Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

BAB II

LANDASAN TEORI

2.1 Pengertian *Mobile Aplication*

Mobile aplication adalah proses pengembangan aplikasi untuk perangkat gengam seperti PDA, asisten *digital* perusahaan atau telepon. Sedangkan menurut Buyens (2001) aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* adalah sebutan untuk aplikasi yang berjalan pada *mobile device*. Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

Aplikasi *mobile* merupakan peningkatan dari sistem perangkat lunak terpadu yang umumnya ditemukan pada PC *desktop*. Pada awal kemunculannya, aplikasi menyediakan fugsionalitas yang terbatas dan terisolasi seperti permainan, kalkulator atau *mobile web browsing* sehingga aplikasi sangat dihindari karena *multitasking* yang memakan sumber daya *hardware* perangkat *mobile* yang awalnya cukup terbatas.

Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemai oleh pengguna telepon seluler, karena dengan memanfaatkan adanya fitur *game, music player*, sampai *video player* membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun. Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, Design *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari *dekstop* sistem. Perangkat *mobile* memiliki memori yang kecil.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.2 Konsep Dasar Android

2.2.1 Pengertian Android

Android merupakan suatu *software stack* yang terdistribusi *open source*. Terdiri dari sistem operasi, *middleware*, dan *key application*. Sistem oprasi Android didesign untuk perangkat *mobile* dan merupakan turunan dari sistem opeasi berbasis kenel *linux*, diantaranaya kenel, pustaka atau *library*, *framework* dengan penambahan *Virtual Machine* (VM). Dengan kata lain, OS GNU-*Linux*. (Istiyanto, 2013).

OS Android tidak memiliki sistem *native* X-*Windows* seperti pada OS *linux* dan tidak mendukung *library* standar GNU seperti GNU C. Aplikasi yang berjalan pada OS GNU-*linux* tidak akan dapat berjalan pada perangkat Android. Sejumlah fitur yang dimiliki *software stack* Android (Meier,2010) diantaranya adalah:

1. Dukungan *aplication framework*, sejumlah fungsi API tersedia untuk kemudahan pengembangan Android.
2. *Virtual Machine* (VM) sebagai *inperter* aplikasi Android. VM mendukung eksekusi aplikasi secara *multitasking defan instance* (*object*) *virtual machine* berjalan terpisah dan konsumsi *resource processor* dan memori rendah.
3. *Intergrated Browser*, dengan *web browser* berbasis *WebKit* dan dapat diintegrasikan dengan aplikasi lainnya dengan mengaktifkan *intent* dari *browser* tersebut.
4. Optimalisasi tampilan grafis, tersedia sejumlah pustaka untuk fungsi grafis 2D dan 3D berbasis OpenGl.
5. *Database Relasional* SQLite, format *standar query* sama seperti bahasa SQL lainnya.
6. Dukungan berbagai format *audio*, video dan *file* gambar seperti MPEG4, H.264, MP3, OGG, AAC, AMR, JPG, PNG dan GIF.
7. Mendukung komunikasi percakapan telepon atau SMS menggunakan jaringan GSM/CDMA.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau State Islamic University of Sultan Syarif Kasim Riau

8. Fleksibilitas jaringan, mendukung berbagai teknologi *nirkabel* seperti *Bluetooth*, ANT+, WiFi, GSM, CDMA, WCDMA, EDGE, EDVO, HSDPA, HSUPA.
9. Mendukung *Device Native*, Seperti GPS, A-GPS,*Compass, Accelerometer*, *Magnometer*.
10. Dapat diintegrasikan dengan *tool-tool developing* diantaranya *tool emulator*, *tool debugging*, *tool profiling* memori dan *tools plugin* IDE *Eclipse*.

2.2.2 Sejarah Android

Pada mulanya terdapat berbagai macam sistem operasi pada perangkat selular, diantaranya sistem operasi *Symbian, Microsoft Windows Mobile, Mobile Linux, iPhone* dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google ikut berkecimpung didalamnya dengan *platform* Android, yang menjanjikan keterbukaan, keterjangkauan, *open source* dan *framework* berkualitas. Pada tahun 2005, Google mengakuisisi perusahaan Android. Untuk memulai pengembangan *platform* Android. Dimana terlibat dalam pengembangan ini Andy Rubin, Rich Miner, Nick Sears dan Chris White.

Pada pertengahan 2007 sekelompok pemimpin industri bersama sama membentuk aliansi perangkat seluler terbuka, *Open Handset Alliance* (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik dengan produk awalnya adalah *platform* Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur *handset* dan pengembang aplikasi.

OHA berkomitmen untuk membuat Android *open source* dengan lisensi Apache versi 2. Android pertama kali diluncurkan pada 5 November 2007 dan *smartphone* pertama yang sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode menggunakan sistem operasi Android dikeluarkan oleh T-*Mobile* dengan sebutan G1 pada bulan September 2008. Hingga saat ini Android telah merilis beberapa versi Android untuk

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

menyempurnakan versi nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, versi 2.0 dirilis pada 26 Oktober 2009 diberi nama *Éclair*. Versi 2.2 dirilis pada 6 Desember 2010 dan diberi nama *Frozen Youghurt* (*Froyo*), versi 3.0 dirilis pada bulan Februari 2011 yang diberi nama *Gingerbread* tetapi pada versi ini dirancang khusus hanya untuk tablet dan versi yang terakhir adalah 4.0 yang akan diluncurkan akhir tahun 2011 (Telaumbanua, 2013).

2.2.3 Arsitektur Android

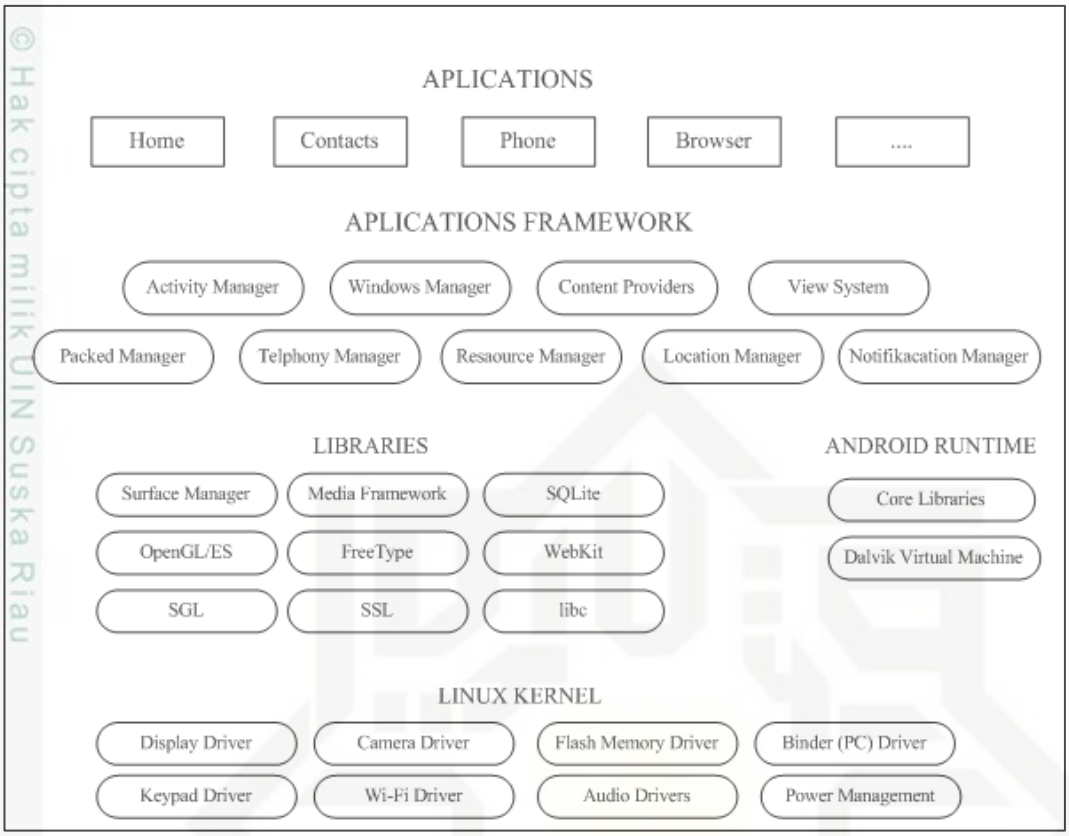
Pada perangkat *Android* terdapat sejumlah *image* yang serupa dalam sistem operasi GNU-*Linux* dengan penambahaan beberapa partisi *image* disesuaikan fitur dan spesifikasi masing-masing perangkat Android diantaranya adalah:

1. *Image Bootloader* untuk fungsi inisiasi *loading boot image* proses *start up* perangkat.
2. *Image Boot* ini sebagai media penyimpanan karnel *linux* dan RAM *Disk*.
3. *Image System* berisikan sistem operasi yang berbasis UNIX dan sejumlah aplikasi *native*.
4. *Image Data* untuk penyimpanan data dilevel *user*.
5. *Recovery Image* berisikan sejumlah *file* untuk proses *recovery* yang akan mereset konfigurasi sistem operasi ke *state default factory*.
6. *Radio Image* yang berisikan sejumlah *file* untuk *lavel stack* radio *image-image* tersebut tersimpan pada *flash memory non-volatile*, yang bersifat *read only* (ROM).

Berikut adalah Gambar 2.1 yang menjelaskan arsitektur Android:

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau



Gambar 2.1 Arsitektur *Android*

Sumber: (Istiyanto, 2013)

Dari Gambar 2.2 secara umum arsitektur Android terdiri dari:

1. *Aplications*

   Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi ini yang telah terdapat pada Android termasuk kalender, kontak, SMS dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan bahasa pemograman Java.

2. *Aplication Framework*

   Pengembangan aplikasi memilik akses penuh ke Android sama dengan aplikasi inti yang telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm menambahkan pemberitahuan ke *status bar* dan lain sebagainya. Arsistektur aplikasi ini dirancang untuk menyederhanakan pengunaan kembali komponen, Aplikasi apa pun dapat mempublikasikan kemempuan dan aplikasi lain dapat menggunakan kemampuan mereka sesuai batasan keamanan.

State Islamic University of Sultan Syarif Kasim Riau

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

3. *Libraries*

   Satu *set libraries* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem Android.

4. *Android Runtime*

   Satu *set libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman Java. Setiap Aplikasi akan berjalan sebagai proses sediri pada *Virtual Machine* (VM).

5. *Linux* Kernel

   Android bergantung pada *linux* versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *snetwork stack* dan modal *driver*. Karnel juga bertindak sebagai antara *hardware* dan seluruh *software*.

2.2.4 Kelebihan Android

Dalam jurnal Telaumbanua (2013) dijelaskan sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya *Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile* (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi ke lebihan Android. Walaupun beberapa itur-fitur yang ada telah muncul sebelumnya pada *platform* lain, Android adalah yang pertama menggabungkan hal seperti berikut:

1. Keterbukaan, bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan *linux* dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royality*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU

lokasi, *database* SQL, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.

4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *flash* menyatu dalam 3D menggunakan OpenGL memungkinkan membuat aplikasi maupun *game* yang berbeda.
6. Portabilitas aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *Virtual Dalvik*, sehingga kode program *portabel* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.3 *Quick Response Code* (QR *Code*)

QR *Code* merupakan teknik yang mengubah data tertulis menjadi kode-kode 2 dimensi yang tercetak kedalam suatu media yang lebih ringkas. QR *Code* adalah *barcode* 2 dimensi yang diperkenalkan pertama kali oleh perusahan Jepang *Denso-Wave* pada tahun 1994. *Barcode* ini pertama kali digunakan untuk pendataan invertaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. QR adalah singkatan dari *Quick Response* karena ditujukan untuk diterjemahkan isinya dengan cepat. QR *Code* merupakan pengembangan dari *barcode* satu dimensi, QR *Code* salah satu tipe dari *barcode* yang dapat dibaca menggunakan kamera *handphone* (Musthofa, 2016).

QR *Code* mampu menyimpan semua jenis data, seperti data angka/numerik, *alphanumeric*, *biner*, kanji/kana. Selain itu QR *Code* memiliki tampilan yang lebih kecil dari *barcode*. Hal ini dikarenakan QR *Code* mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

tampilannya gambar QR *Code* bisa hanya sepersepuluh dari ukuran sebuah *barcode.* Tidak hanya itu QR *Code* juga tahan terhadap kerusakan, sebab QR *Code* mampu memperbaiki kesalahan sampai dengan 30% tergantung dengan ukuran atau versinya. Oleh karena itu, walaupun sebagian simbol QR *Code* kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun (Musthofa, 2016).

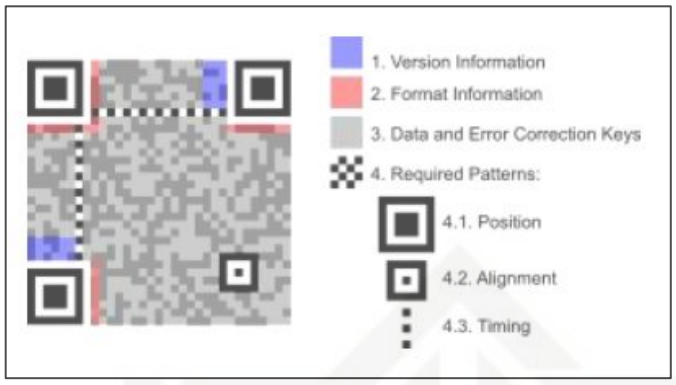
QR *Code* adalah *image* berupa *matriks* dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. QR *Code* merupakan evolusi dari *barcode*. Contoh sebuah QR *Code* dapat dilihat pada Gambar 2.2.

Gambar 2.2 Contoh QR *Code*

Sumber: (Nugraha, 2011)

Tidak seperti *barcode* yang hanya satu sisinya saja yang mengandung data, QR *Code* mempunyai dua sisi yang berisi data dan ini membuat QR *Code* lebih banyak memuat informasi dibandingkan *barcode*. QR *Code* misalnya, dapat menampung informasi berupa *Uniform Resource Locator* (URL) suatu *website* yang nantinya dapat digunakan pada majalah, iklan atau media lainnya. Sehingga ketika seorang pengguna *handphone* berkamera yang mempunyai aplikasi pembaca QR *Code* dapat langsung men-*scan* dan masuk ke *website* yang dimaksud tanpa perlu mengetikkan alamatnya. Kegunaan lain misalnya QR *Code* digunakan untuk menyimpan data teks mengenai informasi produk atau hal lain, SMS atau informasi kontak yang mengandung nama, nomor telepon dan alamat

(Asih, 2011). QR *Code* memiliki bagian-bagian struktur yang akan penulis jelaskan pada Gambar 2.3.



Gambar 2.3 Struktur QR *Code*

Sumber: (Nugraha, 2011)

Berikut ini merupakan penjelasan dari istilah-istilah yang berkenaan dengan Gambar 2.3 sebagai berikut:

1. *Finding Pattern* merupakan pola untuk mendeteksi posisi dari QR *Code*.
2. *Timing pattern* merupakan pola yang digunakan untuk identifikasi koordinat pusat dari QR *Code*, dibuat dalam bentuk modul hitam putih bergantian.
3. *Version Information* merupakan versi dari sebuah QR *Code*, versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (177 x 177) modul.
4. *Quiet Zone* merupakan daerah kosong dibagian terluar QR *Code* yang mempermudah mengenali pengenal QR oleh sensor CCD.
5. QR *Code version* merupakan versi QR *Code*. Pada contoh gambar, versi yang digunakan adalah versi 3 (29 x 29 modul).
6. Data merupakan daerah tempat data tersimpan atau data dikodekan.
7. *Alignment Pattern* merupakan pola yang digunakan untuk memperbaiki penyimpangan QR *Code* terutama distorsi *non linier*.
8. *Format information* merupakan informasi tentang *error correction level* dan *mask pattern*.

Kapasitas data untuk QR *Code* dibandingkan *matriks* kode yang lain dapat dikatakan cukup besar, yaitu dapat menampung 7.089 data numerik, 4.296 data *alphanumerik*, 2.953 data *biner*, atau 1.817 karakter kanji, dengan dukungan

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kecepatan pendekodean dan ukuran cetak yang kecil. Hasil cetakan QR *Code* dikatakan juga tahan terhadap kerusakan sampai dengan 30% agar tetap dapat dibaca. Selain itu, QR *Code* dapat dibaca dari segala arah dengan hasil yang sama sehingga meminimalkan kesalahan baca akibat salah posisi QR *Code* (Kurniati,2011). Berikut adalah Tabel 2.1 yang menjelaskan perbedaan *barcode* dan QR *Code*.

Tabel 2.1 Perbedaan *Barcode* dan QR *Code*

No	*QR Code*	*Barcode*
1	Kapasitas data hingga ratusan data.	Kapasitas data maksimal 20 digit data.
2	Tipe Data yang disimpan beragam mulai dari angka, huruf bahkan sampai huruf Jepang seperti Kanji, Hiragana dan Katakana.	Ukuran cetak *Barcode* minimal adalah 102 piksel.
3	Ukuran cetak untuk QR *Code* dapat jauh lebih kecil karena dapat menyimpan data baik secara horisontal maupun vertikal.	Ukuran cetak *Barcode* minimal adalah 102 piksel.
4	Hasil cetakan di QR *Code* lebih tahan terhadap kerusakan seperti debu sampai robek bahkan data di QR *Code* masih bisa dibaca walaupun sebagian kode sudah rusak atau robek.	Hasil cetakan *Barcode* tidak tahan terhadap kerusakan, jika salah satu bar tertutup atau terputus maka tidak dapat dibaca.
5	QR *Code* dapat dibaca dari segala arah atau sudut (360 derajat) sehingga kemungkinan gagal dalam membaca QR *Code* sangat kecil.	*Barcode* harus berada di depan *scanner* pada saat pembacaan.

2.4 Pariwisata

2.4.1 Pengertian Pariwisata

Berdasarkan Undang-Undang Kepariwisatawan tahun 2009 Nomor 10, wisatawan adalah adalah orang yang melakukan wisata. Sedangkan pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, pemerintah, dan Pemerintah Daerah. Dalam undang-undang tersebut juga disebutkan bahwa daya tarik wisata adalah segala sesuatu yang memiliki keunikan, keindahan, dan nilai

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

yang berupa keanekaragaman kekayaan alam, budaya, dan hasil buatan manusia yang menjadi sasaran atau tujuan kunjungan wisatawan.

Dalam Undang-Undang Kepariwisatawan tahun 2009 nomor 10 telah dijabarkaan kepariwisataan berfungsi memenuhi kebutuhan jasmani, rohani, dan intelektual setiap wisatawan dengan rekreasi dan perjalanan serta meningkatkan pendapatan negara untuk mewujudkan kesejahteraan rakyat dan tujuan dari kepariwisataan adalah untuk meningkatkan pertumbuhan ekonomi, meningkatkan kesejahteraan rakyat, menghapus kemiskinan, mengatasi pengangguran, melestarikan alam, lingkungan dan sumber daya, memajukan kebudayaan, mengangkat citra bangsa, mengangkat citra bangsa dan mempererat persahabatan antar bangsa.

Sedangkan *event* wisata adalah Suatu atraksi yang merupakan sinonim dari pengertian *entertainment*, yaitu sesuatu yang dipersiapkan terlebih dahulu agar dapat dilihat, dinikmati dengan melibatkan orang lain. Namun pada dasarnya obyek wisata dan *event* wisata adalah segala sesuatu yang ada di daerah tujuan wisata yang merupakan daya tarik agar orang-orang mau datang berkunjung ke tempat itu.

2.4.2 Pramuwisata

Pada Undang-Undang Republik Indonesia tentang kepariwisataan Bab IV butir C bahwa pramuwisata termasuk dalam jenis-jenis usaha jasa pariwisata. Hal ini menunjukan bahwa jasa pramuwisata dibutuhkan dalam kegiatan pariwisata. Peranan pramuwisata dalam pariwisata adalah sebagai ujung tombak yang dapat menentukan keberhasilan sebuah pelayanan perjalanan wisata.

Pramuwisata secara umum adalah seseorang yang dibayar untuk menemani wisatawan dalam perjalanan mengunjungi, melihat serta menyaksikan objek dan atraksi wisata. Sedangkan dari sudut pandang wisatawan pramuwisata adalah seseorang yang bekerja pada suatu biro perjalanan atau suatu kantor pariwisata yang bertugas memberikan informasi, petunjuk dan advis secara langsung kepada wisatawan sebelum dan selama perjalanan berlangsung (Mukti, 2011).

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Pengertian pramuwisata menurut Keputusan Menteri Pariwisata, Pos dan Telekomunikasi No: KM.82/PW.102/MPPT-88 tanggal 17 September 1988, yang dimaksud dengan pramuwisata adalah seseorang yang bertugas memberikan bimbingan, penerangan dan petunjuk tentang objek wisata serta membantu segala sesuatu yang diperlukan wisatawan.

Dari beberapa pengertian diatas menunjukan bahwa pramuwisata memiliki tugas yang cukup berat dalam rangka memberikan pelayanan dan informasi kepada wisatawan. Pramuwisata dituntut memiliki wawasan yang luas dan memiliki ketrampilan kerja yang menunjang tugas-tugasnya (Mukti, 2011).

Pramuwisata memiliki tugas yang harus diemban dengan penuh tanggung jawab dan keihklasan. Tugas dan tanggung jawa pramuwisata menurut Menteri Pariwisata, Pos dan Telekomunikasi No: KM.82/PW.102/MPPT-88 tanggal 17 September 1988 adalah sebagai berikut:

1. Mengantar wisatawan, baik rombongan maupun perorangan yang mengadakan perjalanan dengan transportasi yang tersedia.

   Tugas pramuwisata dalam hal ini adalah mengantar wiastwan. Seorang pramuwisata harus menguasai daerah yang sedang dikunjungi baik dalam perjalanan maupun selama di tempat tujuan. Pramuwisata harus mampu memberikan penjelasan hal-hal yang ditemui selama perjalanan berlangsung serta memberikan informasi yang akurat mengenai objek yang dikunjungi.
2. Memberikan penjelasan tentang rencana perjalanan dan objek wisata serta memberikan penjelasan mengenai dokumen perjalanan, akomodasi, transportasi dan fasilitas wisata lainnya. Dalam hal ini pramuwisata harus menjelaskan tentang jadwal perjalanan, tujuan perjalanan, jumlah objek wisata yang akan dikunjungi, memberikan informasi umum mengenai objek yang akan dikunjungi, memberikan informasi tentang dokumen perjalanan yang harus dibawa, informasi akomodasi, memberikan informasi transportasi yang digunakan selama perjalanan dan beberapa informasi mengenai fasilitas yang diberikan sesuai kebutuhan wisatawan.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

3. Memberikan petunjuk tentang objek wisata.

   Seorang pramuwisata harus memberikan informasi tentang objek wisata yang akan dikunjungi berupa informasi tentang bentuk objek wisata, hal-hal yang boleh dan dilarang untuk dilakukan, sehingga wisatawan nyaman dan tidak terjadi pelanggaran terhadap aturan yang telan diinformasikan sebelumnya.

4. Membantu pengurusan barang bawaan wisatawan.

   Dalam tugasnya pramuwisata tidak hanya memberikan informasi tetapi juga memberikan semua pelayanan yang dibutuhkan oleh wisatawan seperti menguruskan barang bawaan, dalam hal ini apabila wisatawan mengalami kesulitan dengan barang bawaannya.

Berdasarakan keahliannya pramuwisata dibedakan menjadi beberapa jenis diantaranya yaitu:

1. Pramuwisata Umum

   Pramuwisata yang mempunyai pengetahuan mengenai kebudayaan, kekayaan alam dan aspirasi kehidupan bangsa atau penduduk sesuatu wilayah secara umum yang mempunyai ijin untuk memberikan bimbingan perjalanan dan penerangan kepariwisataan dengan mempergunakan satu atau beberapa bahasa tertentu terhadap wisatawan baik sebagai perseorangan atau secara berkelompok.

2. Pramuwisata Khusus

   Pramuwisata yang mempunyai pengetahuan mengenai objek wisata secara khusus mendalam mengenai satu atau lebih objek wisata seperti kebudayaan, arkeologi, sejarah teknik perdanganan, keagamaan, ilmiah margasatwa, perburuan, dan lain selanjutnya yang mempunyai ijin usaha atau kegiatan untuk membimbing perjalanan dan memberikan penerangan kepada wisatawan baik sebagai perseorangan atau sebagai kelompok dengan menggunakan satu atau beberapa bahasa tertentu.

3. Pembimbing Darmawisata

   Pramuwisata senior, yang mempunyai kartu tanda pramuwisata untuk memimpin perjalanan satu kelompok wisatawan yang melakukan

perjalanan di suatu wilayah atau beberapa Negara guna memberikan asistensi perjalanan, bimbingan dan penerangan mengenai objek wisata, kebudayaan, kekayaan alam dan aspirasi kehidupan dari penduduk/bangsa wilayah yang dijelajahnya.

4. Pramuwisata Pengemudi

   Pramuwisata yang mempunyai kartu tanda pramuwisata untuk memberikan bimbingan perjalanan dan penerangan umum mengenai objek wisata, kebudayaan, kekayaan alam dan aspirasi kehidupan bangsa di samping kedudukannya sebagai pengemudi kendaraan umum seperti *taxi*, *touring coach*, bus dan sebagainya.

Sedangkan dalam hubungan kerja pramuwisata dibedakan sebagai berikut:

1. Pramuwisata *Tour* Operator

   Pramuwisata yang bertugas mengatur, memimpin dan menyelenggarakan perjalanan wisata termasuk perlengkapan perjalanan, baik di dalam maupun ke luar negeri atas nama perusahaan perjalanan tempat ia berkerja.

2. Pramuwisata Objek Wisata

   Pramuwisata yang memberikan penerangan, menunjukkan dan bimbingan ke suatu atau lebih objek wisata.

3. Pramuwisata Bebas

   Pramuwisata yang tidak terikat dalam kontrak kerja dengan suatu perusahaan biro perjalanan tetapi hanya dibesarkan untuk beroperasi melakukan kegiatan pembimbingan wisatawan dan memberikan penerangan kepada wisatawan bila mana ia disewa atau bertindak untuk salah satu tour operator atau *travel agent* dan mempunyai ijin usaha dan kegiatan untuk memangku tugas pramuwisata.

2.5 Konsep Metodelogi Berorientasi Objek

2.5.1 Pengertian Metodelogi Beorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorintasi objek, pemograman berorientasi objek dan pengujian berorientasi objek.

Menurut Sholiq (2006) metode berorientasi objek atau *object oriented* merupakan paradigma baru dalam rekayasa perangkat lunak yang memandang sistem sebagai kumpulan objek-objek diskrit yang saling berinteraksi. Yang dimaksud dengan berorientasi objek adalah bahwa mengorganisasikan perangkat lunak sebagai kumpulan objek-objek diskrit yang bekerja sama antara informasi atau struktur data dan perilaku yang mengaturnya.

2.5.2 Karakteristik Metodelogi Berorientasi Objek

Karakteristik atau sifat-sifat yang dipunyai sebuah sistem berorientasi objek adalah:

1. Abstraksi

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

2. *Enkapsulasi*

   Pembungkusan atribut data dan layanan yang dipunyai objek, untuk membunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.

3. Pewarisan

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dai objek lain sebagai bagian dari dirinya.

4. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

5. Generalisasi dan Spesialisasi

   Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

6. Komunikasi Antar Objek

   Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

7. *Polymorphism*

   Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.5.3 Keunggulan Metodelogi Berorientasi Objek

Pengembangan sistem dengan metode berorientasi objek dapat memberikan keunggulan-keunggulan sebagai berikut:

1. Meningkatkan Produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

2. Kecepatan Pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan Pemeliharaan

   Karena dengan objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dari pola-pola yang mungkin sering berubah.

4. Adanya Konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.6 Perancangan Sistem

2.6.1 Pengertian Perancangan UML

Unified Modelling Language (UML) adalah sebuah bahasa yang berdasarkan grafik untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *object-oriented*. UML juga merupakan sistem notasi yang membantu pemodelan sistem menggunakan konsep berorientasi objek.

UML adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain (Nugroho, 2005).

UML dibangun atas model 4+1 *view*. Model ini didasarkan pada fakta struktur sebuah sistem dideskripsikan dalam 5 *view*, dimana salah satunya adalah *Use Case View*. Dalam proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Dengan model kita dapat merepresentasikan sesuatu karena :

1. Model mudah dan cepat untuk dibuat.
2. Model bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu.
3. Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu.
4. Model bisa mewakili sesuatu yang nyata maupun tidak nyata.

2.6.2 Diagram-Diagram Dalam UML

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML mempunyai aturan untuk menggabungkan dan mengkombinasikan elemen-elemen tersebut. Dalam membangun suatu model perangkat lunak dengan UML digunakan bentuk-bentuk *diagram* atau simbol untuk merepresentasikan elemen-elemen dalam sistem.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Bentuk diagram yang digunakan untuk merepresentasikannya adalah sebagai berikut:

1. *Use Case Diagram*

Diagram use case merupakan pemodelan untuk kelakuan (*behavior)* sistem informasi yang akan dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Notasi *diagram use case* dapat dilihat dalam Tabel 2.2.

Tabel 2.2 Notasi *Use Case*

Gambar	Nama	Keterangan
	Actor	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	Include	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
	Extend	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

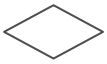
Tabel 2.2 Notasi *Use Case* (Lanjutan)

Gambar	Nama	Keterangan
	Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. *Activity Diagram*

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem sistem. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut adalah Tabel 2.3 notasi-notasi yang digunakan dalam *activity diagram*.

Tabel 2.3 Notasi *Activity Diagram*

Gambar	Nama	Keterangan
	Status awal	Status awal aktivitas *system*, sebuah diagram aktivitas memiliki status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	*Decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir	Status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Maka dari itu untuk menggambarkan *diagram sekuence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* serta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Sequence diagram mendokumentasikan komunikasi/interaksi antar kelas-kelas. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakkan diantara obyek-obyek didalam *use case*. Perlu diingat bahwa di dalam *diagram* ini, kelas-kelas dan aktoraktor diletakkan dibagian atas *diagram* dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertikal terhadap kelas dan *actor*. Berikut notasi *sequence diagram* yang dapat dilihat Tabel 2.4.

Tabel 2.4 Notasi *Sequence Diagram*

Simbol	Nama	Keterangan
	Actor	Orang, proses atau *system* lain yang berinteraksi dengan system informasi dan mendapat manfaat dari system.
	LifeLine	Menyatakan hidup suatu objek
Nama Objek:nama kelas	*Object*	Menyatakan objek yang berinteraksi pesan
	Message	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create <<create>>	*Message*	Objek yang lain, arah panah mengarah pada objek yang di buat

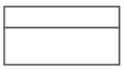
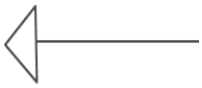
4. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akanm menghasilkan sebuah objek dan merupakan inti dari pengembangan dan *Design* berorientasi objek. *Class* menggambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Berikut adalah notasi pada yang digunakan dalam *class diagram* dapat dilihat dalam Tabel 2.5.

Tabel 2.5 Notasi Pada *Class Diagram*

Simbol	Nama	Keterangan
	*Class*	*Class* adalah blok-blok pembangun pada pemrograman berorientasi obyek.
	Assosiation	Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 *class*
	Composition	Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/*solid*.
	Dependency	Kadangkala sebuah *class* menggunakan *class* yang lain. Penggunaan *dependency* di gunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain.
	Aggregation	*Aggregation* mengindikasikan keseluruhan bagianya *relationship* dan biasanya disebut sebagai relasi "mempunyai sebuah" atau "bagian dari".
	*Generalization*	Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasi obyek.

2.7 Metode Pengembangan Sistem

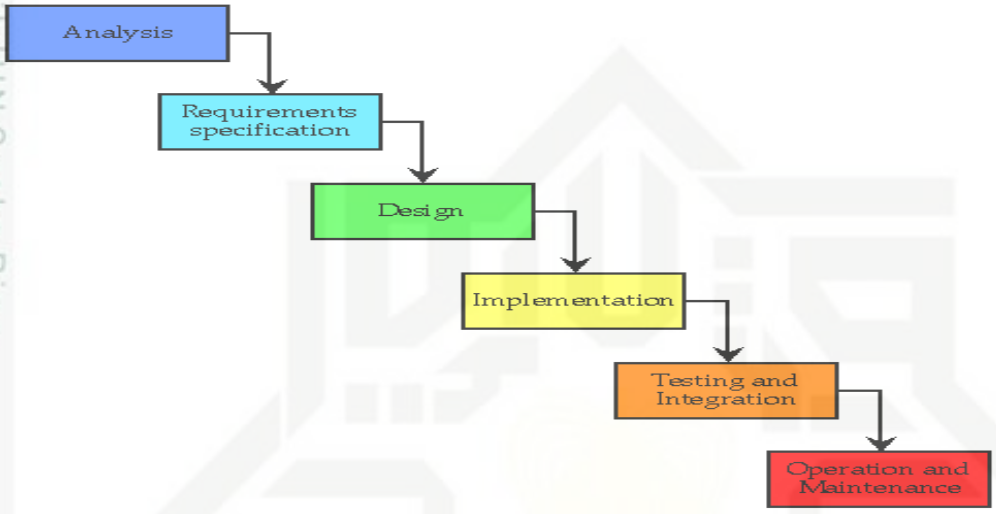
2.7.1 Pengertian Model Pengembangan *Waterfall*

Model *waterfall* adalah model pengembangan perangkat lunak yang paling sering digunakan. Model pengembangan ini bersifat *linear* dari tahap awal pengembangan *system* yaitu tahap perencanaan sampai tahap akhir pengembangan *system* yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya (Andriana, 2014).

Model *Waterfall* adalah merupakan model yang paling populer diantara model pendekatan yang lain didalam model pengembangan sistem dan bekerja

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

secara *linier* dan berurutan di lain pihak model ini merupakan model yang paling awal terbentuk. Pendekatan ini mengikuti suatu paradigma yang dikenal dengan *documentation driver paradigm,* harapanya adalah bahwa disetiap proses pengembangan sistem, setiap langkah yang dilakukan akan dikomentasi atau ditulis dalam bentuk spesifikasi dalam laporan yang detil (Dewanto, 2004).



Gambar 2.4 Ilustrasi Model *Waterfall*

Sumber: (Andriana, 2014)

2.7.2 Proses-Proses Model Pegembangan *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

   Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. *Design*

   Design perangkat lunak adalah proses multi langkah yang fokus pada *design* pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi *design* agar dapat di implementasikan menjadi program pada tahap selanjutnya. *design* perangkat lunak pada tahap ini perlu di dokumentasikan.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

3. Pembuatan Kode Program

   Design harus di translasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan *design* yang telah di buat pada tahap Design.

4. Pengujian

   Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang di inginkan.

5. Pemeliharaan

   Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi denga lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan peerangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.7.3 Kelebihan dan Kekurangan Model *Waterfall*

Pada model *waterfall*, beberapa prinsip utama yaitu *project* dibagi-bagi dalam beberapa *fase* yang saling berurutan, penekanan pada perencanaan, jadwal, *deadline*, *budget*, dan implementasi keseluruhan sistem sekaligus, sekaligus kontrol yang ketat dalam siklus hidup *project* dengan menggunakan bantuan dokumentasi tertulis. Sedangkan kelebihan dari metode *waterfall* meliputi model ini mempunyai kemudahan untuk dimengerti, mudah digunakan, *requirement* dari sistem bersifat stabil, baik dalam manajemen kontrol, serta bekerja dengan baik ketika kualitas lebih diutamakan dibandingkan dengan biaya dan jadwal.

Selain kelebihan-kelebihan di atas, model *waterfall* ini memiliki kekurangan yang sangat banyak. Dikarenakan sangat banyak tim yang telah mengimplementasikan *project*-nya dengan menggunakan model ini. Beberapa

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

diantara kekurangannya itu yakni semua kebutuhan sistem harus diketahui terlebih dahulu, dapat memberikan kesan palsu dari progresnya, tidak menunjukkan menunjukkan prinsif "*Problem Solving*" dalam pengembangan perangkat lunak dikarenakan fase yang harus berurut, integrasi sekaligus di akhir sistem, *customer* hanya memiliki sedikit kesempatan untuk melihat dan me-*review* sistem (yakni di akhir *project*), resiko sangat tinggi, karena *testing* hanya dilakukan pada setiap akhir *fase*, bahkan tidak jarang *testing* hanya dilakukan di akhir-akhir *project*, membutuhkan waktu yang cukup lama (walaupun *project*-nya tidak terlalu besar), perubahan *requirement* dapat merubah keseluruhan proses yang telah dilaksanakan.

2.8 Pengujian Sistem

Pengujian *software* sangat diperlukan untuk memastikan *software* yang sudah dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan *software* seperti halnya analisis, *design*, dan pengkodean (Mustaqbal, 2015).

Pengujian *software* haruslah dilakukan dalam proses rekayasa perangkat lunak atau *software engineering*. Sejumlah strategi pengujian *software* telah diusulkan dalam literatur. Semuanya menyediakan template untuk pengujian bagi pembuat *software*. Dalam hal ini, semuanya harus memiliki karakteristik umum yakni:

1. *Testing* dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasiskan komputer.
2. Teknik *testing* yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. *Testing* diadakan oleh pembuat/pengembang *software* dan untuk proyek yang besar ole group *testing* yang *independent*.

4. *Testing* dan *debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomodasikan pada setiap strategi *testing.*

Terdapat beberapa tipe yang berbeda dalam pengujian *software* yang meliputi studi kelayakan dan simulasi yaitu:

1. Metode *software engineering* menyediakan dasar dari mutu yang mana yang akan dipakai.
2. Metode *analysis, design and construction* berupa tindakan untuk meningkatkan kualitas dengan menyediakan teknik yang seragam dan hasil yang sesuai dengan keinginan.
3. Metode formal *technical reviews* menolong untuk memastikan kualitas kerja produk merupakan hasil konsekuensi dari setiap langkah *software engineering*.
4. Metode *measurement* diberlakukan pada setiap elemen dari konfigurasi *software*.
5. Metode *standards and procedures* membantu untuk memastikan keseragaman dan formalitas dari SQA untuk menguatkan dasar filosofi kualitas total.
6. Metoda *testing* menyediakan cara terakhir dari tingkat kualitas mana yang dapat dicapai dan dengan praktis dapat mengetahui letak *error*.

Langkah-langkah pengujian *software* ada 4 yaitu:

1. *Unit testing-testing* per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian *error* secara maksimum.
2. *Integration testing-testing* penggabungan unit yaitu pengalaman dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan konstruksi program.
3. *High-order test* yaitu terjadi ketika *software* telah selesai diintegrasikan atau dibangun menjadi satu-tidak terpisah-pisah.
4. *Validation test* yaitu menyediakan jaminan akhir bahwa *software* memenuhi semua kebutuhan fungsional, kepribadian dan performa.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.8.1 *Blackbox Testing*

Blackbox testing berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Blackbox testing* bukanlah solusi alternatif dari *whitebox testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *whitebox testing* (Mustaqbal, 2015). *Blackbox testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan terminasi.

Blackbox testing cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *blackbox* memperhatikan struktur kontrol, sehingga perhatian berfokus pada domain informasi. Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *blackbox testing*, antara lain:

1. *Equivalence Partitioning.*
2. *Boundary Value Analysis*/*Limit Testing.*
3. *Comparison Testing.*
4. *Sample Testing.*
5. *Robustness Testing.*
6. *Behavior Testing.*
7. *Requirement Testing.*
8. *Performance Testing.*
9. Uji Ketahanan.
10. Uji Sebab-Akibat.

Pada pengujian *blackbox* pada sistem *backend* yang dilakukan setelah responden melakukan pengujian selanjutnya digunakan Rumus 2.1.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Persentase Kegagalan : $\frac{\text{Jumlah Modul Eror}}{\text{Jumlah Total Modul}}$ x 100% (2.1)

Persentase Keberhasilan : 100-persentase kegagalan

2.8.2 Pengujian *User Acceptance Testing* (UAT)

User Acceptance Testing merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut Lewis, setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *blackbox* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

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). *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 yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna.

2.9 Komponen-Komponen Pendukung

2.9.1 Pengertian *Web*

Web adalah sistem dengan standar yang diterima secara universal untuk menyimpan, menelusuri, memformat, dan menampilkan informasi melalui arsitektur *client*/*server*, *web* bisa menerima semua jenis informasi digital,

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

termasuk teks, hipermedia, grafis dan suara. *Web* menggunakan antarmuka pengguna grafis, sehingga sangat mudah digunakan.

Web didasari oleh bahasa hiperteks standar yang disebut *Hypertext Markup Language* (HTML), yang memformat dokumen dan memadukan *link hiperteks* dinamis ke dokumen-dokumen lainnya yang disimpan di dalam komputer yang sama atau yang berbeda. HTML berasal dari standar *Generalized Markup Language* (SGML) yang lebih rumit, bahasa berbasis teks untuk mendeskripsikan isi dan struktur dari dokumen digital.

HTML adalah *subset* yang lebih sederhana dari SGML dan mencakup tabel, *applet*, aliran teks dalam gambar, superskrip dan *subskrip*. Dengan menggunakan *link* hiperteks ini (yang umumnya berwarna biru, tercetak tebal, dan bergaris bawah, pengguna mengarahkan kursor ke kata yang tersorot (*highlight*), mengetiknya dan kemudian dibawa ke dokumen lain. Pengguna bisa melakukan navigasi di *web* dengan bebas tanpa batasan, dengan mengikuti logika, kebutuhan, dan minat mereka sendiri.

Penawaran informasi melalui *web* memerlukan adanya *home page*, yang merupakan tampilan layar grafis dan teks yang biasanya menyambut pengguna dan menjelaskan organisasi yang membuat alam tersebut. Dalam kebanyakan kasus *home page* akan mengarahkan pengguna ke halaman-halaman lainnya. Semua halaman dari perusahaan atau individu tertentu disebut situs *web*.

Kebanyakan halaman web menyediakan cara menghubungi organisasi atau individu. Orang yang bertanggung jawab atas situs organisasi disebut *webmaster*. Untuk mengakses situs *web* pengguna harus menentukan *Uniform Source Locator* (URL), yang mengarahkan ke alamat dari sumber tertentu di *web. Hypertext Transport Protocol* (HTTP) yang merupakan standar komunikasi yang digunakan untuk mentransfer halaman di bagian WWW di internet. HTTP mendefinisikan bagaimana pesan diformat dan dikirim Serta tindakan apa yang perlu diambil oleh *server web* dan penjelajah dalam merespons berbagai perintah.

Sebagai salah satu jenis dari sistem komputasi klien/*server*, sistem berbasis *web* adalah aplikasi atau layanan yang berada dalam *server* dan dapat diakses dengan menggunakan penjelajah *web* dan karenanya dapat diakses dimana saja

melalui internet maupun intranet. Satu-satunya peranti lunak sisi klien yang dibutuhkan untuk mengakses dan menjalankan aplikasi berbasis *web* adalah lingkungan penjelajah *web*, dan berbagai aplikasi tersebut harus sesuai dengan protokol internatnya.

Perkembangan sistem berbasis *web* yang cepat, serta pengenalan berbagai konsep baru seperti komputasi untuk *utilitas* dan layanan peranti lunak mengubah perusahaan mengatur berbagai sumber daya komputasi mereka. Oleh karena itu, internet, intranet dan kadang ekstranet menjadi bagian yang tidak terpisahkan dalam kebanyakan arsitektur TI. Arsitektur berbasis *web* yang baru mungkin menggantikan berbagai arsitektur lama, atau mengintegrasikan sistem arisan ke dalam strukturnya.

1. Internet

   Internet adalah fasilitas publik yang kooperatif dan dapat berjalan sendiri, yang dapat diakses oleh ratusan dan jutaan orang di seluruh dunia. Secara fisik, internet menggunakan sebagian dari sumber daya total jaringan telekomunikasi publik yang saat ini ada. Secara teknis, yang membedakan internet adalah penggunaan serangkaian protokol, yang disebut TCP/IP.

2. Intranet

   Intranet adalah penggunaan teknologi *web* untuk menciptakan jaringan privat, biasanya dalam sebuah perusahaan. Gerbang keamanan seperti *firewall* digunakan untuk memisahkan intranet dan internet dan untuk secara selektif mengizinkan akses dari luar intranet. Intranet memiliki banyak kegunaan, intranet memungkinkan distribusi *online* secara aman berbagai bentuk informasi internal perusahaan. Intranet digunakan untuk aktivitas *workgroup* dan pendistribusian berbagai proyek secara bersamaan dalam perusahaan.

3. Ekstranet

   Ekstranet menghubungkan beberapa intranet melalui internet, dengan menambahkan ke internet mekanisme keamanan. Ekstranet membentuk jaringan maya yang lebih besar dengan memungkinkan para pengguna

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau State Islamic University of Sultan Syarif Kasim Riau

jarak jauh untuk secara aman terkoneksi melalui internet ke intranet perusahaan.

4. Portal perusahaan

   Portal perusahaan adalah berbagai situs *web* yang menyediakan gerbang ke informasi perusahaan dari sebuah titik akses. Intranet biasanya digabungkan dengan dan diakses melalui portal perusahaan. Portal akan mengumpulkan informasi dari berbagai tempat dan meyajikanya ke pengguna.

2.9.2 *Hypertext Pre Processor* (PHP)

Hypertext Preprocessor (PHP) merupakan bahasa pemrogramman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa.

Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server* (Kurniati, 2011).

Cara kerja PHP yang akan kita bahas pada buku ini adalah PHP sebagai bahasa pemrograman untuk mengembangan aplikasi berbasis web. Karena selain untuk *web programming* PHP juga dapat digunakan untuk mengembangan aplikasi berbasis desktop dan *Command Line Interface* (CLI). Berikut Ini adalah keunggulan PHP yakni:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Dalam sisi pengembangan lebih mudah, karena banyaknya forum dan *developer* yang siap membantu dalam pengembangan.
3. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

4. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.9.3 MySQL

Database merupakan kumpulan data yang setiap perintahnya menggunakan SQL. *Structured Query Langunge* (SQL) merupakan bahasa *American National Standard Institute* (ANSI) yang digunakan untuk melakukan *query* data pada *database*. Hampir semua *software database* mengimplementasikan bahasa ini sebagai komponen utama dari produknya.

My Stucture Query Language (MySQL) merupakan *software* RDBMS *open source*. Pada awalnya MySQL dikembangkan pada platform sistem operasi *linux* namun kemudian dikembangkan untuk penggunaan pada *platform windows*. Sebagai *software database* dengan konsep *database* modern, MySQL memiliki banyak kelebihan antara lain:

1. *Protability*

   MySQL dapat digunakan dengan stabil tanpa kendala, berarti pada berbagai sistem operasi diantaranya seperti *Windows, Linux, Mac OS X Server, Solaris, Amiga HP-UX* dan masih banyak lagi.

2. *Open Source*

   MySQL didistribusikan secara *open source* di bawah lisensi GPL, sehingga dapat memperoleh menggunakannya secara cuma-cuma tanpa dipungut biaya sepeserpun.

3. *Multiuser*

   MySQL dapat digunakan untuk menangani beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini akan memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan dalam waktu yang bersamaan pula.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang cukup menakjubkan dalam menangani *query* sederhana, serta mampu memproses lebih banyak SQL persatuan waktu.

5. *Column Types*

   MySQL didukung tipe kolom (tipe data) yang sangat kompleks.

6. *Command dan Functions*

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

7. *Scalability dan Limits*

   Dalam hal batas kemampuan, MySQL terbukti mampu menangani *database* dalam skala yang besar dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

8. *Interface*

   Sama halnya dengan *software database* lainnya, MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi *Aplication Programming Interface* (API).

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU