BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Menurut penelitian oleh Aldi Apriansyah, dkk (2017) tentang Pengenalan Hewan Dengan Augmented Reality Menggunakan Metode *Marker Less* Pada Platform Android. Penerapan *Augmented Reality* (AR) untuk menghasilkan aplikasi pengenalan hewan dengan konsep aplikasi proses pengenalan hewan, aplikasi di jalankan akan mengenali hewan berdasarkan marker yang berada pada magic book, apabila dikenali maka aplikasi ini akan memberikan informasi berupa objek 3D dan virtual button sebagai audio hewan tersebut. Dengan adanya aplikasi ini Aldi, dkk berharap dapat membantu meminimalisir minimnya alat peraga yang digunakan oleh pengajar.

Menurutpenelitian I Gede Aditya Nugraha, dkk (2016) tentangAplikasi *Android Augmented Reality* pada Museum Bali : Gedung Karangsem dan Gedung Tabanan. Penerapan teknologi *Augmented Reality* (AR) sebagai media promosi Museum Bali, dimana Museum Bali mengalami penurunan pengunjung beberapa tahun terakhir. Aplikasi yang dihasilkan akan mengenali objek 3D yang terdapat pada marker dan memberikan informasi tentang benda – benda yang berada di dalam Museum Bali tersebut, dengan demikian Museum Bali.

Penelitian yang dilakukan oleh Anang Pramono, dkk (2019) Pemanfaatan Augmented Reality Sebagai Media Pembelajaran Pengenalan Buah – Buahan. Penerapan Augmented Reality untuk menghasilkan aplikasi pengenalan buah – buhan sangat direspon baik oleh anak – anak usia dini.konsep aplikasi yang di jalankan akan mengenali buah – buahan berdasarkan marker yang berada pada kartu – kartu bergambar, apabila dikenali maka aplikasi ini akan memberikan informasi berupa objek 3D dan informasi tentang nama buah tersebut.

Menurut penelitian dilakukan oleh Eka Prasetya Adhy Sugara, dkk (2017) Aplikasi Pengenalan Profesi Pekerjaan Bagi Anak Usia Dini Berbasis *Augmented Reality*.Penerapan teknolog i*Augmented Reality* pada pengenalan profesi pekerjaan sebagai media edukasi pembelajaran tentang berbagai profesi seperti astronot, dokter, guru, pilot, polisi dan lain – lain sebagai cita – cita dimasa depan. Aplikasi yang dihasilkan akan mengenali objek 3D berupa animasi profesi pekerjaan berdasarkan marker yang berada pada *magic book***.**

II.2.StudiLiteratur

Kemudian penulis juga melakukan pendekatan teoritis melalui beberapa studi literatur yang berhubungan dengan penelitian yang dilakukan demi mendukung keberhasilan penelitian ini.

II.2.1. Aplikasi

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan yang dapat diartikan juga sebagai program komputer yang dibuat oleh suatu perusahaan computer untuk membantu mengerjakan tugas –

tugas manusia yang terdiri dari beberapa form, dan report, disusun sedemikian rupa sehingga dapat mengakses data. (Wahyuzi Andriansyah, dkk, 2017).

II.2.2. *Augmented Reality*

Augmented Reality menggunakan bahasa pemrograman C# (dibaca "See-Sharp") adalah bahasa pemrograman baru yang diciptakan oleh Microsoft (dikembangkan dibawah kepemimpinan Anders Hejlsberg yang *notabene* juga telah menciptakan berbagai macam bahasa pemrograman termasuk *Borland Turbo C++* dan *Borland Delphi*. (Feby Zulham Adami, 2016).

II.2.3. *Unity* 3D

Unity 3D adalah sebuah aplikasi yang berintegrasi dengan banyak *tools* dan *rapid workflows* yang digunakan untuk membuat konten 3D yang interaktif dan bersifat *multiplatform*. *Unity* termasuk kedalam aplikasi untuk *game engine.* Seperti halnya *game engine* lainya, unity dapat mengolah berbagai bentuk data misalnyaobjek 3D, teksture, suara, dll.Keunggulan dari aplikasi ini yaitu dapat menangani grafik dua dimensi maupun tiga dimensi.Namun, aplikasi ini lebih focus terhadap grafik tiga dimensi. *Unity* 3D memiliki beberapa fitur pendukung yang dapat digunakan untuk merancang game atau aplikasi yang diantaranya yaitu: (Ellinda Dwi Fransiska, dkk, 2017).

II.2.3.1. *Project*

Project adalah kumpulan dari komponen dari aplikasi yang akan dirancang. Pada suatu project terdiri dari *asset, package, scene,* dan lain-lain yang saling mendukung untuk *platform* tertentu. (Ellinda Dwi Fransiska, dkk, 2017).

II.2.3.2. *Asset*

Asset merupakan objek yang menyimpan GameObject , desain, dan lain-lain yang dapat diimport dari berbagai sumber seperti *project* lama atau dari aplikasi lain. (Ellinda Dwi Fransiska, dkk, 2017).

II.2.3.3. *Package*

Package merupakan kumpulan dari *asset-asset* yang dijadikan satu *Package* dapat membantu pengguna untuk mengorganisasikan *assetnya* agar mudah dicari. (Ellinda Dwi Fransiska, dkk, 2017).

II.2.3.4. *Scene*

Scene dapat dikatakan sebagai layar representasi dari aplikasi yang sedang dirancang. *Scene* juga dapat diibaratkan sebagai level dari permainan. *Scene* juga dapat berupa menu dari suatu aplikasi atau option lainnya yang ada. (Ellinda Dwi Fransiska, dkk, 2017).

II.2.4. *Vuforia*

Vuforia adalah *Augmented Reality Software Development Kit(SDK)* untuk perangkat mobile yang memungkinkan pembuatan aplikasi *Augmented Reality*. Dulunya lebih dikenal dengan QCAR *(Qualcomm Company Augmentend Reality). Vuforia* merupakan *Software Development Kit* yang disediakan oleh *Qualcomm* untuk membantu para *developer* membuat aplikasi-aplikasi *Augmented Reality (AR)* di *mobile phones (iOS, Android). AR Vuforia* memberikan cara berinteraksi yang memanfaatkan kamera *mobilephones* untuk digunakan sebagai perangkat masukan, sehingga di layarbisa ditampilkanperpaduanantara dunia nyata dan dunia virtual yang digambar oleh aplikasi(Ellinda Dwi Fransiska, dkk, 2017).

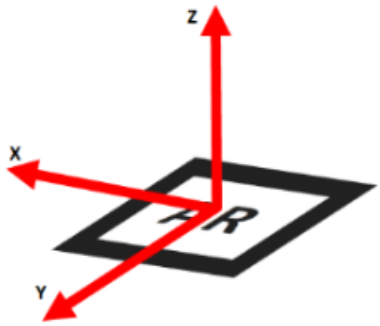
II.2.5. *Marker Based Tracking*

Marker based tracking merupakan metode *Augmented Reality* yang mengenali markerdan mengidentifikasi pola dari marker tersebut untuk menambahkan suatu objek ke lingkungan nyata.*Marker*merupakan ilustrasi persegi hitam dan putih dengan sisi hitam tebal, pola hitam ditengah persegi dan latar belakang putih (Bagus Satria,dkk,2018).Berikutcontoh marker dapatdilihat pada Gambar II.2

Gambar II.1.Contoh Marker

(Sumber : Bagus Satria,dkk:2018)

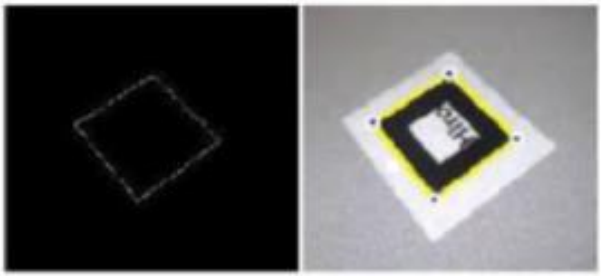
Titik koordinat pada marker berfungsi untuk menentukan posisi dari objek yang akan ditambahkan pada lingkungan nyata. Posisi dari objek akan terletak tegak lurus dengan marker. Objek akan berdiri segaris dengan sumbu Z serta tegak lurus terhadap sumbu X (kanan atau kiri) dan sumbu Y (depan atau belakang) dari koordinat marker. Gambar titik koordinat marker dapat dilihat pada Gambar II.3



Gambar II.2.Titik Koordinat*Virtual*pada Marker

(Sumber : Bagus Satria,dkk:2018)

Proses Tracking, dimana pada tahap ini merupakan tahap prosesor mengolah secara *realtime frame* per *frame* dari tangkapan hasil video. Tahap berikutnya *thres holding image,* dimana tahap ini tiap *frame* video mengalami *thres holding* sehingga menghasilkan gambar hitam putih. Tahap ini bertujuan untuk mengenali segi empat dan pola marker dari video yang telah di tangkap. Proses *contours extraction* dan *corner detection* dapat dilihat pada Gambar II.4



Gambar II.3. *Contours Extraction* dan *Corner Detection*
(Sumber : Bagus Satria,dkk:2018)

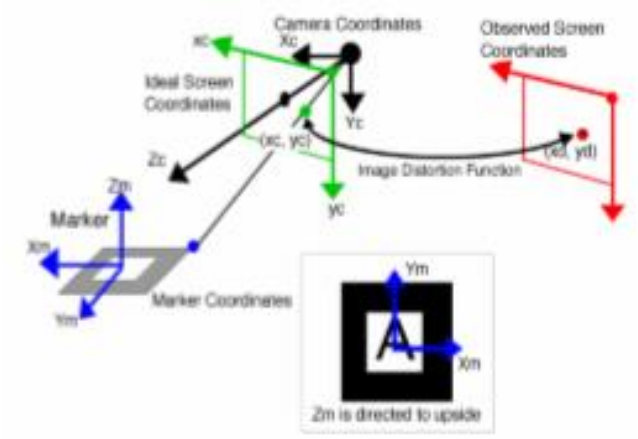
Proses *contours extraction* dan *corner detection*yaitu memanfaatkan gambar hitam putih yang didapat pada tahap kedua untuk mendapatkan koordinat dari empat sisi dan empat titik sudut marker. Berikut gambar dua garis paralel pada *Marker* dapat dilihat pada Gambar II.5

Gambar II.4. Dua garis paralel pada Marker
(Sumber : Bagus Satria,dkk:2018)

Proses berikutnya pada tahap *marker detection* adalah *pattern normalizatiion* dan *template matching* dapat dilihat pada Gambar II.6

Gambar II.5. *Pattern Normalization* dan *Template Matching*
(Sumber : Bagus Satria,dkk:2018)

Tahap terakhir adalah tahap *pose* and *position estimation*. Dimana tahap ini dalam peletakan objek diatas marker. Tahap ini berhubungan antara tiga koordinat, yaitu koordinat dari perangkat tampilan (*observed screen coordinates*). Sistem koordinat dapat dilihat pada Gambar II.7



Gambar II.6. Sistem Koordinat
(Sumber : Bagus Satria,dkk:2018)

II.2.6. Sejarah *Android*

Perjalanan *Android* dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Minner, NickSears dan Chris White mendirikan *Android*.Inc, di California US. Visi *Android* untuk mewujudkan *mobile device* yang lebih peka dan mengerti pemiliknya,kemudian menarikraksasa dunia maya *Google*. *Google* kemudian mengakuisisi *Android* pada Agustus 2005.OS*Android* dibangun berbasis *platform* Linux yang bersifat*open source*,senada denganLinux, *Android* juga bersifat*Open Source.* Dengan nama besar *Google* dan konsep*open source* pada OS Android, tidak membutuhkan waktulama bagi android untuk bersaing dan menyisihkan *Mobile OS* lainnya seperti *Symbian, Windos Mobile,Blackberry* dan iOS. Kinisiapa yang tak kenal *Android* yang telah menjelma menjadi penguasa *Operating System*bagi *Smartphone.*Adapun *Versi - Versi Android*adalah sebagai berikut: (Hendra Nugraha Lengkong, dkk, 2015).

1. *Android 1.5 Cupcake*

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi *Android* pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besardan membuatnya menjadi versi 1.5 *Cupcake* adalah kuekecil yangdipanggang dalam cetakan berbentuk cup.

2. *Android 1.6 Donut*

Android V1.6, codename Donut, dirilis pada 15September 2009. Pada versi ini diperbaikibeberapa kesalahan reboot, perubahan fitur foto danvideo dan integrasi pencarian yang lebih baik. Donatmerupakan panganan berbentuk cincin

bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan toping diatasnya.

3. *Android 2.0/2.1 Eclair*

Android 2.0/2.1 Eclair Dirilis 26 Oktober2009. *Eclair* adalah makanan penutup yakni kue yangbiasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. *Android 2.2 Froyo*

Dirilis 20 Mei 2010 menggunakan *codename* Froyo, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari *Yoghurt. Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah *yoghurt* yang telah mengalami proses pendinginan,sehinggasecara terlihat sama seperti es krim.

5. *Android 2.3 Gingerbread*

Android versi 2.3 *Gingerbread* dirilis resmi tanggal6 Desember 2010. *Gingerbread* merupakan jenis kuekering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. *Android 3.0 Honeycomb*

Dirilis tanggal 22 February 2011. Hadalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, *Honeycomb*/saranglebah, sereal ini terbuat dari potongan jagungberbentuksarang lebah dengan rasa madu.

7. *Android 4.0 Ice Cream Sandwich*

Android 4.0-4.0.2 API Level 14 dan 4.0.3API Level 15 pertama dirilis 19 Oktober 2001. Dinamai*Ice Cream Sandwich*. *Ice Cream Sandwich*es krim,

biasanya rasa vanilla yang terjepit di antara duakue coklat, dan biasanya berbentuk persegi panjang.

8. *Android 4.1 Jelly bean*

Android Jelly Bean diluncurkan pertama kali padaJuli 2012, dengan berbasis *Linux Kernel*dari *Android* 4.1 API Level 16, *Android* 4.2 API Level 17, *Android* 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. *Android 4.4 KitKat*

Android 4.4 *Kitkat* API level 19.Googlemengumumkan *Android KitKat* (dinamai dengan izin *Nestle* dan *Hershey*) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merksebuah coklat yang dikeluarkan oleh *Nestle*. Rilis berikutnya setelah nama *KitKat* diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai '*Pie'.*

10. *Android 5.0 (Lollipop)*

Mulai beberapa *versi* ke belakang, *Android* dan *Google* pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun.Termasuk *Android 5.0 & 5.1 Lollipop* yang dirilis dan diresmikan pada Juni 2014.Bisa dibilang *Android 5.0 & 5.1 Lollipop* menjadi pionir dibuatnya *smartphone flagship* dengan spesifikasi cukup mumpuni.*Versi Android* ini sudah mendukung arsitektur 64-bit yang sudah memungkinan penggunaan RAM di atas 3GB. Salah satunya *ASUS Zenfone 2* yang sudah mengusung RAM 4GB saat itu.

11. *Android 6.0 (Marshmallow)*

Android 6.0 Marshmallow menjadi suksesor dari *versi Android* sebelumnya.Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint* sensor sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint* sensor ini dapat digunakan untuk otentikasi *Google Play Store* dan pembelian dengan menggunakan *Android Pay*.

12. *Android 7.0 (Nougat)*

Untuk saat ini, sistem operasi *Android* ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini.*Android 7.0 & 7.1 Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan *ikon* robot *Android* dengan batangan *Nougat*.Sistem *operasi Android 7.0 & 7.1 Nougat* mengalami perubahan dari segi tampilan antar muka.Selain itu ada juga fitur *splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus.

13. *Andorid 8.0 (Oreo)*

Android 8.0 & 8.1 Oreo menjadi sistem operasi Android paling terbaru yang banyak digunakan saat ini.Sistem operasi ini dirilis secara stabil mulai Agustus 2017 sudah mengalami pembaruan lewat *versi Android 8.1 Oreo*. *Android 8.0 & 8.1 Oreo* menjadi *versi Android* kedua yang menggunakan makanan manis dari nama brand terkenal setelah *Android 4.4 KitKat*. Sistem operasi ini menawarkan pengalaman *multitasking* yang makin mumpuni

dibanding *versi* sebelumnya. Selain itu ada juga *Project Treble* yang memungkinkan pengguna mendapat pembaruan lebih cepat.

14. *Android 9.0 (Pie)(Android.com, 2019)*

Kemudianada*Android 9.0 Pie* yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi *Android*ini memberi banyak ubahan, terutama untuk *HP* dengan desain baru. Misal *Android 9.0 Pie* memberikan navigasi berupa *gesture* yang menggantikan tombol fisik *Home, Back,* dan *Recent Apps*.*Fitur* lainnya yang cukup berguna adalah sistem *notifikasi,* pengatur kecerahan, hingga sistem *screenshoot* terbaru yang lebih memudahkan pemakainya.

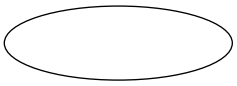
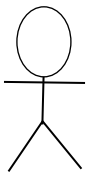
II.2.7. *Unified Modeling Language* (UML)

UML merupakan metodelogi dalam mengembangkan system berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. (Ade Hendini, 2016).

II.2.7.1.*Use Case Diagram*

Use case diagram merupakan suatu pemodelan untuk system informasi aplikasi yang ingin kita buat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Simbol U*se Case Diagram* dapat dilihat pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

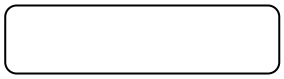
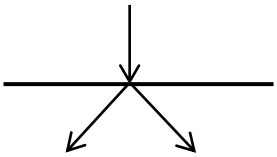
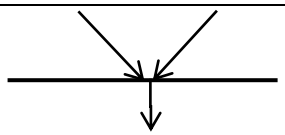
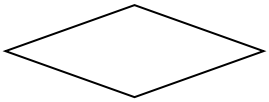
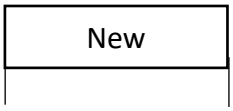
Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini:2016 : 2)

II.2.7.2. *Activity Diagram*

Activity Diagram menggambarkan kegiatan aktivitas dari sebuah sistem atau proses bisnis. Simbol *Activity Diagram* dapat dilihat pada Tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

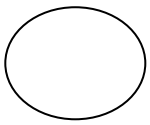
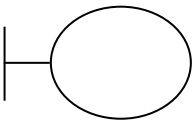
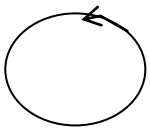
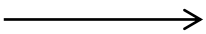
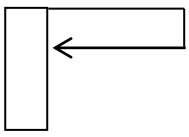
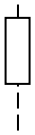
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini:2016 : 2)

II.2.7.3.*Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol *Sequence Diagram* dapa dilihat pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

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

	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Ade Hendini:2016 : 2)

II.2.7.4. *Class Diagram*

Class diagram 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* secarakhasmeliputi: Kelas (*Class*), Relasi*Assosiation*, *Generalitation* dan *Aggregation,* attribut (*Attributes*), operasi (*operation/method*) dan *Visibility,* tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti. Simbol *Multiplicity Class diagram* dapat dilihat pada Tabel II.4.

Tabel II.4. *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

(Sumber : Ade Hendini:2016 : 2)