan tropis. Sehingga pada pembangunan aplikasi rekomendasi ini, pemanfaatan teknologi sensor bisa digunakan untuk mengukur suhu disekitarnya agar bisa merekomendasikan tanaman yang sesuai dengan lingkungan tersebut dengan satuan yang digunakan adalah derajat *celcius*.

3. *Humidity Sensor*

Humidity Sensor adalah alat yang digunakan untuk mengukur kelembaban yang ada pada lingkungan sekitarnya, sensor ini cocok untuk mengukur tingkat kelembaban sekitar sehingga kecocokan suatu tanaman dapat. Kelembaban udara adalah jumlah kandungan uap air yang ada dalam udara. Kandungan uap air di udara berubah-ubah bergantung apda suhu Makin tinggi suhu, makin banyak kandungan uap airnya. Kelembapan udara ada 2 jenis sebagai berikut:

a. Kelembapan mutlak (absolut) yaitu bilangan yang menunjukkan jumlah uap air dalam satuan gram pada satu meter kubik udara.
b. Kelembapan relatif (nisbi), yaitu angka dalam persen yang menunjukkan perbandingan antara banyaknya uap air yang benar-benar dikandung udara pada suhu tertentu dan jumlah uap air maksimum yang dapat dikandung.

Kelembaban yang dibutuhkan tanaman berbeda-beda tergantung pada jenisnya. jika ingin mendapatkan produktifitas yang optimal, tanaman ada yang membutuhkan kelembaban yang tinggi dan ada juga yang membutuhkan kelembaban yang rendah. Kelembaban udara berpengaruh terhadap penguapan pada permukaan tanah dan penguapan pada daun. Contohnya pohon, bila kelembaban tinggi maka pertumbuhan pohon akan terganggu karena tidak seimbangnya antara unsur air dan cahaya. Tetapi kelembaban udara yang tinggi sangat mempengaruhi pertumbuhan organ vegetatif pada pohon.

Kelembaban udara akan berpengaruh terhadap laju penguapan atau transpirasi. Jika kelembaban rendah, maka laju transpirasi meningkat dan penyerapan air dan zat-zat mineral juga meningkat. Hal itu akan meningkatkan ketersediaan nutrisi untuk pertumbuhan tanaman. Dan sebaliknya, jika kelembaban tinggi, maka laju transpirasi rendah dan penyerapan zat-zat nutrisi juga rendah . Hal ini akan mengurangi ketersediaan nutrisi untuk pertumbuhan tanaman sehingga

pertumbuhannya juga akan terhambat. Selain itu, kelembaban yang tinggi akan menyebabkan tumbuhnya jamur yang dapat merusak atau membusukkan akar tanaman. Dan apabila kelembabannya rendah akan menyebabkan timbulnya hama yang dapat merusak tanaman. Pada pembangunan aplikasi ini, sensor berfungsi untuk mendeteksi seberapa tinggi atau rendahnya tingkat kelembaban suatu tempat, sehingga aplikasi akan merekomendasikan tanaman yang cocok dengan lingkungannya dengan satuan yang digunakan adalah persen (%).

2.4 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemanduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak. Diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (Object Manajement Group). Tujuan utama dari perancangan UML adalah:

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorogn pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

Kegunaan diagram pada pemodelan adalah untuk formalisasi ekspresi model objek secara koheren. UML menyediakan sejumlah diagram untuk mengekspresikan pemodelan berorientasi objek yang dilakukan. yaitu, diagram Kelas, Objek, Komponen, Deployment, *Use case*, Sekuen,

Kolaborasi, *Statechart*, dan diagram Aktivitas [13]. Pada penelitian ini, diagram yang digunakan adalah diagram *Use case*, Kelas, Sekuen dan Aktivitas, dimana pada pembangunan aplikasi ini UML digunakan untuk membantu pembagian lingkup sistem sehingga lebih mudah diatur.

1. Diagram *Use Case*

Diagram use case merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. *Use case* adalah interaksi antara aktor eksternal dan sistem digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Diagram use case menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Elemen-elemen yang ada pada diagram *use case* adalah aktor, use case, dan hubungan ketergantungan, generalisasi dan asosiasi. [13]

a. Aktor

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu siapa atau apa yang menggunakan sistem. Aktor berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan. Use case selalu diawali oleh aktor yang mengirim pesan [13].

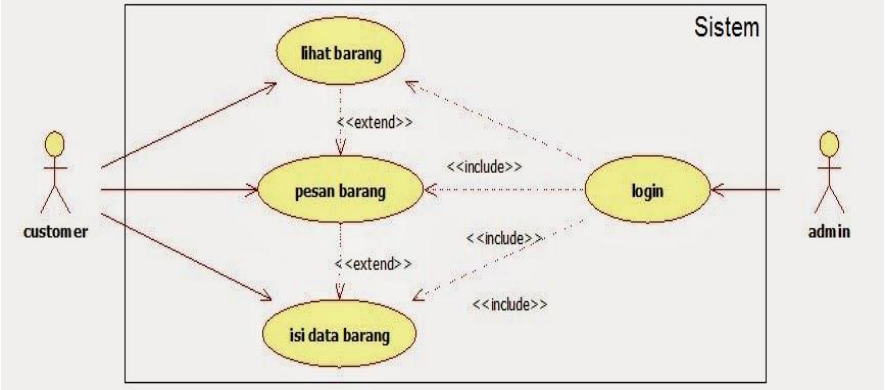
b. Use Case

Use case adalah cara spesifikasi penggunaan sistem oleh aktor. *Use case* menspesifikasikan perilaku sistem atau bagian sistem dan merupakan deskripsi sekumpulan sekuen aksi termasuk varian-varian yang dilakukan sistem untuk memproduksi hasil atau nilai ke aktor [13].

c. Hubungan Antar Use Case

Keterhubungan antar *use case* dengan *use case* lain berupa generalisasi antara *use case*, yaitu *include* dan *extend*. *Include* adalah perilaku *use case* merupakan bagian dari *use case* yang lain. Sedangkan *extend* adalah perilaku *use case* merupakan perilaku *use case* yang lain [13].

Berikut contoh dari *use case* :



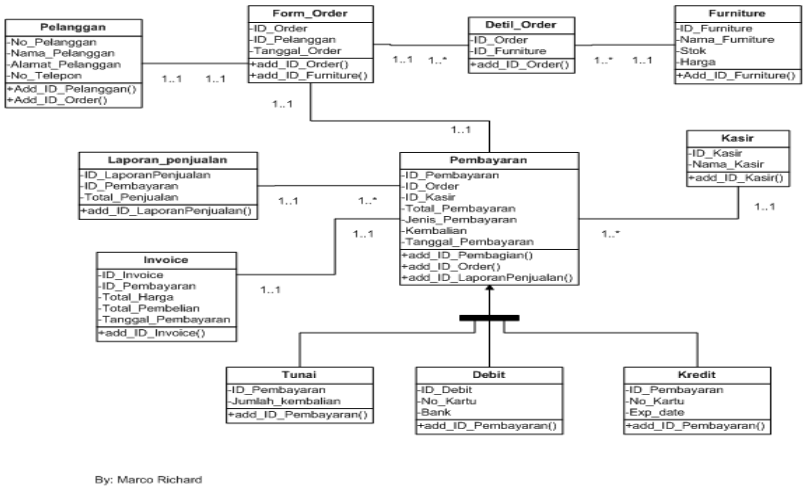
Sumber Gambar : http://wawasan-ilmiah.blogspot.co.id/2014/12/perbedaan-include-dan-extend-di-use-case.html

Gambar 3.4 Contoh Use Case Diagram

2. Diagram Kelas

Diagram kelas merupakan diagram paling umum dipakai di semua pemodelan berorientasi objek. Diagramkelas menunjukan aspek statik sistem terutama untuk mendukung kebutuhan fungsional sistem. Kebutuhan fungsional berarti layanan-layanan yang harus disediakan sistem ke pemakai. Kelas merupakan elemen terpenting di sistem berorientasi objek [13].

Dalam diagram kelas, kelas memiliki nama kelas. nama kelas harus unik karena nama akan menjadi identifier di program, memiliki atribut yang merupakan properti bernama dikelas yang mendeskripsikan range nilai yang dipunyai instan kelas. Kelas dapat mempunyai sejumlah atribtu atau tidak sama sekali, memiliki ketampakan (*visibility*) yang digunakan untuk menspesifikasikan ketampakan atributdan operasi dari kelas-kelas lain yaitu public, protected dan private.

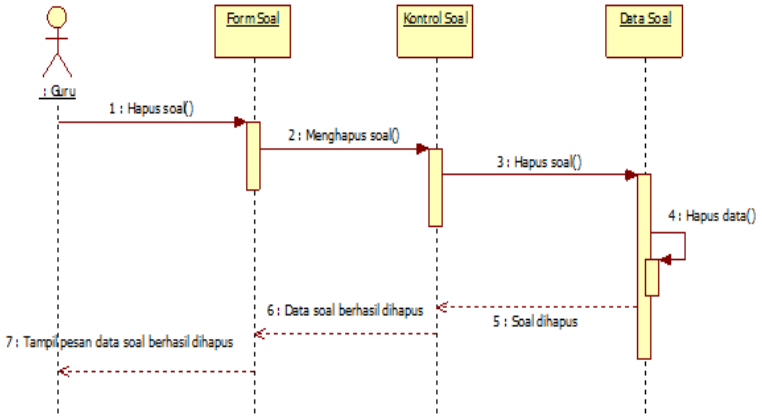


Sumber Gambar : http://marcomarentek.blogspot.co.id/2013/04/business-driven-technology-class.html

Gambar 2.5 Contoh Diagram Kelas

3. Diagram Sekuen

Diagram sekuen digunakan untuk memodelkan skenario penggunaan. Penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Diagram sekuean menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagi panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan [13].



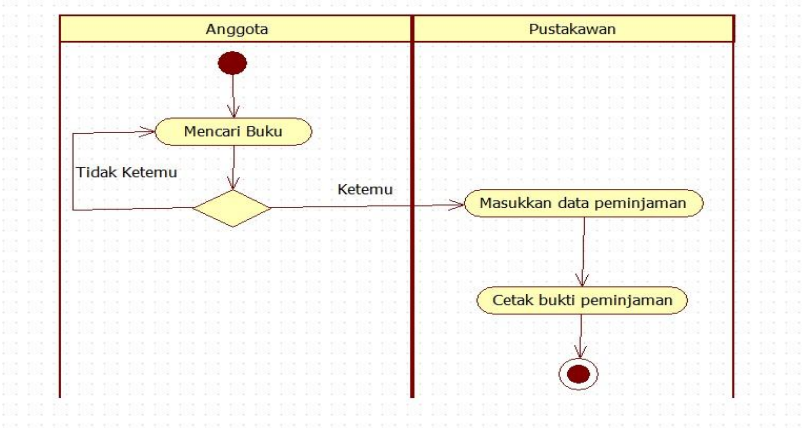
Sumber Gambar : http://www.tutorialkampus.com/2014/05/tutorial-diagram-sequence.html

Gambar 2.6 Contoh Diagram Sekuen

4. Diagram Aktivitas

Diagram aktivitas pada umumnya adalah diagram flowchart yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain.

Diagram aktivitas berfokus pada aktivitas-aktivitas, potongan-potongan dari proses yang boleh jadi berkorespondensi dengan metode – metode atau fungsi-fungsi anggota dan pengurutan dari aktivitas-aktivitas ini. Elemen-elemen yang ada pada diagram aktivitas adalah *aktivity state* dan *action state,* transisi, dan objek [13].



Sumber Gambar : http://tugas-kuliah-stmik.blogspot.co.id/2013/04/activity-diagram-uml.html

Gambar 2.7 Contoh Diagram Kelas

2.5 *Database*

Database atau Basis data terdiri atas dua kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar,bunyi, atau kombinasinya.

Basis Data merupakan kumpulan data yang terorganisasi untuk melayani berbagai aplikasi pada saat bersamaan dengan melakukan penyimpanan dan pengelolaan data sehingga data tersebut terlihat di satu lokasi. Prinsip utama basis

data adalah pengaturan data atau arsip dengan tujuan utama untuk kemudahan dan kecepatan dalam pengambilan kembali data atau arsip. Dapat disimpulkan bahwa Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan disimpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya.

Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi [14]. Pada pembangunan aplikasi ini, *database* digunakan untuk menampung dan mengolah data yang dibutuhkan oleh aplikasi ini.

2.5.1 *Database* Management System (DBMS)

Database Management System (DBMS) adalah sebuah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, membuat, menjaga, dan mengontrol akses ke basis data [15].

Biasanya, sebuah DBMS menyediakan fasilitas – fasilitas sebagai berikut :

1. Fasilitas yang mengizinkan pengguna untuk mendefinisikan basis data biasanya dengan menggunakan *Data Definition Language* (DDL). DDL mengizinkan pengguna untuk menentukan tipe data dan strukturnya, serta batasan aturan mengenai data yang akan disimpan ke dalam basis data.
2. Fasilitas yang mengizinkan pengguna untuk *Insert* (memasukkan), *Update* (memperbaharui), *Delete* (menghapus), dan *Retrieve* (memperoleh kembali) data dari basis data, biasanya dengan menggunakan *Data Manipulation Language* (DML). Ada juga fasilitas yang melayani pengaksesan data yang disebut *query language*. Fasilitas yang menyediakan akses kontrol ke dalam basis data yang meliputi:

a. *Security system*, mencegah pengguna yang tidak memiliki hak akses untuk memasuki basis data.
b. *Integrity system*, menjaga konsistensi data yang tersimpan.
c. *Concurrency control system*, mengizinkan akses ke basis data secara bersama.
d. *Recovery control system*, memperbaiki dan mengembalikan basis data ke dalam kondisi semula apabila terjadi kerusakan pada *hardware* atau *software.*
e. *User-accessible catalog*, berisi deskripsi data pada basis data.

Contoh – contoh perintah dasar *Data Manipulation Language* (DML) :

a. *Insert*

```
INSERT INTO Jurusan (KodeJur,
NamaJur)
VALUES ('12','Teknik Infromatika');
```

b. *Update*

```
UPDATE Mahasiswa
    SET Nama='Melati A Pertiwi',
    Alamat='Jl. Sapan 15'
WHERE Nim='10113356';
```

c. *Delete*

```
DELETE FROM Mahasiswa
WHERE Nim='10113356';
```

2.5.2 Model Basis Data

Abdul Kadir (2003) mengemukakan bahwa, model basis data relasional memiliki beberapa definisi penting sebagai berikut [16]:

1. Kumpulan objek atau relasi untuk menyimpan data
2. Kumpulan dari operator yang melakukan suatu aksi terhadap suatu relasi untuk menghasilkan relasi-relasi lain
3. Basis data relasional harus mendukung integritas data sehingga data tersebut harus akurat dan konsisten.

Basis data relasional memiliki fungsi-fungsi kegunaan sebagai berikut:

1. Mengatur penyimpanan data
2. Mengontrol akses terhadap data
3. Mendukung proses menampilkan dan memanipulasi data.

Beberapa istilah mengenai basis data relasional antara lain:

1. Tabel : Merupakan struktur penyimpanan dasar dari basis data relasional, terdiri dari satu atau lebih kolom (*column*) dan nol atau lebih baris (*row*).
2. *Row* (baris) : Baris merupakan kombinasi dari nilai-nilai kolom dalam tabel; sebagai contoh, informasi tentang suatu departemen pada tabel Departmen. Baris seringkali disebut dengan "*record*".
3. *Column* (kolom) : Kolom menggambarkan jenis data pada tabel; sebagai contoh, nama departemen dalam tabel Departmen. Kolom di definisikan dengan nama kolom dan tipe data beserta panjang data tertentu.
4. *Field* : *Field* merupakan pertemuan antara baris dan kolom. Sebuah field dapat berisi data. Jika pada suatu *field* tidak terdapat data, maka field tersebut dikatakan memiliki nilai "*null*".
5. *Primary key* : *Primary key* atau kunci utama merupakan kolom atau *kumpulan kolom* yang secara unik membedakan antara baris yang satu dengan lainnya; sebagai contoh adalah kode departemen. Kolom dengan kategori ini tidak boleh mengandung nilai "null", dan nilainya harus *unique* (berbeda antara baris satu dengan lainnya).
6. *Foreign key* : *Foreign key* atau kunci tamu merupakan kolom atau kumpulan kolom yang mengacu ke *primary key* pada tabel yang sama atau tabel lain. *Foreign key* ini dibuat untuk memaksakan aturan-aturan relasi pada basis data. Nilai data dari *foreign key*

harus sesuai dengan nilai data pada kolom dari tabel yang diacunya atau bernilai “null”.

Pada pembangunan aplikasi ini, pemodelan basis data digunakan untuk memodelkan alur pembuatan *database* untuk aplikasi. Pemodelan ini dilakukan agar pembangunan aplikasi tidak terjadi redudansi atau penumpukan data. Jika dilakukan pemodelan tersebut, maka akan mencegah terjadinya kesalahan data.

2.6 MySQL

MySQL adalah perangkat lunak *database server* atau sebut saja *Database Smart*. *Database* ini semakin lama semakin populer. Dengan menggunakan *database* ini, data semakin aman dan berdaya guna. *Database* ini juga banyak dipakai pada web *database* sehingga data semakin terintegrasi antara *database* dekstop dengan *database* web. Untuk menggunakan *database* MySQL harus menginstalasinya dahulu ke komputer. Keistimewaan MySQL Sebagai *server database* dengan konsep *database* modern, MySQL memiliki keistimewaan. Pada pembangunan aplikasi ini menggunakan database MySQL, karena dengan menggunakan MySQL pembangunan *database* lebih mudah dan terjamin keamanannya. Beberapa keistimewaan dimiliki MySQL sebagai berikut :

1. *Portability*

   Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti *Windows*, *Linux*, FreeBSD, Mac OS X *Server*, Solaris, Amiga, HP-Unix, dan lain-lain.

2. *Open Source*

   MySQL merupakan *database open source* (gratis), sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3. *Multiuser*

   MySQL merupakan *database* yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan pula.

4. *Performace Tuning*

MySQL mempunyai kecepatan yang cukup baik dalam menangani *query-query* sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.

5. *Column Type*

Column Type pada *database* MySQL ini, didukung dengan tipe-tipe data yang sangat kompleks.

6. *Command dan Functions*

MySQL *server* memiliki operator dan fungsi secara penuh yang mendukung perintah, seperti pada perintah SELECT dan WHERE dalam query.

7. *Security*

Sistem Security pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan subnetmask, hostname dan izin akses *user* dengan sistem perizinan yang mendetil serta *password terenkripsi*.

8. *Scalability dan Limits*

MySQL mempunyai kemampuan menangani *database* dalam skala cukup besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.

9. *Connectivity*

Adanya kemampuan pada MySQL dengan melakukan koneksi dengan client menggunakan protokol TCP/IP, *Unix socket* (Unix), atau *Named Pipes* (NT).

10. *Localization*

Localization pada MySQL adalah, mempunyai kemampuan dalam mendeteksi kesalahan (*error code*) pada client menggunakan lebih dari dua puluh bahasa.

11. *Interface*

Interface pada MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemograman menggunakan fungsi API (*Application Programming Interface*).

12. *Clients dan Tools*

Clients dan Tools pada *database* MySQL dilengkapi berbagai tools yang dapat digunakan untuk administrasi *database*.

13. Struktur Tabel

MySQL memiliki struktur tabel cukup baik serta cukup fleksibel sehingga memudahkan pengguna untuk menggunakannya, misalnya ketika menangani *alter table*.

2.7 Web *Service*

Web service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* di atas jaringan. Web *service* mempunyai alat penghubung yang diuraikan di dalam format *machine-processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan Web *service* di dalam cara yang ditentukan oleh deskripsinya yang menggunakan pesan SOAP, secara khas disampaikan menggunakan HTTP dengan XML *serialization*, bersama dengan standar lain yang terkait dengan web.

Untuk menjalankan fungsinya, *web service* memerlukan agen. Agen adalah bagian perangkat lunak atau perangkat keras yang mengirimkan dan menerima pesan. Agen dapat ditulis dengan berbagai bahasa pemrograman. Dan dapat berganti-ganti bahasa pemrograman dengan fungsi yang sama. Tujuan *web service* adalah untuk menyediakan beberapa fungsi atas nama pemiliknya, seseorang atau organisasi seperti bisnis atau perorangan. Provider entity adalah organisasi atau orang yang menyediakan agen yang sesuai untuk menerapkan *service* tertentu. *Requester* entity adalah seseorang atau organisasi yang berkeinginan untuk menggunakan *web service provider entity*. Itu akan menggunakan *request*er agent untuk menukar pesan dengan provider agent milik provider entity.

Dalam pertukaran pesan agar berhasil, *reques*ter entity dan provider entity harus dulu sepakat menggunakan semantik dan mekanisme yang sama dalam pertukaran pesan. Semantik dalam web *service* adalah ekspektasi tentang perilaku (*behavior*) *service*. Ini semacam kontrak antara *requester entity* dan *provider entity* mengenai kegunaan dan konsekuensi dari interaksi tersebut. Mekanisme pertukaran pesan didokumentasikan di *Web Service Description* (WSD). WSD adalah spesifikasi mesin yang dapat berproses pada *interface* web *service*. menggambarkan format pesan, tipe data, protokol transpor, dan format serialisasi yang digunakan antara *requester agent* dan *provider agent* [17]. Pada pebangunan aplikasi ini, web service berfungsi untuk menjembatani antara program *frontend* dan *backend*.

2.8 Java

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Beberapa keunggulan java yaitu, Java merupakan bahasa yang sederhana. Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori.

Java merupakan bahasa berorientasi objek (OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas

dapat mewarisi kelas lain [18]. Pada pembangunan aplikasi ini menggunkan bahasa java, alasan penggunaan bahasa java tersebut karena bahasa java lebih sederhana dan mudah dipelajari.

2.9 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. JSON terbuat dari dua struktur [19]:

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 [19]:

1. Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).
3. 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.
4. 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.
5. Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail encoding yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan [19]. Pada pebangunan aplikasi ini dibutuhkan pemanfaatan JSON, karena aplikasi dibangun menjadi aplikasi *frontend* dan *backend* maka proses pertukaran data antara *web service* dan aplikasi *mobile* membutuhkan format data JSON.

2.9 HTML

HTML (*Hypertext Markup Language*) adalah suatu bahasa yang digunakan untuk menulis halaman web. HTML dirancang untuk digunakan tanpa tergantung pada suatu platform tertentu. Dokumen HTML adalah suatu dokumen teks biasa, dan disebut sebagai markup language karena mengandung tanda- tanda (tag) tertentu yang digunakan untuk menentukan tampilan suatu teks tersebut dalam suatu dokumen.

Pada dokumen HTML yang termasuk sistem *hypertext* , kita tidak harus membaca dokumen tersebut secara urut dari atas ke bawah atau sebaliknya, tetapi kita dapat menuju pada topik tertentu secara langsung dengan menggunakan teks penghubung yang akan membawa anda ke suatu topik

atau dokumen lain secara langsung. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu SGML (*Standart Generalized Markup Language*). Sejak awal perkembangan sampai sekarang ini telah tersedia bermacam-macam level (versi) HTML, ada HTML level 1.0, HTML 2.0, HTML 3.0, dan HTML 4.0. **[**20**]** Pada pebangunan aplikasi ini, HTML ini digunakan untuk membangun tampilan *beckend* yaitu berupa *website*.

2.9.1 Bagian HTML

HTML terdiri dari beberapa bagian yang fungsinya sebagai penanda suatu kelompok perintah tertentu, misalnya kelompok perintah form yang ditandai dengan kode <form>, judul dengan <title> dan sebagainya. Untuk lebih lanjut mengenai bagian - bagian HTML perhatikan skema dibawah ini :

```
<html>
  <head>
<title>.....</title>
  </head>
<body>
...isi dari halaman
web...
</body>
</html>
```

Keterangan :

1. Dokumen HTML selalu diawali dengan tanda tag pembuka<html> dan diakhiri dengan tanda tag penutup </html>.
2. Pada elemen head <head></head>, dapat kita sisipkan kode-kode untuk menuliskan keterangan tentang dokumen HTML. Atau dapat juga kita sisipkan scripts-scripts pemograman web seperti *JavaScript*, VB Scripts, atau CSS untuk menambah daya tarik pada situs yang kita buat agar lebih menarik dan dinamis.
3. Elemen body <body></body> berisi tag-tag untuk isi atau layout tampilan situs kita, seperti: , <table></table>, <form></form>.

Jadi jelas bahwa elemen adalah suatu bagian yang besar yang terdiri dari kode-kode tag tersebut. Sedangkan tag hanyalah merupakan bagian dari elemen. Tag adalah kode-kode yang digunakan untuk men-setting dokumen HTML. [20]

2.10 PHP

PHP (*Perl Hypertext Preprocessor*) adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis". Pada pebangunan aplikasi ini, PHP digunakan untuk pembangunan web agar web tersebut lebih dinamis. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis, contoh *syntax* dasar php sebagai berikut : [21]

```
<html>
<body>

  <?php
    echo    "Hello    World";
  ?>

</body>
</html>
```

Adapun kelebihan-kelebihan dari PHP yaitu:

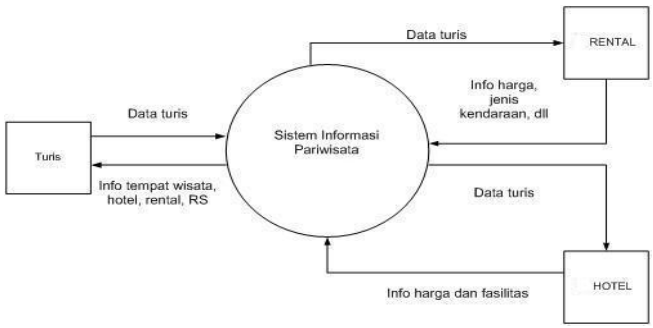
1. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada web *server* yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada *apache* yang bersifat open *source*.
3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembanganya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

5. PHP dapat berjalan pada 3 operating sistem, yaitu: *Linux, unux,* dan *windows*, dan juga dapat dijalankan secara runtime pada suatu console.

2.11 Diagram Konteks

Context Diagram merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran data menuju dan dari sistem diketahui menganalisis dari wawancara dengan user dan sebagai hasil analisis dokumen. [22] Pada pebangunan aplikasi ini, diagram konteks bermanfaat sebagai pengatur sebuah pembangunan sistem agar tersusun sesuai dengan kegunaannya.

Contoh diagram konteks bisa dilihat pada gambar 2.8 sebagai berikut.



Gambar 2.8 Contoh Diagram Konteks

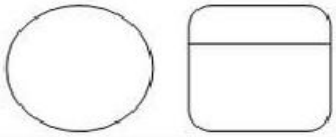
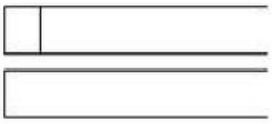
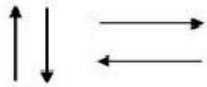
2.12 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. *Data Flow Diagram* (DFD) menunjukkan arus data, penyimpanan data, dan proses yang merespon data yang masuk dan merubahnya.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut disimpan. Kelebihan dari DFD yaitu dapat menggambarkan sistem secara terstruktur dengan mengubah sistem menjadi level yang lebih rendah, sedangkan kekurangannya adalah tidak dapat menunjukan proses pengulangan, proses keputusan dan tidak menunjukan. [22] Pada pebangunan aplikasi ini, DFD berfungsi sebagai pengatur pembangunan sistem untuk perlevelnya agar tersusun sesuai dengan kegunaannya.

Berikut adalah simbol -simbol yang ada dalam DFD, dapat dilihat pada tabel 2.1 sebagai berikut.

Tabel 2.1 Simbol – Simbol DFD

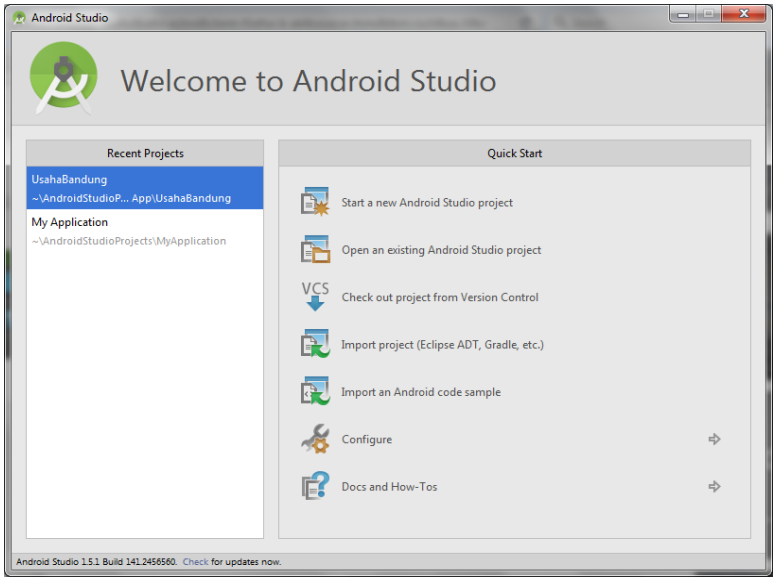
Simbol	Keterangan
	*External Entity*, merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain.
	*Process*, merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan
	*Data Store* (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual
	*Data Flow* (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

2.13 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Pada pebangunan aplikasi ini, android studio berfungsi sebagai alat pembangunan aplikasi rekomendasi ini, karena android studio memiliki banyak fitur yang bisa dimanfaatkan.

Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya [23]:

1. Sistem pembuatan berbasis *Gradle* yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat penguji dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*



Gambar 2.8 : Android Studio

2.14 Sistem

Sistem adalah kumpulan dari komponen-komponen yang saling berhubungan yang saling berinteraksi untuk melakukan suatu tugas untuk mencapai suatu tujuan. Dalam sistem terdapat 3 komponen dasar yang terdapat didalamnya seperti:

1. *Input*, memasukkan elemen-elemen (data mentah) yang akan diproses.
2. *Process*, proses transformasi *input* menjadi *output.*
3. *Output*, mengirimkan elemen-elemen (data mentah) yang telah diproses ke tujuannya.

Jadi, sistem adalah sekumpulan komponen yang saling terkait dan bekerja sama melakukan suatu tugas untuk mencapai suatu tujuan [24] [25]. Pada pembangunan aplikasi ini, sistem komputer brfungsi sebagai pengolahan dan pembangunan sistem karena sifatnya yang cepat, tepat dan memiliki tingkat akurasi yang tinggi.

2.12.1 Kualitas Sistem

Kualitas sistem merupakan ciri karakteristik kualitas yang diinginkan dari sistem informasi itu sendiri dan kualitas informasi yang diinginkan informasi karakteristik produk. Suatu produk yang sama dapat memiliki penilaian kualitas yang berbeda dari masing-masing orang, hal ini disebabkan karena persepsi tiap orang mengenai kualitas berbeda-beda atau dapat dikatakan bahwa kualitas itu sesuatu hal yang relatif dan sangat subjektif [26].

Kualitas sistem memerlukan indikator untuk dapat mengukur seberapa besar kualitas dari sistem. Indikator diperlukan karena kualitas sistem merupakan variabel laten yang tidak dapat diukur secara langsung. Pembangunan aplikalsi ini harus memiliki kualitas sistem yan bagus, karena akan berpengaruh pada proses pengolahan dan pembangunan aplikasi tersebut. Indikator kualitas sistem diwujudkan dalam seperangkat pertanyaan kualitas sistem yang dapat diukur melalui beberapa indikator sebagai berikut [27]:

1. *Ease of Use* (Kemudahan Penggunaan)

   Suatu sistem informasi dapat dikatakan berkualitas jika sistem tersebut dirancang untuk memenuhi kepuasan pengguna melalui kemudahan dalam menggunakan sistem informasi tersebut. Davis mengungkapkan kemudahan yang dipersepsikan adalah tingkatan dimana seseorang percaya bahwa pengunaan suatu sistem tertentu dapat menjadikan orang tesebut bebas dari usaha (*free of effort*). Bebas dari usaha yang dimaksudkan adalah bahwa saat seseorang menggunakan sistem, ia hanya memerlukan sedikit waktu untuk mempelajari sistem tersebut karena sistem tersebut sederhana, tidak rumit, dan mudah dipahami, sudah dikenal (familiar).

   Kemudahan penggunaan dalam konteks ini bukan saja kemudahan untuk mempelajari dan menggunakan suatu sistem tetapi juga mengacu pada kemudahan dalam melakukan suatu pekerjaan atau tugas dimana pemakaian suatu sistem akan semakin memudahkan seseorang dalam bekerja dibanding mengerjakan secara manual. Pengguna sistem informasi mempercayai bahwa sistem informasi yang lebih fleksibel, mudah dipahami dan mudah pengoperasiannya sebagai karakteristik kemudahan penggunaan.

2. *Response Time* (Kecepatan Akses)

   Kecepatan akses merupakan salah satu indikator kualitas sistem informasi. Jika akses sistem informasi memiliki kecepatan yang optimal maka layak dikatakan bahwa sistem informasi yang diterapkan memiliki kualitas yang baik. Kecepatan akses akan meningkatkan kepuasan pengguna dalam menggunakan sistem informasi.

3. *Reliability* (Keandalan Sistem)

   Sistem informasi yang berkualitas adalah sistem informasi yang dapat diandalkan. Jika sistem tersebut dapat diandalkan maka sistem informasi tersebut layak digunakan. Keandalan sistem informasi dalam konteks ini adalah ketahanan sistem informasi dari kerusakan dan kesalahan. Keandalan sistem informasi ini juga dapat dilihat dari

sistem informasi yang melayani kebutuhan pengguna tanpa adanya masalah yang dapat mengganggu kenyamanan pengguna dalam menggunakan sistem informasi.

4. *Flexibility* (Fleksibilitas)

   Fleksibilitas suatu sistem informasi menunjukkan bahwa sistem informasi yang diterapkan tersebut memiliki kualitas yang baik. Fleksibilitas yang dimaksud adalah kemampuan sistem informasi dalam melakukan perubahan-perubahan kaitannya dengan memenuhi kebutuhan pengguna. Pengguna akan merasa lebih puas menggunakan suatu sistem informasi jika sistem tersebut fleksibel dalam memenuhi kebutuhan pengguna.

2.15 Informasi

Informasi menurut Turban et al [28] dalam penelitian Fanny dan Eko [29], merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya. Sedangkan menurut Jogiyanto, Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Dapat dikatakan bahwa data merupakan bahan mentah, sedangkan informasi adalah bahan jadi atau bahan yang telah siap digunakan, Jadi, sumber dari informasi adalah data.

Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian nyata adalah perubahan dari suatu nilai yang disebut dengan transaksi [30]. Pada pembangunan aplikasi ini, informasi yang dihasilkan adalah *list* rekomendasi tanaman hias yang dibutuhkan.

2.15.1 Kualitas Informasi

Untuk memberikan manfaat, informasi harus memiliki kualitas. Kualitas merupakan segala sesuatu yang dapat memenuhi keinginan dan kebutuhan dari pengguna. Indikator yang dapat digunakan untuk mengukur kualitas informasi adalah *relevance* [31], *accurate* [27] [31], *completeness* [27] [31], *timeliness* [31],

dan *understandability* [26] [32] dari informasi yang dihasilkan. Kualitas informasi pada pembangunan aplikasi ini harus dijaga tingkat akurasinya, agar aplikasi tidak salah merekomendasikan tanaman.

1. *Relevance*

   Kualitas informasi yang dihasilkan oleh aplikasi rekomendasi pencarian tanaman ini akan dikatakan baik apabila relevan terhadap kebutuhan pengguna. Aplikasi digunakan pelanggan yang akan melakukan pencarian tanaman hias, sehingga informasi yang disampaikan harus sesuai dengan yang dibutuhkan dalam proses tersebut.

2. *Accurate*

   Informasi yang akurat akan sangat bermanfaat untuk pengambilan keputusan pelanggan [26]. Informasi yang akurat berarti harus bebas dari kesalahan dan tidak menyesatkan pengguna serta menggambarkan dengan jelas maksud dari informasi yang dihasilkan oleh aplikasi.

3. *Completeness*

   Informasi yang dihasilkan oleh aplikasi dapat berkualitas apabila informasi yang dihasilkan lengkap. Kelengkapan informasi mencakup seluruh informasi yang dibutuhkan oleh pelanggan dalam menggunakan aplikasi.

4. *Timeliness*

   Kualitas informasi akan dikatakan baik apabila informasi yang dihasilkan bersifat tepat waktu. Informasi yang sampai kepada pelanggan tidak boleh telat, karena untuk digunakan sebagai pengambilan keputusan, informasi yang digunakan harus tepat waktu dan *uptodate*.

5. Understandability

   Informasi harus dapat dengan mudah untuk digunakan dan dipahami oleh para penggunanya.

2.14 Pengujian Alpha dan Beta

Acceptance testing biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan software dan hardware komersial, *acceptance test* biasanya disebut juga "*alpha tests*" (yang dilakukan oleh pengguna *in-house*) dan "*beta tests*" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). Alpha dan beta test biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. *Acceptance* testing mencakup data, *environment* dan skenario yang sama atau hampir sama pada saat *live* yang biasanya berfokus pada skenario penggunaan produk tertentu.

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Pada pembangunan aplikasi ini, pengujian *alpha* beta digunakan untuk mengetahui fungsional setiap menu telah berjalan baik atau tidak [27].

2.15 Skala Likert (Kuisioner)

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial. Dalam penelitian gejala sosial, peneliti telah menetapkan secara spesifik skalanya dan selanjutnya disebut variabel penelitian. Pada pembangunan aplikasi ini, skala *likert* digunakan untuk mengetahui masalah-masalah yang akan dijadikan sebagai acuan pembangunan aplikasi tersebut. Dengan menggunakan skala *Likert*, variabel yang akan diukur dijabarkan menjadi dimensi, lalu dimensi dijabarkan menjadi subvariabel dan subvariabel dijabarkan lagi menjadi indikator yang dapat diukur. Akhirnya, indikator-indikator yang terukur dapat menjadi titik tolak untuk membuat item instrumen berupa pernyataan atau pertanyaan yang perlu dijawab oleh responden. Setiap jawaban dihubungkan dengan bentuk pernyataan atau dukungan sikap yang diungkapkan dengan kata-kata sebagai berikut [33]:

1. Sangat setuju (SS)
2. Setuju (S)
3. Setuju tidak, tidak setuju pun tidak, alias netral (N)
4. Tidak setuju (TS)
5. Sangat tidak setuju (STS)

Untuk keperluan analis kuantitatif, maka jawaban itu dapat diberi skor, misalnya:

1.	Setuju/selalu/sangat positif diberi skor	5
2.	Setuju/sering/positif diberi skor	4
3.	Ragu-ragu/kadang-kadang/netral diberi skor	3
4.	Tidak setuju/hampir tidak pernah/negative diberi skor	2
5.	Sangat tidak setuju/tidak pernah/sangat positif diberi skor	1

Teknik pengumpulan data angket, contohnya diberikan kepada 100 orang pegawai yang diambil secara acak. Dari 100 orang pegawai setelah dilakukan analisi misalnya

25 orang pegawai menjawab SS
40 orang pegawai menjawab ST
5 orang pegawai menjawab RG
20 orang pegawai menjawab TS
10 orang pegawai menjawab STS

Berdasarkan data tersebut 65 orang atau 65% pegawai menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas pegawai setuju dengan adanya sistem informasi.

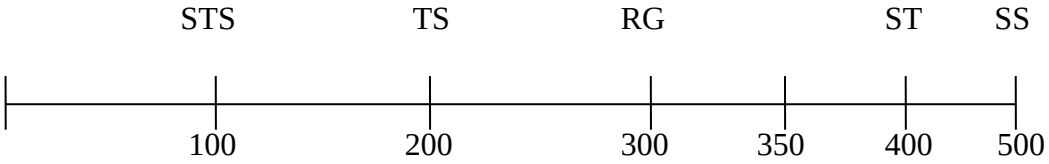
Data tersebut juga dapat dianalisis berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan maka

Jumlah skor untuk 25 pegawai menjawab SS	= 25 x 5 = 125
Jumlah skor untuk 40 pegawai menjawab ST	= 40 x 4 = 160
Jumlah skor untuk 5 pegawai menjawab RG	= 5 x 3 = 15
Jumlah skor untuk 20 pegawai menjawab TS	= 20 x 2 = 50
Jumlah skor untuk 10 pegawai menjawab STS	= 10 x 1 = 10
Jumlah	= 350

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (SS) skor tertinggi jumlah skor rendah 1 x 100 = 100 (STS).

Jadi berdasarkan data itu maka tingkat persetujuan terhadap sistem informasi yang dibuat = (350 : 500) x 100% = 70%

Secara kontinum dapat digambarkan seperti berikut:



Jadi berdasarkan data yang diperoleh dari 100 responden maka data 350 terletak pada daerah setuju.

Bila berdasarkan pada kelompok responden, maka dapat diketahui bahwa :

1. 25 prosen menyatakan sangat setuju (25/100 = 25%)
2. 40 prosen menyatakan setuju (40/100 = 40%)
3. 5 prosen menyatakan ragu-ragu (5/100 = 5%)
4. 20 prosen menyatakan tidak setuju (20/100 = 20%)
5. 10 prosen menyatakan sangat tidak setuju (10/100 = 10%)

a. Contoh bentuk pilihan ganda

   Berilah satu jawaban terhadap pertanyaan berikut sesuai dengan pendapat anda, dengan member tanda lingkaran pada nomor jawaban yang tersedia.

1. Apakah anda setuju, sistem informasi yang dibuat mampu memudahkan dan mangatasi masalah yang ada sebelumnya?
    a. Sangat tidak setuju
    b. Tidak setuju
    c. Ragu-ragu
    d. Setuju
    e. Sangat setuju

Akan tetapi dengan bentuk pilihan ganda itu, maka jawaban dapat diletakkan pada tempat yang berbeda-beda. Semisal untuk jawaban diatas "sangat setuju" diletakkan di paling akhir untuk selanjutnya jawaban "sangat tidak setuju diletakkan di paling awal. Dalam penyusunan instrumen untuk variabel tertentu, sebaiknya butir butir pertanyaan dibuat dalam bentuk kalimat positif, netral, ataupun negative sehingga responden dapat menjawab dengan serius dan konsisten.

Dengan cara demikian maka kecendrungan responden untuk menjawab kolom tertentu dari bentuk checklist dapat dikurangi. Dengan model ini juga selalu membaca pertanyaan setiap item instrumen dan juga jawabannya. Pada bentuk *checklist*, sering jawaban tidak dibaca, karena letak jawaban sudah menentu. Tetapi dengan bentuk *checklist* juga akan didapat keuntungan dalam hal ini singkat dalam pembuatannya, hemat kertas, mudah mentabulasikan data, dan secara visual lebih menarik. Data yang diperoleh dari skala tersebut adalah berupa interval.