

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terkait

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal local yaitu:

**Tabel II.1. Penelitian Terkait**

No	Penulis	Judul	Hasil Penelitian
1	Yulianto (2018)	Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal	Kota Samarinda terdapat lebih 20 lapangan futsal, pengguna membutuhkan informasi atau kriteria untuk memilih lapangan futsal terbaik sesuai kebutuhan sedangkan informasi tentang lokasi. Penelitian ini menerapkan formula Haversine untuk pencarian lokasi terdekat, GoogleMaps sebagai pembangun peta digital, dan dikembangkan berbasis website
2	Dzakaul Malik (2019)	Sistem Pemesanan Makanan Tradisional Berbasis Android Menggunakan Metode Haversine Formula	Pengembangan sistem pemesanan makanan tradisional di Kabupaten Pandeglang Provinsi Banten ini berbasis android. Hasil aplikasi pada penelitian ini digunakan untuk membantu penjual makanan tradisional di Kabupaten Pandeglang mempromosikan makanan tradisional yang mereka jual serta meningkatkan retensi pelanggannya dengan memberi kemudahan para pembeli untuk mendapatkan informasi dan memesan makanan tradisional di Kabupaten Pandeglang
3	Bahryan Purmadipta (2016)	Sistem Informasi Geografis Perumahan dan Fasilitas Sosial Terdekat	Saat ini developer masih menggunakan brosur atau media cetak lainnya untuk mempromosikan keunggulan lokasi yang dimiliki perumahannya, hal kurang efektif dan efisien. Disamping itu masyarakat pendatang di Kota Pontianak cenderung kesulitan dalam mencari perumahan maupun fasilitas sosial

		dengan Metode Haversine Formul	terdekat. Oleh karena itu, perlu bantuan teknologi untuk mengatasi masalah yang dialami developer dan masyarakat pada umumnya
4	Abadi Nugroho (2020)	Penerapan Metode Haversine Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana	Tujuan dari penelitian ini adalah membuat aplikasi yang dapat memberikan informasi bencana dan mencari titik kumpul dengan jarak terdekat menggunakan metode Haversine Formula serta teknologi Google Maps. Haversine Formula merupakan sebuah persamaan dalam navigasi dengan cara memberikan jarak radius (lingkaran besar) antara dua titik pada permukaan bola (bumi) berdasarkan lintang dan bujur. Pembuatan aplikasi ini dibangun dengan Android Developer Tools dengan bahasa pemrograman Java.
5	Ida Bagus Gede Purwania (2020)	Rancang Bangun Aplikasi Peringatan Dan Mitigasi Gempa Bumi Berbasis Mobile Hybrid	Penelitian ini dilakukan untuk menghasilkan aplikasi yang dapat memberikan peringatan ke masyarakat ketika terjadi gempa bumi dan memberikan informasi lokasi evakuasi yang terdekat. Selain itu aplikasi ini dirancang agar dapat terhubung dengan teman atau kerabat dekat saat terjadi bencana gempa bumi.

## II.2. Uraian Teoritis

### II.2.1. Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Pengertian perancangan menurut para ahli diantaranya adalah sebagai berikut :

1. Menurut Varzello/John Reuter III perancangan adalah tahap setelah analisis dari siklus pengembang sistem : Pendefinisian dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “ Mengembarkan bagaimana suatu sistem dibentuk”.

2. Menurut John Buch & Gary Grudnitski perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
3. Menurut George M. Scott perancangan adalah menentukan bagaimana sistem akan menyelesaikan apa yang mesti diselesaikan ; tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. (Kholik Hidayatulloh, et. al. ; 2020 : 20)

## **II.2.2. Pengertian Sistem Informasi**

Sistem adalah kumpulan suatu jaringan yang saling berinteraksi atau terhubung untuk melakukan suatu kegiatan untuk mencapai suatu tujuan yang diharapkan. Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Pengertian Sistem Informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi, dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Wili Wildaningsih ; 2018 : 182)

### II.2.3. Sistem Zonasi

Penerimaan peserta didik tahun 2018/2019/dan 2019/2020 di atur dengan Peraturan Menteri Pendidikan dan Kebudayaan Nomor 17 Tahun 2017 tentang Penerimaan Peserta Didik Baru (PPDB) pada Taman Kanak-Kanak, Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas, Sekolah Menengah Kejuruan atau bentuk lain sederajat. Peraturan penerimaan peserta didik baru ini disebut dengan Sistem Zonasi. Sistem Zonasi itu adalah sistem penerimaan peserta didik disetiap awal tahun pembelajaran sebagai masuk di satu lembaga pendidikan melalui mekanisme baik itu jejaring (*daring online*) maupun luar jejaring (*luring offline*) dengan memberikan kesempatan kepada anak-anak yang berada di daerah zonasi tanpa melihat pada nilai, namun lebih melihat pada akses mereka lebih dekat dengan sekolah.

Ketentuan sistem zonasi dicantumkan dalam pasal 15 sampai dengan pasal 17 Permendikbud No 17 Tahun 2017 dimana sekolah wajib menerima paling sedikit 90% peserta didik yang berdomisili pada radius zona terdekat dari sekolah. Keterangan domisili dibuktikan dengan kartu keluarga yang diterbitkan paling lambat 6 bulan sebelum pelaksanaan Penerimaan Peserta Didik Baru. Bicara soal radius zona terdekat ditetapkan sendiri oleh pemerintah daerah masing-masing sesuai dengan kondisi setempat. Namun, apabila berdasarkan analisis kebutuhan masih belum dapat menampung peserta didik yang tersedia sesuai dengan ketentuan zonasi, sekolah dapat melaksanakan secara bertahap disesuaikan dengan kesiapan masing-masing daerah.

Lagi-lagi dalam konsep soal sistem zonasi tersebut diuraikan kembali bahwa sekolah yang diselenggarakan oleh pemerintah daerah dapat menerima calon peserta didik melalui:

- a. Jalur prestasi yang berdomisili di luar radius zona terdekat dari sekolah paling banyak 5% (lima persen) dari total jumlah keseluruhan peserta didik yang diterima.
- b. Jalurbagi calon peserta didik yang berdomisili di luar zona terdekat dari sekolah dengan alasan khusus meliputi pemindahan domisili orangtua/wali peserta didik atau terjadi bencana alam/sosial, paling banyak 5% (lima persen) dari total jumlah keseluruhan peserta didik yang diterimah.

Ditegaskan lagi pada Permendikbud Nomor 14 tahun 2018 pasal sebagaimana tertuang sebagai berikut:

- (1) Sekolah yang diselenggarakan oleh pemerintah daerah wajib menerima calon pesertadidik yang berdomisili pada radius zona terdekat dari Sekolah paling sedikit sebesar 90% (sembilan puluh persen) dari total jumlah keseluruhan peserta didik yang diterima.
- (2) Domisili calon peserta didik sebagaimana dimaksud pada ayat (1) berdasarkan alamat pada kartu keluarga yang diterbitkan paling lambat 6 (enam) bulan sebelum pelaksanaan PPDB.
- (3) Radius zona terdekat sebagaimana dimaksud pada ayat (1) ditetapkan oleh pemerintah daerah sesuai dengan kondisi di daerah tersebut berdasarkan:
  - a. ketersediaan anak usia sekolah di daerah tersebut; dan
  - b. jumlah ketersediaan daya tampung dalam rombongan belajar pada

masing-masing sekolah

- (4) Dalam menetapkan radius zona sebagaimana dimaksud pada ayat (3), pemerintah daerah melibatkan musyawarah/kelompok kerja kepala Sekolah.
- (5) Bagi Sekolah yang berada di daerah perbatasan provinsi/kabupaten/kota, ketentuan persentase dan radius zona terdekat sebagaimana dimaksud pada ayat (1) dapat diterapkan melalui kesepakatan secara tertulis antarpemerintah daerah yang saling berbatasan.
- (6) Sekolah yang diselenggarakan oleh pemerintah daerah dapat menerima calon peserta didik melalui:
  - a. jalur prestasi yang berdomisili diluar radius zona terdekat dari Sekolah paling banyak 5% (lima persen) dari total jumlah keseluruhan peserta didik yang diterima; dan b
  - b. jalur bagi calon peserta didik yang berdomisili diluar zona terdekat dari Sekolah dengan alasan khusus meliputi perpindahan domisili orangtua/wali peserta didik atau terjadi bencana alam/sosial, paling banyak 5% (lima persen) dari total jumlah keseluruhan peserta didik yang diterima. (Amirulah Datuk ; 2020 : 23-25)

#### **II.2.4. Sistem Informasi Geografis**

SIG adalah sistem komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi-informasi yang berhubungan dengan permukaan bumi. Pada dasarnya, istilah sistem informasi

geografi merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografi. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG. Dengan melihat unsur-unsur pokoknya, maka jelas SIG merupakan salah satu sistem informasi. SIG merupakan suatu sistem yang menekankan pada unsur informasi geografi. Istilah “geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau tertukar hingga timbul istilah yang ketiga, geospasial. Ketiga istilah ini mengandung pengertian yang sama di dalam konteks SIG. Penggunaan kata “geografis” mengandung pengertian suatu persoalan mengenai bumi: permukaan dua atau tiga dimensi. Istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui. (Koko Mukti Wibowo : 2015)

#### **II.2.5. Algoritma Haversine**

Formula Haversine adalah persamaan penting dalam sistem navigasi, nantinya Formula Haversine akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (longtitude) dan garis lintang (latitude). Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Ríos di tahun 1801. (Yulianto, 2018)

Istilah haversine ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Ríos menggunakan haversine pertama kali dalam penelitiannya tentang “Masalah Utama Astronomi Nautical“, Proc. Royal Soc, Dec 22. 1796. Haversine digunakan untuk menemukan jarak antar bintang. Formula Haversine adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Formula Haversine merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. (Yulianto, 2018)

Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus haversine: (Yulianto, 2018)

$$\Delta\text{lat} = \text{lat}2 - \text{lat}1 \quad \dots\dots\dots(1)$$

$$\Delta\text{long} = \text{long}2 - \text{long}1 \quad \dots\dots\dots(2)$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2(\Delta\text{long}/2) \quad \dots\dots\dots(3)$$

$$c = 2 \arctan^2(\sqrt{a}, \sqrt{1-a}) \quad \dots\dots\dots(4)$$

$$d = R \cdot c \quad \dots\dots\dots(5)$$

Dimana :

R = jari-jari bumi sebesar 6371(km)

$\Delta\text{lat}$  = besaran perubahan latitude

$\Delta\text{long}$  = besaran perubahan longitude

C = kalkulasi perpotongan sumbu

d = jarak (km) 1 derajat = 0.0174532925 radian

### **II.2.6. PHP (*HypertextPreprosesor*)**

*PHP* adalah pemrograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. *PHP (HypertextPreprosesor)* adalah skrip bersifat *server-side* yang di tambahkan ke dalam *HTML*. *PHP* sendiri merupakan singkatan dari *PersonalHomePageTools*. (Nofyat, et. al. ; 2018 : 13)

### **II.2.7. Database**

*Database* merupakan sekumpulan dari data yang terintegrasi. Sistem *Database* merupakan sistem komputerisasi yang bertujuan untuk menyimpan sejumlah data sehingga memudahkan pengguna untuk mendapatkan dan mengupdate informasi sesuai dengan kebutuhan Informasi yang disimpan dalam *Database* dapat berupa *text* maupun angka. Salah satu penggunaan *Database* dalam kehidupan sehari-hari adalah pada aplikasi *software* pembayaran oleh kasir supermarket. Ketika melakukan pembayaran dikasir, kasir tersebut melakukan pekerjaan dengan cara menempelkan *bar-code* di salah satu sensor *infrared*. Kemudian muncul harga dari barang tersebut dalam komputer. Sehingga kasir tersebut tidak perlu mengetik ulang barang yang yang dibeli oleh pembeli tersebut.

Aplikasi pembayaran oleh kasir merupakan salah satu contoh dari penggunaan *Database* yang terintegrasi oleh komputer. Dalam aplikasi komputer tersebut telah terekam data-data barang beserta harga, sehingga kasir dengan mudah mendapatkan data barang serta saat melakukan input barang, maka *Database* tersebut secara otomatis terupdate. Salah satu keuntungan dari *Database* adalah dapat mempersingkat waktu pekerjaan. Sebuah *Database* dapat menampung data yang kompleks yang dapat digunakan secara internal maupun eksternal. *Software* yang digunakan oleh suatu organisasi untuk mengendalikan *Database* disebut dengan *Database Management System* atau dapat disingkat dengan DBMS. DBMS merupakan suatu kumpulan program yang memudahkan pengguna untuk membuat dan mengelola *Database*. Data yang ada pada DBMS dapat dimanipulasi atau dikelola dengan menggunakan *query*, dimana dengan *query* tersebut pengguna dapat mendapatkan data spesifik yang dibutuhkan, serta dapat mengupdate informasi dalam data-data tersebut. (Intan Mardiono ; 2019 : 103-104)

### **II.2.8. MySQL**

MySQL adalah sebuah perangkat lunak yang terdapat didalam sistem manajemen basis data SQL (database management system) atau yang biasa disebut DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. Dalam membuat mysql yang tersedia di dalam perangkat lunak yang terletak di GPL atau yang biasa di sebut dengan General Public License. Dan tetapi mysql ini dapat menjual

dibawah komersial dalam kasus pengguna bagi yang tidak sama cocok dengan penggunaan General Public License. (Harry Dhika ; 2019 : 107)

### **II.2.9. XAMPP**

XAMPP di artikan sebagai software web apache yang terdapat database mysql dan php programming. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di linux dan windows. Keuntungannyahanya menginstall sekali yang sudah tersedia di dalam web apache, database MySQL, dan PHP programming (php4 dan php5). Hanya bedanya kalau yang versi untuk windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Versi windows ini memiliki kelebihan tersendiri antara lain yaitu mempunyai fitur yang dapat mengaktifkan sebuah server secara free, dan jika di bandingkan dengan linux sangat jauh sekali karena linux masih berupa perintah-perintah di console. Oleh karena itu yang versi untuk linux sulit untuk dioperasikan. Dulu XAMPP untuk linux dinamakan LAMPP, sekarang berganti nama menjadi XAMPP for Linux. (Harry Dhika ; 2019 : 107-108)

### **II.2.10. *Unified Modeling Language (UML)***

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem

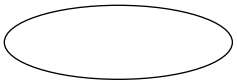
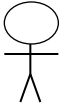
berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2015 : 95).



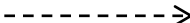
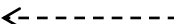
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*

*Use case* diagram merupakan pemodelan untuk kelakuan (*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 dapat dilihat pada tabel II.1 dibawah ini :

**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, dan 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




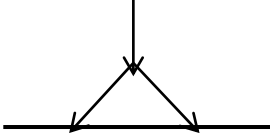
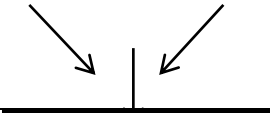
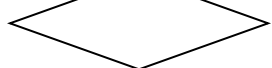
	berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidिकासikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidिकासikan 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.

(Sumber : Urva dan Siregar, 2016 : 94)

## 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* dapat dilihat pada tabel II.2 dibawah ini :

**Tabel II.3. Simbol Activity Diagram**

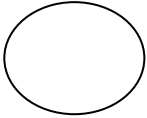
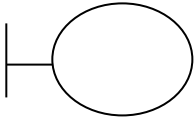
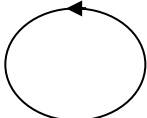
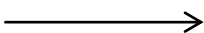
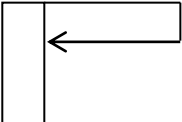


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel 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> .
New Swimlane	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

(Sumber : Urva dan Siregar, 2016 : 94)

### 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* dapat dilihat pada tabel II.3 dibawah ini :

**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> .

(Sumber : Urva dan Siregar, 2016 : 95)

#### 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 yang dapat dilihat pada tabel II.4 dibawah ini :

**Tabel II.5. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
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 : Urva dan Siregar, 2016 : 95)*