



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Perangkat Lunak

Sukanto dan Shalahuddin (2016:2), Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan model desain, dan cara penggunaan (*user manual*).

Ladjamudin (2013:20), *Software* merupakan kumpulan dari perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

Dari beberapa definisi diatas penulis menyimpulkan bahwa perangkat lunak adalah program komputer yang berisi perintah yang ditulis dengan aturan tertentu untuk menjalankan suatu tugas tertentu.

2.1.2. Pengertian Komputer

Menurut Erzha (2014:7), “Komputer adalah suatu alat untuk pengolahan informasi sesuai prosedur yang telah dirumuskan.”

Irwansyah dan Moniaga (2014:02), Komputer adalah perangkat elektronik yang beroperasi di bawah kendali instruksi yang memiliki memori, dapat menerima dan mengolah data sesuai dengan aturan yang ditetapkan, menghasilkan hasil atau informasi dan menyimpan hasil tersebut untuk digunakan di masa depan atau di kemudian hari.

Dengan demikian Komputer dapat diartikan sebagai alat elektronik yang beroperasi di bawah kendali instruksi sesuai prosedur yang telah dirumuskan untuk menghasilkan informasi yang dapat digunakan di kemudian hari.

2.1.3. Pengertian Data

McLeod dalam Yakub (2012:5), Data adalah kenyataan yang menggambarkan adanya suatu kejadian (*event*), data terdiri dari fakta (*fact*) dan angka yang secara relatif tidak berarti bagi pemakai.



Asropudin (2013:22), Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.

Dari beberapa definisi diatas penulis menyimpulkan bahwa data adalah sekumpulan fakta yang harus diolah terlebih dahulu agar menghasilkan suatu informasi.

2.1.4. Pengertian Basis Data (*Database*)

Yakub (2012:55), Basis Data merupakan koleksi dari data-data yang terorganisasi dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi.

Kristanto (2011:21), *Database* merupakan kumpulan file yang saling terintegrasi, namun database tidak akan dapat diakses oleh siapapun tanpa adanya software aplikasi.

Badiyanto (2013:57), *Database* bisa dikatakan sebagai suatu kumpulan dari data yang tersimpan dalam tabel dan diatur atau diorganisasikan sehingga data tersebut bisa diambil atau dicari dengan mudah dan efisien.

Dari beberapa definisi diatas penulis menyimpulkan bahwa basis data (*database*) adalah tempat yang digunakan untuk menyimpan data-data agar lebih terorganisir.

2.1.5. Pengertian Pengolahan Data

Kristanto (2011:8), Pengolahan Data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Sutabri (2012:6) berpendapat bahwa, Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data.

a) Penyimpanan (*Storage*)

Penyimpanan data meliputi pekerjaan pengumpulan (*filling*), pencairan (*searcing*), dan pemeliharaan (*maintenance*). Data disimpan dalam suatu



tempat yang lazim dinamakan “*file*”. *File* dapat berbentuk map, *ordner*, *disket*, *tape*, *hard disk*, dan lain sebagainya.

b) Penanganan data (*data handling*)

Penanganan data meliputi berbagai kegiatan seperti: pemeriksaan (*verifying*), perbandingan (*comparing*), pemilihan (*sorting*), peringkasan (*extracting*), dan penggunaan (*manipulating*). Pemeriksaan data mencakup pengecekan data yang muncul pada berbagai daftar yang berkaitan atau yang datang dari berbagai sumber, untuk mengetahui berbagai sumber untuk mengetahui perbedaan atau ketidaksesuaian, pemeriksaan ini dilakukan dengan kegiatan pemeliharaan *file* (*file maintenance*).

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Asropudin (2013:6), “Aplikasi adalah *Software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms-Word, Ms-Excel.

Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya Ms-Word, Ms-Excel”.

Dari beberapa definisi diatas penulis menyimpulkan bahwa Aplikasi adalah program computer yang dibuat oleh perusahaan computer untuk mengerjakan tugas-tugas tertentu.

2.2.2. Pengertian Pelayanan

Mardikawati dan Farida (2013:68), “Pelayanan merupakan perilaku produsen dalam rangka memenuhi kebutuhan dan keinginan konsumen demi tercapainya kepuasan pada konsumen itu sendiri”.

Koestanto dan Yuniati (2014:4), “Pelayanan merupakan salah satu variabel yang diikutsertakan dalam pemasaran maupun dalam hal pengkonsumsian produk



oleh konsumen karena pelayanan yang baik kepada pelanggan merupakan suatu nilai tambah perusahaan”.

Dengan demikian pelayanan dapat diartikan sebagai perilaku produsen dalam rangka memenuhi kebutuhan dan keinginan konsumen karena pelayanan yang baik kepada konsumen merupakan nilai tambah bagi perusahaan.

2.2.3. Pengertian Jasa

Dharmesta dalam Yuliana (2012:41) menyebutkan bahwa, “Jasa adalah suatu kegiatan yang dapat berwujud maupun tidak berwujud yang dilakukan untuk melayani konsumen”.

Kotler dalam Talumewo dkk. (2014:1586) menyatakan bahwa, “Jasa: Setiap tindakan atau kegiatan yang dapat ditawarkan oleh satu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun. Produksinya dapat dikaitkan atau tidak dapat dikaitkan dengan suatu produk fisik”.

Dari definisi diatas, disimpulkan bahwa jasa adalah tindakan atau kegiatan tidak berwujud maupun berwujud yang pada dasarnya dilakukan untuk melayani konsumen.

2.2.4. Pengertian Pengiriman

Menurut Kamus Besar Bahasa Indonesia, “Pengiriman adalah proses, cara, perbuatan mengirimkan”.

2.2.5. Pengertian Surat

Menurut Pusat Pembinaan dan Pengembangan Bahasa Departemen Pendidikan Nasional dalam Purwanto (2014:97), “Surat dapat didefinisikan sebagai sarana komunikasi yang digunakan untuk menyampaikan informasi tertulis oleh suatu pihak kepada pihak lain”.

Mariskha (2015:2) berpendapat bahwa, “Surat adalah komunikasi tertulis untuk menyampaikan pesan kepada pihak lain, yang memiliki persyaratan khusus



yaitu penggunaan kode dan notasi (lampiran dan perihal), penggunaan kertas, penggunaan model dan bentuk, pemakaian bahasa yang khas serta pencantuman tanda tangan”.

Dari definisi diatas penulis menyimpulkan bahwa surat adalah sarana komunikasi tertulis untuk menyampaikan informasi kepada pihak lain.

2.2.6. Pengertian Paket

Menurut Kamus Besar Bahasa Indonesia, “Paket adalah barang yg dikirimkan dlm bungkus melalui pos atau perusahaan ekspedisi”.

2.2.7. Pengertian Teknologi

Waridah (2017:280), Teknologi adalah metode ilmiah untuk mencapai tujuan praktis; ilmu pengetahuan terapan.

Sujatmiko (2012:269), Teknologi adalah penerapan keilmuan yang mempelajari dan mengembangkan kemampuan dari suatu rekayasa dengan langkah dan teknik tertentu dalam suatu bidang.

Dari beberapa definisi diatas penulis menyimpulkan bahwa teknologi adalah metode ilmiah yang mempelajari dan mengembangkan kemampuan dari suatu rekayasa dengan langkah dan teknik tertentu.

2.2.8. Pengertian Android

Juhara, (2016:1), Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama.

Salbino (2010:7) , Android merupakan sistem operasi berbasis Linux yang bersifat terbuka (*open source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet.

Dari beberapa definisi diatas penulis menyimpulkan bahwa android adalah sistem operasi berbasis Linux yang bersifat terbuka (*open source*) untuk perangkat mobile (*smartphone*) selain *mobile* khusus seperti Windows Phone dan iOS.



Mobile Phone yang digunakan harus bersifat touchscreen sehingga sesuai dengan kegunaan android yaitu memudahkan.

2.2.8.2. Generasi Android

Ponsel pertama yang memakai sistem Operasi Android adalah HTC Dream yang di rilis pada tanggal 22 Oktober 2008 dan pada awal tahun 2009 mulailah para pengembang ponsel menggunakan OS android ini dan di perkirakan setidaknya 18 ponsel bersistem OS Android rilis di awal tahun 2009.

Menurut Salbino (2010:17), perkembangan android adalah sebagai berikut:

1. Android versi 1.0

Android 1.0, merupakan versi komersial pertama Android, dirilis 23 September 2008 dengan kode nama Apple pie serta ukuran layar 320x480 HVGA. Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android versi 1.1

Android versi 1.1 di rilis pada 9 Maret 2009 oleh Google. Android ini telah disupport oleh Google Mail Service dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

3. Android versi 1.5 Cup Cake

Android Cup Cake di rilis pada pertengahan Mei 2009, masih oleh Google Inc. Android ini telah dilengkapi software development kit dengan berbagai pembaharuan termasuk penambahan beberapa fitur lain, yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube, upload gambar ke Picasa langsung dari telepon, serta mendapat dukungan Bluetooth A2DP.

4. Android versi 1.6 Donut

Android Donut di rilis pada September 2009 menampilkan proses pencarian yang lebih baik dari versi-versi sebelumnya. Android Donut juga memiliki fitur-fitur tambahan seperti galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang



terintegrasikan; Text-to-speech engine; dial kontak; teknologi *text to change speech*, baterai indikator dan kontrol applet VPN.

5. Android versi 2.0/2.1 Eclair

Android Eclair dirilis pada 3 Desember 2009. Perubahannya antara lain : pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. android Eclair adalah android pertama yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan user interface.

6. Android versi 2.2 Froyo (*Frozen Yogurt*)

Android Froyo dirilis pada 20 Mei 2012. Versi ini memiliki kecepatan kinerja dan aplikasi 2 sampai 5 kali dari versi-versi sebelumnya. Selain itu ada penambahan fitur-fitur baru seperti dukungan Adobe Flash 10.1, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan Wifi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

7. Android versi 2.3 Gingerbread

Android Gingerbread di rilis pada 6 Desember 2010. Perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

8. Android versi 3.0/3.1 Honeycomb

Android Honeycomb di rilis pada awal 2012. Versi Android yang dirancang khusus untuk device dengan layar besar seperti Tablet PC. Fitur baru pada Android Honeycomb antara lain dukungan terhadap processor multicore dan



grafis dengan hardware acceleration. User interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Tablet pertama yang memakai Honeycomb adalah tablet Motorola Xoom yang dirilis bulan Februari 2011.

9. Android versi 4.0 ICS (*Ice Cream Sandwich*)

Android Ice Cream Sandwich diumumkan secara resmi pada 10 Mei 2011 pada ajang Google I/O Developer Conference (*San Fransisco*), pihak Google mengklaim Android Ice Cream Sandwich akan dapat digunakan baik di smartphone ataupun tablet. Android Ice Cream Sandwich membawa fitur Honeycomb untuk smartphone serta ada penambahan fitur baru seperti 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. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

10. Android versi 4.1 Jelly Bean

Android Jelly Bean juga diluncurkan pada acara Google I/O 10 Mei 2011 yang lalu. Keunggulan dan fitur baru versi ini, diantaranya peningkatan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Sistem Operasi Android Jelly Bean 4.1 pertama kali digunakan dalam produk tablet Asus, yakni Google Nexus 7.

11. Android versi 4.2 Jelly Bean

Fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (*dalam tablet saja*), widget terbaru. Android 4.2 pertama kali dikenalkan melalui LG Google Nexus 4.

12. Android versi 4.3 Jelly Bean

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Fransisco. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.



13. Android versi 4.4 Kitkat

Google mengumumkan Android 4.4 KitKat (*dinamai dengan izin dari Nestle dan Hershey*) pada 3 September 2013, dirilis pada tanggal 31 Oktober 2013. Keunggulannya diantaranya pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan, Optimasi kinerja dengan spesifikasi perangkat yang lebih rendah, NFC Host Card Emulation sebagai emulator kartu pintar, WebViews berbasis Chromium, Sensor batching, Step Detector, dan Counter API, peningkatan tampilan mode layar penuh, Penyeimbang audio, pemantauan audio, dan peningkatan suara audio, dukungan Bluetooth Message Acces Profile (*MAP*)

14. Android versi 5.0 Lollipop

Android Lollipop 5.0, diresmikan Juni 2014, tersedia resmi sebagai update over-the-air (OTA) November 2014, memiliki fitur antarmuka yang di desain ulang, dibangun dengan desain responsif, yang kemudian dilanjutkan versi 5.0, 5.0.1, 5.0.2, 5.1, dan terakhir 5.1.1. Fitur yang disediakan seperti Peningkatan Kunci Layar, Penambahan multi-bahasa, Peningkatan antar muka tampilan layar serta Peningkatan notifikasi sistem.

15. Android versi 6.0 Marshmallow

Pertama kali diperkenalkan Mei 2015 dengan kode nama Android M, secara resmi dirilis pada Oktober 2015. Android Marshmallow berfokus terutama pada peningkatan pengalaman pengguna secara keseluruhan dari versi Lollipop, memperkenalkan arsitektur baru, termasuk bantuan kontekstual, sistem manajemen daya dengan mengurangi aktivitas latar belakang saat perangkat tidak secara fisik digunakan, dukungan asli untuk pengenalan sidik jari dan USB, kemampuan untuk bermigrasi data dan aplikasi ke kartu microSD dan menggunakannya sebagai penyimpanan utama, serta perubahan internal lainnya. Kemudian dilanjutkan dengan versi 6.0.1

16. Android versi 7.0 Nougat

Android Nougat adalah sistem operasi terbaru yang memperkenalkan perubahan penting untuk platform dan pengembangan, termasuk kemampuan



untuk menampilkan beberapa aplikasi di layar sekaligus dalam tampilan layar terpisah, dukungan untuk balasan pemberitahuan, serta lingkungan berbasis “Java OpenJDK” dan dukungan untuk render grafis “Vulkan API”, dan pembaruan sistem mulus pada perangkat yang didukung.

Dari penjelasan diatas, versi android yang akan digunakan pada pembuatan aplikasi ini yaitu dari android yang menggunakan versi ice cream sampai android yang di update itu sendiri.

2.2.9. Pengertian Perseroan Terbatas (PT)

Berdasarkan Pasal 1 UU PT No. 40/2007, Pengertian Perseroan Terbatas (Perseroan) adalah badan hukum yang merupakan persekutuan modal, didirikan berdasarkan perjanjian, melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham, dan memenuhi persyaratan yang ditetapkan dalam undang-undang ini serta peraturan pelaksanaannya.

Dari definisi diatas dapat disimpulkan bahwa perseroan terbatas (PT) adalah badan hukum yang mempunyai modal bersama yang akan dibagikan sesuai saham.

2.2.10. Pengertian PT. Pos Indonesia (Persero)

PT. Pos Indonesia (Persero) adalah sebuah BUMN (Badan Usaha Milik Negara) yang bergerak di bidang industri pelayanan jasa.

2.2.11. Pengertian Aplikasi Pelayanan Jasa Pengiriman surat dan paket Berteknologi Android pada PT. Pos Indonesia (Persero) Palembang

Pengertian Aplikasi Pelayanan Jasa Pengiriman surat dan paket Berteknologi Android pada PT. Pos Indonesia (Persero) Palembang adalah Suatu Aplikasi yang digunakan untuk mempermudah pelanggan dalam melakukan pemesanan jasa pengiriman surat dan paket, serta mempermudah bagian pelayanan dalam mencari pelanggan dengan mudah di wilayah Palembang menggunakan teknologi android.



2.3. Teori Khusus

2.3.1. Pemrograman Berorientasi Objek

2.3.1.1. Pengertian Pemrograman Berorientasi Objek

Sukamto dan Shalahuddin (2016:100), berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Dari definisi diatas dapat disimpulkan bahwa pemrograman berorientasi objek adalah program komputer dari berbagai objek yang melakukan suatu tindakan terhadap masing-masing objek

2.3.1.2. Ciri Pemrograman Berorientasi Objek

Siallagan (2009:149), ciri-ciri atau karakteristik pemrograman berorientasi objek, antara lain:

a. Abstraksi (*Abstraction*)

Abstraksi adalah pengabstrakan atau melakukan seleksi terhadap aspek-aspek tertentu suatu masalah. Abstraksi digunakan untuk penyembunyian kerumitan dari suatu proses. Sebagai contoh, dalam membuat suatu sistem, ada tombol-tombol yang dapat digunakan. Operator atau pengguna tidak perlu berpikir tentang pembuatan tombol tersebut, tetapi yang penting mereka dapat menggunakannya.

b. Pembungkusan (*Encapsulation*)

Pembungkusan sering pula disebut pengkapsulan. Artinya, proses membuat paket (memaketkan) data objek bersama dengan metode-metodenya. Berdasarkan kode program, proses memisahkan aspek-aspek objek dilakukan dengan pembungkusan. Proses pembungkusan itu sendiri merupakan cara atau mekanisme untuk melakukan abstraksi.

c. Pewarisan (*Inheritance*)

Pewarisan adalah memberikan atau mewariskan sesuatu kepada keturunan berikutnya. Misalnya, seorang anak pasti akan mewarisi beberapa sifat atau



perilaku yang dimiliki oleh ibu/bapaknya. Dalam konteks ini, suatu kelas dalam program dapat diturunkan menjadi kelas-kelas baru lainnya yang akan mewarisi beberapa sifat atau perilaku dari kelas induknya.

d. Polimorfisme (*Polymorphism*)

Polimorfisme adalah suatu kejadian ketika objek dapat mengungkap banyak hal melalui satu cara yang sama.

2.3.1.3. Metodologi Pengembangan Perangkat Lunak dengan RUP

Sukamto dan Shalahudin (2016:128) dalam *Rational Unified Process* terdapat empat tahap pengembangan perangkat lunak yaitu:

a. *Inception* (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini:

- Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya).
- Membangun kasus bisnis yang dibutuhkan.

b. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

c. *Construction* (Konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.



d. *Transition* (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.3.2. Pengujian Perangkat Lunak

2.3.2.1. Pengertian Pengujian Perangkat Lunak

Sukanto dan Shalahudin (2016:272), Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*). Dapat juga dikatakan sebagai berikut:

- a. Verifikasi: “Apakah produk dibangun dengan benar?” (lebih ke arah apakah proses pengembangan produk sudah benar dan telah berhasil mengimplementasikan fungsi yang benar)
- b. Validasi: “Apakah sudah membangun produk yang benar?” (lebih ke arah hasil produk apakah sudah sesuai dengan yang diinginkan)

2.3.2.2. Metode *Black Box Testing*

Sukanto dan Shalahudin (2016:275), *Black Box Testing* (pengujian kotak hitam), yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.



Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah:

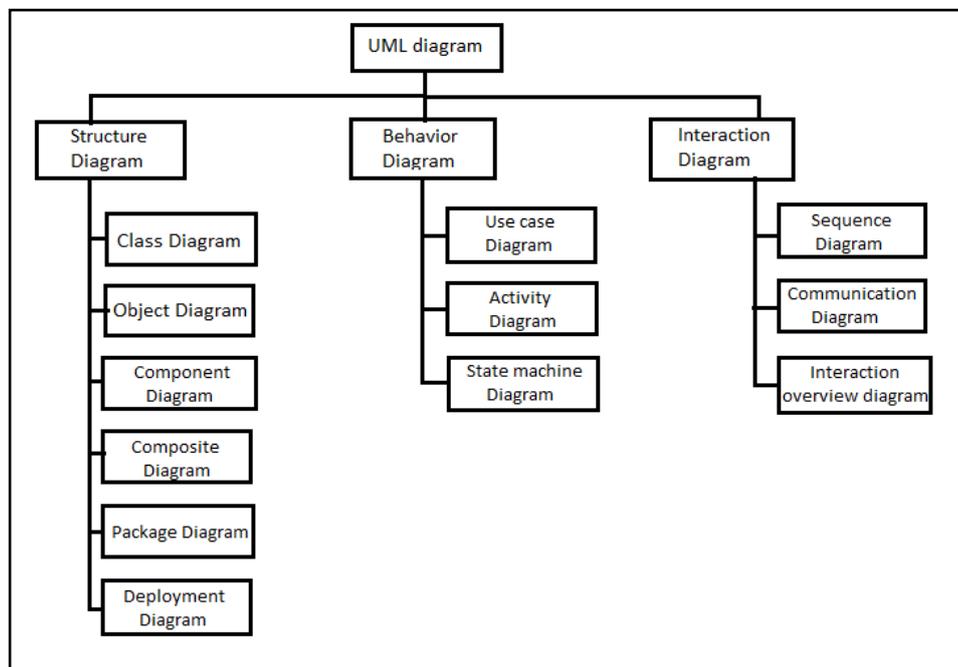
- a. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar
- b. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.3.3. *Unified Modeling Language (UML)*

Sukamto dan Shalahuddin (2016:133), *UML (Undefined Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.3.3.1. Macam-macam Diagram *UML*

Sukamto dan Shalahudin (2016:140), menyatakan pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.1 *Macam-macam Diagram UML*

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. *Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.3.2. *Use Case Diagram*

Munawar (2005:62), *use case* adalah deskripsi fungsi dari sebuah system dari perspektif pengguna.

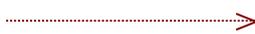


Sukanto dan Shalahuddin (2016:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-simbol dalam Use case

| Simbol | Deskripsi |
|--|---|
| Use Case  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| Aktor / actor  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| Asosiasi / association  | Komunitas antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor. |
| Ekstensi / extend <<extend>>  | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tamhanan itu; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya. |

Lanjutan Tabel 2.1 Simbol-simbol dalam *Use case*

| | |
|--|--|
| <p>Generalisasi / <i>Generalization</i></p>  | <p>Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p> |
| <p><i>Include</i></p> <p><<include>></p>  | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini; <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p> |

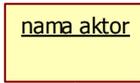
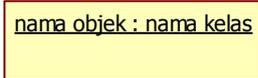
(Sumber: Sukamto dan Shalahuddin, 2016:156)

2.3.3.3. *Sequence Diagram*

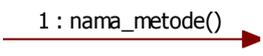
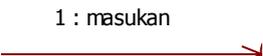
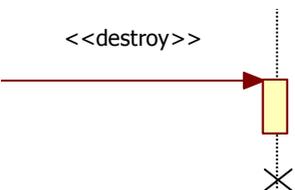
Munawar (2005:87), *sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

Sukamto dan Shalahuddin (2016:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2.2 Simbol-simbol dalam *Sequence Diagram*

| Simbol | Deskripsi |
|---|--|
| <p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>tanpa waktu aktif</p> | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p> |
| <p>Garis hidup / <i>lifeline</i></p>  | <p>menyatakan kehidupan suatu objek.</p> |
| <p>Objek</p>  | <p>Menyatakan objek yang berinteraksi pesan.</p> |
| <p>Waktu aktif</p>  | <p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p> |
| <p>Pesan tipe create</p>  | <p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p> |

Lanjutan Tabel 2.2 Simbol-simbol dalam *Sequence Diagram*

| | |
|---|---|
| <p>Pesan tipe call</p>  | <p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p> |
| <p>Pesan tipe send</p>  | <p>Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p> |
| <p>Pesan tipe return</p>  | <p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p> |
| <p>Pesan tipe destroy</p>  | <p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.</p> |

(Sumber: Sukamto dan Shalahuddin, 2016:165)

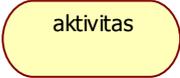
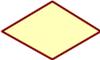
2.3.3.4. Activity Diagram

Munawar (2005:109), *activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus.

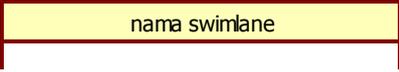
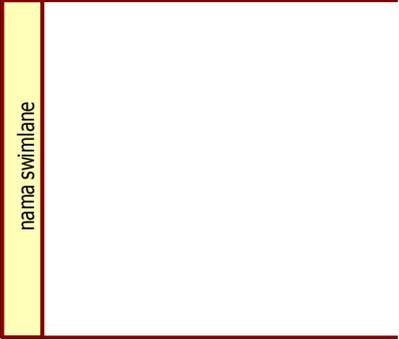


Sukamto dan Shalahuddin (2016:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Tabel 2.3 Simbol-simbol dalam Activity Diagram

| Simbol | Deskripsi |
|--|--|
| Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan / <i>decision</i>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

Lanjutan Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

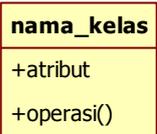
| | |
|--|---|
| <p>Swimlane</p>  <p>atau</p>  | <p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p> |
|--|---|

(Sumber: Sukamto dan Shalahuddin, 2013:162)

2.3.3.5. Class Diagram

Sukamto dan Shalahuddin (2013:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.4 Simbol-Simbol *Class Diagram*

| Simbol | Deskripsi |
|--|------------------------------------|
| <p>Kelas</p>  | <p>Kelas pada struktur sistem.</p> |

Lanjutan Tabel 2.4 Simbol-Simbol *Class Diagram*

| | |
|---|---|
| <p>Antar muka/ <i>interface</i></p>  | <p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p> |
| <p>Asosiasi / <i>association</i></p>  | <p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p> |
| <p>Asosiasi berarah / <i>directed association</i></p>  | <p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p> |
| <p>Generalisasi</p>  | <p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).</p> |
| <p>Kebergantungan / <i>dependency</i></p>  | <p>Relasi antarkelas dengan makna kebergantungan antarkelas.</p> |
| <p>Agregasi / <i>aggregation</i></p>  | <p>Relasi antarkelas dengan makna semua-bagian (whole-part).</p> |

(Sumber: Sukamto dan Shalahuddin, 2013:146)



2.4. Teori Program

2.4.1. Pemrograman Java

2.4.1.1. Pengertian Pemrograman Java

Sukanto dan Shalahuddin (2016:103), java merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas.

Asropudin (2013:52), java adalah bahasa pemrograman untuk menciptakan isi yang aktif dalam halaman *web*, juga dapat dijalankan dalam semua komputer.

2.4.1.2. Pengelompokan Tipe Data dalam Java

Purnama (2007:34), java mengenal dua jenis tipe data:

- a. Tipe data primitif, yaitu tipe data yang diadopsi dari tipe data klasik. Tipe data ini diadopsi dari berbagai bahasa pendahulu Java, antara lain C++ dan Pascal.
- b. Tipe data objek, yaitu tipe data berbentuk *class* yang merupakan ciri khas dari pemrograman berorientasi objek. Banyak dari tipe data ini yang disediakan untuk mendukung operasional tipe data primitif.

Beberapa tipe data yang sering digunakan adalah sebagai berikut:

Tabel 2.5. Tipe data dalam Java

| Tipe data | Class | Keterangan |
|-----------|--------------|----------------------------|
| Int | Integer | Tipe data bilangan bulat |
| Double | Double | Tipe data bilangan pecahan |
| Char | Character | Tipe data karakter tunggal |
| Boolean | Boolean | Tipe data logika |
| | String | Tipe data string statis |
| | StringBuffer | Tipe data string dinamis |



2.4.1.3. Operator dalam Java

Siallagan (2009:50-65), berikut ini adalah operator-operator dalam Java:

a. Operator Aritmatika

Operator aritmatika adalah operator-operator yang digunakan untuk mengoperasikan perhitungan (aritmatika). Bahasa pemrograman *java* menyediakan operator-operator aritmatika untuk memanipulasi variabel data.

Tabel 2.6 *Operator Aritmatika*

| Operator | Keterangan |
|----------|---------------------|
| + | Penjumlahan |
| - | Pengurangan |
| * | Perkalian |
| / | Pembagian |
| % | Modulus (sisa bagi) |

b. Operator Relasional

Operator relasional adalah operator hubungan (relasi) yang membandingkan kedua nilai *operand* dan hasilnya berupa nilai *boolean*, yaitu benar (*true*) atau salah (*false*).

Tabel 2.7 *Operator Relasional*

| Operator | Keterangan |
|----------|---|
| == | Sama dengan (membandingkan bukan penugasan) |
| != | Tidak sama dengan |
| > | Lebih besar |
| < | Lebih kecil |
| >= | Lebih besar sama dengan |
| <= | Lebih kecil sama dengan |

c. Operator Logika/*Boolean*

Operator logika adalah operator yang digunakan terhadap *operand* bertipe *Boolean* yang hasilnya benar (*true*) atau salah (*false*).

**Tabel 2.8** Operator Logika

| Operator | Keterangan |
|----------|------------|
| & | Logika AND |
| | Logika OR |
| ^ | Logika XOR |
| ! | Logika NOT |

2.4.2. Eclipse

Menurut Nofriadi (2015:4), Eclipse merupakan sebuah editor yang digunakan membuat project berbasis *embedded system* dan *mobile* seperti Android, Nokia, atau Blackberry. Ruang lingkup kerja Eclipse (*workspace*) adalah direktori untuk proyek atau proyek explorer, menu, atau toolbar, area editor, dan juga console debugging.

Menurut Fatima (2011:2), Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*).

Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mult-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan *open source*. *Open source* berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang



membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plug-in*.

2.4.2.1. Sejarah Eclipse

Menurut Fatima (2011:2), Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.

2.4.2.2. Arsitektur Eclipse

Menurut Fatima (2011:3), Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah dipasang (diinstal). Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP).

Berikut ini adalah komponen yang membentuk RCP:

1. *Core platform*
2. OSGi
3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE adalah :

1. Terbuka (*open*),
2. Mudah diperluas (*extensible*) untuk apa saja, dan
3. Tidak untuk sesuatu yang spesifik.



2.4.2.3. Versi-versi Eclipse

Menurut Fatima (2011:4), Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse.

Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

Tabel 2.9. Versi-versi Eclipse

| Kode Peluncuran | Tanggal Peluncuran | Platform | Nama Proyek |
|--|---------------------|----------|-------------------|
| Eclipse 3.0 | 28 Juni <u>2004</u> | 3.0 | |
| Eclipse 3.1 | <u>28 Juni 2005</u> | 3.1 | |
| Callisto | 30 Juni <u>2006</u> | 3.2 | Callisto projects |
| Lanjutan Tabel 2.9. Versi-versi Eclipse | | | |
| Europa projects | <u>29 Juni 2007</u> | 3.3 | <u>Europa</u> |
| Ganymede | 25 Juni <u>2008</u> | 3.4 | Ganymede projects |
| Galileo Galileo projects | <u>24 Juni 2009</u> | 3.5 | |

2.4.3. MySQL

2.4.3.1. Pengertian MySQL

Sukamto dan Shalahuddin (2016:46), SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.



Hendry (2015:7), MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.4.3.2. Keistimewaan MySQL

Hendry (2015:7), MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana. Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi

MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).



7. Keamanan

MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala cukup besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu table, serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa indonesia belum termasuk di dalamnya.

11. Antarmuka

MySQL memiliki antarmuka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada, disertakan petunjuk online.

13. Struktur Tabel

MySQL memiliki struktur tabel lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya seacam PostgreSQL ataupun Oracle.