

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Tabel II.1. Penelitian Terdahulu

No	Nama	Judul	Hasil Penelitian
1.	Etri Pujiati, dkk (2016)	Peramalan Dengan Menggunakan <i>Metode Double Exponential Smoothing</i> Dari <i>Brown</i> (Studi Kasus: Indeks Harga Konsumen (IHK) Kota Samarinda)	Hasil peramalan IHK Kota Samarinda dari bulan Januari sampai dengan Maret 2015 menggunakan parameter $\alpha = 0,61$ dengan metode <i>Double Exponential Smoothing</i> dari <i>Brown</i> menunjukkan bahwa IHK Kota Samarinda adalah mengalami peningkatan tiap bulannya, dimana diramalkan IHK Kota Samarinda pada bulan Januari sampai dengan Maret 2015 secara berturut-turut sebesar 121,44, 123,06, dan 124,68.
2.	Kurniagara	Penerapan <i>Metode</i>	Hasil Penelitian ini dapat diperoleh

	(2017)	<i>Exponential Smoothing</i> Dalam Memprediksi Jumlah Siswa Baru (Studi Kasus: Smk Pemda Lubuk Pakam)	Laporan jumlah siswa baru dalam satu periode diperlukan untuk melakukan peramalan jumlah siswa baru pada periode berikutnya. Data siswa baru ini nantinya akan di analisa dengan <i>Metode Exponential Smoothing</i> untuk menghasilkan peramalan jumlah siswa baru pada periode berikutnya.
3	Annastasya Lieberty, dkk (2015)	Sistem Informasi Meramalkan Penjualan Barang Dengan <i>Metode Double Exponential Smoothing</i> (Studikases: PD. Padalarang Jaya)	Hasil penelitian ini tentang Aplikasi membantu pengelola perusahaan untuk dalam membantu melakukan peramalan penjualan barang yang dapat memudahkan pengguna dalam penentuan stok barang.

4	Hendini Ade(2016)	Pemodelan UML Sistem Informasi Monitoring Penjualan dan stok barang (Studikusus: Distro Zhezha Pontianak	Penelitian ini menghasilkan system informasi monitoring penjualan dan stok barang ini untuk mempermudah pelaku usaha dalam memantau atau mengetahui penjualan dan stok barang ditiap cabang.
5	Khairun Nisa Ulfa, dkk (2016)	Perancangan Aplikasi Prediksi Jumlah Siswa Baru Pada Yayasan Cerdas Murni Menggunakan <i>Exponential Smoothing</i> .	Metode <i>Exponential Smoothing(Single Exponential Smoothing)</i> dapat diterapkan untuk memprediksi jumlah siswa baru dengan melakukan perhitungan <i>forecase</i> alpha dari 0.1 sampai 0.9 secara acak. Kemudian masing-masing hasil perhitungan diuji nilai eror-nya. Yaitu dengan cara melakukan perhitungan <i>error Mean Square Error (MSE)</i> , dimana yang memiliki nilai eror terkecil merupakan hasil yang terbaik.

II.2. Landasan Teori

Berikut adalah penjelasan dari beberapa teori yang berhubungan dengan judul pada penelitian ini yaitu :

II.2.1. Sekolah Menengah Kejuruan (SMK)

Sekolah Menengah Kejuruan (SMK) merupakan salah satu lembaga pendidikan yang berupaya menciptakan sumber daya manusia yang memiliki kemampuan, keterampilan, dan keahlian sehingga lulusannya dapat mengembangkan keterampilannya agar dapat terjun dalam dunia kerja maupun untuk melanjutkan ke jenjang pendidikan yang lebih tinggi. Menurut Peraturan Pemerintah No. 29 Tahun 1990 pasal 2 ayat (1) menyebutkan bahwa sekolah kejuruan bertujuan untuk meningkatkan pengetahuan siswa dalam pengembangan diri dan untuk meningkatkan kemampuan siswa sebagai anggota masyarakat. Menurut Peraturan Pemerintah No. 29 Tahun 1990 pasal 3 ayat (2) disebutkan bahwa sekolah kejuruan bertujuan untuk menyiapkan siswa dalam memenuhi lapangan kerja, menyiapkan siswa agar mampu memiliki karir, dan menyiapkan tamatan agar menjadi warga Negara yang produktif, adaptif, dan normatif (Kurniagara, 2017: 214)

II.2.2. Peramalan

Peramalan (*forecasting*) merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien khususnya dalam bidang ekonomi. Dalam organisasi moderen mengetahui keadaan yang akan datang tidak saja penting untuk melihat yang baik atau buruk tetapi juga bertujuan untuk melakukan persiapan peramalan.(Yamit, 2003:214). Peramalan adalah prediksi, proyeksi atau estimasi tingkat kejadian yang tidak pasti dimasa yang akan datang. Ketepatan secara mutlak dalam memprediksi

peristiwa dan tingkat kegiatan yang akan datang adalah tidak mungkin dicapai, oleh karena itu ketika perusahaan tidak dapat melihat kejadian yang akan datang secara pasti, diperlukan waktu dan tenaga yang besar agar mereka dapat memiliki kekuatan untuk menarik kesimpulan terhadap kejadian yang akan datang.

II.2.3. Metode Pemulusan (*Smoothing*)

Menurut Subagyo (2002:34), metode *smoothing* merupakan teknik meramal dengan cara mengambil rata-rata dari nilai-nilai beberapa periode yang lalu untuk menaksir nilai pada suatu periode yang akan datang.

II.2.3.1 Pemulusan Eksponensial (*Exponential Smoothing*)

Pemulusan eksponensial (*exponential smoothing*) adalah suatu metode yang menunjukkan pembobotan menurun secara eksponensial terhadap nilai pengamatan yang lebih tua. Oleh karena itu metode ini disebut prosedur *exponential smoothing*. Seperti halnya dengan *moving average*, metode *exponential smoothing* terdiri atas tunggal, ganda, dan metode yang lebih rumit. Semuanya mempunyai sifat yang sama, yaitu nilai yang lebih baru diberikan bobot yang relatif lebih besar dibanding nilai pengamatan yang lebih lama. (Makridakis, Wheelwright dan McGee, 2003:34).

II.2.3.2 Pemulusan Eksponensial Ganda (*Double Exponential Smoothing*) dari Brown

Metode Double Exponential Smoothing merupakan model linear

yang dikemukakan oleh Brown. Dalam metode ini dilakukan proses smoothing dua kali. Dasar pemikiran metode pemulusan eksponensial linear dari Brown adalah serupa dengan rata-rata bergerak linear, karena kedua nilai pemulusan tunggal dan ganda ketinggalan dari data yang sebenarnya jika terdapat unsur trend. Perbedaan antara nilai pemulusan tunggal dan ganda dapat ditambahkan dengan nilai pemulusan tunggal dan disesuaikan untuk trend.

Adapun rumus dari perhitungan metode *Double Exponential Smoothing* adalah sebagai berikut : (Cahyarizki Adi Utama, 2016:148)

1. Pemulusan Eksponensial Tunggal:

$$S'_t = aX_t + (1 - a) S'_{t-1}$$

Dimana S'_t adalah nilai pemulusan eksponensial

2. Pemulusan Eksponensial Ganda:

$$S''_t = aS'_t + (1 - a) S''_{t-1}$$

Dimana S''_t adalah nilai pemulusan eksponensial ganda

3. Pemulusan Trend:

$$a_t = S'_t + (S'_t - S''_t) = 2 S'_t - S''_t$$

Dimana a_t adalah Nilai Konstanta

4. Pemulusan Trend:

$$b_t = \frac{a}{1-a} (S'_t - S''_t)$$

Dimana b_t adalah Nilai *Slope*

5. Ramalan

$$F_t + m = a_t + b_t(m)$$

Dimana m adalah jumlah periode ke muka yang akan diramalkan.

Keterangan :

- ' = Nilai pemulusan tunggal pada waktu ke- t
- " = Nilai pemulusan ganda pada waktu ke- t
- = Data aktual pada waktu ke- t
- , = Konstanta pemulusan
- (+) = Hasil peramalan
- = Periode masa mendatang
- α = Parameter *Exponential* dengan nilai antara 0–1 (Reyhan, 2019:4935).

II.2.7. Visual Basic

Visual Studio 2010 merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi desktop (*Windows Form*, *CommandLine* (Console), Aplikasi Web, *Windows Mobile* (Paket PC). Visual Studio 2010 memiliki lebih dari satu kompiler, SDK (*Software Development Kit*), dan Dokumentasi Tutorial (MSDN Library). Kompiler yang dimasukkan kedalam Visual Studio 2010 antara lain Visual Basic, Visual C#, Visual C++, Visual InterDev, Visual J++, Visual F#, dan Visual Source Safe, dan banyak yang lainnya. Dan semua itu sudah terpaket dan diperuntukkan kedalam *platform .Net Framework 4.0* atau versi yang lebih tinggi.

Visual Studio 2010 memiliki lebih dari satu kompiler, SDK (*Software Development Kit*), dan Dokumentasi Tutorial (MSDN Library). Kompiler

yang dimasukkan kedalam Visual Studio 2010 antara lain *Visual Basic*, *Visual C#*, *Visual C++*, *Visual InterDev*, *Visual J++*, *Visual F#*, dan *Visual Source Safe*, dan banyak yang lainnya. Dan semua itu sudah terpaket dan diperuntukkan kedalam *platform .Net Framework 4.0* atau versi yang lebih tinggi. Visual studio ini dapat digunakan untuk membuat aplikasi yang berbasis desktop yang merupakan *platform windows*, namun juga dapat dijalankan dalam bentuk *Microsoft Intermediate Language* diatas *.Net Framework*. Selain itu Visual Studio juga dapat digunakan untuk membuat aplikasi yang dapat dijalankan diatas windows mobile yang berjalan diatas *.Net Compact Framework*.

Visual Studio 2010 terbagi menjadi beberapa tipe diantaranya :

1. Visual Studio 2010 *Express Edition* yang bisa digunakan secara gratis tanpa memberikan royalti kepada Microsoft Inc.
2. Visual Studio *Standard Edition*.
3. Visual Studio 2010 *Professional Edition* 4. Visual Studio 2010 *Ultimate Edition*. (Rolly Yesputra, 2017:1).

adapun software dan hardware yang dibutuhkan adalah sebagai berikut :

1. Software yang dibutuhkan :
 - a. Windows 7 atau Windows 8
 - b. Visual Studio 2010
 - c. Jika ada permasalahan, install manual *.Net Framework 4.xx*
2. Hardware Yang dibutuhkan :
 - a. Prosesor 1.5 GHz

b. Memory 2 GB

3. Monitor 10". (Rolly Yesputra, 2017:7).

II.2.8. *SQL Server*

SQL merupakan bahasa standar untuk sistem manajemen *database relasional*. Perintah *SQL* digunakan untuk melakukan tugas-tugas seperti perbaruan data, atau mengambil data dari database. Beberapa sistem manajemen *database relasional* umum yang menggunakan *SQL* adalah: *Oracle, Sybase, Microsoft SQL Server, Access, MySQL* dan lain-lain. Meskipun sebagian besar sistem *database* menggunakan *SQL*, kebanyakan dari mereka juga memiliki ekstensi tambahan milik mereka sendiri yang biasanya hanya digunakan pada sistem mereka. Meskipun demikian, perintah-perintah *SQL standar* seperti *Select, Insert, Update, Delete, Create* dan *Drop*, dapat digunakan pada seluruh sistem *database relasional* yang ada saat ini. Pada bab ini akan dibahas dasar-dasar perintah *SQL* serta penggunaannya dalam operasi *database*. Lebih lanjut, *query* adalah pernyataan yang meminta pemanggilan informasi tertentu dari *database*. Sebagian dari DML dinamakan bahasa *query*. Meskipun tidak terlalu tepat, orang sering menyebut seluruh DML sebagai bahasa *query*. (Lantip Diat Prasajo, 2014:10).

1. **Kebutuhan Sistem**

Kebutuhan Sistem Sebelum melakukan instalasi, perlu diketahui spesifikasi system operasi dan perangkat keras (*hardware*) minimal yang

diperlukan oleh *SQL Server* sehingga proses instalasi bisa berhasil. Untuk sistem operasi, *SQL Server 2008 Express Edition* mendukung beberapa versi *Windows*, yaitu: *Windows 7*, *Windows Server 2003*, *Windows Server 2008*, *Windows Server 2008 R2*, dan *Windows Vista*.

SQL Server R2 ini merupakan *SQL Server* versi ringkas, sehingga memiliki berbagai keterbatasan jika dibandingkan dengan versi lengkapnya. Salah satu keterbatasan yang penulis temukan saat menggunakan *SQL Server* versi ini adalah hanya bisa mengeksekusi paket data kurang dari 10 GB (sepuluh giga byte). Namun demikian, *SQL Server Express Edition* lebih dari memadai untuk digunakan sebagai sistem basis data untuk aplikasi yang cukup kompleks namun tidak begitu memerlukan tingkat keamanan yang tinggi. Karena versi ringkas, tentunya kebutuhan perangkat keras tidak sama dengan versi lengkapnya.

Berikut adalah spesifikasi perangkat keras minimal yang diperlukan untuk bisa melakukan instalasi *SQL Server 2008 R2*.

- a. Prosesor : Pentium III atau prosesor setara lainnya dengan kecepatan 1 GHz atau lebih tinggi.
- b. Memori: Minimal 500 MB untuk *SQL Server Express* dengan *Tools* dan *Advanced Services*, dan 4 GB untuk *SQL Server* dengan *Reporting Services*.
- c. Harddisk : 2.2 GB, belum termasuk space untuk sistem operasi dan *database* aplikasi yang akan Anda kembangkan menggunakan *SQL Server*. (Lantip Diat Prasojo, 2014:72).




II.2.9. UML


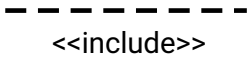

Unified Modeling Language (UML) adalah suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. *Unified Modeling Language* (UML) terdiri dari beberapa jenis, namun dalam rancangan ini jenis UML yang digunakan yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

1. *Diagram Use Case*

Use case adalah abstraksi dari interaksi antara *system* dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

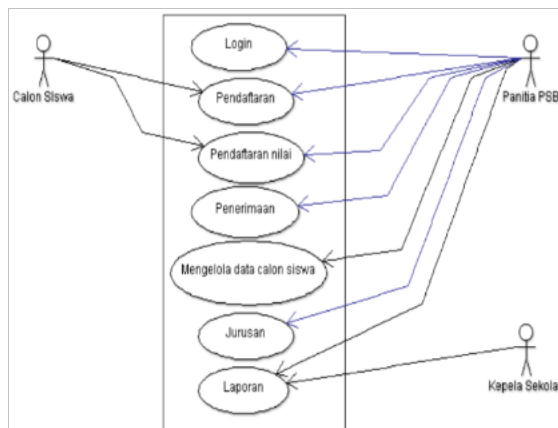
Tabel II.2. Simbol-simbol Use Case

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem.
	Aktor adalah orang atau sistem yang melakukan tugas atau kegiatan.
	Asosiasi antara actor dan <i>use case</i> interaksi secara langsung.

	Asosiasi antara actor dan <i>use case</i> interaksi secara pasif dengan sistem.
	<i>Include</i> , pemanggilan <i>use case</i> oleh <i>use case</i> lain.
	<i>Extend</i> , perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini ; 2016:108)

Contoh *Use Case diagram* dapat dilihat pada gambar II.1. dibawah ini :



Gambar II.1. Use Case Diagram Penerimaan Siswa Baru




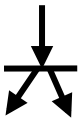

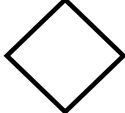
(Sumber : Sri Rahayu,2017:258)

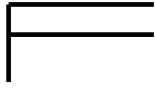
2. Activity Diagram

Activity diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan

aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan paralel. Pada dasarnya diagram Activity sering digunakan oleh *flowchart*, diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagasimana aktifitas-aktifitas tersebut bergantung satu sama lain.

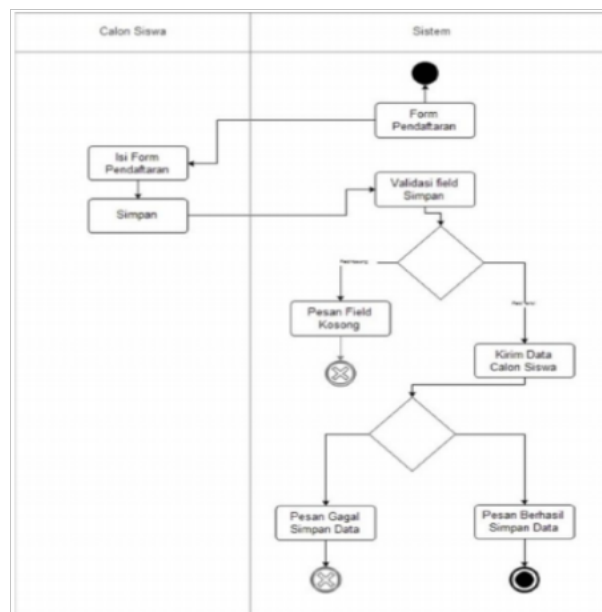
Tabel II.3. Simbol – Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , diletakan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> merupakan akhir dari aktifitas.
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> atau percabangan digunakan untuk menunjukan kegiatan yang dilakukan secara parallel.
	<i>Join</i> (penggabungan) atau <i>raked</i> digunakan untuk menunjukan adanya dekomposisi.
	<i>Decision Points</i> menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .

	<p><i>Swimlane</i> merupakan pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p>
---	---

(Sumber : Ade Hendini, 2016:109)

Contoh *Activity Diagram* dapat dilihat pada gambar II.2. dibawah ini :



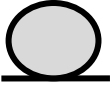
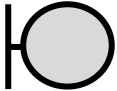


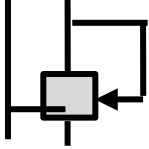


Gambar II.2 *Activity Diagram Login Calon Siswa*

(Sumber : Ismail Mohidin, 2018:41)

3. *Sequence Diagram*

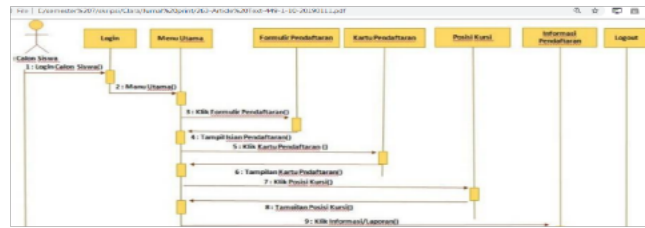
Sequence Diagram merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Tabel II.4. Simbol-simbol Sequence Diagram

Gambar	Keterangan
	<p><i>Entity Class</i> merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas – entitas yang membentuk gambaran awal sistem.</p>
	<p><i>Boundary Class</i> berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem.</p>
	<p><i>Control Class</i> merupakan suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.</p>
	<p><i>Message</i> merupakan simbol untuk mengirim pesan antar <i>class</i>.</p>
	<p>Recursive menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i> mewakili sebuah eksekusi operasi dan kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i> merupakan garis titik – titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Ade Hendini, 2016:110)

Contoh *Sequence Diagram* dapat dilihat pada gambar II.3. dibawah ini :



Gambar II.3. *Sequence Diagram* Aktor Calon Siswa

(Sumber : Ismail Mohidin, 2018:41)

4. *Class Diagram*

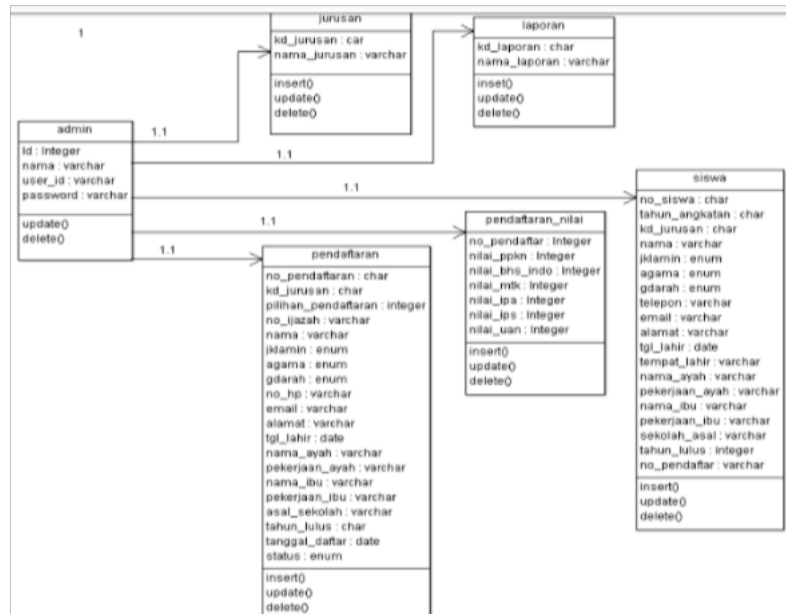
Class Diagram merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* adalah kondisi objek yang dinyatakan dalam attribute/properties. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

Tabel II.5. Simbol – Simbol *Class Diagram*

<i>Multiplicity Diagram</i>	<i>Class</i>	Keterangan
1		Satu dan hanya satu.
0..*		Boleh tidak ada atau satu atau lebih.
1..*		1 atau lebih.
0..1		Boleh tidak ada maksimal 1.
n..n		Batasan antara. Contoh 2.4 mempunyai arti minimal 2 maksimal 4.

(Sumber : Ade Hendini, 2016:111)

Contoh *Class Diagram* dapat dilihat pada gambar II.4. dibawah ini :



Gambar II.4. *Class Diagram* Penerimaan Siswa Baru

(Sumber : Ramdhan, Rizal: 2016:258)