

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Dalam penelitian ini penulis memaparkan satu penelitian terdahulu yang membahas tentang Perancangan Brosur Interaktif Berbasis *Augmented Reality* (Raymond Rumajar, Arie Lumenta, ST, MT, Brave A. Sugiarso, ST, MT. 2015) menghasilkan sebuah aplikasi yang mampu merancang sebuah brosur *virtual* yang dapat memberikan informasi produk yang lengkap secara 3D, sehingga konsumen dapat mengetahui secara jelas berbagai produk yang ditawarkan. Maka, munculah sebuah teknologi baru dalam bidang perdagangan yang memanfaatkan teknologi *Augmented Reality* sehingga menghasilkan suatu brosur interaktif yang dapat membuat suatu proses bisnis dalam melakukan promosi produk menjadi lebih efektif. Aplikasi di implementasikan dengan penggabungan dari fungsi-fungsi dari pemanggilan kamera, pendeteksian *marker*, memunculkan objek, dan kontrol objek hingga menjadi satu kesatuan aplikasi *Augmented Reality*.

Penelitian selanjutnya yang berjudul Pembuatan *Augmented Reality* tentang Pengenalan Hewan Untuk Anak Usia Dini Berbasis Android Menggunakan Metode *Image Tracking* Vuforia (Riana Indriani, Bayu Sugiarto, Agus Purwanto, 2016) yang menghasilkan sebuah aplikasi *Augmented Reality* pengenalan hewan secara umum kepada anak-anak.

Nurul Huda dan Fitri Purwaningtias membuat penelitian yang berjudul Perancangan Aplikasi Pembelajaran Pengenalan Huruf Dan Angka Berbasis

Augmented Reality(Nurul Huda, Fitri Purwaningtias, 2017) yang merancang sebuah aplikasi *Augmented Reality* pengenalan huruf dan angka, namun sayangnya penelitian ini terbatas hanya pada pembuatan *interface* saja.

Selanjutnya penelitian yang membahas tentang pengenalan satwa langka, penelitian ini berjudul Rancang Bangun Aplikasi Pengenalan Satwa Langka di Indonesia Menggunakan *Augmented Reality*(Christian O. Karundeng, Dringhuzen J. Mamahit, Brave A. Sugiarto, 2018) mampu menghasilkan aplikasi *Augmented Reality* yang dapat mengenalkan satwa langka sehingga dapat mengenalkan dan mengingatkan kembali kepada manusia satwa langka yang dilindungi di Indonesia, objek yang ditampilkan pada penelitian ini dibuat menggunakan *software* blender dengan ukuran objek yang kecil sehingga menghemat penyimpanan.

Dan terakhir, penelitian yang berjudul Edukasi Tata Surya Menggunakan Teknologi *Augmented Reality*(Maura Widyaningsih, Wina Wulandari, 2019) yang mampu menciptakan sebuah aplikasi edukasi untuk pengenalan tata surya dengan teknologi AR sehingga dapat menarik minat belajar anak-anak agar tidak mengalami kejenuhan. Proses pengembangan aplikasi ini dimulai dengan pembuatan objek 3D, perancangan aplikasi, dan perancangan *marker* yang bertujuan untuk menampilkan objek planet yang berbentuk 3 dimensi dengan animasinya serta menampilkan suara yang menggunakan bantuan *marker* yang dibuat disesuaikan dengan gambar planet-planetnya.

II.2. Landasan Teori

Berikut merupakan beberapa teori yang mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa tinjauan pustaka pada penelitian ini yaitu:

II.2.1. Rancang Bangun

Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan. Rancang bangun berawal dari kata desain yang artinya perancangan, rancang, desain, bangun. Sedangkan merancang artinya mengatur, mengerjakan atau melakukan sesuatu dan perancangan artinya proses, cara, perbuatan merancang. Dapat disimpulkan arti kata desain adalah proses, cara, perbuatan dengan mengatur segala sesuatu sebelum bertindak atau merancang. Rancang Bangun adalah kemampuan untuk membuat beberapa alternatif pemecahan masalah . Pendapat lain, rancang bangun adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis (Ariansyah, A., Fajriyah, F. and Prasetyo, F.S., 2017).

II.2.2. Aplikasi

Aplikasi adalah program yang siap pakai yang direka untuk melaksanakan tugas tertentu suatu fungsi bagi pengguna atau aplikasi yang lain dapat digunakan oleh sasaran yang dituju. Aplikasi merupakan sekumpulan elemen yang saling berinteraksi dan saling berketerkaitan antara satu dengan yang lainnya dalam melakukan kegiatan secara bersama-sama untuk mencapai tujuan tertentu. Sebagai

contoh aplikasi komputer yang terdiri dari *software*, *hardware*, *brainware*. Ketiga unsur tersebut saling dan bertanggung jawab dalam proses input atau masukan dan output atau keluaran (Chandra, Y.I. and Kosdiana, K., 2019).

II.2.3. *Augmented Reality*

Augmented Reality adalah penggabungan benda - benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejukan yang efektif. (Nurul Huda, Fitri Purwaningtias, 2017).

Konsep *Augmented Reality* pertama kali dikemukakan oleh Ronald T. Azuma pada tahun 1997 saat ia bekerja di perusahaan Boeing. Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep AR yaitu :

- a. Mampu mengkombinasikan dunia nyata dan dunia maya,
- b. Mampu memberikan informasi secara interaktif dan *realtime*,
- c. Mampu menampilkan dalam bentuk 3D (tiga dimensi).

Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan *virtual* sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan objek) lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (*virtual*). Informasi tentang objek dan

lingkungan disekitar kita dapat ditambahkan ke dalam sistem AR yang kemudian informasi tersebut ditampilkan diatas *layer* dunia nyata secara *real-time* seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh objek *virtual* membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata (Rumanjar. R, Lumenta. A dan Sugiarto B. A, 2015).

AR memiliki tiga karakteristik yaitu bersifat interaktif (meningkatkan interaksi dan persepsi pengguna dengan dunia nyata), menurut waktu nyata (*real time*) dan berbentuk 3 dimensi. Terdapat dua metode AR yang dikembangkan saat ini yaitu *Marker Based Tracking* dan *Markless AR*. Disini penulis menggunakan metode *Marker Based Tracking*. *Marker Based Tracking* adalah AR yang menggunakan *marker* atau penanda objek dua dimensi yang memiliki suatu pola yang akan dibaca komputer melalui media webcam atau kamera yang tersambung dengan komputer(Haryani. P dan Triyono. J, 2017).

II.2.3.1. Arsitektur *Augmented Reality*

Arsitektur teknologi *Augmented Reality* menurut Indriani. R, Sugiarto. B dan Purwanto. A, (2016) seperti yang terlihat pada gambar II.1 berikut ini:



Gambar II.1. Arsitektur *Augmented Reality*

a. *Input*

Input dapat berupa apa saja, contoh *marker*, gambar 2D, gambar 3D, *sensor wifi*, *sensor* gerakan, GPS, dan *sensor-sensor* yang lain.

b. Kamera

Kamera disini sebagai perantara untuk *input* yang berupa gambar, *marker*, gambar 2D maupun 3D.

c. Prosessor

Prosessor dibutuhkan untuk memproses *input* yang masuk dan kemudian memberikannya ke tahapan *output*.

d. *Output*

Dapat berupa *Head Mounted Display* (HMD), *monitor*, seperti *monitor TV*, LCD, *monitor* ponsel.

II.2.4. Unggas

Unggas adalah jenis ternak bersayap dari kelas *Aves* yang telah didomestikasi dan cara hidupnya telah diatur oleh manusia dengan tujuan untuk memberikan nilai ekonomis dalam bentuk daging, telur dan jasa. Telur merupakan salah satu bahan makanan sempurna yang mengandung nilai gizi yang dibutuhkan oleh makhluk hidup (Suryandari, L., Erina, E., Darniati, D., Safika, S., Asmilia, N. and Salim, M.N., 2018).

II.2.5. Silabus

Silabus adalah rencana pembelajaran pada suatu dan/atau kelompok mata pelajaran/tema tertentu yang mencakup standar kompetensi, kompetensi dasar, materi pokok/pembelajaran, kegiatan pembelajaran, indikator, penilaian, alokasi waktu, dan sumber/bahan/alat belajar. Silabus merupakan penjabaran standar kompetensi dan kompetensi dasar ke dalam materi pokok/pembelajaran, kegiatan pembelajaran, dan indikator pencapaian kompetensi untuk penilaian (Faridah, 2019).

Berdasarkan hasil wawancara dan analisis dokumen yang telah dilakukan, Penulis mendapatkan kesimpulan bahwa aplikasi *augmented reality* yang dirancang sesuai dengan silabus, rencana pelaksanaan pembelajaran(RPP) dan lembar kerja peserta didik(LKPD) di SD Negeri 095249 Tapak Kuda untuk materi pelajaran Ilmu pengetahuan Alam, Tema 5, Subtema 1 Pembelajaran 1 yang membahas tentang jenis dan spesies hewan sehingga dapat menunjang proses belajar mengajar.

Sesuai dengan kurikulum 2013 yang bertujuan untuk mempersiapkan manusia agar memiliki kemampuan dan menjadi warga Negara yang beriman, produktif, kreatif, inovatif, dan efektif, serta mampu berkontribusi kepada kehidupan bermasyarakat, berbangsa, bernegara dan peradaban dunia. Proses pembelajaran pada Kurikulum 2013 menekankan pada pendekatan saintifik. Pendekatan saintifik merupakan kegiatan pembelajaran yang dirancang untuk lebih melibatkan siswa secara aktif. Kurikulum 2013 juga dirancang untuk mengembangkan keseimbangan antara pengembangan sikap spiritual dan sosial,

rasa ingin tahu, kreativitas, kerja sama dengan kemampuan intelektual dan psikomotorik (Anggreani, 2018).

II.2.6. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005.

Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi (Aziz, M.M. dan Guntara, R.G., 2019).

Sejarah sistem operasi Android dimulai sejak diumumkannya *Open Handset Alliance* di akhir tahun 2007. Sebenarnya ide untuk membuat sistem operasi *open source* buat peranti *embedded* sudah ada sejak lama. Karena *backing* dari Google yang sangat agresiflah, maka Android bisa sangat terkenal hanya dalam beberapa tahun saja. Beberapa perusahaan telekomunikasi kini mulai memiliki atau menawarkan perangkat android dalam produk – produknya. Tidak

hanya ponsel, tapi perangkat lain seperti tablet, *notebook*, televisi, bahkan hingga ke peranti *embedded* di *mobile* memiliki sistem operasi android. Sebenarnya android pada awalnya tidak dikembangkan oleh google, tapi dikembangkan oleh sebuah perusahaan bernama Android.Inc.

Karena google melihat banyaknya *user* yang *online* dengan perangkat mobile, maka google mengira bahwa perangkat mobile ini memiliki masa depan yang cerah. Sehingga Android Inc diakuisisi oleh Google di tahun 2005. Pada waktu yang sama, Apple di tahun 2007 juga mengembangkan sebuah sistem operasi iOS untuk iPhone – nya dengan fitur – fitur yang sangat inovatif, seperti *multitouch* dan adanya *open market* untuk aplikasi.

Tiap variasi dari aspek – aspek tersebut bisa menyebabkan munculnya perilaku yang tidak terprediksi. Namun mengecek satu – satu untuk variasi tersebut juga tidak memungkinkan. Karena jumlahnya yang sangat banyak. Karena itu, sejak awal Android sudah diatur agar bisa memiliki tampilan yang seragam, dan *user experience* yang sama di semua *platform*. Caranya dengan melakukan abstraksi pada perbedaan *hardware*. Sistem operasi Android menutup aplikasi dari modifikasi yang *hardware specific*. Tapi di sisi lain juga memungkinkan fleksibilitas untuk melakukan *tune up* pada aspek yang diperlukan. Google menyediakan banyak kemudahan untuk pengembang *software* pihak ketiga, seperti *plugin Android Development Tool* (ADT) untuk *Eclipse* (dan juga *tool standalone*) termasuk di dalamnya kemampuan untuk *logging* secara *realtime*, *emulator* realistis yang menjalankan kode ARM *native* dan pelaporan error.

Berikut ini adalah perkembangan Android sejak tahun 2009 sampai dengan sekarang:

Tabel II.1 Versi Android

(Sumber : <https://developer.android.com/about/dashboards/index.html>)

No	Versi	Nama Kode	Tanggal Rilis	Level API
1	1.5	Cupcake	30 April 2009	3
2	1.6	Donut	15 September 2009	4
3	2.0–2.1	Eclair	26 Oktober 2009	7
4	2.2	Froyo	20 Mei 2010	8
5	2.3–2.3.2	Gingerbread	6 Desember 2010	9
6	2.3.3–2.3.7	Gingerbread	9 Februari 2011	10
7	3.1	Honeycomb	10 Mei 2011	12
8	3.2	Honeycomb	15 Juli 2011	13
9	4.0.3–4.0.4	Ice Cream Sandwich	16 Desember 2011	15
10	4.1	Jelly Bean	9 Juli 2012	16
11	4.2	Jelly Bean	13 November 2012	17
12	4.3	Jelly Bean	24 Juli 2013	18
13	4.4	KitKat	31 Oktober 2013	19
14	5.0-5.1	Lollipop	15 Oktober 2014	21
15	6.0	Marshmallow	19 Agustus 2015	23
16	7.0	<i>Nougat</i>	23 Agustus 2016	24
17	7.1	<i>Oreo</i>	30 Januari 2017	25
18	8.0	<i>Oreo</i>	21 Agustus 2017	26
19	8.1	<i>Oreo</i>	5 Desember 2017	27
20	9	<i>Pie</i>	6 Agustus 2018	28
21	10	<i>Pie</i>	September 3, 2019	29

II.2.7. Vuforia

Vuforia adalah *Augmented Reality* Software Development Kit (SDK) untuk perangkat mobile yang memungkinkan pembuatan aplikasi AR. SDK Vuforia juga tersedia untuk digabungkan dengan unity yaitu bernama Vuforia AR Extension for Unity. Vuforia merupakan SDK yang disediakan oleh Qualcomm untuk membantu para developer membuat aplikasi-aplikasi *Augmented Reality* (AR) di mobile phones (iOS, Android). SDK Vuforia sudah sukses dipakai di beberapa aplikasi-aplikasi mobile untuk kedua platform tersebut (Pujabaladika, A.H. and Anifah, L., 2020).

AR Vuforia memberikan cara berinteraksi yang memanfaatkan kamera mobile phones untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda tertentu, sehingga di layar bisa ditampilkan perpaduan antara dunia nyata dan dunia yang digambar oleh aplikasi. Dengan kata lain, Vuforia adalah SDK untuk computer vision based AR. Vuforia ini sendiri merupakan SDK yang dikembangkan oleh Qualcomm untuk membantu pengembang dalam menciptakan aplikasi atau game yang memiliki teknologi *Augmented Reality*. Tentunya aplikasi maupun game yang dibuat dengan teknologi ini akan terlihat lebih interaktif dan hidup (Pujabaladika, A.H. and Anifah, L., 2020).

II.2.8. Grafik Komputer

Grafik komputer adalah sebuah ilmu komputer yang berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital. Bentuk sederhana dari

grafik komputer adalah grafika komputer 2D yang kemudian berkembang menjadi grafik komputer 3D. Teknologi yang sedang berkembang dalam bidang grafik komputer yaitu pemodelan 3D. Dengan adanya pemodelan 3D ini banyak hal yang dapat dimudahkan contohnya dalam bidang industri dimana digunakan untuk pemodelan arsitektur yang akan dibuat seperti bangunan, jembatan, gedung pencakar langit dan lainnya, dalam bidang kesehatan juga digunakan untuk pemodelan organ tubuh manusia, pada bidang perfilman digunakan sebagai karakter dan objek untuk animasi(Charley Corputty, Adang Suhendra, 2015).

Grafika Komputer bertujuan menghasilkan citra (lebih tepat disebut grafik atau picture) dengan primitif-primitif geometri seperti garis, lingkaran dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh data deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan virtual reality(Umam, K. dan Negara, B.S., 2016).

II.2.9. Unity

Unity merupakan *game engine* yang dikembangkan oleh Unity Technologies. *Software* ini pertama kali diluncurkan pada tahun 2005 dan menjadi salah satu dari sekian banyak *game engine* yang dipakai oleh banyak pengembang game profesional di dunia. Unity merupakan alat bantu pengembangan game dengan kemampuan rendering. Unity dapat menciptakan sebuah program interaktif tidak hanya 2 dimensi, tetapi juga dalam bentuk 3 dimensi. Unity adalah sebuah

game engine yang memungkinkan untuk membuat sebuah game 3D dengan mudah dan cepat (Sidiq, P. and Mustafidah, H., 2020).

Secara default, Unity telah diatur untuk pembuatan game bergenre *first person shooting* (FPS), namun unity juga bisa digunakan untuk membuat game bergenre *role playing game* (RPG), *adventure*, *arcade*, *racing* dan *action*. Selain itu unity merupakan sebuah *engine multiplatform* yang memungkinkan game yang dibangun untuk berbagai platform seperti Windows, Mac, Linuk Standalone, Android, iOS webplayer, PS3, Xbox 360 dan juga wii. Unity sendiri juga memiliki *asset Store*. Asset Store merupakan sebuah situs yang memungkinkan kita untuk mendapatkan aset-aset yang dibutuhkan dalam pembuatan sebuah program sendiri, di *asset store* disediakan berbagai jenis kebutuhan mulai dari model karakter, *sound FX*, *script* dan juga *prefarb* (Sidiq, P. and Mustafidah, H., 2020).

II.2.10. Unified Modeling Language (UML)

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak (Urva. G dan Siregar. H. F, 2015).

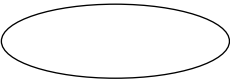
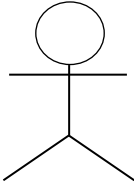
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Urva. G dan Siregar. H. F, 2015).



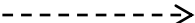
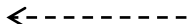
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu :

Tabel II.2. Simbol *Use Case*



Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .


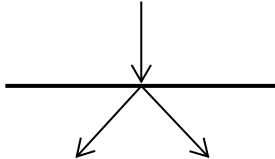
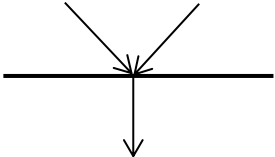
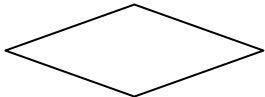
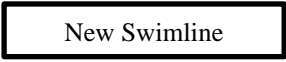


	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

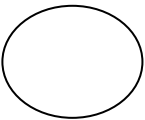
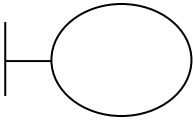
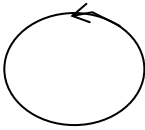

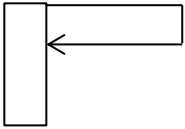


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.

	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
	Signal sebagai pengirim pesan dari aktifitas yang terjadi. Signal pengirim yang digambarkan dengan convex poligon.
	Signal sebagai penerima pesan dari aktifitas yang terjadi. Signal penerima yang digambarkan dengan poligon terbuka.

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.5. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4