BAB II

LANDASAN TEORI

2.1 Pengertian Perangkat Lunak Aplikasi

Menurut Abdul Kadir (Pengenalan Teknologi Informasi 2003:230) Perangkat lunak aplikasi (*application software*) adalah program yang biasa dipakai oleh pemakai untuk melakukan tugas-tugas yang spesifik, misalnya untuk membuat dokumen, memanipulasi data, atau membuat laporan keuangan.

Sebuah program komputer tanpa tersosialisasi dengan dokumentasinya maka belum disebut perangkat lunak aplikasi (*application software*). Sebuah perangkat aplikasi juga sering disebut dengan sistem perangkat lunak aplikasi. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sistem Perangkat Lunak Aplikasi berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (c*ustomer).* Pelanggan (c*ustomer)* adalah orang atau organisasi yang dengan sukarela mengeluarkan uang untuk memesan atau membeli perangkat lunak. *User* atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk atau memudahkan pekerjaannya.

2.2 Pengertian Reservasi

Kata reservasi berasal dari bahasa Inggris, *reservation*. Menurut Kamus Inggris Indonesia (2000:480), *reservation* memiliki arti pesanan tempat.

Jadi pengertian reservasi disini adalah memesan tempat atau waktu kunjungan agar bisa disesuaikan dengan jadwal yang telah ditentukan sehingga bisa mendapatkan fasilitas yang diinginkan sesuai dengan yang telah di pesan.

2.3 Pengertian Hotel

Menurut kamus Bahasa Indonesia (2013) Hotel merupakan bangunan dengan jumlah kamar yang banyak disewakan sebagai tempat menginap dan tempat makan bagi orang yang sedang dalam perjalanan.

Secara harfiah, kata hotel berasal dari bahasa latin yaitu *hospitium*,yang artinya ruang tamu. Kata ini kemudian mengalami proses perubahan pengertian dan untuk membedakan *guest house* dengan *mansion house* yang berkembang saat itu, maka rumah besar disebut hostel. Hostel disewakan pada masyarakat umum untuk menginap dan beristirahat sementara waktu,dan dikoordinir oleh seorang *host*. Seiring perkembangan dan adanya tuntutan terhadap kepuasan,dimana orang tidak menyukai peraturan yang terlalu banyak pada hostel, maka kata hostel kemudian mengalami perubahan, yakni penghilangan hurug “s” pada kata hostel sehingga menjadi hotel.

Jadi dapat disimpulkan pengertian hotel adalah suatu bagunan yang menyediakan jasa penginapan, makanan dan minuman, serta jasa lainnya yang diperuntukan bagi umum dan dikelola secara komersial.

2.3.1 Klasifikasi Hotel

Hotel dapat diklasifikasikan menjadi 8 kategori, yaitu :

1. Berdasarkan Luas dan Jumlah Kamar
   a. Hotel Kecil (*small* hotel) adalah hotel yang memiliki lebih dari 25 kamar atau kurang dari 100 kamar.
   b. Hotel Menengah (*above average* hotel) ialah hotel yang memiliki lebih dari 100 kamar dan kurang dari 300 kamar.
   c. Hotel Besar (*large* hotel), yaitu hotel yang memiliki lebih dari 300 kamar.
2. Berdasarkan Jenis Tamu yang Menginap
   a. Hotel Keluarga (*family* hotel) adalah hotel yang dirancang untuk keluarga.
   b. Hotel Bisnis (bisnis hotel) adalah hotel yang dirancang untuk para usahawan.
   c. Hotel Wisata (*tourist* hotel) adalah hotel yang dirancang untuk para wisatawan.
   d. Hotel Transit (*transit* hotel) adalah hotel yang dirancang khusus untuk orang-orang yang melakukan persinggahan sementara dalam suatu perjalanan.
   e. Hotel Perawatan Kesehatan (*cure* hotel) adalah hotel yang dirancang untuk orang-orang yang sedang menginginkan penyembuhan dari suatu penyakit atau meningkatkan kesehatannya.
   f. Hotel Konvensi (*convention* hotel) adalah hotel yang dirancang untuk keperluan orang-orang yang menyelenggarakan konvensi.

3. Berdasarkan Lama Tamu Menginap
    a. *Transient* hotel, yaitu hotel dimana para tamunya menginap hanya untuk satu atau dua malam.
    b. *Semi-residential* hotel, yaitu hotel dimana para tamunya lebih dari 12 malam sampai satu minggu.
    c. *Residential* hotel, yaitu hotel dimana para tamunya menginap untuk jangka waktu lama, lebih dari satu minggu.
4. Berdasarkan Lokasi
    a. *Mountain* hotel, adalah hotel yang terletak didaerah pegunungan.
    b. *Beach* hotel, adalah hotel yang terletak ditepi pantai.
    c. *City* hotel, adalah hotel yang berlokasi dperkotaan.
    d. *Highway* hotel, adalah hotel yang terletak ditepi jalan bebas hambatan dan biasanya diantara dua kota.
    e. *Airport* hotel, adalah hotel terletak tidak jauh dari *airport*.
    f. *Resort* hotel, adalah hotel yang berlokasi dikawasan wisata.
5. Berdasarkan Lamanya Buka dalam Setahun
    a. *Seasonal* hotel, yaitu hotel yang hanya buka pada waktu-waktu tertentu dalam setahun (3 bulan, 6 bulan, 9 bulan).
    b. *Year-round* hotel, yaitu hotel yang buka sepanjang tahun.
6. Berdasarkan Tarif Kamar
    a. *Economy class* hotel, ialah hotel yang memiliki tarif kamar kelas ekonomi (harga kamar relatif murah).
    b. *First class hotel*, yaitu hotel dengan tarif kamar mahal.

c. *Deluxe/luxury* hotel, yaitu hotel yang memiliki harga kamar sangat mahal.

7. Menurut Bintang (*Star*)

   Berdasarkan Surat Keputusan Menteri Perhubungan No. PM. 10 / P.V.301 / Pht /77 tanggal 22 Desember 1977 tentang Peraturan Industri Perhotelan dan Klasifikasi hotel antara lain ditentukan menurut bintang, yaitu mulai bintang satu hingga lima. Perbedaan bintang tersebut terlihat dari fasilitas, peralatan dan mutu serta standar pelayanan. Penentuan kelas atau bintang diadakan setiap tiga tahun sekali dan ditetapkan oleh Keputusan Direktur Jenderal Pariwisata dalam bentuk sertifikat.

8. Berdasarkan Unsur atau Komponen Harga Kamar (*Type of Plan*)

   a. *European Plan* Hotel (*EP*), ialah hotel yang menetapkan bahwa harga kamar hanya untuk kamar saja.

   b. *Continental Plan* Hotel (*CP*), ialah hotel yang menetapkan bahwa harga kamar termasuk makan pagi.

   c. *Modified American Plan* Hotel (*MAP*), ialah hotel yang menetapkan bahwa harga kamar termasuk dua kali makan (makan pagi, siang atau malam).

   d. *Full American Plan* (*FAP*), yaitu harga kamar termasuk tiga kali makan.

2.4 Pengertian Android

Menurut Nazruddin Safaat H (Pemrograman aplikasi *mobile smartphone* dan *tablet PC* berbasis android 2012:1) android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi.

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/*smartphone*. 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. Di lain pihak, Google merilis kode–kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Didunia ini terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal *Open Handset Distribution (OHD).*

Kelebihan-kelebihan Android dibandingkan dengan *mobile platform* yang ada dipasaran, yaitu :

1. Merupakan pengembang platform bebas yang berbasis *linux* dan *open source.*
2. Sebuah komponen berbasis arsitektur yang terinspirasi oleh *mashup* internet. Bagian dari satu aplikasi dapat dikembangkan, digantikan, ditingkatkan

komponennya sesuai dengan versi yang diinginkan sehingga dapat memunculkan berbagai kreativitas pada *mobile application.*

3. Memiliki layanan berbasis lokasi menggunakan GPS atau triangulasi menara sel untuk menyesuaikan lokasi anda berada.
4. Memungkinkan anda untuk memanfaatkan kekuatan penyimpanan lokal untuk berhubungan dengan komputasi dan sinkronisasi.
5. Android dioptimalkan untuk perangkat daya rendah dan rendah memori, yang ada pada *platform* sebelumnya tidak terpikirkan.
6. Mendukung grafis dan suara berkualitas tinggi dan lebih halus.
7. Memiliki probabilitas diberbagai perangkat keras saat ini dan masa depan.
8. Program yang terisolasi dapat memberikan tingkat stabilitas sistem yang tidak terlihat sebelumnya pada *smartphone.*
9. Android menawarkan berbagai cara untuk berinteraksi dengan *mobile application* bersama dengan dasar-dasar teknis untuk membuatnya.
10. Dalam pengembangannya, hanya diperlukan satu komputer yang ter-*install* SDK Android dan emulator.

Adapun kekurangan-kekurangan Android adalah sebagai berikut:

1. *Handset* Android dibuat oleh banyak vendor dengan ukuran layar yang berbeda, sebagian menggunakan *keyboard* fisik, sebagian lagi hanya menggunakan *touch screen.* Versi Android yang terpasang di *handset* juga berbeda-beda, hal ini menyebabkan pengalaman pemakai Android tidak seragam, hal ini juga sekaligus menyulitkan *programmer* dalam membuat

program, karena apabila versi sistem operasi tidak sama, program tersebut tidak bisa di-*install*.

2. Android adalah produk *open source* dari Google dan *Open Handset Alliance*, karena itu aplikasi yang ada dalam Android sangat menyatu dengan layanan Google, contohnya *email* menggunakan *Gmail*, kalender menggunakan Google *Calender*, bahkan sampai *contact* pun disimpan di *Gmail Contact*. Bagi penggemar Google ini merupakan keunggulan sendiri, namum bagi pengguna yang tidak familiar dengan produk Google, hal in cukup merepotkan karena mereka harus menyesuaikan keadaan.
3. Android cukup aktif dikembangkan, hampir tiap tahun mereka merilis sistem operasi baru, pada tahun 2010 Google merilis *Froyo (2.2)*, *Gingerbread (2.3)* dan *Honeycomb (3.0)*, pada tahun 2011 merilis *Ice Cream Sandwich (4.0).* Namun dari sisi *vendor* atau operator, *update* ke *user* membutuhkan waktu berbulan-bulan, sebagai contoh, *Froyo* dirilis pada Mei 2010, namun pengguna Android kebanyakan baru meng-*update* setelah September.

2.4.1 Android SDK

Menurut Nazruddin Safaat H. (Pemrograman aplikasi *mobile smartphone* dan *tablet PC* berbasis android 2012:5) Android SDK (*Software Development Kit*) adalah *tools* API yang diperlukan untuk pengembangan atau pembangunan suatu aplikasi pada *platform* android menggunakan bahasa pemrograman *Java*.

Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa

pemrograman *Java*. Sebagai *platform* aplikasi *netral*, android memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan bawaan *handphone/smartphone*.

Beberapa fitur-fitur android yang paling penting adalah :

a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
b. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile.*
c. *Integrated browser* berdasarkan *engine opensource WebKit.*
d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES* 1,0 (Opsional akselerasi *hardware*).
e. SQLite untuk penyimpanan data.
f. Media *support* yang mendukung *audio, video*, dan berbagai format gambar (MPEG4, H.264, MP3,AAC, AMR, JPG, PNG, GIF), GSM *Telephony* (tergantung *hardware*).
g. GSM, *Bluetooth*, EDGE, 3G, dan WiFi (tergantung *hardware*).
h. Kamera, *Global Positioning System* (GPS), kompas, dan *accelerometer* (tergantung *hardware*).
i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk *IDE Eclipse*.

2.4.2 ADT (*Android Development Tools*)

Menurut Nazruddin Safaat H (Pemrograman aplikasi *mobile smartphone* dan *tablet PC* berbasis android 2012 : 6) ADT (*Android Development Tools*)

adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse.

Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project* android, membuat GUI aplikasi, dan menambahkan komponen – komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui eclipse. Dengan ADT juga kita dapat melakukan pembuatan *package* android (.*apk*) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Mengembangkan aplikasi android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi android. Berikut adalah versi ADT untuk eclipse yang sudah dirilis :

a. ADT 12.0.0 (*July* 2011)
b. ADT 11.0.0 (*June* 2011)
c. ADT 10.0.1 (*March* 2011)
d. ADT 10.0.0 (*February* 2011)
e. ADT 9.0.0 (*January* 2011)
f. ADT 8.0.1 (*December* 2010)
g. ADT 8.0.0 (*December* 2010)
h. ADT 0.9.9 (*September* 2010)
i. ADT 0.9.8 (*September* 2010)
j. ADT 0.9.7 (*May* 2010)
k. ADT 0.9.6 (*March* 2010)
l. ADT 0.9.5 (*December* 2009)

m. ADT 0.9.4 (*October* 2009)

2.4.3 Komponen Aplikasi Android

Menurut Nazruddin Safaat H (Pemrograman aplikasi *mobile smartphone* dan *tablet PC* berbasis android (2012:9) ada empat jenis komponen pada aplikasi android yaitu:

a. Activites

Suatu *activity* akan menyajikan *user interface (UI)* kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bias jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *acivity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface (UI)* saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain kita dapat melakukannya dengan satu *event*, misalnya klik tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method Activity.setContentView(). ContentView* adalah objek yang berada pada *root* hirarki.

b. Service

Service tidak memiliki *Graphic Users Interface (GUI)*, tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya media *player* sedang memutar

lagu dari *list* yang ada, Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu misalnya, atau menulis sms sambil *player* sedang jalan. Untuk menjaga musik tetap di jalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

c. *Broadcast Receiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *Broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh *camera*, atau pengubahn referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

Broadcast receiver tidak memiliki *user interface (UI)*, tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat dan lain sebagainya.

d. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file* system seperti *database SQLite*. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan

untuk mengakses data kontak dan navigasi, maka di sinilah fungsi *content provider*.

2.4.4 Versi Android

Telepon pertama yang memakai system operasi android adalah *HTC Dream*, yang dirilis pada Oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua *vendor seluler* di dunia mengggunakan android sebagai *operating system*.

Adapun versi-versi android yang pernah dirilis adalah sebagai berikut:

a) Versi 1.0 dirilis 23 September 2008

b) Versi 1.1 dirilis 9 Maret 2009

c) Versi 1.5 (*Cupcake*)

d) Versi 1.6 (*Donut*)

e) Versi 2.0 / 2.1 (*Eclair*)

f) Versi 2.2 (*Froyo:Frozen Yoghurt*)

g) Versi 2.3 (*Gingerbread*) yaitu versi yang digunakan saat membangun aplikasi ini.

h) Versi 3.0 / 3.1 / 3.2 (*Honeycomb*)

i) Versi 4.0 (*IceCreamSandwich*)

j) Versi 4.1 (*Jelly Bean*)

k) Versi 4.2 (*Key Lime Pie*)

2.4.5 Eclipse

Menurut Nazruddin Safaat H (Pemrograman aplikasi *mobile smartphone* dan *tablet PC* berbasis android 2012:16) Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk pengembangan *java*/android yang *free* dan dapat di-*download* di ***http://www.eclipse.org/downloads/***, versi eclipse yang ada sekarang sudah banyak seperti *Eclipse Helios* (eclipse versi 3.6), *Eclipse Galileo* (eclipse versi 3.5), dan *Eclipse Ganymede* (Eclipse versi 3.4), kita dapat melakukan instalasi salah satu versi, versi eclipse 3.4 sudah *support* dengan *Android Development Tools (ADT)* untuk membuat eclipse dapat digunakan untuk *coding project Android*.

Berikut ini adalah sifat dari Eclipse:

a. *Multi-platform* : Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*.

b. *Multi-language* : Eclipse dikembangkan dengan bahasa pemrograman *Java*, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis Bahasa pemrogrman lainnya, seperti *C/C++*, *Cobol,Phyton,Perl,PHP* dan lain sebagainya.

c. *Multi-role* : Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, tes perangkat lunak, pengembangan *web*, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in.*

2.5 Pengertian Java

Menurut Abdul kadir (Dasar Pemrograman Java 2 2005:2) Java adalah bahasa pmrograman serbaguna, Java dapat digunakan untuk membuat suatu program sebagaimana anda membuatnya dengan bahasa seperti *Pascal* atau *C++*.

Yang lebih menarik, Java juga mendukung sumberdaya Internet yang saat ini popular, yaitu *World Wide Web* atau yang sering disebut *web* saja. Java juga mendukung aplikasi klien/server, baik dalam jaringan *local* (LAN) maupun jaringan berskala luas (WAN).

Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada *C* dan *C++* namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*.

2.6 Pengertian MySql

Menurut Budi Raharjo (2011 :21) MySql merupakan *software* RDBMS (atau server *database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangata besar, dapat diakses oleh banyak *user* (*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Saat ini, MySql banyak digunakan diberbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai industri, baik industri kecil, menengah, maupun besar.

Lisensi MySql terbagi menjadi dua. Anda dapat menggunakan MySql sebagai produk *open source* dibawah *GNU General Public Licensi* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySql versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataanya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik.

2.6.1 Keunggulan MySql

Beberapa keunggulan dari MySql yaitu :

a) Cepat, handal dan mudah dalam penggunaannya

b) Didukung oleh berbagai bahasa

c) Mampu membuat tabel berukuran sangat besar

d) Lebih Murah

e) Melekatnya integrasi PHP dengan MySql

2.7 Pengertian PHP

Menurut Abdul Kadir (2008 : 2) yang dimaksud dengan PHP merupakan singkatan dari *Hypertext Preprocessor*, Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server.

Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman *web*. Pada prinsispnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Clod Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip *Perl* yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya. Yakni pada tahun 1994. Skrip - skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*Personal Home Page*". Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah pemograman PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan - perhitungan yang kompleks sambil jalan.

2.8 Pengertian 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 :

- 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*.
- 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 pemrograman *modern* mendukung struktur 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.

2.9 Pengertian UML

Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami".

Menurut Rosa A. S (Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek 2013:137) UML (*Unified Modeling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa "*Unified Modelling Language* (*UML*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*)".

2.9.1 Diagram – Diagram dalam UML

Adapun diagram – diagram dalam UML mempunyai beberapa diagram antara lain yaitu:

1) *Use case Diagram*

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa yang berhak menggunakan fungsi-fungsi itu.

2) *Activity Diagram*

Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

3) *Class Diagram*

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

4) *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario pada *use case.*

5) *Component Diagram*

Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

6) *Deployment Diagram*

Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

a. Sistem tambahan (*embedded system)* yang menggambarkan rancangan *device, node,* dan *hardware.*
b. Sistem *client/server.*
c. Sistem terdistribusi murni
d. Rekayasa ulang aplikasi.

Adapun tujuan penggunaan UML yaitu :

a. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
b. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
c. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
d. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail.

Sedangkan perangkat lunak yang mendukung pembuatan diagaram *UML* yaitu

a) *StarUML* (*http://StarUML.sourceforge.net/en/)*

StarUML adalah sebuah proyek *Staropen source* untuk mengembangkan cepat, *fleksibel*, *extensible*, *featureful*, dan bebas-tersedia *UML* / *platform MDA* berjalan pada *platform Win32* dan *Win64*. Tujuan dari proyek *StarUML* adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat UML komersial seperti *Rational Rose*.

b) *ArgoUML* (*http://ArgoUML.tigris.org/)*

ArgoUML adalah *Staropen source UML modeling tool* terkemuka dan termasuk dukungan untuk semua diagram UML standar 1,4. Ini berjalan pada setiap *platform* Java dan tersedia dalam bahasa sepuluh. *ArgoUML* ditulis seluruhnya di Java dan menggunakan Java Kelas *Foundation*. Hal ini memungkinkan *ArgoUML* untuk berjalan di hampir semua *platform*.