

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Peneliti menggunakan telah penelitian yang berasal dari skripsi terdahulu yang berkaitan dengan permasalahan pada penelitian yang berjudul “Aplikasi Panduan Budidaya Tanaman Sayuran Berbasis *Android*”. Dari penelitian tersebut penelitian ini diharapkan sebagai bahan perbandingan antara penelitian terdahulu dengan penelitian yang nantinya akan dirancang. Beberapa telah penelitian tersebut di antaranya :

Menurut Sumardjo, dkk (2011) informasi pertanian merupakan salah satu faktor yang penting dalam proses produksi usaha tani. Informasi pertanian juga merupakan aplikasi pengetahuan yang mendorong dan menciptakan peluang untuk pembangunan dan pengurangan kemiskinan. Integrasi yang efektif dalam sektor pertanian menuju pada pertanian berkelanjutan melalui penyiapan informasi pertanian yang tepat waktu yang relevan dapat memberikan informasi yang tepat kepada petani dalam proses pengambilan keputusan berusahatani untuk meningkatkan produktivitasnya.

Menurut Tamba (2007), selama ini pemberdayaan pertanian umumnya belum secara nyata memberdayakan para petani. Selama ini petani masih belum bisa mendapatkan informasi yang dibutuhkan. Salah satu strategi pemberdayaan adalah bagaimana membuat petani mampu memperbaiki kehidupannya sendiri dalam arti tahu, termotivasi, dan mampu mengembangkan usaha taninya dengan

mencari dan memanfaatkan informasi pertanian yang tersedia. Pemerintah berkewajiban memberikan pelayanan informasi pertanian bagi para petani sebagaimana yang diamanatkan dalam UU No. 16 Tahun 2006 tentang Sistem Penyuluhan Pertanian, Perikanan, dan Kehutanan.

2.2. Landasan Teori

Adapun teori-teori penunjang yang digunakan penulis untuk mengerjakan Tugas Akhir ini, antara lain:

2.2.1 Arsitektur *Android*

1. *Android Operating System*

Android adalah sebuah *operating system* yang digunakan untuk menjalankan sebuah *smartphone*. *Android* pada awalnya dikembangkan oleh *Android Inc.*, yang kemudian dibeli oleh *google* pada tahun 2005 setelah diberikan dukungan secara finansial oleh *google*. Sistem operasi *android* tersebut resmi dirilis pada tahun 2007, bersamaan dengan didirikannya sebuah perusahaan *Open Handset Alliance*, konsorsium dari beberapa perusahaan-perusahaan perangkat keras, perangkat lunak, serta telekomunikasi yang memiliki tujuan untuk memajukan standar terbuka dari perangkat seluler.

Android merupakan sistem operasi yang bersifat *open source*, yang berarti, bahwa perangkat lunaknya dapat dimodifikasi dan dikembangkan secara bebas. Saat ini sudah ada banyak sekali komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman *Java*. Pada bulan Oktober 2013, ada lebih dari satu juta

aplikasi yang tersedia untuk *android*, dan sekitar 50 miliar aplikasi telah diunduh dari *Google Play*, toko aplikasi *android*.

Berikut adalah kelebihan dan kekurangan dari sistem operasi *android* yaitu:

2.2.2 Kelebihan Android

1. User Friendly

Kata *User Friendly* sangat melekat pada sistem operasi *Windows* milik *Microsoft*, ibarat pengguna dengan sangat mudah mengoperasikan komputernya hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada *android* yang berjalan pada *smartphone*.

2. Notifications

Pengguna dapat dengan sangat mudah mendapatkan notifikasi dari *smartphone android* dengan mengatur beberapa akun pada sebuah aplikasi.

3. Tampilan

Android tidak kalah bagus dari IOS milik Apple, karena memang dari awal *android* hampir mengusung teknologi IOS, hanya saja dapat dibilang ini versi murahnya.

4. Open Source

Operating system ini memang dibuat *open source* oleh penciptanya, karena memang berbasis kernel Linux dan sangat banyak *Custom Room* yang dibuat untuk masingmasing perangkat *android*.

5. Aplikasi

Sangat banyak aplikasi yang disajikan bahkan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar.

2.2.3 Kekurangan *Android*

1. *Update System*

Untuk melakukan *update system* dapat dibilang tidak mudah, pengguna harus menunggu dari masing-masing Vendor untuk merilis Update Versi yang terbaru.

2. Baterai Cepat Habis

Masalah ini sering terjadi jika pengguna menyalakan paket data dan menggunakan *widget* serta aplikasi yang berjalan secara berlebihan.

3. *Lag*

Sebenarnya ada kaitannya dengan spesifikasi dari masing-masing perangkat, namun ada kalanya *android* ini tidak bersahabat dengan beberapa aplikasi dikarenakan Ram ataupun Prosesornya yang kurang memadai.

a. *Library*

Library adalah program yang berisikan data *script* dari program *java* dan merupakan fitur yang sudah otomatis beroperasi dalam android studio, *library* ini sendiri pada android studio bersifat *plug and play* jadi hanya perlu mengunduh dan memanggil dalam perintah program (Nazarudin, 2011).

b. *Kernel*

Linux Kernel adalah lapisan dimana inti dari sistem operasi *android* itu berada, berisi file sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. *Linux kernel* yang digunakan *android kernel* adalah 3.10.28 (Nazarudi, 2011).

Sejak tahun 2008, *android* mulai secara bertahap melakukan sejumlah pembaruan atau *update* untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur baru, memperbaiki *bug* pada versi *android* yang sebelumnya. Setiap versi yang dirilis dinamakan secara *alfabetis* dengan berdasarkan nama sebuah makanan pencuci mulut, seperti *cupcake*, *donut*, dan sebagainya. Berikut nama-nama versi *android*:

- a. *Android* (1.0)
- b. *Cupcake* (1.2 – 1.5)
- c. *Donut* (1.6)
- d. *Éclair* (2.0 – 2.1)
- e. *Froyo* (2.2 – 2.2.3)
- f. *Gingerbread* (2.3 – 2.3.7)
- g. *Honeycomb* (3.0 – 3.2.6)
- h. *Ice Cream Sandwich* (4.0 – 4.0.4)
- i. *Jelly Bean* (4.1 – 4.3)
- j. *Kit Kat* (4.4+)
- k. *Lollipop* (5.0)
- l. *Marshmallow* (6.0)

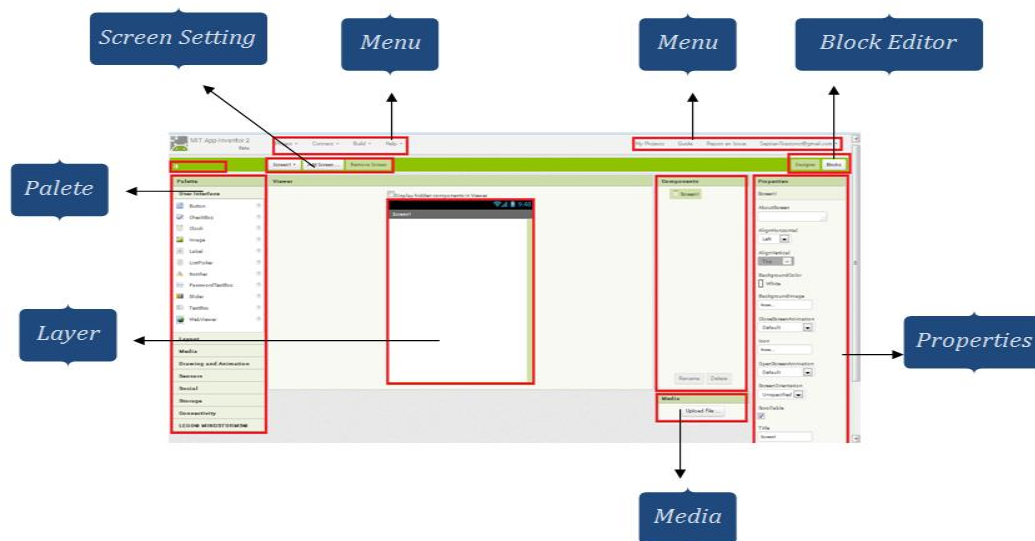
2.3 MIT App Inventor 2

App Inventor for Android adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-*maintenance* oleh Massachusetts Institute of Technology (MIT). *App Inventor* menggunakan bahasa *Kawa Language*

Framework dan *Kawa's dialect* yang dikembangkan oleh Per Bothner. Kedua aplikasi tersebut dijadikan sebagai compiler dan menerjemahkan *Visual Block Programming*.

2.3.1 Area kerja MIT App Inventor 2

Untuk dapat menggunakan *App Inventor 2* diperlukan pengenalan tentang area kerja dari *App Inventor 2* tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 2.1.



Gambar 2.1. Area Kerja *App Inventor 2*

Berikut ini penjelasan dari masing-masing elemen yang terdapat pada area kerja *App Inventor 2*:

1. *Screen Setting* merupakan sebuah kelompok yang berguna untuk mengatur layar, menambah layar, dan menghapus layar.
2. *Palette* adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain.

3. Menu merupakan sekelompok menu yang berguna dalam membuat *project* baru, proses *debugging*, konversi file apk, dll.
4. *Block Editor* adalah suatu tombol untuk masuk ke halaman kode blok untuk proses pengkode-an.
5. *Properties* : untuk mengatur komponen yang telah di buat menjadi desain di layer.
6. Media : Tempat dimana untuk meng-upload file.
7. Layer : Area untuk men-desain.

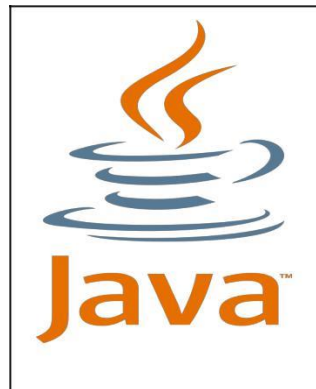
2.3.2 Java

Sejarah *Java*

Java dalam ilmu komputer, merupakan bahasa pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh Sun Microsystem Inc., yang saat *Java* diciptakan, dipimpin oleh James Gosling (Nugroho, 2008). Bahasa pemrograman java tercipta berawal dari sebuah perusahaan Sun Microsystem yang ingin membuat sebuah bahasa pemrograman yang dapat berjalan di semua *device* tanpa harus terikat oleh platform yang digunakan oleh *device* tersebut, terlaksanalah sebuah proyek yang dipelopori oleh Patrick Naughton, James Gosling, Mike Sheridan dan Bill Joy pada tahun 1991, maka terciptalah bahasa pemrograman *java* yang awalnya bernama "Oak".

Java adalah sebuah teknologi dimana pada teknologi tersebut mencakup *java* sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman 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 pemrograman *java* (Rickyanto, 2003). Alasan terbesar dalam pembuatan bahasa pemrograman *java* adalah keinginan akan terbentuknya suatu bahasa pemrograman yang bisa berjalan di berbagai perangkat tanpa harus terikat oleh *platform*, sehingga *java* ini bersifat *portable* dan *platform independent* (tidak tergantung mesin atau sistem operasi). Berikut ini adalah ikon dari bahasa pemrograman *java* :



Gambar 2.2. Contoh *Icon Java*

Karakteristik *Java*

Adapun karakteristik-karakteristik dari *java* secara singkat akan diuraikan di bawah ini (Rickyanto, 2003).

a. Sederhana

Bahasa pemrograman *java* banyak menggunakan sintaks dari C++ yang sudah terkenal sehingga *java* tidak menyulitkan bagi para programmer.

b. Berorientasi Obyek

Java merupakan bahasa pemrograman yang berorientasi obyek secara murni yang artinya dalam pemrograman java adalah obyek kecuali tipe data primitif.

c. Dapat didistribusikan dengan mudah

Java didesain untuk berjalan pada lingkungan yang terdistribusi seperti halnya internet.

d. Aman

Pemrograman *java* memiliki *library security* serta *policy* yang membatasi akses applet di komputer *client*.

e. *Interpreted*

Java dapat dieksekusi pada *platform* yang berbeda-beda dengan melakukan interpretasi pada *bytecode*.

f. *Portable*

Kemampuan *java* yang bisa dijalankan di berbagai macam *platfor*

g. *Multithreading*

Kemampuan untuk meng-*handle* serta menjalankan banyak *threads* dalam waktu yang sama.

h. Dinamis

Bahasa pemrograman *java* merupakan teknologi yang terus berkembang.

i. Netral terhadap arsitektur perangkat keras

Dapat berjalan di setiap komputer yang memiliki arsitektur yang berbeda-beda.

j. Robust

Dapat membantu programmer dalam menghasilkan program secara cepat dan handal karena dapat mencegah berbagai *error* yang mungkin terjadi dengan adanya berbagai proses pengecekan awal pada kompilasi.

2.3.3 *Smartphone*

Ponsel pintar atau yang lebih populer dengan *smartphone* merupakan sebuah ponsel yang memiliki OS (*Operating System*) yang terdiri dari atas kombinasi dari fitur sebuah PC (*Personal Computer*) dan fitur yang terdapat pada ponsel juga tentunya. Sebagian besar *smartphone* dapat digunakan untuk mengakses internet, dengan layar sentuh, dan dilengkapi dengan kamera. *Mobile Operating System* yang digunakan untuk *smartphone* terdiri dari beberapa jenis, yaitu:

1. *Android*
2. *IOS*
3. *Windows Phone*
4. *Blackberry*
5. *Firefox OS*
6. *Sailfish OS*
7. *Ubuntu Touch*

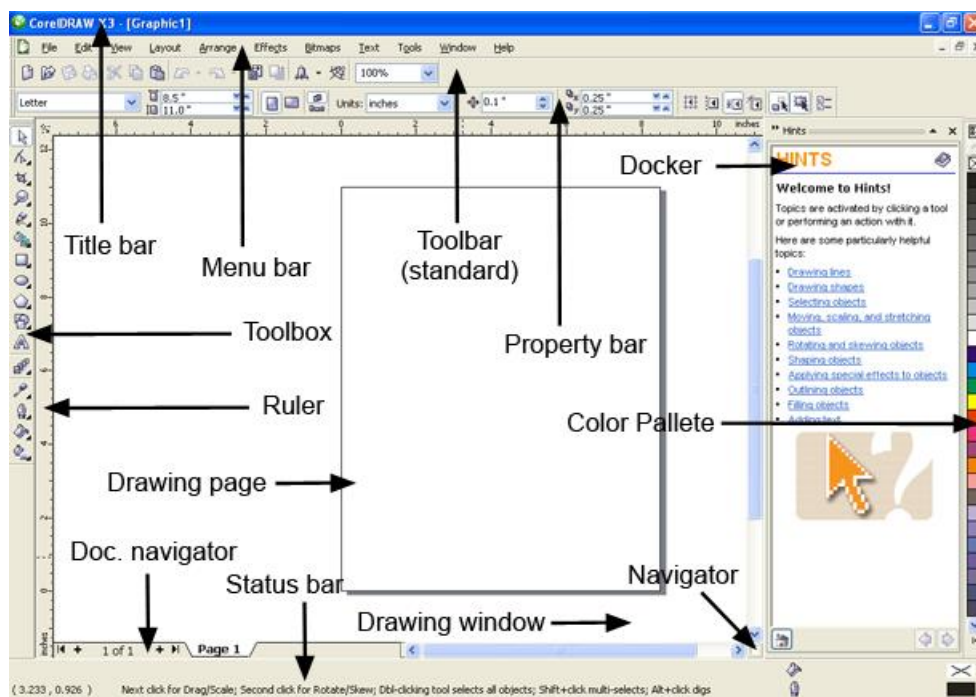
2.4 *Corel Draw X7*

Corel Draw merupakan salah satu program komputer desain grafis yang sudah banyak dikenal dan digunakan oleh pada desainer grafis profesional.

Program ini dibuat oleh sebuah perusahaan software yang berkantor di Ottawa, Kanada. *Corel Draw* memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan di dalam bidang publikasi, percetakan atau pekerjaan di bidang lain yang membutuhkan proses visualisasi. Menurut saya ini sangat membantu sekali dalam pekerjaannya berhubungan dengan menggambar.

2.4.1 Area kerja *Corel Draw*

Untuk dapat menggunakan *Corel Draw* diperlukan pengenalan tentang area kerja dari *Corel Draw* tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 2.3.



Gambar 2.3 Area Kerja *Corel Draw*

Berikut ini penjelasan dari masing-masing elemen yang terdapat pada area kerja *Corel Draw*.

1. *Title Bar*

Title Bar merupakan area yang berisi judul/nama file yang sedang dibuka di *Corel Draw*.

2. *Menu Bar*

Menu Bar merupakan area yang berisi menu-menu perintah yang terorganisasi sesuai dengan kategori masing-masing.

3. *Toolbar / Standar bar*

Toolbar berisikan ikon-ikon yang mempunyai fungsi tertentu dan berfungsi untuk mengakses perintah atau *tool* tanpa harus melalui menu *bar*.

4. *Property Bar*

adalah bagian yang berisi informasi mengenai *tool*/perintah yang aktif (*Active Tool*) atau sedang kita gunakan. Isinya pun selalu berubah-ubah sesuai dengan *tool* yang sedang kita gunakan.

5. *Window Control*

adalah sekumpulan tombol yang berfungsi untuk mengatur tampilan jendela *Corel Draw* atau jendela *drawing*.

6. *Tool Box*

merupakan bagian utama *Corel Draw* yang biasa disebut kotak perkakas/peralatan dimana di dalamnya terdapat tool-tool yang digunakan untuk mendesain di *Corel Draw*. Secara default hanya satu ikon *tool* saja yang ditampilkan, beberapa *tool* lainnya tersembunyi di dalamnya. Tanda bahwa di dalam ikon *tool* tersebut masih ada *tool* yang tersembunyi adalah tanda segitiga kecil di sebelah kanan bawah ikon. Cara memunculkan ikon tersebut cukup

dengan klik tanda segitiga tersebut atau klik dan tahan pada *tool* yang ada tanda segitiganya. Secara *default* letak *tool box* berada di sisi kiri jendela *Corel Draw*.

7. *Drawing Window*

adalah area di luar *Drawing Page* yang bisa juga digunakan untuk mendesain.

8. *Drawing Page*

Drawing Page merupakan area lembar kerja berbentuk segiempat yang ukurannya bisa disesuaikan sesuai keinginan. Area ini juga bisa disebut *Printable Area* karena hanya objek yang berada didalamnya saja yang bisa dicetak, jika suatu objek berada diluar area *Drawing Page* ini maka tidak akan ikut tercetak dan langsung dipindah ke *Master Page* secara otomatis.

9. *Ruler*

Ruler merupakan alat bantu yang berbentuk mirip penggaris, berfungsi untuk mengukur suatu objek secara pasti dan presisi. secara *default* *Ruler* ini sudah tersedia secara *Vertikal* dan *Horizontal* (di bagian atas dan kiri jendela *Corel Draw*).

10. *Color Palette*

Color Palette merupakan bagian yang berisi sekumpulan warna yang bisa digunakan langsung untuk mewarnai objek kurva atau teks dengan cara klik objeknya terlebih dahulu lalu klik warna yang diinginkan.

11. *Docker*

Merupakan jendela yang berisi perintah dan pengaturan yang relevan dengan alat/*tool* tertentu.

12. *Document Navigator*

Document Navigator adalah bagian yang berisi tombol untuk berpindah dari satu halaman ke halaman yang lainnya, *Document Navigator* ini berada di pojok kiri bawah *Drawing Window*.

13. *Status Bar*

Status Bar berfungsi untuk menampilkan status objek yang sedang aktif dan letak posisi kursor di jendela *Corel Draw*.

14. *Navigator*

Navigator adalah tombol yang berada dipojok kanan bawah *Drawing Window* yang dapat membantu pengguna untuk bergerak menjelajahi area *Drawing Page*. Cara menggunakannya cukup dengan klik dan tahan lalu gerakkan sesuai keinginan.

2.4.2 *Adobe photoshop CS5*

Adobe photoshop merupakan salah satu program pengolah gambar standar professional. Tutorial *adobe photoshop* merupakan perangkat lunak editor citra buatan *adobe* sistem yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Hal tersebut dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Photoshop CS5*

Berikut ini elemen penjelasan dari masing-masing area kerja *photoshop* di antaranya dapat dilihat sebagai berikut.

- *Tool Box* bagian ini merupakan berisi tentang berbagai jenis *tools* yang digunakan untuk melakukan perubahan, manipulasi, seleksi, menghapus dan sebagainya.
- *Status Bar* adalah keterangan suatu *tools* yang sedang berjalan/aktif, serta keterangan gambar yang sedang aktif
- Di dalam Menu File terdapat pilihan seperti *Save As* (menyimpan gambar), *Open* (Membuka obyek yang akan diedit), *Export*, *Import*, *Print* (cetak hasil karya) dll.
- Di dalam Menu *Image* terdapat pilihan seperti mengatur warna, *Hue/Saturation*, *Brightness/Contrast*, *Image Size* (ukuran *image*)

- .Layer merupakan tempat didalam Menu *Layer* yang digunakan untuk pilihan seperti menambahkan *layer*, menghapus *layer*, menambahkan efek pada *layer*, dan lain sebagainya.

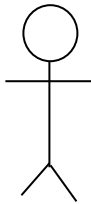
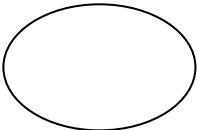

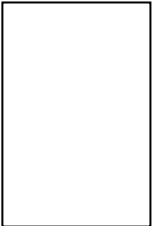
2.4.3 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Berikut di bawah ini adalah Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

a. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah keterangan tentang tabel *Use Case Diagram*.

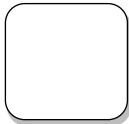



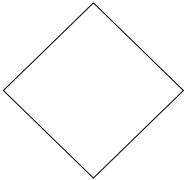
Tabel 2.1 *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Orang yang berada di luar system
	<i>User case</i>	Digunakan untuk fungsional system
	<i>Association</i>	Apa yang dihubungkan antara objek satu dengan objek yang lain
	Sistem	Batasan gambaran sistem dalam sebuah kasus

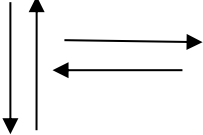
a. Activity Diagram

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Tabel 2.2 Activity Diagra

Simbol	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing antarmuka saling berinteraksi satu sama lain
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial node</i>	Bagaimana obyek dibentuk atau diawali
	<i>Activity final node</i>	Bagaimana obyek diawali dan di akhiri
	<i>Decision</i>	Bagaimana menggambarkan keputusan/tindakan yang harus diambil pada kondisi tertentu

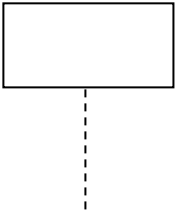
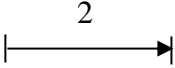
Tabel 2.2 Lanjutan *Activity Diagram*

	<p><i>Line connector</i></p>	<p>Digunakan untuk menghubungkan satu simbol dengan simbol yang lain</p>
---	------------------------------	--

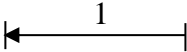
b. *Sequence Diagram*

Suatu *sequence* diagram adalah suatu penyajian perilaku yang tersusun sebagai rangkaian langkah-langkah percontohan dari waktu ke waktu. *Sequence diagram* digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil.

Tabel 2.3 *Sequence Diagram*

Simbol	Nama	Kegunaan
	<p><i>Life Line</i></p>	<p>Obyek <i>entity</i>, antarmuka yang saling berinteraksi</p>
	<p><i>Message</i></p>	<p>Spesifikasi dari komunikasi antar obyek yang memuat informasi tentang aktifitas yang terjadi</p>

Tabel 2.3 Lanjutan *Sequence Diagram*

	<i>Message</i>	Spesifikasi dari komunikasi antar obyek yang memuat informasi tentang aktifitas yang terjadi
---	----------------	--

2.5 Basis Data (*Database*)

Database merupakan kumpulan informasi yang disimpan dalam sebuah komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data (*database*) tersebut (Fathansyah, 2012). Penerapan *database* dalam suatu informasi disebut dengan *database* sistem. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

Database adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi. Selain itu ada pula yang mendefinisikan *database* sebagai kumpulan file, tabel, atau arsip yang saling terhubung yang disimpan dalam media elektronik. *Database* terbentuk dari sekelompok data-data yang memiliki jenis atau sifat yang sama. Sebagai contoh data nama, data kelas, data alamat dikelompokkan dalam data baru yaitu mahasiswa. Demikian juga, kumpulan dari 37 data-data mahasiswa, data-data dosen, data-data keuangan dan lainnya dapat dikumpulkan lagi menjadi kelompok besar, misalkan data-data jurusan atau fakultas pada

sebuah Universitas. Bahkan dalam perkembangannya, data-data tersebut dapat berbentuk berbagai macam data, misalkan dapat berupa program, lembaran-lembaran untuk *entry* (memasukkan) data, laporan-laporan. Semuanya itu dapat dikumpulkan menjadi satu yang disebut dengan *database*. *Database* perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data supaya diorganisasikan sedemikian rupa, sehingga membentuk informasi yang lebih berkualitas. Perancangan basis data merupakan upaya untuk membangun sebuah *database* dalam suatu lingkungan bisnis. Untuk membangun sebuah basis data terdapat tahapan-tahapan yang perlu dilalui yaitu:

1. Perencanaan basis data
2. Mendefinisikan sistem
3. Analisa dan mengumpulkan kebutuhan
4. Perancangan basis data
5. Perancangan aplikasi
6. Membuat *prototype*
7. Implementasi
8. Konversi data
9. Pengujian
10. Pemeliharaan operasional