

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya yang pertama dengan judul “Sistem Penjadwalan Mata Kuliah pada Jurusan Pendidikan Teknik Elektro Fakultas Teknik Universitas Negeri Makassar Menggunakan PHP” dilakukan oleh Selviyanti dkk [4]. Sistem ini merupakan pengembangan dari sistem sebelumnya, dengan permasalahan utama yaitu lamanya waktu, tenaga, dan tingkat ketelitian yang dibutuhkan dalam pembuatan jadwal kuliah, serta hasil yang belum optimal. Oleh karena itu, penulis melakukan penelitian dan pengembangan dengan menggunakan metode ADDIE, yang menghasilkan beberapa hal sebagai berikut:

1. Pengembangan penjadwalan mata kuliah pada Jurusan Pendidikan Teknik Elektro Fakultas Teknik Universitas Negeri Makassar menggunakan PHP dapat memudahkan pengguna dalam melakukan dan menyampaikan informasi terkait penjadwalan mata kuliah.
2. Hasil pengujian sistem menggunakan standar ISO 9126 dengan 4 karakteristik, yaitu functionality (100%), reliability (100%) tanpa adanya error, serta melalui pengujian white box tidak ditemukan kesalahan logika. Sistem ini juga portabel dan dapat digunakan di berbagai platform, seperti Windows 7, Windows 10, Android Samsung Galaxy Note 4, iPhone 6S, dan Mac High Sierra. Pengujian usability menunjukkan hasil rata-rata tanggapan responden sebesar 71.60 dengan kategori sangat baik.

Penelitian kedua dengan judul “Rancang Bangun Sistem Penjadwalan pada SMA Muhammadiyah Al Kautsar Program Khusus Kartasura” dilakukan oleh Novendius Eka Saputra dkk [5]. Sebelum adanya sistem ini, terdapat beberapa masalah dalam penjadwalan yang belum terkomputerisasi di sekolah tersebut. Hal ini menyebabkan seringnya terjadi kesalahan manusia (*human error*) dan perubahan jadwal yang mendadak, yang kemudian memerlukan penyusunan ulang jadwal yang memakan waktu yang cukup lama. Dalam penelitian ini, penulis membangun sebuah sistem informasi dengan menggunakan metode

waterfall. Berdasarkan hasil pengujian yang telah dilakukan, pembangunan sistem ini berhasil berjalan dengan baik.

Penelitian ketiga dengan judul “Rancang Bangun Sistem Informasi Penjadwalan Mata Pelajaran Berbasis Web” dilakukan oleh Manase Sahat H Simarankir dkk [6]. Permasalahan yang dihadapi adalah pelaksanaan penjadwalan masih dilakukan secara manual menggunakan Microsoft Excel dan Microsoft Word sehingga dinilai kurang efektif. Maka dari itu, dibuatlah sistem informasi penjadwalan yang terkomputerisasi dengan menggunakan metode pengembangan sistem waterfall. Hasil dari penelitian ini adalah sistem informasi berbasis web yang dapat membantu proses pengolahan data dengan lebih terstruktur dan meminimalisir terjadinya kehilangan data.

Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian sebelumnya. Fokus penulis adalah mengembangkan sebuah sistem yang dapat mengatasi permasalahan dalam pembuatan penjadwalan Ujian Tengah Semester dan Ujian Akhir Semester yang memakan waktu dan rentan terjadi kesalahan manusia. Selain itu, masalah lain yang akan ditangani adalah kurangnya efektivitas dalam persebaran informasi yang masih mengandalkan aplikasi general. Selain itu, aplikasi ini akan dibuat agar dapat mencetak kartu ujian hanya untuk mahasiswa yang telah melunasi pembayaran. Aplikasi yang akan dibuat menggunakan bahasa pemrograman PHP dan *framework* laravel dengan menerapkan metode *prototype*.

2.2 Landasan Teori

Dalam menunjang penelitian ini, maka diperlukannya teori-teori mendasar. Teori-teori yang digunakan dalam penelitian ini adalah:

2.2.1 Sistem Informasi

Sistem informasi adalah hubungan sekumpulan komponen yang terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya data [7]. Sistem informasi juga dapat berarti kumpulan komponen yang saling terhubung untuk mengelola informasi, mulai dari pengumpulan, pemrosesan, penyimpanan, hingga distribusi informasi.

2.2.2 Penjadwalan

Penjadwalan merupakan kegiatan yang sangat mudah ditemui di instansi pendidikan, salah satunya di PNC. Beberapa contoh penjadwalan di PNC ialah jadwal akademik, jadwal sidang, dan jadwal ujian untuk mata kuliah berbasis teori.

Di sisi lain, penjadwalan memiliki arti pengalokasian sumber daya dalam rentang waktu tertentu untuk menyelesaikan serangkaian kegiatan [8]. Kegiatan yang dimaksud dalam penelitian ini adalah penjadwalan ujian untuk mata kuliah berbasis teori. Penjadwalan ujian untuk mata kuliah berbasis teori ini memiliki keterbatasan sumber daya berupa mata kuliah, ruangan, sesi, pengawas 1, dan pengawas 2.

Penjadwalan terbagi menjadi 2 bagian yaitu *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk dipenuhi). *Hard constraints* adalah batas-batas yang harus diterapkan pada penjadwalan dan harus dipenuhi. Sebuah solusi hanya dapat dikatakan benar jika dalam solusi tersebut sama sekali tidak ditemukan *hard constraint* yang terlanggar [9]. *Hard constraints* yang ditetapkan dalam pembangunan Sistem Penjadwalan Ujian Tengah Semester dan Ujian Akhir Semester adalah sebagai berikut :

1. Mata kuliah dengan dosen pengajar yang sama harus dilaksanakan pada sesi yang bersamaan.
2. 1 kelas menetap dalam 1 ruangan selama ujian berlangsung.
3. 1 ruangan hanya dapat digunakan dalam tiap 1 sesi.
4. Pengawas 1, dan pengawas 2 hanya dapat berada di 1 ruangan tiap sesinya dan bukan merupakan dosen pengampu dari mata kuliah yang diujikan.

Adapun *soft constraint* yang diterapkan dalam sistem ini, yaitu jika pengawas berhalangan hadir, maka pihak panitia dan BAAK akan mengkoordinir situasi. Hal itu dilakukan dengan cara memilih sumber daya manusia yang tersedia baik dari panitia ujian maupun pengawas lain yang tersedia tanpa merubah data jadwal pengawas yang telah ditetapkan.

2.2.3 Ujian

Ujian merupakan suatu cara untuk mengadakan penilaian yang berbentuk suatu tugas yang harus dikerjakan pelajar ataupun yang berkepentingan sehingga menghasilkan suatu nilai tentang tingkah laku

atau prestasi pelajar tersebut yang kemudian dapat dibandingkan dengan hasil tiap pelajar maupun standar yang berlaku [10].

Di PNC, terdapat dua jenis ujian, yaitu ujian teori dan ujian praktik. Ujian praktik dijadwalkan sesuai dengan waktu yang telah ditetapkan, namun pelaksanaannya tidak dilakukan secara serentak. Sementara itu, ujian teori dilakukan secara serentak dan sesuai dengan penjadwalan yang telah disiapkan oleh panitia ujian.

2.2.4 Ujian Tengah Semester (UTS)

Ujian Tengah Semester (UTS) adalah kegiatan akademis yang wajib dilakukan oleh setiap pelajar dalam menyelesaikan studinya sebagai bentuk evaluasi semester berjalan (tengah semester) dengan menekankan pada aspek kognitif. Tidak terkecuali mahasiswa perguruan tinggi. UTS di perguruan tinggi dilaksanakan setelah proses belajar mengajar berjalan kurang lebih tiga bulan [11]. UTS di PNC biasanya dilaksanakan setelah 8 kali pertemuan perkuliahan atau pada pertengahan periode semester tertentu.

2.2.5 Ujian Akhir Semester (UAS)

Ujian Akhir Semester (UAS) adalah kegiatan akademis yang wajib dilakukan oleh setiap pelajar dalam menyelesaikan studinya sebagai bentuk evaluasi akhir semester dengan menekankan pada aspek kognitif. Tidak terkecuali mahasiswa perguruan tinggi. UAS di perguruan tinggi dilaksanakan setelah semua proses belajar mengajar setiap semester selesai [12]. UAS di PNC biasanya dilaksanakan setelah 16 kali pertemuan perkuliahan atau pada akhir periode semester tertentu.

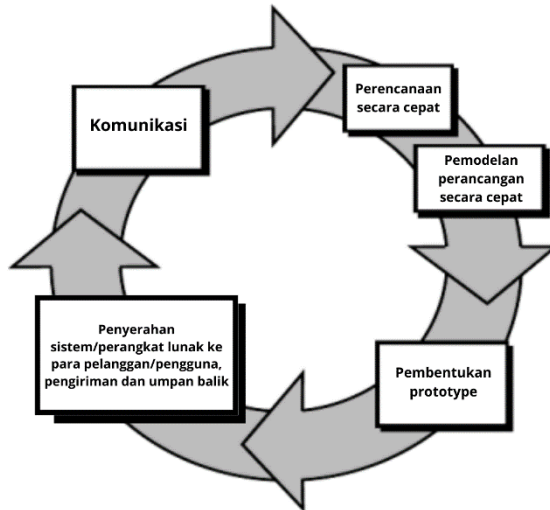
2.2.6 Rekayasa Web

Rekayasa web adalah sebuah pendekatan sistematis, disiplin, dan terukur dalam pengembangan, operasi, dan pemeliharaan aplikasi berbasis web. Rekayasa web juga berarti subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi web [13]. Berikut ini adalah metode pengembangan sistem dan pengujian sistem yang akan digunakan:

A. Metode Pengembangan Sistem

Pengembangan metode pengembangan sistem yang akan digunakan pada sistem ini yaitu metode pengembangan perangkat lunak Software Development Life (SDLC) model *prototype*. Metode ini

membantu pengembang dalam berkomunikasi dengan pengguna supaya mendapatkan hasil yang sesuai dengan keinginan pengguna melalui rancangan tampilan fungsionalitas aplikasi. Berikut ini adalah tahapan metode *prototype* menurut Pressman (2012) [3]:



Gambar 2. 1 Metode *Prototype* [3]

Berdasarkan model *prototype* yang telah digambarkan di atas, dapat diuraikan pembahasan disetiap tahapannya sebagai berikut [3]:

1. **Komunikasi**
 Pada tahap ini penulis bekerja sama dengan pengguna dalam mengidentifikasi kebutuhan dan tujuan utama sistem yang akan dikembangkan. Penulis menggunakan informasi ini untuk merancang dan mengembangkan *prototype* awal yang sederhana. Tujuan dari tahap ini yaitu untuk memahami kebutuhan dan ketentuan yang akan diterapkan dengan jelas, serta untuk dijadikan acuan dalam mengembangkan *prototype* secara lengkap.
2. **Perencanaan secara Cepat**
 Pada tahap ini, dilakukan perencanaan awal dalam pembuatan *prototype* pertama yang akan digunakan sebagai gambaran awal

bagi pengguna. Meskipun *prototype* awal ini belum lengkap, namun upaya dilakukan untuk memenuhi sebanyak mungkin kebutuhan pengguna. Tahapan iterasi selanjutnya pada tahap ini melibatkan perencanaan singkat guna menghasilkan *prototype* yang sesuai dengan kebutuhan pengguna.

3. **Pemodelan Perancangan secara Cepat**
Dalam tahap ini, dilakukan pemodelan alur dan desain sistem dengan ringkas. Tujuan dari tahap pemodelan ini adalah untuk menyajikan gambaran alur dan antarmuka perangkat lunak yang dapat dilihat oleh pengguna.
4. **Pembentukan *Prototype***
Tahap ini digunakan untuk merancang dengan cara mengkodekan sistem dan menguji sistem yang sedang dikembangkan, meskipun masih belum memiliki fungsi yang sempurna dan belum memiliki fitur yang lengkap. *Prototype* pada iterasi selanjutnya akan diperbaiki sesuai dengan kebutuhan dari pengguna.
5. **Penyerahan sistem/ perangkat lunak ke para pelanggan/ pengguna, pengiriman dan umpan balik**
Prototype kemudian diserahkan kepada pengguna untuk dievaluasi, dengan tujuan mendapatkan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Iterasi terjadi ketika penulis melakukan perbaikan terhadap *prototype* berdasarkan umpan balik yang diterima.




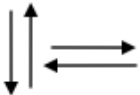
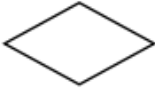
B. Metode Pengujian Sistem



Pengujian yang akan digunakan sistem ini yaitu pengujian *black box*. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak, dan kemudian keluaran dari perangkat lunak diuji kelayakannya. Keuntungan dari *black box testing* yaitu pengujian ini berasal dari sudut pandang pengguna dan tidak memerlukan pengetahuan yang tinggi dalam pemrograman sehingga lebih efisien dalam menemukan ketidaksesuaian spesifikasi yang diinginkan pengguna. Sedangkan kekurangan dari pengujian *black box* ini ialah cakupan pengujian dinilai terbatas karena hanya fungsional perangkat lunak yang diujikan [14].

2.2.7 Flowchart

Flowchart atau diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. *Flowchart* digunakan oleh analis sistem sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Oleh sebab itu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. *Flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung [15]. Berikut ini simbol-simbol pada *flowchart diagram* [16]:

Tabel 2. 1 Simbol *Flowchart* [16]

No	Simbol	Nama	Keterangan
1		<i>Terminal</i>	Awal atau akhir dari suatu prosedur.
2		Proses	Proses operasional yang terjadi dalam komputer.
3		<i>Output/Input</i>	Memasukkan atau mengeluarkan data tanpa melihat jenis perangkat yang digunakan.
4		Garis Alur	Menggambarkan alur atau aliran program.
5		<i>Decision</i>	Menandakan suatu kondisi yang mengarahkan pada dua kondisi YA/TIDAK.

No	Simbol	Nama	Keterangan
6		Simbol Dokumen	Untuk mencetak dokumen berupa kertas.
7		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer

2.2.8 Pemrograman Berbasis Objek (PBO)

Pemrograman Berbasis Objek (PBO) adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. PBO dapat diartikan sebagai paradigma atau teknik pemrograman yang berorientasi kepada objek [17]. Metodologi pengembangan sistem yang berorientasi objek memiliki beberapa konsep dasar yang harus dipahami, antara lain:

- a. **Kelas (*Class*)**
Kelas adalah kumpulan objek dengan karakteristik yang sama. Sebuah kelas memiliki atribut, operasi/*method*, relationship dan arti. Suatu kelas dapat diturunkan ke kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.
- b. **Objek (*Object*)**
Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi dan dimusnahkan.
- c. **Metode (*Method*)**
Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi berfungsi untuk memanipulasi objek itu sendiri.
- d. **Atribut (*Attribute*)**
Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut mempunyai

- secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga enkapsulasi.
- e. **Enkapsulasi (*Encapsulation*)**
Pembungkusan atribut data dan layanan (operasi - operasi) yang dimiliki objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.
 - f. **Pewarisan (*Inheritance*)**
Mekanisme yang memungkinkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.
 - g. **Antarmuka (*Interface*)**
Antarmuka atau interface biasa digunakan agar kelas lain tidak dapat mengakses langsung ke suatu kelas, melainkan hanya mengakses antarmukanya.
 - h. ***Reusability***
Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut. Misalkan pada sebuah aplikasi peminjaman buku diperlukan pada kelas anggota, maka ketika membuat aplikasi VCD, kelas anggota ini bisa digunakan kembali dengan sedikit perubahan untuk aplikasi penyewaan VCD tanpa harus membuat dari awal lagi.
 - i. **Komunikasi Antar Objek**
Komunikasi antara objek dilakukan melalui pesan (*message*) yang dikirim dari satu objek ke objek yang lainnya.
 - j. **Polimorfisme (*Polymorphism*)**
Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.
 - k. ***Package***
Package adalah sebuah kontainer yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas bernama sama disimpan dalam package yang berbeda.

2.2.9 Unified Modelling Language (UML)





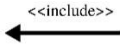
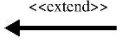
UML (Unified Modeling Language) adalah metode pemodelan secara visual untuk sarana perancangan sistem berorientasi objek. UML juga dapat diartikan sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem

software [18]. Berikut ini beberapa model diagram yang ada di dalam UML:

a) *Use Case Diagram*

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor [19]. Simbol-simbol pada *use case diagram* digambarkan dalam tabel 2.2.







Tabel 2. 2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Menggambarkan peran orang atau tokoh yang ada dalam sistem, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2		<i>Use case</i>	Interaksi antara sistem dan aktor yang dinyatakan dengan kata kerja.
3		Asosiasi	Penghubung antara aktor dengan <i>use case</i> .
4		Generalisasi	Spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

b) *Sequence Diagram*

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan [20]. Berikut ini simbol-simbol pada *sequence diagram*:

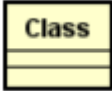

Tabel 2. 3 Simbol *Sequence Diagram* [20]





No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan sebuah objek atau tokoh yang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menunjukkan hubungan yang akan dilakukan dalam sistem.
3		<i>Boundary Class</i>	Menggambarkan scope atau biasanya berupa form.
4		<i>Control Class</i>	Menggambarkan hubungan antara boundary dengan tabel.
5		<i>Life Line</i>	Menggambarkan tempat dimulai dan berakhirnya <i>message</i> .
6		<i>Message</i>	Menggambarkan terjadinya pengiriman pesan.

c) Class Diagram

Class diagram merupakan sebuah kelas yang menggambarkan struktur dan penjelasan kelas, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain [21]. Berikut ini simbol-simbol pada *class diagram* [22]:

Tabel 2. 4 Simbol *Class Diagram* [22]

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Penggambaran nama kelas, atribut, atau properti atau data dan method
2		<i>Association</i>	Hubungan antara objek satu dengan objek lainnya.

No	Simbol	Nama	Keterangan
3		<i>Navigable Association</i>	Relasi antar kelas yang memiliki arti suatu kelas digunakan oleh kelas lain.
4		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
5		<i>Dependency</i>	Relasi antar kelas dengan kebergantungan antar kelas
6		<i>Aggregation</i>	Relasi antar kelas dengan semua bagian (<i>whole-part</i>).

2.2.10 Framework Laravel

Laravel adalah *framework* web PHP yang gratis dan bersifat *open source* dan berada di bawah lisensi Massachusetts Institute of Technology (MIT). *Framework* ini dikembangkan oleh Taylor Otwell dan digunakan untuk mengembangkan aplikasi web dengan arsitektur MVC (*Model-View-Controller*). MVC adalah sebuah arsitektur yang memisahkan komponen utama aplikasi, yaitu tampilan (*View*), logika pengontrol (*Controller*), dan penanganan data (*Model*) [23].

Laravel juga merupakan sebuah *framework* yang mudah dipahami dan menyediakan kemudahan dalam hal otentikasi, *routing*, manajemen sesi, *caching*, serta beberapa fitur lainnya. *Framework* ini juga menyediakan fitur-fitur tambahan seperti migrasi database dan dukungan untuk pengujian unit yang sangat berguna bagi pengembang dalam membangun aplikasi yang kompleks [24].

2.2.11 Database


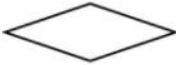


Database atau basis data merupakan kumpulan data yang terhubung secara logis yang dipakai bersama dan deskripsi dari data ini dirancang untuk memenuhi kebutuhan informasi sebuah organisasi [25]. Pada penelitian Sistem Informasi Penjadwalan UTS dan UAS ini,

database yang akan digunakan ialah MariaDB. Adapun diagram ERD yang dapat digunakan sebagai alat bantu dalam pembuatan database.

1. ERD (Entity Relationship Diagram)

ERD (Entity Relationship Diagram) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Berikut ini simbol-simbol yang ada pada ERD [26]:

Tabel 2. 5 Simbol *Entity Relationship Diagram* [26]

No	Simbol	Nama	Keterangan
1		Entitas	Kumpulan objek yang dapat diidentifikasi secara unik.
2		Relasi	Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan yang ada meliputi: satu ke satu, satu ke banyak, dan banyak ke banyak.
3		Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
4		Garis	Hubungan antara entitas dan atributnya dan himpunan entitas dengan himpunan relasi.

2.2.12 DBMS

Database Management System (DBMS) adalah perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil [27]. Berikut ini beberapa contoh DBMS :

1. Oracle
2. MySQL
3. Postgre SQL

~Halaman Sengaja Dikosongkan~