

BAB II

LANDASAN TEORI

2.1 Rancang Bangun

Menurut Pressman tahun 2002 menyatakan bahwa, Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Menurut Ladjamudin tahun 2005 menyatakan bahwa, Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. Menurut Pressman tahun 2002 menyatakan bahwa, Bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian. Menurut *Whitten et al* tahun 2004 menyatakan bahwa, Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain.

2.2 Reservation

Menurut Edwin tahun 2000 menyatakan bahwa, Reservasi adalah suatu perjanjian pemesanan tempat antar dua (2) pihak atau lebih. Perjanjian atau pemesanan tersebut dapat berupa perjanjian atas pemesanan suatu ruangan, kamar, tempat duduk dan lainnya pada waktu atau periode tertentu dan disertai dengan produk jasanya.

2.3 Hotel

Menurut Bagyono tahun 2012 menyatakan bahwa, Hotel adalah suatu perusahaan yang dikelola oleh pemiliknya dengan menyediakan pelayanan dengan menyediakan pelayanan makanan, minuman dan fasilitas kamar untuk tidur kepada orang-orang yang sedang melakukan perjalanan dan

mampu membayar dengan jumlah yang wajar sesuai dengan pelayanan yang diterima tanpa adanya perjanjian khusus.

2.4 Aplikasi

Menurut Prayitno tahun 2009 menyatakan bahwa, Aplikasi berasal dari kata *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.

2.5 Android

Menurut Sugeng Purwantoro, Heni Rahmawati & Achmad Thermizi tahun 2013 menyatakan bahwa, *Android* merupakan suatu *software* (perangkat lunak) yang digunakan pada *mobile device* (perangkat berjalan) yang meliputi sistem operasi, *middleware* dan aplikasi inti. Menurut Satyaputra & Aritonang tahun 2014 menyatakan bahwa, *Android* adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara *device* (piranti) dan penggunaannya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*.

Sistem operasi *Android* ini bersifat *open source* sehingga programmer berbondong-bondong untuk membuat aplikasi maupun memodifikasi sistem ini. Para programmer memiliki peluang yang sangat besar untuk terlibat mengembangkan aplikasi *Android* karena alasan *open source* tersebut. Sebagian besar aplikasi yang terdapat dalam *Play Store* bersifat gratis dan ada juga yang berbayar.

2.5.1. Sejarah Android

Menurut Satyaputra & Aritonang tahun 2014 menyatakan bahwa, Perkembangan *Android* dimulai dengan berdirinya *Android Inc.* pada Oktober 2003 dengan tujuan *Mobile Device* yang lebih pintar

untuk menyaingi *Symbian* dan *Windows Mobile* yang populer pada saat itu dimana *iPhone* dan *Blackberry* belum dirilis.

Pada tahun 2005, *Android* diakuisisi oleh Google, pengembangan terus dilanjutkan sampai *Android* versi beta diluncurkan pada tanggal 5 November 2007. Hingga saat ini tanggal 5 November diperingati sebagai hari jadi *Android*. Seminggu setelahnya yaitu pada tanggal 12 November 2007 *Android SDK (software Development Kit)* diluncurkan, sehingga pengguna dapat membuat dan mengembangkan aplikasi-aplikasi *Android* mereka sendiri.

2.5.2. Versi Android

Menurut Nazaruddin tahun 2012 menyatakan bahwa, Terdapat beberapa versi android seperti dibawah ini ialah :

1. Android versi 1.1

Pada 9 maret 2009, Google merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.

2. Android versi 1.5

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan *Android* dan *SDK (software Development Kit)* dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube dan gambar ke *picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik disbanding sebelumnya, penggunaan baterai indikator dan *control applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan pada CDMA.

4. Android versi 2.0/2.1

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Éclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2MP, *Digital Zoom*, dan *Bluetooth* 2.1

5. Android versi 2.2

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portable, dan kemampuan auto update dalam aplikasi Android Market.

6. Android versi 2.3

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini, antara lain peningkatan kemampuan permainan, peningkatan fungsi copy paste, layar antar muka

didesain ulang, dukungan format video VP8 dan WebM, dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis.

8. Android versi 4.0

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

9. Android versi 4.1

Diumumkan pada tanggal 27 Juni 2012, android versi Jelly Bean merupakan peningkatan dari versi sebelumnya, dimana pada versi ini fungsi dan kinerja antarmuka pengguna sudah ditingkatkan menjadi lebih baik lagi sepertiantisipasi sentuh, triple buffering, perpanjangan waktu sync, dan peningkatan frame rate hingga 60fps untuk menciptakan tampilan yang lebih halus.

10. Android versi 4.4

Pada tanggal 3 September 2013 android dengan versi KitKat memiliki beberapa pembaruan antara lain, Pembaruan antarmuka dengan bar status dan navigasi transparan pada layar

depan, Optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah, Kerangka kerja pencetakan, NFC *Host Card Emulation* sebagai emulator kartu pintar, WebViews berbasis *Chromium*, perluasan fungsionalitas bagi layanan pendengar notifikasi, API umum untuk mengembangkan dan mengelola klien pesan teks, kemampuan untuk menentukan aplikasi SMS standar, Kerangka kerja baru untuk transisi UI, Kerangka kerja akses penyimpanan untuk mengambil konten dan dokumen dari sumber lain yang saling terintergrasi.

2.5.3. Komponen Aplikasi Android

Menurut Arif Akbarul Huda tahun 2013 menyatakan bahwa, Komponen aplikasi merupakan bagian penting dari sebuah *Android*. Setiap komponen mempunyai fungsi yang berbeda, dan antara komponen atau dengan yang lainnya bersifat saling berhubungan. Berikut ini komponen aplikasi yang harus diketahui, yaitu :

1. *Activites*. *Activity* merupakan satu halaman antarmuka yang bisa digunakan oleh *user* (pengguna) untuk berinteraksi dengan aplikasi. Biasanya dalam satu *activity* terdapat *button, spinner, list view, edit text*, dan sebagainya. Satu aplikasi dalam *Android* dapat terdiri atas lebih dari satu *Activity*.
2. *Services*. *Services* merupakan komponen aplikasi yang dapat berjalan secara *background*, misalnya digunakan untuk memuat data dari *server database*. Selain itu, aplikasi pemutar music atau radio juga memanfaatkan servis supaya aplikasinya bisa tetap berjalan meskipun pengguna melakukan aktivitas dengan aplikasi lain.
3. *Contact Provider*. Komponen ini digunakan untuk mengelola data sebuah aplikasi, misalnya kontak telepon. Siapapun bisa membuat aplikasi *Android* dan dapat mengakses kontak yang tersimpan pada sistem *Android*. Oleh karena itu, agar dapat

mengakses kontak, *user* (pengguna) memerlukan komponen *contact provider*.

4. *Broadcast Receiver*. Fungsi komponen ini sama seperti bahasa terjemahannya yaitu penerima pesan. Kasus baterai lemah merupakan kasus yang sering dialami *handphone Android*. Sistem *Android* dirancang untuk menyampaikan “pengumuman” secara otomatis jika baterai habis. Apabila aplikasi yang dibuat dilengkapi dengan komponen *broadcast receiver*, maka *user* (pengguna) dapat mengambil tindakan menyimpan kemudian menutup aplikasi atau tindakan yang lain.

2.5.4. Kelebihan dan Kelemahan Android

Menurut Zuliana & Irwan Padli tahun 2013 menyatakan bahwa, *Android* merupakan sistem operasi yang dirancang oleh salah satu pemilik situs terbesar di dunia yaitu Google. Seiring berjalannya waktu, *Android* telah berevolusi menjadi sistem yang luar biasa dan banyak diminati oleh pengguna *smartphone* karena mempunyai banyak kelebihan. Namun, dibalik popularitas *platform Android* yang disebut sebagai teknologi canggih ini pastilah memiliki kekurangan. Berikut adalah kelebihan dan kelemahan *Android* :

1. Kelebihan *Android*

- a) Lengkap : Para pengembang dapat melakukan pendekatan yang komprehensif ketika sedang mengembangkan *platform Android*. *Android* merupakan sistem operasi yang aman dan banyak menyediakan *tools* guna membangun *software* dan menjadikan peluang untuk para pengembang aplikasi.
- b) *Android* bersifat terbuka (*Open Source Platform*) : *Android* berbasis linux yang bersifat terbuka atau *open source* maka dapat dengan mudah untuk dikembangkan oleh siapa saja.
- c) *Free Platform* : *Android* merupakan *platform* yang bebas untuk para pengembang. Tidak ada biaya untuk membayar

lisensi atau biaya royalty. *Software Android* sebagai *platform* yang lengkap, terbuka bebas, dan informasi lainnya dapat diunduh secara gratis dengan mengunjungi website developer.android.com

d) Sistem Operasi Rakyat : Ponsel *Android* tentu berbeda dengan *iPhone Operating System (IOS)* yang terbatas pada *gadget* dari Apple, lain halnya dengan *Android* yang memiliki banyak produsen, mulai dari HP china seperti Evercross, INfinix, Xiaomi hingga Samsung dengan harga yang cukup terjangkau.

2. Kelemahan *Android*

- a) *Android* selalu terhubung dengan internet. *Handphone* bersistem *Android* ini sangat memerlukan koneksi internet yang aktif.
- b) Banyaknya iklan yang terpampang diatas atau dibawah aplikasi. Walaupun tidak ada pengaruhnya dengan aplikasi yang sedang dipakai tetapi iklan ini cukup mengganggu.
- c) Baterai yang cepat habis.

2.6 Android Studio

Android Studio adalah lingkungan pengembangan android baru berdasarkan IntelliJ IDEA. Mirip dengan *eclipse* dengan ADT plugin, *Android Studio* menyediakan alat pengembang android terintegrasi untuk pengembangan dan debugging. Diatas kemampuan yang anda harapkan dari IntelliJ, *Android Studio* menawarkan :

- Berbasis *Gradle* membangun dukungan.
- *Refactoring* android, spesifik dan perbaikan yang cepat.
- Alat *lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi dan masalah lainnya.
- *ProGuard* dan aplikasi penanda tangan kemampuan.

- Penyihir berbasis template untuk membuat desain android umum dan komponen.
- Sebuah *layout* editor yang memungkinkan anda untuk *drag and drop* UI komponen, *layout* pratinjau pada beberapa konfigurasi layar, dan banyak lagi.

2.7 Java

Menurut Abdul Kadir tahun 2005 menyatakan bahwa,Java adalah bahasa pemrograman serbaguna yang dapat digunakan untuk membuat suatu program. Menurut Isak Rickyanto tahun 2005 menyatakan bahwa, Java merupakan teknologi dimana teknologi tersebut mencakup java sebagai bahasa pemograman yang memiliki sintaks dan aturan pemograman tersendiri, juga mencakup java sebagai *platform* dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemograman java.

2.8 Android Software Development Kit (SDK)

Menurut Nazrudin Safaat tahun 2011 menyatakan bahwa, Android SDK adalah *tool API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang release oleh google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemograman java.

2.9 XAMPP

Menurut Riyanto tahun 2010 menyatakan bahwa, XAMPP adalah merupakan paket PHP dan MySQL, berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket.

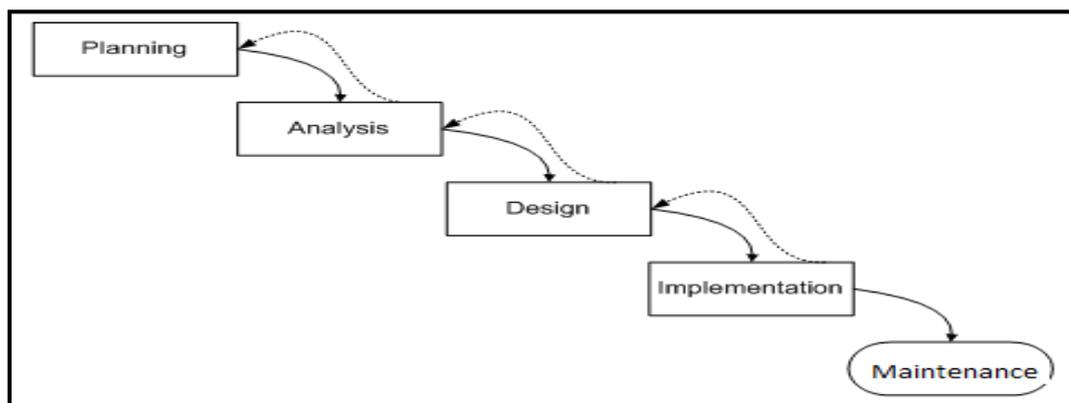
2.10 HTML5

Menurut Raharjo,H dan RK tahun 2010 menyatakan bahwa, HTML (*HyperText Markup Language*) merupakan *file* teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser*. Setiap informasi yang tampil di *web* selalu dibuat menggunakan kode HTML. Oleh karena itu, dokumen HTML sering disebut juga sebagai *web page* (halaman *web*).

Secara umum, dokumen HTML terbagi atas dua bagian, yaitu bagian *header* (kepala) dan *body* (badan). Bagian *header* diawali dengan *tag* `<head>` dan diakhiri dengan *tag* `</head>`. Sedangkan bagian *body* diawali dengan *tag* `<body>` dan ditutup dengan *tag* `</body>`. Kedua bagian tersebut diapit oleh *tag* `<html>` dan `</html>`.

2.11 Metode Pengembangan Perangkat Lunak

Menurut S,Rosa & M tahun 2016 menyatakan bahwa,Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun :



Gambar 2.1 Metode *waterfall*

2.11.1 Planning (Perencanaan)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa,Tahap perencanaan merupakan proses penting untuk mengetahui mengapa sistem harus dibuat dan menentukan bagaimana cara membangun sistem tersebut. Langkah pertama dari proses tersebut adalah dengan mengidentifikasi peluang apakah dapat memberikan kemungkinan biaya rendah tetapi menghasilkan keuntungan.

2.11.2 Analysis (Analisis)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2.11.3 Design (desain)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean.

2.11.4 *Implementation / coding*(pengujian)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, Merupakan tahap berikutnya untuk menerjemahkan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan. Semua tahap ini desain perangkat lunak sebagai sebuah program lengkap atau unit program.

2.11.5 Maintenance (pemeliharaan)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dan pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tapi untuk membuat perangkat lunak baru.

2.12 Basis Data

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data dalam sistem informasi meliputi :

1. Memasukkan, menyimpan, dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

2.13 DBMS

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara
4. Mampu menangani backup data

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational* DBMS yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relational* DBMS atau RDBMS.

Pada tahun 1980an mulai berkembang *Object Oriented* DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek. Saat ini OODBMS juga cukup berkembang namun belum dapat menggeser kepopuleran RDBMS.

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan di dunia saat ini, yaitu :

- a. Oracle
- b. Microsoft SQL Server
- c. IBM DB2
- d. Microsoft Access

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut :

- a. MySQL
- b. PostgreSQL
- c. Firebird
- d. SQLite

2.14 SQL

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS . SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini :

| NO | Tahun | Nama |
|----|-------|----------|
| 1 | 1986 | SQL-86 |
| 2 | 1989 | SQL-89 |
| 3 | 1992 | SQL-92 |
| 4 | 1999 | SQL:1999 |
| 5 | 2003 | SQL:2003 |
| 6 | 2006 | SQL:2006 |
| 7 | 2008 | SQL:2008 |
| 8 | 2011 | SQL:2011 |

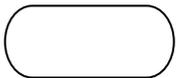
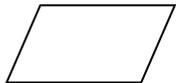
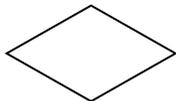
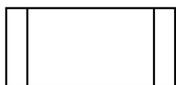
Tabel 2.1 (SQL)

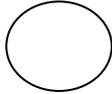
Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

2.15 Diagram Alir Flowchart

Menurut Wibowo tahun 2009 menyatakan bahwa, Flowchart atau bagan alir adalah representasi grafik dari sistem yang mendeskripsikan relasi fisik di antara *entitas-entitas* intinya. Bagan alir dapat digunakan untuk menyajikan aktivitas manual, aktivitas pemrosesan komputer, atau keduanya. Bagan alir dokumen (dokumen *flowchart*) digunakan untuk menggambarkan elemen-elemen dari sistem manual, termasuk catatan

akuntansi (dokumen, jurnal, buku besar, dan file), departemen organisasi yang terlibat dalam proses dan aktivitas (baik yang bersifat administrative maupun fisik) yang dilakukan dalam departemen tersebut. Adapun symbol bagan alir program dapat dilihat pada tabel 2.2 sebagai berikut.

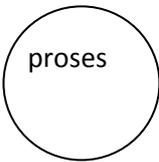
| Simbol | Keterangan |
|---|--|
| <p><i>Terminator</i></p>  | Digunakan untuk memberikan awal dan akhir suatu proses. |
| <p>Proses</p>  | Menunjukkan proses dari operasi program komputer. |
| <p><i>Preparation</i></p>  | Proses inisialisasi/pemberian harga awal. |
| <p><i>Input/Output Data</i></p>  | Proses <i>input/output</i> data, parameter, informasi. |
| <p>Document</p>  | <i>Document</i> merupakan symbol untuk data yang berbentuk informasi. |
| <p>Garis alir</p>  | Digunakan untuk menunjukkan arus dari proses. |
| <p><i>Decision</i></p>  | Digunakan untuk suatu penyeleksian kondisi di dalam program. |
| <p>Proses terdefinisi</p>  | Simbol yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain. |

| | |
|--|--|
| <p><i>on page reference</i></p>  | <p>Suatu prosedur akan masuk dan keluar melalui <i>symbol</i> ini dalam lembar yang sama</p> |
| <p><i>off page reference</i></p>  | <p>Merupakan symbol untuk masuk dan keluarnya suatu prosedur pada lembar kertas yang lain.</p> |

Tabel 2.2 Bagan Alir Program (*Program Flowchart*)

2.16 Data Flow Diagram (DFD)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, *Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Notasi-notasi pada DFD adalah seperti tabel 2.3 dibawah ini :

| Notasi | Keterangan |
|---|---|
|  | <p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harus menjadi fungsi atau prosedur didalam kode program.</p> |
|  | <p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat mejadi table-tabel basis data yang dibutuhkan, table-tabel ini juga harus sesuai dengan perancangan table-tabel pada basis data (ERD)</p> |
|  | <p>Entitas luar (<i>external entity</i>) atau masukan atau keluaran atau orang-orang yang memakai atau berinteraksi dengan perangkat lunak yang dmodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan</p> |
|  | <p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses kemasukan (<i>input</i>) atau keluaran (<i>ouput</i>)</p> |

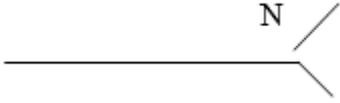
Tabel 2.3 Simbol Data *Flow Diagram*

2.17 Entity Relationship Diagram (ERD)

Menurut S,Rosa & M tahun 2016 menyatakan bahwa, pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori

himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Notasi-notasi pada ERD adalah seperti ditunjukkan pada tabel 2.4 dibawah ini

| Simbol | Keterangan |
|--|---|
| <p data-bbox="295 481 478 515"><i>Entitas /entity</i></p> <div data-bbox="391 716 670 840" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p data-bbox="406 750 582 784">Nama_entitas</p> </div> | <p data-bbox="805 481 1220 1008">Entitas/entity merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer;penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel</p> |
| <p data-bbox="295 1086 391 1120">Atribut</p> <div data-bbox="343 1108 638 1232" style="border: 1px solid black; border-radius: 50%; padding: 5px; width: fit-content; margin: 0 auto;"> <p data-bbox="406 1142 566 1176">Nama_atribu</p> <p data-bbox="406 1187 422 1220">†</p> </div> | <p data-bbox="805 1086 1220 1232">Atribut atau field;atau kolom data yang butuh disimpan dalam suatu entitas.</p> |
| <p data-bbox="295 1254 574 1288">Atribut Kunci Primer</p> <div data-bbox="319 1344 694 1489" style="border: 1px solid black; border-radius: 50%; padding: 5px; width: fit-content; margin: 0 auto;"> <p data-bbox="391 1388 622 1422"><u>Nama kunci primer</u></p> </div> | <p data-bbox="805 1254 1220 1512"><i>Field</i> atau kolom yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id</p> |
| <p data-bbox="295 1601 686 1635">Atribut Multi nilai/<i>multivalue</i></p> <div data-bbox="295 1657 710 1814" style="border: 3px double black; border-radius: 50%; padding: 5px; width: fit-content; margin: 0 auto;"> <p data-bbox="391 1702 550 1736">Nama_atribut</p> </div> | <p data-bbox="805 1534 1220 1736"><i>Field</i> atau kolom data yang butuh disimpan dalam suatau entitas yang dapat memiliki nilai lebih dari satu.</p> |

| | |
|---|---|
| <p>Relasi</p>  | <p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p> |
| <p>Asosiasi / Association</p>  | <p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.</p> |

Tabel 2.4 Notasi pada Model ERD

2.18. Penelitian Terdahulu

Pada bagian ini akan membahas mengenai tabel penelitian terdahulu yang berhubungan dengan karya ilmiah penulis, yang terdapat pada table 2.5 berikut :

| No | Nama | Judul | Tahun Terbit | Keterangan |
|----|--------------|---|---|---|
| 1 | Ayu Ningtias | Membangun <i>M-Reservation</i> Pada Pecel Lele Haji Fuad Bandar Lampung | Teknik Informatika IBI Darmajaya / 2017 | Aplikasi reservasi dan pembayaran secara online pada rumah makan Haji Fuad Bandar Lampung. Masalah utamanya adalah antrian panjang pelanggan yang sering kali harus mengantri lama untuk memesan dan membayar makanan mereka. Selain itu aplikasi ini juga menjadi sarana promosi dan |

| | | | | |
|---|-------------|--|---|--|
| | | | | pencatatan bagi rumah makan Haji Fuad. |
| 2 | Eka Yuliana | Rancang Bangun Aplikasi Transaksi Pulsa Elektronik Telepon Sellular Berbasis Android | Teknik Informatika IBI Darmajaya / 2016 | Pengisian pulsa elektronik merupakan salah satu cara melakukan transaksi isi ulang pulsa ke suatu nomor <i>handphone</i> tertentu dengan metode elektronik. Pengisian pulsa dengan metode elektronik dapat dilakukan dengan melakukan pengiriman pesan singkat menggunakan teknologi SMS (<i>Short Message Service</i>). Pesan akan dikirim dengan format tertentu ke nomor <i>handphone</i> server pulsa melalui <i>SMS center</i> , format pesan ditentukan oleh masing-masing server pulsa cara ini biasanya digunakan oleh <i>counter</i> pulsa dalam melakukan transaksi pulsa. |

Tabel 2.5 Penelitian terdahulu