BAB 2
TINJAUAN PUSTAKA

2.1 Profil Sekolah

Profil sekolah akan menjelaskan mengenai sejarah singkat sekolah serta visi dan misi dari SMP N 14 Bandung.

2.1.1 Sejarah Singkat Sekolah

SMP Negeri 14 Bandung didirikan pada tahun 1962 sebagai bagian dari SMPN VII, Jl Ambon Bandung. Mula-mula dibangun 4 lokal untuk ruang kelas dan sebuah ruang kecil untuk kantor. Pada tahun 1965 ditambah 2 ruang kelas sederhana, lalu dilanjutkan tahun 1967 dibangun lagi 3 lokal dan pada tahun 1968 ditambah lagi 3 lokal.

Pada tahun 1971 sekolah filial SMPN II diresmikan sebagai sekolah induk yang berdiri sendiri dengan nama SLTPN 14 Bandung berdiri sebagai sekolah mandiri tanggal 20 oktober 1971 dengan SK No 0187/1971.

SMP Negeri 14 Bandung menempati suatu area strategis di Kota Bandung dengan luas tanah 3569 m2 dengan luas bangunan 1497 m2 selain dari itu SMPN 14 Bandung terus berbenah dalam rangka meningkatkan pelayanan terhadap siswa di tahun ajaran 2006/2007 membangun ruang kelas dalam rangka terpenuhinya rencana SMP agar dapat terselenggara proses kegiatan pembelajaran di pagi hari di tahun ajaran 2007/2008.

2.1.2 Visi dan Misi Sekolah

Visi dari SMP Negeri 14 Bandung adalah “SMP Negeri 14 Bandung unggul dalam prestasi akademik, non-akademik, religi, sosial budaya, berwawasan international dan berwawasan lingkungan”

Sedangkan misi dari SMP Negeri 14 ini adalah sebagai berikut :

a. Melaksanakan pembelajaran yang interaktif dan menyenangkan

b. Memfasilitasi siswa dalam mengembangkan potensi dibidang agama, seni, dan olahraga sesuai minat dan bakat
c. Mengembangkan dan mencintai budaya Sunda, dan dapat mengimplementasikannya dalam kehidupan sehari-hari
d. Membentuk peribadi yang berakhlakul karimah (baik dan terpuji)
e. Membuka jaringan kerjasama dengan klub olahraga dan sanggar kesenian
f. Menyediakan fasilitas dan layanan jaringan informasi bagi orangtua dan siswa

2.2 Landasan Teori

Landasan teori merupakan bagian yang akan membahas tentang uraian teori yang berhubungan dengan pembangunan sistem. Landasan teori yang akan dibahas adalah media pembelajaran, *mobile learning*, media interaktif, geografi, android, basis data, perancangan sistem, dan *tools* yang digunakan.

2.2.1 Media Pembelajaran

Media adalah kata jamak dari medium, yang artinya perantara. Dalam proses komunikasi, media hanyalah satu dari empat komponen yang harus ada. Komponen yang lain, yaitu : sumber informasi, informasi dan penerima informasi. Seandainya satu dari empat komponen tersebut tidak ada, maka proses komunikasi tidak mungkin terjadi. Interaksi dan saling ketergantungan keempat komponen tersebut dapat terlihat pada Gambar 2.1



Gambar 2.1 Proses Komunikasi [15]

Dalam pembelajaran (*instructional*), sumber informasi adalah dosen, guru, instruktur, peserta didik, bahan bacaan dan sebagainya. Menurut

Schramm (1977), media pembelajaran adalah teknologi pembawa pesan (informasi) yang dapat dimanfaatkan untuk keperluan pembelajaran. Briggs (1977) mendifinisikan media pembelajaran sebagai sarana fisik untuk menyampaikan isi / materi pembelajaran. Sedang menurut Arief S. Sadiman (1986) media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyalurkan pesan dari pengirim ke penerima, sehingga dapat merangsang pikiran, perasaan, perhatian dan minat siswa sehingga proses belajar terjadi.

2.2.1.1 Manfaat Media Pembelajaran

Secara umum manfaat media dalam pembelajaran adalah memperlancar interaksi guru dan siswa, dengan maksud membantu siswa belajar secara optimal. Namun demikian, secara khusus manfaat media pembelajaran dikemukakan oleh Kemp dan Dayton (1985), yaitu :

1. Penyampaian materi pembelajaran dapat diseragamkan

   Guru mungkin mempunyai penafsiran yang beraneka ragam tentang sesuatu hal. Melalui media, penafsiran yang beragam ini dapat direduksi dan disampaikan kepada siswa secara seragam.
2. Proses pembelajaran menjadi lebih menarik

   Media dapat menyampaikan informasi yang dapat didengar (audio) dan dapat dilihat (visual), sehingga dapat mendeskripsikan prinsip, konsep, proses atau prosedur yang bersifat abstrak dan tidak lengkap menjadi lebih jelas dan lengkap.
3. Proses pembelajaran menjadi lebih interaktif

   Jika dipilih dan dirancang dengan benar, media dapat membantu guru dan siswa melakukan komunikasi dua arah secara aktif. Tanpa media, guru mungkin akan cenderung berbicara “satu arah” kepada siswa.
4. Jumlah waktu belajar-mengajar dapat dikurangi

   Sering kali terjadi, para guru banyak menghabiskan waktu untuk menjelaskan materi ajar. Padahal waktu yang dihabiskan tidak perlu

sebanyak itu, jika mereka memanfaatkan media pembelajaran dengan baik.

5. Kualitas belajar siswa dapat ditingkatkan
   Penggunaan media tidak hanya membuat proses pembelajaran lebih efisien, tetapi juga membantu siswa menyerap materi ajar secara lebih mendalam dan utuh.
6. Proses pembelajaran dapat terjadi dimana saja dan kapan saja
   Media pembelajaran dapat dirancang sedemikian rupa sehingga siswa dapat belajar dimana saja dan kapan saja mereka mau, tanpa tergantung pada keberadaan guru.
7. Sikap positif siswa terhadap proses belajar dapat ditingkatkan
   Dengan media, proses pembelajaran menjadi lebih menarik. Dan hal ini dapat meningkatkan kecintaan dan apresiasi siswa terhadap ilmu pegetahuan dan proses pencarian ilmu.
8. Peran guru dapat berubah ke arah yang lebih positif dan produktif
   Dengan media, guru tidak perlu mengulang-ulang penjelasan dan mengurangi penjelasan verbal (lisan), sehingga guru dapat memberikan perhatian lebih banyak kepada aspek pemberian motivasi, perhatian, bimbingan dan sebagainya.

2.2.1.2 Pembelajaran Berbasis Komputer

Menurut Kearsley secara garis besar tujuan dari penggunaan komputer sebagai alat bantu dalam dunia pendidikan atau disebut dengan pelatihan berbasis komputer atau CBT (*Computer Based Training*) adalah untuk mencapai cara belajar yang efektif (peningkatan hasil belajar mengajar) dan efisien (penggunaan sumber daya yang terbatas seperti manusia, waktu, peralatan, dan sebagainya) [4]. Sedangkan tujuan digunakan CAI atau perangkat ajar sebagai media pelengkap pembelajaran yang lebih khususnya ada sepuluh yaitu [5] :

1. Peningkatan pengawasan

CBT melakukan peningkatan pengawasan dalam hal memperbaiki pemakain atau penyelesaian suatu materi, peningkatan standarisasi suatu pelatihan, dan pengawasan terhadap kemampuan pengguna.

2. Pengurangan sumber daya manusia

   Dengan menyajikan perangkat ajar maka pengajaran secara individu akan menyebabkan kebutuhan tenaga ajar dapat dikurangi.

3. Individualisasi

   Siswa dapat belajar sendiri lebih cepat, dan dapat memilih topik yang ada dan mengembangkan cara belajar sesuai dengan topik yang diinginkan.

4. Ketepatan waktu pelatihan

   Masalah utama dalam pelatihan yaitu harus dapat memberikan materi pada waktu yang tepat dan dengan adanya perangkat ajar dapat memberikan pelatihan yang cepat.

5. Pengurangan waktu latihan

   Menurut hasil suatu penelitian, pelatihan menggunakan komputer lebih cepat 30 persen bila dibandingkan tanpa menggunakan komputer. Misalnya bagi orang yang sudah bekerja dapat juga belajar sewaktu-waktu dengan menggunakan komputer tanpa menghabiskan waktu di perjalanan menuju ke tempat pelatihan.

6. Perbaikan hasil kinerja

   Dengan perangkat ajar dapat memperbaiki kualitas pelatihan dan meningkatkan hasil kerja secara langsung maupun tidak langsung. Secara langsung dapat melatih para siswa dalam keahlian khusus yang diperlukan dalam suatu hal, sedangkan secara tidak langsung dapat menyediakan pelatihan yang lebih umum dari biasanya.

7. Kenyamanan pengguna

   Perangkat ajar akan membantu para pengguna komputer juga menghadapi masalah dalam sistem penggunaan komputer

8. Mengubah cara belajar

Cara belajar para siswa tidak hanya bisa belajar disekolah tetapi juga bisa belajar sendiri dengan menggunakan perangkat ajar yang ada.

9. Peningkatan kepuasan belajar

   Perangkat ajar lebih bersifat interaktif sehingga membuat para pelajar merasa puas dimana timbulnya motivasi belajar bagi para siswa juga sangat penting selama waktu belajar.

10. Pengurangan waktu pengembangan

    Pengembangan suatu materi seperti perubahan peralatan buku-buku pelajaran, suatu prosedur bahan atau materi, dan latian. Membuat perbaikan seperti ini akan menyulitkan karena akan menyababkan pelatihan menjadi tidak efektif dan memerlukan biaya yang cukup besar. Bila digunakan perangkat ajar maka hanya dilakukan perbaikan pada data pusat saja dan dapat dilakukan dengan mudah dan waktu relatif lebih cepat.

2.2.2 Mobile Learning

Mobile learning (*m-learning*) merupakan model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi. Pada konsep pembelajaran tersebut *mobile learning* membawa manfaat ketersediaan materi ajar yang dapat di akses setiap saat dan visualisasi materi yang menarik. Istilah *M-Learning* atau *Mobile Learning* merujuk pada penggunaan perangkat genggam seperti PDA, ponsel, laptop dan perangkat teknologi informasi yang akan banyak digunakan dalam belajar mengajar, dalam hal ini kita fokuskan pada perangkat *handphone* (telepon genggam). Tujuan dari pengembangan *mobile learning* sendiri adalah proses belajar sepanjang waktu (*long life learning*), siswa/mahasiswa dapat lebih aktif dalam proses pembelajaran, menghemat waktu karena apabila diterapkan dalam proses belajar maka mahasiswa tidak perlu harus hadir di kelas hanya untuk mengumpulkan tugas, cukup tugas tersebut dikirim melalui aplikasi pada

mobile phone yang secara tidak langsung akan meningkatkan kualitas proses belajar itu sendiri [6].

Mobile Learning memiliki tiga fungsi dalam kegiatan pembelajaran di dalam kelas (*classroom instruction*), yaitu sebagai berikut :

1. Suplemen (tambahan)

   Mobile Learning berfungsi sebagai suplemen (tambahan), yaitu: peserta didik

   mempunyai kebebasan memilih, apakah akan memanfaatkan materi *mobile learning* atau tidak. Dalam hal ini, tidak ada kewajiban/keharusan bagi peserta didik untuk mengakses materi *mobile learning*. Sekalipun sifatnya opsional, peserta didik yang memanfaatkannya tentu akan memiliki tambahan pengetahuan atau wawasan.

2. Komplemen (pelengkap)

   Mobile Learning berfungsi sebagai komplemen (pelengkap), yaitu: materinya

   diprogramkan untuk melengkapi materi pembelajaran yang diterima peserta didik di dalam kelas. Di sini berarti materi *mobile learning* diprogramkan untuk menjadi materi *reinforcement* (penguatan) atau remedial bagi peserta didik di dalam mengikuti kegiatan pembelajaran konvensional.

3. Substitusi (pengganti)

   Beberapa perguruan tinggi di negara-negara maju memberikan beberapa alternatif model kegiatan pembelajaran kepada para peserta didik /siswanya. Tujuannya agar para peserta didik dapat secara fleksibel mengelola kegiatan perkuliahannya sesuai dengan waktu dan aktifitas sehari-hari peserta didik. Ada tiga alternatif model kegiatan pembelajaran yang dapat dipilih peserta didik, yaitu:

   1) sepenuhnya secara tatap muka (konvensional)
   2) sebagian secara tatap muka dan sebagian lagi melalui internet
   3) sepenuhnya melalui internet.

2.2.3 Media Interaktif

Menurut Seels dan Glasgow dalam Arsyad (2006:36) mengemukakan bahwa media interaktif merupakan sistem media penyampaian yang menyajikan materi video rekaman dengan pengendalian komputer kepada penonton (mahasiswa) yang tidak hanya mendengar dan melihat video dan suara, tetapi juga memberikan respon yang aktif dan respon itu yang menentukan kecepatan dan sekuensi penyajian. Kamus Besar Bahasa Indonesia (KBBI) media interaktif adalah alat perantara atau penghubung berkaitan dengan komputer yang bersifat saling melakukan aksi antar-hubungan dan saling aktif.

2.2.4 Geografi

Geografi adalah ilmu yang menelaah bumi sebagai tempat tinggal atau ruang huni bagi manusia dan manusia sebagai penghuni bumi [7]. Tujuan pembelajaran Geografi meliputi tiga aspek, yaitu pengetahuan, keterampilan, dan sikap [8].

a. Pengetahuan
    1) Mengembangkan konsep dasar Geografi yang berkaitan dengan pola keruangan dan proses-prosesnya.
    2) Mengembangkan pengetahuan sumber daya alam, peluang, dan keterbatasannya untuk dimanfaatkan.
    3) Mengembangkan konsep dasar Geografi yang berhubungan denganlingkungan sekitar dan wilayah negara/dunia.
b. Keterampilan
    1) Mengembangkan keterampilan mengamati lingkungan fisik, lingkungan sosial, dan lingkungan binaan.
    2) Mengembangkan keterampilan mengumpulkan, mencatat data, dan informasi yang berkaitan dengan aspek-aspek keruangan.
    3) Mengembangkan keterampilan analisis, sintesis, kecenderungan, dan hasil-hasil dari interaksi berbagai gejala geografis.

c. Sikap

1) Menumbuhkan kesadaran terhadap perubahan fenomena geografi yang terjadi di lingkungan sekitar.
2) Mengembangkan sikap melindungi dan tanggung jawab terhadap kualitas lingkungan hidup.
3) Mengembangkan kepekaan terhadap permasalahan dalam hal pemanfaatan sumber daya.
4) Mengembangkan sikap toleransi terhadap perbedaan sosial dan budaya.
5) Mewujudkan rasa cinta tanah air dan persatuan bangsa.

2.2.5 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware,* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan 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 open source pada perangkat mobile. Di lain pihak, Google merilis kode – kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Di dunia 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 sebagai *Open Handset Distribution* (OHD) [9].

2.2.5.1 Arsitektur Android

Secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut [9] :

1. *Applications dan Widgets*

   Applications dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja dimana biasanya kita *download* aplikasi kemudian dengan kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semuanya aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Applications Frameworks*

   Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembangnya atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memliki 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 (*reuse*).

   Komponen – komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

   a. *Views*
   b. *Content Provider*
   c. *Resource Manager*
   d. *Notification Manager*

e. Activity Manager

3. Libraries

Libraries ini adalah *layer* dimana 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, serta:

a. *libraries* media untuk pemutaran media audio dan video
b. *libraries* untuk manajemen tampilan
c. *libraries* Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D
d. *libraries* SQLite untuk dukungan *database*
e. *libraries* SSL dan WebKit terintegrasi dengan web browser dan security
f. *libraries* LiveWebcore mencakup modern web browser dan security
g. *libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API‟s.

4. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan di mana 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:

a. Core *Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine* Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.
b. *Dalvik Virtual Machine*: Virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux Kernel

Linux Kernel adalah *layer* di mana inti dari *operating system* dari Android itu berada. Berisi *file system* yang mengatur sistem *processing, memory, resourse, drivers,* dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6.

2.2.5.2 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode java dikompilasi bersama dengan *file resources* yang dibutuhkan oleh aplikasi di mana prosesnya di-*package* oleh *tools* yang dinamakan "apt tools" ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat di *install* di perangkat mobile.

Ada beberapa jenis komponen pada aplikasi Android yaitu [9] :

a. Activities

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* 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 even, misalnya click 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 User 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 pengguna 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 pengubahan 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 diterima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* 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 sistem seperti *database* SQLite. Content Provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*. Misalnya kita menggunakan aplikasi yang membutuhkan peta (*Map*). Atau aplikasi yang membutuhkan mengakses data kontak dan navigasi, maka disinilah fungsi *content provider*.

2.2.6 Basis Data

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain – lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

2.2.6.1 Tujuan Basis Data

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuannya, syarat sebuah basis data yang baik adalah sebagai berikut [10] :

1. Tidak adanya redundansi dan inkonsistensi data

   Redundasi terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya, ada data mahasiswa yang memuat NIM, nama, alamat, dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah, dan nilai. Pada kedua data tersebut kita temukan ada atribut nama. Jika terjadi keadaan seperti ini, informasi nama seorang mahasiswa disimpan di beberapa tempat. Maka data tersebut masih mengandung redundansi yang harus dihindari pada perancangan basis data. Akibat dari redundansi adalah inkonsistensi data atau data yang tidak konsisten. Redundansi juga mungkin terjadi pada tabel yang sama.
2. Kesulitan pengaksesan data

   Basis data bisa dihubungkan dengan program aplikasi sehingga memudahkan pengguna dalam mengakses informasi. Misalnya program

aplikasi untuk kasir yang terhubung dengan basis data. Pengguna cukup menggunakan fasilitas pencarian ataupun laporan yang tersedia pada program aplikasi untuk menapatkan informasi atok, laporan penjualan, dan lain – lain.

Dalam basis data, informasi yang diperoleh dari kumpulan data bisa berupa keseluruhan data, sebagian data, data dengan filter tertentu, data yang terurut, ataupun *data summary*.

3. *Multiple User*

Basis data memungkinkan penggunaan data bersama – sama oleh banyak pengguna pada saat bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak *client*, kita sudah menyediakan akses ke semua pengguna dari komputer *client* ke sumber informasi yaitu basis data. Tentu saja pengaksesan oleh pengguna – pengguna ini disesuaikan dengan hak aksesnya.

2.2.6.2 Manfaat Basis Data

Banyak manfaat yang dapat kita peroleh dengan menggunakan basis data. Manfaat/kelebihan basis data di antaranya adalah [10] :

a. Kecepatan dan kemudahan (*speed*)
b. Kebersamaan pemakai (*sharability*)
c. Pemusatan kontrol data
d. Efisiensi ruang penyimpanan (*space*)
e. Keakuratan (a*ccuracy*)
f. Ketersediaan (*availability*)
g. Kelengkapan (*completeness*)
h. Keamanan (*security*)
i. Kemudahan dalam pembuatan program aplikasi baru
j. Pemakaian secara langsung
k. Kebebasan Data (*data independence*)

l. User view

2.2.6.3 Sistem Basis Data

Sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data (SMBD). Komponen – komponen sistem basis data meliputi [10] :

1. Perangkat keras (*hardware*) sebagai pendukung operasi pengolahan data
2. Sistem operasi (*operating system*) atau perangkat lunak untuk mengelola basis data
3. Basis data (*database*) sebagai inti dari sistem basis data
4. *Database Management System* (DBMS) adalah *software* yang menangani semua akses ke basis data. Contoh dari DBMS antara lain Microsoft SQL Server 2000, Oracle, MySQL, Interbase, Paradox, Microsoft Access, dan lain – lain.
5. Pemakai (*user*) merupakan orang atau sistem yang akan mengakses dan merubah isi basis data.
6. Aplikasi lain yaitu *software* yang dibuat untuk memberikan *interface* kepada *user* sehingga lebih mudah dan terkontrol dalam mengakses basis data. Aplikasi ini merupakan komponen tambahan dalam sistem basis data yang sifatnya opsional.

2.2.7 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah sebuah konsep pemrograman untuk membuat kode program yang lebih terstruktur, terkelompok berdasarkan objek – objek yang terlibat sehingga bagian – bagiannya dapat digunakan untuk pembuatan aplikasi lain. Pemrograman berorientasi objek membagi – bagi kode program aplikasi menjadi kumpulan bungkusan benda / objek dipandang dari sudut pandang aplikasi komputer dan proses yang dilakukan di dalam aplikasi.

Sebuah bahasa pemrograman dinyatakan mendukung pemrograman berorientasi objek jika bahasa itu mendukung syarat – syarat pemrograman berorientasi objek sebagai berikut [11] :

a. Enkapsulasi (*encapsulation*)

   Mampu membungkus atribut dan metode – metode dalam sebuah kelas dan dapat mencegah pengaksesan langsung atribut dan metode – meode yang ingin dilindungi di dalam sebuah kelas.

b. Pewarisan (*inheritance*)

   Memungkinkan adanya pendefinisian kelas baru yang memiliki sifat – sifat turunan dari kelas lain.

c. Polimorfisme (*polymorphism*)

   Memungkinkan pembuatan pengaksesan metode – metode dengan nama yang sama namun berbeda parameter masukan atau berbeda kelas.

2.2.8 Unified Modeling Language (UML)

Pada bagian ini akan dijelaskan pengertian dari *Unified Modelling Language* (UML) dan jenis – jenis diagram UML

2.2.8.1 Pengertian UML

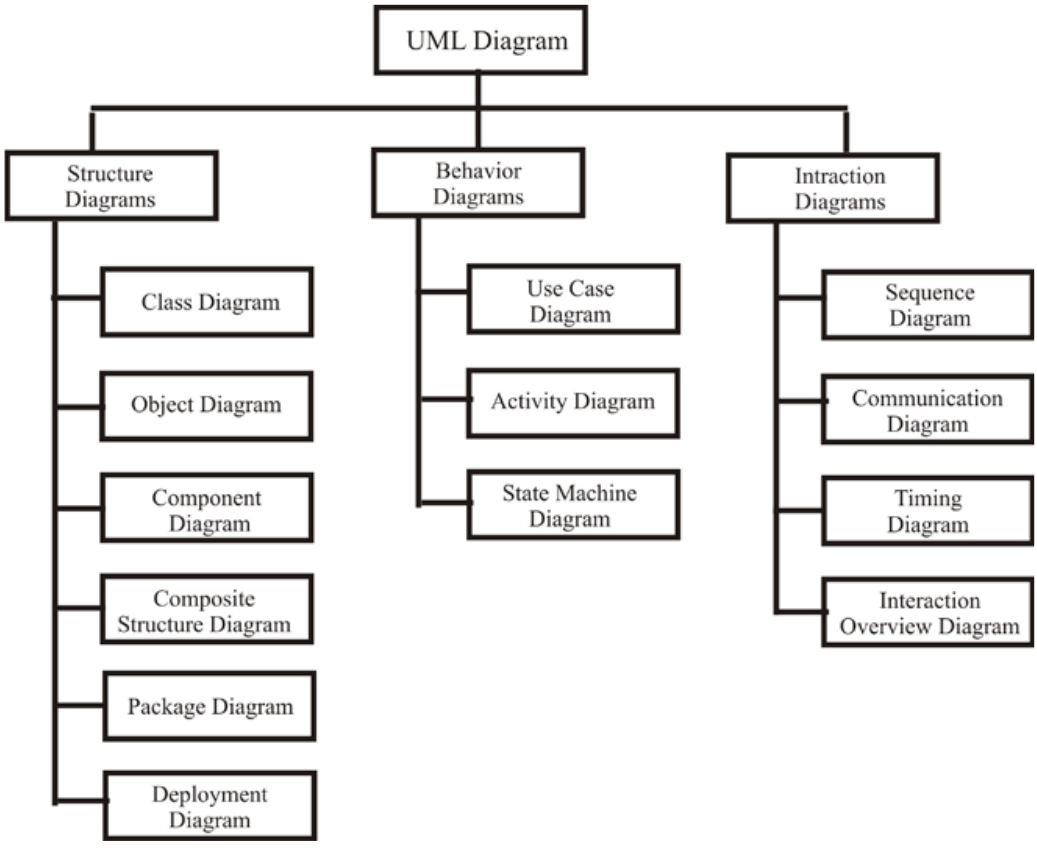
Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tools* untuk mendukung pengembangan sistem tersebut.

UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk

menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi.

UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untukmemahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.2.8.2 Diagram UML



Gambar 2.2 Diagram UML [12]

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagaram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

Berikut ini penjelasan dari masing-masing diagram :

1. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi.

2. Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

3. Component Diagram

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

4. Composite Structure Diagram

Composite structure diagram baru mulai ada pada UML versi 2.0. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang terhubung.

5. *Package Diagram*

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML.

6. *Deployment Diagram*

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut: sistem tambahan yang menggambarkan rancangan, *device*, *node*, dan *hardware*, sistem *client*/*server*, sistem terdistribusi murni, serta rekayasa ulang aplikasi.

7. *Use Case Diagram*

Use case diagram 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. Ada dua hal utama pada *use case* yaitu pendefinisian aktor (orang, proses, atau sistem lain yang berinterkasi dengan sistem informasi yang akan dibuat) dan *use case* (fungsionalitas sebagai unit-unit yang saling bertukar pesan antarunit dan aktor).

8. *Activity Diagram*

Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

9. *State Machine Diagram*

State machine diagram digunakan untuk menggambarkan perubahan status atau transisi dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu graf berarah.

10. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan

diterima objek. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri.

11. Communication Diagram

Communication diagram menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

12. Timing Diagram

Timing diagram merupakan diagaram yang fokus pada penggambaran terkait batasan waktu. *Timing diagram* digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu.

13. Interaction Overview Diagram

Interaction overview diagram adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi. Interaksi diagram dapat meliputi diagram sekuen, diagram komunikasi, dan *timing diagram.*

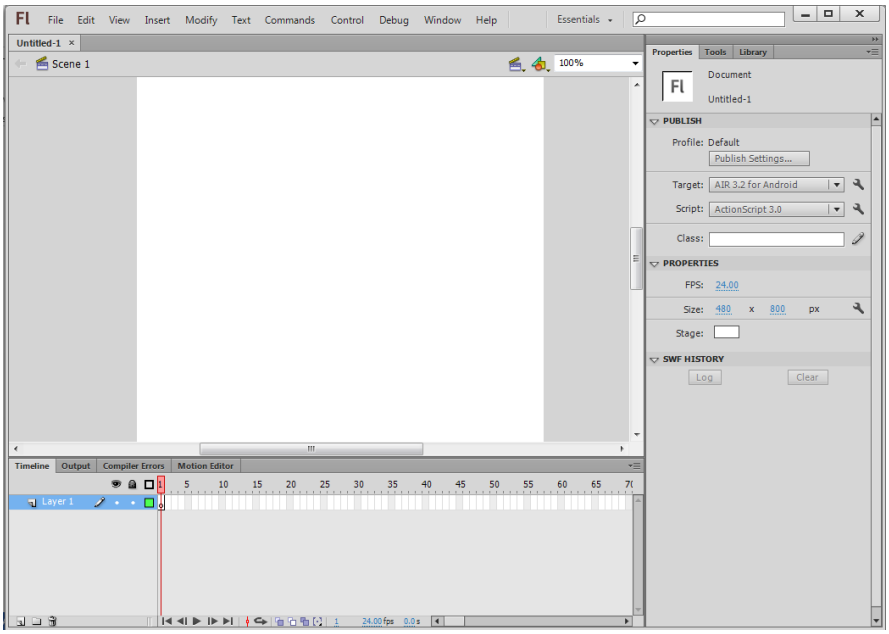
2.2.9 *Tools* yang digunakan

Tools yang digunakan dalam pembangunan aplikasi *mobile learning* interaktif materi geografi pada mata pelajaran ilmu pengetahuan sosial kelas VII berbasis android ini diantaranya adalah Adobe Flash Professional CS6, pemrograman ActionScript 3.0, dan *database* SQLite.

2.2.9.1 Adobe Flash Professional CS6

Adobe Flash Professional adalah salah satu program unggulan dari Adobe Systems yang khusus digunakan untuk membuat animasi gambar vektor. Animasi yang dihasilkan menggunakan file extension.swf dan dapat dijalankan di *web browser* yang telah ter-*install* Adobe Flash Player. Flash telah menjadi standar program pembuatan animasi seperti membangun *website*, membuat *movie*, animasi logo, *game*, *banner*, menu *interaktif*, *e-*

card sampai pengembangan aplikasi untuk *mobile*, *smartphone*, dan komputer tablet [13].



Gambar 2.3 Tampilan Halaman Utama Adobe Flash CS6

Berikut ini adalah beberapa komponen kerja dari Adobe Flash Professional CS6 :

a) *Toolbox* adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membantu suatu desain animasi mulai dari tombol seleksi, *pen*, pensil, *text*, *3D rotation*, dan lain-lain.

b) *Timeline* berguna untuk menentukan durasi animasi, jumlah *layer*, *frame*, menempatkan *script* dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang dibuat akan diatur dan ditempatkan pada *layer* dalam *timeline*.

c) *Stage* adalah lembar kerja yang digunakan untuk membuat atau mendesain objek yang akan dianimasikan.

d) *Panel properties* berguna untuk menampilkan parameter dari sebuah tombol yang terpilih sehingga dapat memodifikasi dan memaksimalkan fungsi dari tombol tersebut.

e) Efek *filter* adalah bagian dari panel propertis yang menampilkan berbagai jenis efek *filter* yang dpat digunakan untuk mempercantik tampilan objek.
f) *Motion editor* berguna untuk melakukan kontrol animasi yang telah dibuat.
g) *Panel motion* presets menyimpan format animasi yang telah jadi dan siap digunakan sewaktu-waktu jika diperlukan.

2.2.9.2 ActionScript 3.0

ActionScript adalah bahasa pemrograman Flash yang berguna untuk membuat animasi interaktif seperti mengontrol *movie,* objek, tombol, dan video. Pada ActionScript 3.0, kode tidak dapat ditulis pada objek tapi hanya dapat ditulis pada Frame. ActionScript akan ditulis dalam sebuah panel yaitu Actions Panel, untuk mengaktifkan panel ini tekan F9 [13] .

2.2.9.3 SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Richard Hipp.

Tidak seperti pada paradigma *client-server* umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead, latency times,* dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.