BAB II

LANDASAN TEORI

2.1. Konsep Dasar Aplikasi

A. Aplikasi

Aplikasi berasal dari bahasa Inggris yaitu *application* yang artinya penerapan, lamaran atau pengguna. Menurut Hendrayudi (2009:143) menerangkan bahwa "aplikasi adalah kumpulan perintah yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)".

Jadi aplikasi secara umum dapat didefinisikan sebagai program yang dapat digunakan untuk menjalankan perintah dari pengguna dengan tujuan mendapatkan hasil yang lebih akurat.

B. *Android*

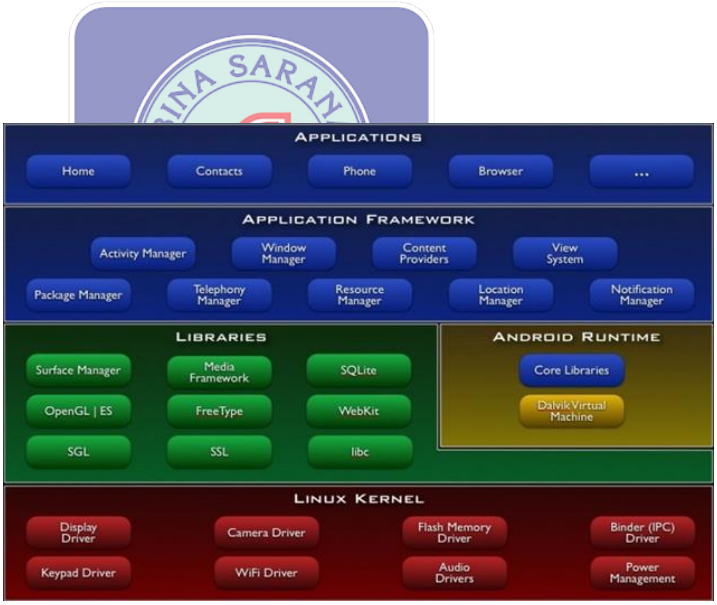
Menurut Hermawan (2011:1) mengemukakan bahwa " *Android* merupakan *OS Mobile* yang berkembang dewasa ini *Android* menawarkan sebuah lingkungan yang berbeda untuk pengembangan". Setiap aplikasi memiliki tingkatan yang sama. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga.

Fitur yang tersedia pada *Android* Adalah:

1. *Framework aplikasi* : memungkinkan pengguna dan pemindahan dari komponen yang tersedia
2. *Dalvik virtual machine* : *virtual machine* yang dioptimalkan untuk perangkat *mobile.*

3. Grafik : grafik 2D dan grafik 3D yang disarankan pada *library OpenGL*.
4. SQLite : untuk penyimpanan data.
5. Mendukung media : audio, video, dan berbagai format gambar (MPEG4, H.26, MP3, AAC, AMR, JPG, PNG, GIF).
6. DSM, *Bluetooth*, EDGE, 3D, and WiFi (tergantung *hadware*)
7. *Camera*, *Global Positioning System (GPS)*, *compass*, dan *accelerometer* (tergantung *hadware*)
8. Lingkungan pengembangan yang kaya, termasuk emulator, peralatan, *debugging*,dan *pligin* untuk *Eclipse* IDE.

Sistem oprasi *Android* dibangun berdasarkan kernel Linuk dan memiliki arsitektur.



Sumber : Setphanus Hermawan S (2011:6)

Gambar II.1.

Arsitektur *Android*

Penjelasan Gambar

1. *Linux Kernel*

Android bergantung pada *Linuk versi* 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack* dan model *driver*. *Kernel* juga bertindak sebagai lapisan antara *hardware* dan seluruh *software*.

2. *Libraries*

Layar di mana fitur-fitur *android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan Aplikasinya. Satu *set libraries* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem *Android*.

3. Android *Run Time*

Satu *set libraries* inti yang menyediakn sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemograman *Java*. Setiap aplikasi akan berjalan sebagai proses sendiri pada *Dalvik Virtual Machine* (VM).

4. *Application Framework*

Pemgembangan aplikasi memiliki akses penuh ke *Android* sama dengan aplikasi inti yang telah tersedia. Pengembangan dapat dengan mudah mengakses informasi lokal, mengatur alarm, menambahkan pemberitahuan ke situs *bar* dan lain sebagainya. Arsitektur aplikasi ini dirancang untuk menyederhanakan kembali komponen, aplikasi apa pun dapat mempublikasikan kemampuan mereka sesuai batasan keamanan. Dasar dari aplikasi adalah seperangkat layanan dan sistem, yaitu berbagai *View* yang digunakan untuk membangun *User Interface* (UI), *Content Provider* yang memungkinkan aplikasi berbagai data *Resource Manager* menyediakan akses bukan kode seperti grafis, *string* dan *layout*,

Notification Manager yang akan membuat aplikasi dapat menampilkan tanda pada status bar dan *Activity Manager* yang berguna mengatur daur hidup dari aplikasi.

5. *Aplications*

Serangkaian aplikasi akan dapat pada perangkat *mobail*. Aplikasi inti yang telah terdapat pada *Android* termasuk kalender, kontak, SMS, dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan bahasa pemrograman *Java*.

1. Daftar versi *Android*
    a. Versi rilis perakomersial (2007-2008)
        1) *AndroidAlpha*
        2) *AndroidBeta*
    b. Versi API *Android* menurut level API (*Application Programing Interface)* (Stephanus Hermawan S,2011)
        1) *Android*versi 1.0
        2) *Android*versi 1.1
        3) *Android*versi 1.5 (*Cupcake*)
        4) *Android*versi 1.6 (*Donut*)
        5) *Android*versi 2.0/2.1 (*Eclair*)
        6) *Android*versi 2.2 (*Froyo:Frozen Yoghurt*)
        7) *Android*versi 2.3 (*Gingerbread*)
        8) *Android*versi 3.0 (*Honeycomb*)
        9) *Android*versi 4.0 (*Ice Cream Sandwitch*)
        10) *Android*versi 4.1 (*Jelly Bean*)
        11) *Android*versi *Kitket*

BINA SARANA INFORMATIKA UNIVERSITAS

12) *Android*versi *Lolipop*

Perbedaan dari *Android versi* 1.0 sampai versi *Lollipop* yaitu tampilan, fitur, dan bentuk. Dari tahun ke tahun tampilan layar dari *Android* terus menerus berubah dari layar kecil sampai menggunakan layar yang besar. Dari fitur *Android* verai 2.3 ke atas lebih baik dibandingkan dengan *Android* versi sebelumnya, pada bentuk *Android* versi 4.0 keatas lebih beragam, dari bentuk yang tipis bahkan tahan terhadap air.

C. Komponen Dasar

Aplikasi *android* ditulis dalam bahasa pemograman *java*. *Java* mengkompilasi kode bersama dengan data *resuarce* dan *file* dibutuhkan oleh aplikasi dimasukan ke dalam *Android, file* arsip ditandai dengan .apk.

Komponen aplikasi pada *android* terdiri dari 4 komponen utama yaitu :

1. *Activities*

   Activities merupakan potongan kode *excutuble* yang menyajikan *user interface* secara visual dimulai oleh pengguna maupun sistem oprasi dan berjalan selama diperukan. *Activities* biasanya sesuai dengan tampilan layar. Masing-masing *activities* menunjukan satu layar untuk pengguna. *Activities* yang tidak aktif dijalankan dapat dimatikan oleh sistem oprasi untuk menghemat memori.

2. *Service*

   Service tidak memiliki visual *user interfaces*, melainkan berjalan di latar belakang untuk waktu yang tidak terbatas. Contoh dari *server* adalah MP3 *player* yang akan memainkan *file* MP3 sesuai urutan *file*, walaupun pengguna menggunakan aplikasi lain.

3. *Broadcast Receiver*

Broadcast Receiver merupakan komponen yang menerima dan beraksi untukmenyiarkan pengumuman. Banyak siaran berasal dalam kode sistem misalnya pengumuman bahwa zona waktu telah berubah, batrai lemah, bahwa gambar telah selesai diambil camera atau bahwa pengguna mengubah preferensi bahasa. Aplikasi juga melakukan siaran misalnya untuk memberikan aplikasi lain mengetehui bahwa beberapadata telah di *download* ke perangkat dan tersedia sehingga dapat digunakan.

4. *Conten Provider*

   Conten Provider diciptakan untuk berbagi data dengan *activities* yang lain atau *service*. Sebuah *conten provider* menggunakan antar muka standar dalam bentuk URI untuk memenuhi permintaan data dari aplikasi lain.

2.2. Peralatan pendukung *(Tools System)*

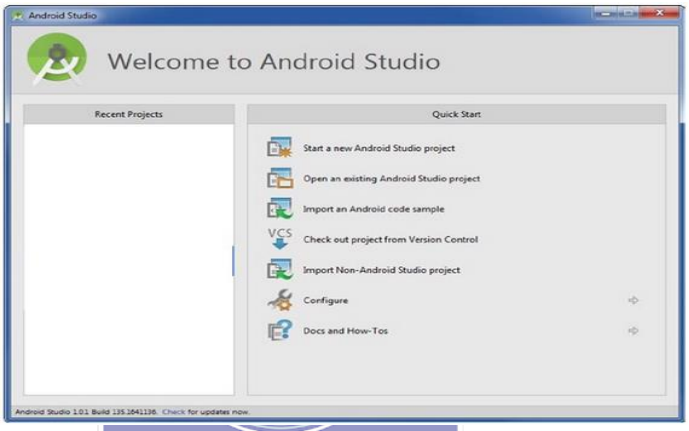
Peralatan pendukung *(Tools System)* adalah alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menggunakan secara tepat arti dan fungsinya. Adapun peralatan pendukung *(Tools System)* yang dijelaskan sebagai model sistem yang dirancang adalah sebagai berikut :

A. *Android Software Development kit (Android SDK)*

Menurut Safaat (2014:5) mengungkapkan bahwa, “Android SDK” merupakan suatu *tools* yang diperlukan untuk mengembangkan aplikasi berbasis Android menggunakan bahasa pemograman *Java”*. Pada saat ini Android SDK telah menjadi alat bantu dan *Application progremming Interface (API)* untuk mengembangkan aplikasi berbasis *Android*.

B. *Android Studio*

Sebuah *Integrated Development* (IDE) untuk *Android Development* yang diperkenalkan *google* pada acara google I/O 20013. *Android Studio* merupakan pengembangan dari *Eclipse* IDE, dan dibuat berdasarkan *Java* populer, yaitu Intellij IDEA. *Android Studio* merupakan IDE resmi untuk mengembangkan aplikasi *Android*.



Sumber : Aplikasi *Android Studio*

Gambar II.2.

Tampilan Awal *Android Studio*

C. *Java*

Menurut Utomo (2013:1) menyimpulkan bahwa: *Java* merupakan salah satu bahasa pemrograman yang bersifat *multiplatform* dengan slogan dari para pengembanganya adalah *'Write once run everywhere'* sehingga aplikasi yang dikembangkan menggunakan bahasa *java* akan dapat diijinkan pada berbagai macam *platform* atau sistem oprasi. Hal ini menjadi salah satu solusi dari berbagai macam bahasa pemrograman yang ada di dunia IT saat ini, yang biasanya hanya dapat dijadikan pada suatu sistem operasi saja dan tidak dapat dijadikan di sistem operasi yang lain.

Bahasa *java* memberi harapan menjadi perekat *universal* yang mengkoneksi pemakai dengan informasi dari *web server*, basis data, penyedia informasi, dan sumber-sumber lain. Ada dua pengertian dari *java* yaitu:

1. Sebagai bahasa pemrograman

   Java merupakan bahasa pemrograman berorientasi objek yang sintaknya mengikuti bentuk C dan C++ sehingga bagi para *progremmer* bahasa C tidak akan kesulitan ketika akan bermigrasi ke bahasa *java* karena sintaknya hampir sama.

2. Sebagai *platform* yang menjalankan program aplikasi lain yang dibangun menggunakan bahasa *java.*

   Pada *platform* lain memerlukan proses secara bentuk fisik dan sistem oprasi, misal prosesor intel dengan sistem oprasi Windows 7. Berbeda dengan *java* ketika berfungsi sebagai *platform* terdiri dari sebuah mesin *virtual* dan media untuk melakukan eksekusi.

BINA SARANA BSI INFORMATIKA UNIVERSITAS

D. *SQLite Database*

Menurut Hermawan (2011:119) mengemukakan bahwa "*SQLite Database* merupakan *interface* yang ada pada sistem oprasi Android yang digunakan untuk membuat *relationaldatabase". SQLite* menyokong implementasi dari *SQL* yang kaya untuk apa pun yang dibutuhkan oleh aplikasi *mobile*. Seperti aplikasi memiliki *database* sendiri dengan pengaturan gkap.

Dengan menggunakan *SQLite*, dapat dibuat *database* untuk aplikasi yang digunakan untuk menyimpan dan mengatur data aplikasi terstruktur. *Database Android* tersimpan di *folder /data/data/<package_name>/database device/emulator*. Secara *default*, semua *database* bersifat *private* yang hanya

dapat diakses oleh aplikasi yang membuatnya. Desain *database* dengan baik termasuk normalisasi cukup penting untuk mengurangi *redundancy*.

Database atau sering juga disebut *basis* data adalah sekumpulan informasi yang disimpan dalam komputer secara sistematik dan merupakan sumber informasi yang dapat diperiksa menggunakan suatu program komputer (Madcoms 2011:12).

E. DB *Browser for SQLite*

DB Browser for SQLite is a high quality, visual, open sourcetool to sreate, desigen, and edit database files compatible with SQLite. It is for users and developers wanting to create database, search and edit data. It is uses a familiar spreadsheet-like interface, and you don't need to learn coplicated SQL command. Control and wizards are available for users to"

1. *Create and compact database file*
2. *Create, define, modify and delete tables*
3. *Create, define and delete indexes*
4. *Browser, edit, add and delete record*
5. *Search record*
6. *Import and export record as text*
7. *Import and export tables form/to CSV files*
8. *Import and export database form/to SQL dump files*
9. *Issue SQL queries and inspect the results*
10. *Examine a log of all SQL commands issued by the application*

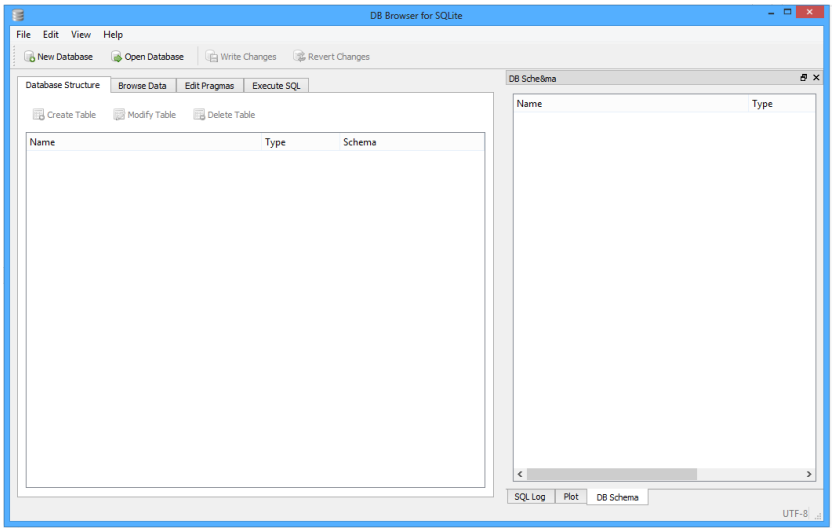
BINA SARANA INFORMATIKA BSI UNIVERSITAS

DB Browser for SQLite adalah sebuah program visual kualitas tinggi dan bersifat *open source* yang bersifat untuk membuat, mendesain dan mengedit *file database* yang bersifat kompatibel dengan *SQLite*.

Program tersebut ditunjukan untuk pengguna dan pengembang dalam membuat database, mencari dan mengedit data. Program *DB Browser for SQLite* menggunakan *interface* atau tampilan seperti halnya *spreadsheet* dan anda tidak Beberapa fungsi atau fitur yang tersedia untuk pengguna diantaranya:

1. Membuat dan *compact file database*
2. Membuat, mendefinisikan dan menghapus tabel
3. Membuat, mendefinisikan dan menghapus *indeks*
4. Mengisi, mengedit, menambah, dan menghapus *record*
5. Mencari *record*
6. Import dan ekspor *record* sebagai *teks*
7. Import dan ekspor tabel dari atau ke *file CSV*
8. Import dan ekspor *database* dari atau ke *file* dump *SQL*
9. Mengeluarkan pertanyaan *SQL* dan memberikan hasil
10. Memberikan *log* dari semua perintah *SQL* yang dikeluarkan oleh aplikasi

BINA SARANA INFORMATIKA UNIVERSITAS



Gambar II.3.
Tampilan *DB Browser for SQLite*

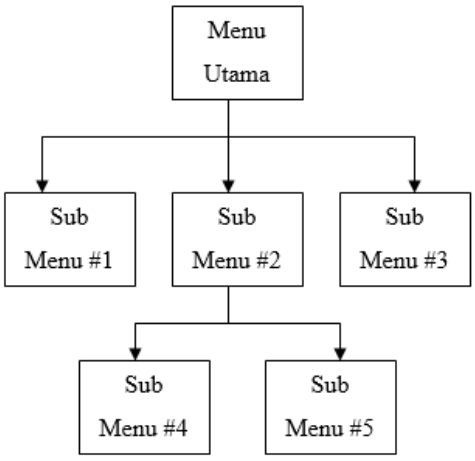
F. HIPO *(Hierarchy Plus Input-Process-Output)*

Menurut Praptiningsih (2010:3) mengemukakan bahwa "HIPO *(Hierarchy Plus Input-Process-Output)* merupakan alat bantu yang digunakan untuk membuat spesifikasi program yang merupakan struktur yang berisi diagram dimana program ini berisi *input* yang diproses dan menghasilkan *output*".

HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program. Penggunaan HIPO ini mempunyai sasaran utama sebagai berikut :

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari program.
2. Untuk menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukan *statemen-statemen* program yang digunakan untuk melaksanakan fungsi tersebut.

3. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan pemakai.



Sumber: http://elib.unikom.ac.id/

Gambar II.4.
HIPO (*Hierarchy Plus Input-Process-Output*)

UNIVERSITAS

G. *Flowchart*

Menurut Sitorus (2015:14) menerangkan bahwa *"Flowchart* mengembangkan aturan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol tertentu".

Flowchart ini akan menunjukkan alur didalam program secara logika. *Flowchart* ini selain dibutuhkan sebagai alat komunikasi, juga diperlukan sebagai dokumentasi. Aturan-aturan dalam perancangan *flowchart* tersebut yaitu :

1. *Flowchart* digambarkan dengan orientasi dari atas ke bawah dan dari kiri ke kanan.
2. Setiap kegiatan/proses dalam *flowchart* harus ditanyakan secara eksplisit.
3. Seriap *flowchart* harus dimulai dari satu *start state* dan berakhir pada suatu atau lebih terminal akhir atau terminator.
4. Gunakan *connector* dan *off-page connector state* dan label yang sama untuk menunjukan keterhubungan antar path algoritma yang terputus, misalnya sebagai akibat pindah atau ganti halaman.

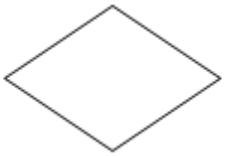
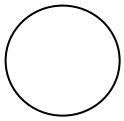
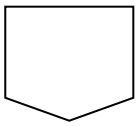
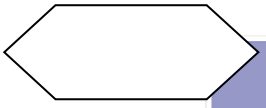
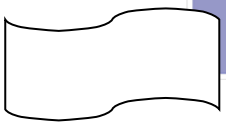
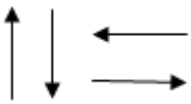
Tujuan dari *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah sederhana, terurai, rapi dan jelas menggunakan simbol-simbol yang standar.

Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang dipakai antara lain :

BINA SARANA BSI INFORMATIKA UNIVERSITAS

Tabel II.1. Simbol Bagian Aliran Sistem *(System Flowchart)*

No	Simbol	Nama	Fungsi
1		*Terminal*	Menyatakan permulaan atau akhir suatu program.
2		*Input* atau *Output*	Menyatakan proses *input* atau *output* tanpa tergantung jenis peralatannya.
3		*Process*	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.

4		*Decision*	Menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya atau tidak.
5		*Connector*	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		*Offline Connector*	Menyatakan sambungan dari proses lainnya dalam halaman yang berbeda.
7		*Predefined Process*	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8		*Subroutine*	Simbol yang menyatakan bagian dariprogram (*subprogram*)
9		*Punch Tape*	Menyatakan *input* atau *output* yang menggunakan pita kertas berlubang.
10		*Document*	Mencetak keluaran dalam bentuk dokumen (melalui printer).
11		*Flow*	Menyatakan jalannya arus suatu proses.

Sumber: Ewolf Community (2012:17)

BINA SARANA INFORMATIKA UNIVERSITAS

H. *Adobe photoshop CS6*

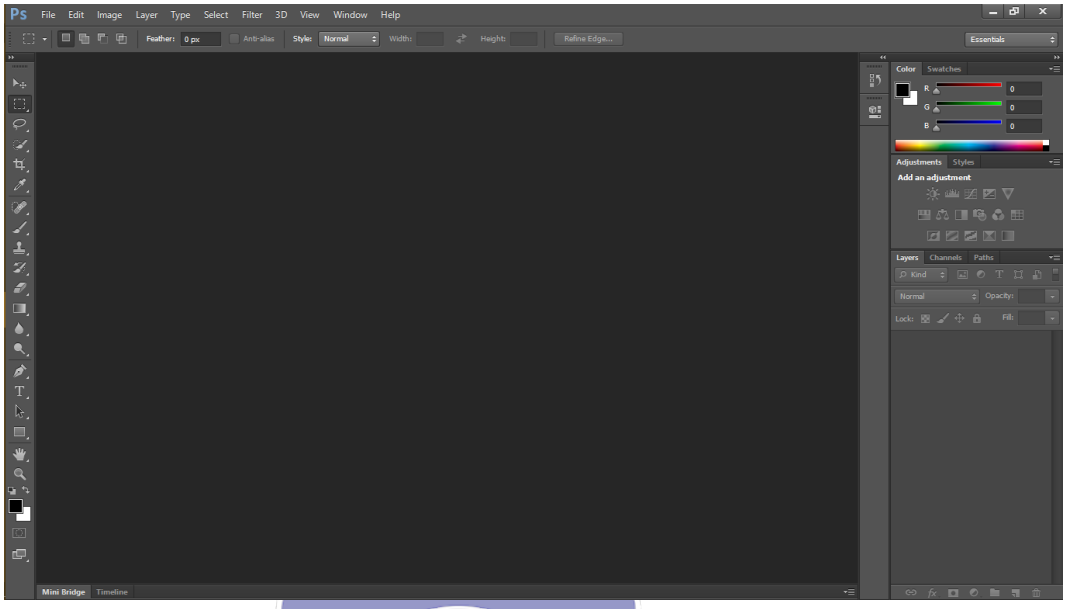
Adobe photoshop CS6 digunakan untuk mengedit tulisan, gambar atau grafik agar tampilan *website* lebih menarik. Menurut MADCOMS (2012:2) mendefinisikan bahwa "*Adobe Photoshop CS6* merupakan sebuah program yang sangat terkenal dikalangan para desainer grafis dan fotografer". Karena kecanggihan dan fasilitasnya yang lengkap, maka *Adobe Photoshop CS6* menjadi pilihan pertama untuk memanipulasi gambar atau foto menjadi sebuah hasil karya yang indah dan menakjubkan.

Berikut ini adalah beberapa fitur dan fasilitas baru yang terdapat dalam *Adobe photoshop CS6*:

1. Tampilan *Interface Adobe Photoshop CS6* yang sekarang berwarna hitam.
2. *Panel Layer* yang sekarang memiliki tambahan fitur dan lebih canggih.
3. *Perspectife Ceopp Tool* yang melengkapi fasilitas *cropping* di dalam *Adobe Photoshop CS6*
4. Hadirnya *Content-Aware Move Tool* yang semakin memudahkan kita dalam manipulasi gambar atau foto.
5. Mengatur gambar menjadi lebih mudah dengan hadirnya perintah *Export/impor Present* dan *Migrate presents* di dalam menu *edit*.
6. Hadirnya fasilitas *Color lookup* di dalam menu *Adjustment* yang menambah variasi dalam pengaturan warna.
7. Sistem *Cropping* yang berbeda dari versi-versi sebelumnya.
8. Perintah menu *File – Automate – PDF presentation* yang digunakan untuk menyimpan hasil pekerjaan dalam *PDF document.*
9. Tambahan *Field Blur* dan *Iris Blur* yang menambah variasi efek blur.

BINA SARANA BSI INFORMATIKA UNIVERSITAS

10. Tambahan menu *Type*, yang semakin memudahkan kita dalam desain tipografi.



Sumber (MADCOMS, 2012:3)

Gambar II.5.
Tampilan *Adobe Photoshop CS6*

BINA SARANA INFORMATIKA
UNIVERSITAS